

2014 Prosper Loan Analysis in R

Raul Maldonado

- Background
- Import Data and Tools
 - Libraries
 - Import Data
- Prosper Loan Dataframe Characteristics
- Data Cleaning
 - Is there anything suspicious about the Data?
 - Basic summary of Null Values
 - How should we resolve this null value issue?
 - Null data decision
 - Filling in the missing data?
- Subset Dataframes
 - Prosper Customer Subset
 - Prosper Investor Subset
- Analysis
 - Prosper Customer Analysis
 - Histograms
 - Loan Status
 - Credit Grade
 - Estimated Loss
 - Current Delinquencies Histogram
 - Scatterplots
 - Loan Status vs Borrower Rate
 - Loan Status vs Estimated Loss
 - Loan Status vs Prosper Score
 - Borrower Rate vs Estimated Loss
 - Current Delinquencies vs Prosper Score
 - Current Delinquencies vs Prosper Score by Loan Status
 - Estimated Loss vs Borrower Rate Colored by Prosper Score
 - Prosper Investor Analysis
 - Histograms
 - Investors
 - Is Borrower a Homeowner?
 - Bank Card Utilization
 - Scatterplots
 - Investors vs Prosper Score
 - Investors vs Is Borrower a Homeowner
 - Investors vs Current Delinquencies
 - Investors (\log_{10}) vs Current Delinquencies
 - Investors vs Bank Card Utilization

- Summary
- Conclusion
- Reflections
- Resources
 - Data Dictionary

Dedication:

I dedicate this to the times I longed to become a data analyst while I worked in non-analytic financial positions.

“I wasn’t hungry for money. I was hungry for success” -Raul M.

Enjoy

Background

Prosper (<https://www.prosper.com>) is advertised as Americas first marketplace lending platform, with over \$9 billion in funded loans.

Prosper allows people to invest in each other in a way that is financially and socially rewarding. On Prosper, borrowers list loan requests between \$2,000 and \$35,000 and individual investors invest as little as \$25 in each loan listing they select. Prosper handles the servicing of the loan on behalf of the matched borrowers and investors.

We explore the 2014 Prosper data containing 113,937 loans with 81 variables on each loan, including loan amount, borrower rate (or interest rate), current loan status, borrower income, borrower employment status, borrower credit history, and the latest payment information.

Import Data and Tools

Libraries

We want to bring in the following libraries:

1. knitr (General-purpose tool for dynamic report generation)
2. dplyr (data manipulation library)
3. tidyr (data manipulation library)
4. ggplot2 (visualization library)
5. gridExtra (grid visualizations)

These libraries should sufficiently allow us to manipulate and explore the dataset.

Import Data

After importing the csv file “propserLoanData.csv,” we observe a glimpse of what the prosper data “df” will be below:

```

##          ListingKey ListingNumber           ListingCreationDate
## 1 1021339766868145413AB3B      193129 2007-08-26 19:09:29.263000000
## 2 10273602499503308B223C1     1209647 2014-02-27 08:28:07.900000000
##   CreditGrade Term LoanStatus           ClosedDate BorrowerAPR BorrowerRate
## 1          C    36 Completed 2009-08-14 00:00:00      0.16516      0.158
## 2          36 Current                  NA            0.12016      0.092
##   LenderYield EstimatedEffectiveYield EstimatedLoss EstimatedReturn
## 1       0.138                      NA             NA            NA
## 2       0.082                      0.0796        0.0249      0.0547
##   ProsperRating..numeric. ProsperRating..Alpha. ProsperScore
## 1                 NA                      NA            NA
## 2                 6                      A             7
##   ListingCategory..numeric. BorrowerState Occupation EmploymentStatus
## 1                 0                     CO    Other  Self-employed
## 2                 2                     CO Professional Employed
##   EmploymentStatusDuration IsBorrowerHomeowner CurrentlyInGroup GroupKey
## 1                 2                   True        True
## 2                 44                  False       False
##   DateCreditPulled CreditScoreRangeLower
## 1 2007-08-26 18:41:46.780000000      640
## 2 2014-02-27 08:28:14                  680
##   CreditScoreRangeUpper FirstRecordedCreditLine CurrentCreditLines
## 1       659      2001-10-11 00:00:00          5
## 2       699      1996-03-18 00:00:00         14
##   OpenCreditLines TotalCreditLinespast7years OpenRevolvingAccounts
## 1         4                  12              1
## 2        14                  29             13
##   OpenRevolvingMonthlyPayment InquiriesLast6Months TotalInquiries
## 1         24                  3               3
## 2        389                  3               5
##   CurrentDelinquencies AmountDelinquent DelinquenciesLast7Years
## 1         2                  472              4
## 2         0                  0               0
##   PublicRecordsLast10Years PublicRecordsLast12Months
## 1         0                  0               0
## 2         1                  0               0
##   RevolvingCreditBalance BankcardUtilization AvailableBankcardCredit
## 1         0                  0.00            1500
## 2      3989                  0.21            10266
##   TotalTrades TradesNeverDelinquent..percentage. TradesOpenedLast6Months
## 1        11                  0.81            0
## 2        29                  1.00            2
##   DebtToIncomeRatio IncomeRange IncomeVerifiable StatedMonthlyIncome
## 1       0.17 $25,000-49,999            True      3083.333
## 2       0.18 $50,000-74,999            True      6125.000
##   LoanKey TotalProsperLoans TotalProsperPaymentsBilled
## 1 E33A3400205839220442E84            NA            NA
## 2 9E3B37071505919926B1D82            NA            NA

```

```

## OnTimeProsperPayments ProsperPaymentsLessThanOneMonthLate
## 1 NA NA
## 2 NA NA
## ProsperPaymentsOneMonthPlusLate ProsperPrincipalBorrowed
## 1 NA NA
## 2 NA NA
## ProsperPrincipalOutstanding ScorexChangeAtTimeOfListing
## 1 NA NA
## 2 NA NA
## LoanCurrentDaysDelinquent LoanFirstDefaultedCycleNumber
## 1 0 NA
## 2 0 NA
## LoanMonthsSinceOrigination LoanNumber LoanOriginalAmount
## 1 78 19141 9425
## 2 0 134815 10000
## LoanOriginationDate LoanOriginationQuarter MemberKey
## 1 2007-09-12 00:00:00 Q3 2007 1F3E3376408759268057EDA
## 2 2014-03-03 00:00:00 Q1 2014 1D13370546739025387B2F4
## MonthlyLoanPayment LP_CustomerPayments LP_CustomerPrincipalPayments
## 1 330.43 11396.14 9425
## 2 318.93 0.00 0
## LP_InterestandFees LP_ServiceFees LP_CollectionFees
## 1 1971.14 -133.18 0
## 2 0.00 0.00 0
## LP_GrossPrincipalLoss LP_NetPrincipalLoss
## 1 0 0
## 2 0 0
## LP_NonPrincipalRecoverypayments PercentFunded Recommendations
## 1 0 1 0
## 2 0 1 0
## InvestmentFromFriendsCount InvestmentFromFriendsAmount Investors
## 1 0 0 258
## 2 0 0 1

```

Observe that we only called two rows from an 81 variable dataset.

Additionally, we look at the feature types for the first 10 rows and first 3 columns

```

## 'data.frame': 10 obs. of 3 variables:
## $ ListingKey : Factor w/ 113066 levels "00003546482094282EF90E
## 5",...: 7180 7193 6647 6669 6686 6689 6699 6706 6687 6687
## $ ListingNumber : int 193129 1209647 81716 658116 909464 1074836 7508
## 99 768193 1023355 1023355
## $ ListingCreationDate: Factor w/ 113064 levels "2005-11-09 20:44:28.8470000
## 00",...: 14184 111894 6429 64760 85967 100310 72556 74019 97834 97834

```

Prosper Loan Dataframe Characteristics

The following are the column names, number of rows, and number of columns respectively.

Column Names:

```
## [1] "ListingKey"
## [2] "ListingNumber"
## [3] "ListingCreationDate"
## [4] "CreditGrade"
## [5] "Term"
## [6] "LoanStatus"
## [7] "ClosedDate"
## [8] "BorrowerAPR"
## [9] "BorrowerRate"
## [10] "LenderYield"
## [11] "EstimatedEffectiveYield"
## [12] "EstimatedLoss"
## [13] "EstimatedReturn"
## [14] "ProsperRating..numeric."
## [15] "ProsperRating..Alpha."
## [16] "ProsperScore"
## [17] "ListingCategory..numeric."
## [18] "BorrowerState"
## [19] "Occupation"
## [20] "EmploymentStatus"
## [21] "EmploymentStatusDuration"
## [22] "IsBorrowerHomeowner"
## [23] "CurrentlyInGroup"
## [24] "GroupKey"
## [25] "DateCreditPulled"
## [26] "CreditScoreRangeLower"
## [27] "CreditScoreRangeUpper"
## [28] "FirstRecordedCreditLine"
## [29] "CurrentCreditLines"
## [30] "OpenCreditLines"
## [31] "TotalCreditLinespast7years"
## [32] "OpenRevolvingAccounts"
## [33] "OpenRevolvingMonthlyPayment"
## [34] "InquiriesLast6Months"
## [35] "TotalInquiries"
## [36] "CurrentDelinquencies"
## [37] "AmountDelinquent"
## [38] "DelinquenciesLast7Years"
## [39] "PublicRecordsLast10Years"
## [40] "PublicRecordsLast12Months"
## [41] "RevolvingCreditBalance"
## [42] "BankcardUtilization"
## [43] "AvailableBankcardCredit"
## [44] "TotalTrades"
## [45] "TradesNeverDelinquent..percentage."
## [46] "TradesOpenedLast6Months"
## [47] "DebtToIncomeRatio"
## [48] "IncomeRange"
```

```

## [49] "IncomeVerifiable"
## [50] "StatedMonthlyIncome"
## [51] "LoanKey"
## [52] "TotalProsperLoans"
## [53] "TotalProsperPaymentsBilled"
## [54] "OnTimeProsperPayments"
## [55] "ProsperPaymentsLessThanOneMonthLate"
## [56] "ProsperPaymentsOneMonthPlusLate"
## [57] "ProsperPrincipalBorrowed"
## [58] "ProsperPrincipalOutstanding"
## [59] "ScorexChangeAtTimeOfListing"
## [60] "LoanCurrentDaysDelinquent"
## [61] "LoanFirstDefaultedCycleNumber"
## [62] "LoanMonthsSinceOrigination"
## [63] "LoanNumber"
## [64] "LoanOriginalAmount"
## [65] "LoanOriginationDate"
## [66] "LoanOriginationQuarter"
## [67] "MemberKey"
## [68] "MonthlyLoanPayment"
## [69] "LP_CustomerPayments"
## [70] "LP_CustomerPrincipalPayments"
## [71] "LP_InterestandFees"
## [72] "LP_ServiceFees"
## [73] "LP_CollectionFees"
## [74] "LP_GrossPrincipalLoss"
## [75] "LP_NetPrincipalLoss"
## [76] "LP_NonPrincipalRecoverypayments"
## [77] "PercentFunded"
## [78] "Recommendations"
## [79] "InvestmentFromFriendsCount"
## [80] "InvestmentFromFriendsAmount"
## [81] "Investors"

```

Number of rows: 113937

Number of columns 81

This 113,937 by 81 loan data is gargantuan. We only outputted the first three rows, above. Ingesting this dataset, we would undergo heavy time costs to our analysis. To improve computing performance and minimize time on project, for the sake of this analysis, we create several subsets from the Prosper loan dataframe “df.”

Note: What’s that “NA” entry that frequently comes up?

Data Cleaning

Notice how we didn't introduce the data as "tidy", "clean", or even mention how we extracted the data. I.e., We didn't introduce any audited data, in which could imply our data has not been maintained or corrected to analysis standards.

Is there anything suspicious about the Data?

We observe the total amount of null values in the Prosper dataframe df 1078519

We have 1078519 total missing values in our dataset.

Below is a breakdown of the total null values per column in the df dataframe:

	df_na_count
##	0
## ListingKey	0
## ListingNumber	0
## ListingCreationDate	0
## CreditGrade	0
## Term	0
## LoanStatus	0
## ClosedDate	0
## BorrowerAPR	25
## BorrowerRate	0
## LenderYield	0
## EstimatedEffectiveYield	29084
## EstimatedLoss	29084
## EstimatedReturn	29084
## ProsperRating..numeric.	29084
## ProsperRating..Alpha.	0
## ProsperScore	29084
## ListingCategory..numeric.	0
## BorrowerState	0
## Occupation	0
## EmploymentStatus	0
## EmploymentStatusDuration	7625
## IsBorrowerHomeowner	0
## CurrentlyInGroup	0
## GroupKey	0
## DateCreditPulled	0
## CreditScoreRangeLower	591
## CreditScoreRangeUpper	591
## FirstRecordedCreditLine	0
## CurrentCreditLines	7604
## OpenCreditLines	7604
## TotalCreditLinespast7years	697
## OpenRevolvingAccounts	0
## OpenRevolvingMonthlyPayment	0
## InquiriesLast6Months	697
## TotalInquiries	1159
## CurrentDelinquencies	697
## AmountDelinquent	7622
## DelinquenciesLast7Years	990
## PublicRecordsLast10Years	697
## PublicRecordsLast12Months	7604
## RevolvingCreditBalance	7604
## BankcardUtilization	7604
## AvailableBankcardCredit	7544
## TotalTrades	7544
## TradesNeverDelinquent..percentage.	7544
## TradesOpenedLast6Months	7544
## DebtToIncomeRatio	8554

```

## IncomeRange 0
## IncomeVerifiable 0
## StatedMonthlyIncome 0
## LoanKey 0
## TotalProsperLoans 91852
## TotalProsperPaymentsBilled 91852
## OnTimeProsperPayments 91852
## ProsperPaymentsLessThanOneMonthLate 91852
## ProsperPaymentsOneMonthPlusLate 91852
## ProsperPrincipalBorrowed 91852
## ProsperPrincipalOutstanding 91852
## ScorexChangeAtTimeOfListing 95009
## LoanCurrentDaysDelinquent 0
## LoanFirstDefaultedCycleNumber 96985
## LoanMonthsSinceOrigination 0
## LoanNumber 0
## LoanOriginalAmount 0
## LoanOriginationDate 0
## LoanOriginationQuarter 0
## MemberKey 0
## MonthlyLoanPayment 0
## LP_CustomerPayments 0
## LP_CustomerPrincipalPayments 0
## LP_InterestandFees 0
## LP_ServiceFees 0
## LP_CollectionFees 0
## LP_GrossPrincipalLoss 0
## LP_NetPrincipalLoss 0
## LP_NonPrincipalRecoverypayments 0
## PercentFunded 0
## Recommendations 0
## InvestmentFromFriendsCount 0
## InvestmentFromFriendsAmount 0
## Investors 0

```

Basic summary of Null Values

The mean NA count for all columns is 1.331504910^{4}

13315.05 is the estimated average amount of null values in some column. What percentage of our df datafram is this?

Dividing the number of rows of the datafram by the mean of the columns, we observe 0.1168633 entries missing.

How should we resolve this null value issue?

We have two choices:

1. Should we delete customer data(rows) where some total null values of a given column are significantly high, such as +1000 points?
2. Fill in the missing non-null value entries for each column?

Null data decision

Say you purchase a meal for \$14.99. You eat a majority of it, but then become full. Do you eat the food, or leave it be?

This is subjective, though I feel I would eat the food because I am lower income individual that worked hard for every single dollar—and I certainly don't want food to go to waste(that is, I don't want my money to go to waste)

Similarly, this estimated 11% loss of data appears to be insignificant, but preservation of such details is important in our analysis. Obtaining the data is a great ordeal, and we should preserve the, once was difficult, capute of information.

So, for the sake of minimizing data loss, we fill in the null values of the dataframe.

Filling in the missing data?

We will use the fill function from tidy to conveniently fill in missing data with the last recorded non-null value in each category.

Example: if row 81 column 8 had an entered value 3.14159, and had several consecutive null values after, the fill function will fill in the null values with 3.14159. (Note: This filling in of data induces bias. However, we preserve data. A detailed discussion will be made at the ending of this report)

After filling in the data, we again observe the total amount and per-category null values in the Prosper dataframe df was 64.

	df_na_count
##	0
## ListingKey	0
## ListingNumber	0
## ListingCreationDate	0
## CreditGrade	0
## Term	0
## LoanStatus	0
## ClosedDate	0
## BorrowerAPR	0
## BorrowerRate	0
## LenderYield	0
## EstimatedEffectiveYield	1
## EstimatedLoss	1
## EstimatedReturn	1
## ProsperRating..numeric.	1
## ProsperRating..Alpha.	0
## ProsperScore	1
## ListingCategory..numeric.	0
## BorrowerState	0
## Occupation	0
## EmploymentStatus	0
## EmploymentStatusDuration	0
## IsBorrowerHomeowner	0
## CurrentlyInGroup	0
## GroupKey	0
## DateCreditPulled	0
## CreditScoreRangeLower	0
## CreditScoreRangeUpper	0
## FirstRecordedCreditLine	0
## CurrentCreditLines	0
## OpenCreditLines	0
## TotalCreditLinespast7years	0
## OpenRevolvingAccounts	0
## OpenRevolvingMonthlyPayment	0
## InquiriesLast6Months	0
## TotalInquiries	0
## CurrentDelinquencies	0
## AmountDelinquent	0
## DelinquenciesLast7Years	0
## PublicRecordsLast10Years	0
## PublicRecordsLast12Months	0
## RevolvingCreditBalance	0
## BankcardUtilization	0
## AvailableBankcardCredit	0
## TotalTrades	0
## TradesNeverDelinquent..percentage.	0
## TradesOpenedLast6Months	0
## DebtToIncomeRatio	0

```

## IncomeRange 0
## IncomeVerifiable 0
## StatedMonthlyIncome 0
## LoanKey 0
## TotalProsperLoans 4
## TotalProsperPaymentsBilled 4
## OnTimeProsperPayments 4
## ProsperPaymentsLessThanOneMonthLate 4
## ProsperPaymentsOneMonthPlusLate 4
## ProsperPrincipalBorrowed 4
## ProsperPrincipalOutstanding 4
## ScorexChangeAtTimeOfListing 16
## LoanCurrentDaysDelinquent 0
## LoanFirstDefaultedCycleNumber 15
## LoanMonthsSinceOrigination 0
## LoanNumber 0
## LoanOriginalAmount 0
## LoanOriginationDate 0
## LoanOriginationQuarter 0
## MemberKey 0
## MonthlyLoanPayment 0
## LP_CustomerPayments 0
## LP_CustomerPrincipalPayments 0
## LP_InterestandFees 0
## LP_ServiceFees 0
## LP_CollectionFees 0
## LP_GrossPrincipalLoss 0
## LP_NetPrincipalLoss 0
## LP_NonPrincipalRecoverypayments 0
## PercentFunded 0
## Recommendations 0
## InvestmentFromFriendsCount 0
## InvestmentFromFriendsAmount 0
## Investors 0

```

We have a total of 16 missing values. It's interesting that there is still some residue of null values. However, we will not pursue this interesting route, for the sake of simplicity in this project.

Note: If one were to go more in depth, one would observe these remaining outlier variables, reiterate, and then repeat process.

Recalling the two solutions for dealing with missing data, we decide to simply delete these remaining 16 values. Though these 16 persons with some missing value can have an influence in our analysis without them, we assume their impact to the analysis to be unlikely.

Below is the fraction at which they represent our dataset 1.404284810^{-4}

This removal is a small, small, risk we are taking.

After removing the residual nulls,

We have 0 Null values remaining

After removing the 16 customer information, rows, we proceed to subsetting our data.

Subset Dataframes

We generate two subset data from our Prosper loan dataset.

These two subsets are based on the idea of perspectives of a Prosper loan.

1. How does one Prosper customer interact/feel towards the loan structure?
2. How does one Prosper investor interact/feel towards the loan structure?

Prosper Customer Subset

The Prosper customer subset “loan_customer_perspective” will contain the following variables:

1. The Credit Grade
2. Loan Status
3. Borrower Rate
4. Estimated Loss
5. Prosper Rating
6. Available Bank Card-Credit

Below is the first few rows of the customer perspective dataset.

	CreditGrade	LoanStatus	BorrowerRate	EstimatedLoss	ProsperScore
## 11833		Chargedoff	0.3177	0.1625	2
## 104541	A	Completed	0.1300	0.1175	2
## 66036		Completed	0.0628	0.0200	10
## 47481		Current	0.1840	0.0749	2
## 35465	A	Completed	0.1272	0.1625	3
## 13413	D	Completed	0.2300	0.0749	4
## CurrentDelinquencies					
## 11833		0			
## 104541		4			
## 66036		0			
## 47481		0			
## 35465		0			
## 13413		0			

Furthermore, the following is table statistics for this dataset:

```

##   CreditGrade          LoanStatus      BorrowerRate
## :29826    Current       :19821 Min.   :0.0000
##  C      : 2003 Completed     :13370 1st Qu.:0.1349
##  D      : 1762 Chargedoff    : 4209 Median :0.1840
##  B      : 1516 Defaulted     : 1791 Mean    :0.1933
##  HR     : 1281 Past Due (1-15 days) :  286 3rd Qu.:0.2509
##  A      : 1203 Past Due (91-120 days): 124 Max.   :0.4800
## (Other): 2409 (Other)        : 399
##   EstimatedLoss    ProsperScore CurrentDelinquencies
## Min.   :0.00490 Min.   : 1.000 Min.   : 0.000
## 1st Qu.:0.04240 1st Qu.: 4.000 1st Qu.: 0.000
## Median :0.06990 Median : 6.000 Median : 0.000
## Mean   :0.08032 Mean   : 5.954 Mean   : 0.606
## 3rd Qu.:0.11200 3rd Qu.: 8.000 3rd Qu.: 0.000
## Max.   :0.36600 Max.   :11.000 Max.   :83.000
##

```

We observe a frequency table for Credit Grade and Loan Status, and numerical calculations for the remainder. I.e., we see counts for the first two features, but see mean, range, etc in the other variables.

Recall factor sets. The first features are factor variables, while the remainder are numerical.

In regards to the statistical information, the minimum, 1st quantile, median, and 3rd quantile are 0. However, mean is non-zero value. We interpret this as the Current Delinquencies feature have several zeros.

Moreover, we observe “other” classification for Credit Grade and Loan Status features. We explore these observations through visual analysis in the “Prosper Customer Analysis” section.

Prosper Investor Subset

The Prosper investor subset “loan_investor_perspective” will contain the following variables:

1. Employment Status
2. Bank Card Utilization
3. Investors
4. LP_Collection Fees
5. Prosper Score
6. IsBorrowerHomeowner
7. CurrentDelinquencies

Below is the first few rows of the loan_customer_perspective dataset

```

##          ProsperScore EmploymentStatus IsBorrowerHomeowner
## 91520              2           Employed             True
## 113155              7           Employed             True
## 17888              9        Full-time            False
## 102009              6        Full-time            False
## 75138               8           Other              True
## 3698                4        Full-time            True
##          BankcardUtilization Investors LP_CollectionFees
## 91520                 1.02       39         0.000
## 113155                 0.28      130         0.000
## 17888                 1.09       31         0.000
## 102009                 0.83       1         0.000
## 75138                 0.63       1         0.000
## 3698                 0.00      27     -123.274
##          CurrentDelinquencies
## 91520                  0
## 113155                  0
## 17888                  0
## 102009                  6
## 75138                  0
## 3698                   1

```

```

##          ProsperScore EmploymentStatus IsBorrowerHomeowner
## Min.    : 1.000   Employed      :11770   False: 9844
## 1st Qu.: 4.000   Full-time     : 4660   True :10156
## Median  : 6.000   Self-employed: 1082
## Mean    : 5.951   Not available: 918
## 3rd Qu.: 8.000   Other        : 701
## Max.    :11.000
##                   :        401
##                   : (Other)      : 468
##          BankcardUtilization Investors LP_CollectionFees
## Min.    :0.0000   Min.    : 1.00   Min.    :-9274.75
## 1st Qu.:0.3100   1st Qu.: 2.00   1st Qu.: 0.00
## Median :0.6000   Median  : 44.00  Median  : 0.00
## Mean   :0.5612   Mean    : 80.52  Mean   : -14.96
## 3rd Qu.:0.8400   3rd Qu.:114.00  3rd Qu.: 0.00
## Max.   :5.8300   Max.    :1189.00 Max.   : 0.00
##
##          CurrentDelinquencies
## Min.    : 0.0000
## 1st Qu.: 0.0000
## Median : 0.0000
## Mean   : 0.5754
## 3rd Qu.: 0.0000
## Max.   :51.0000
## 

```

We observe a frequency table for the “Employment Status” and “Borrower Homeowner?” features, and numerical calculations for the remainder. I.e., we see counts for the first two features, but see mean, range, etc in the other variables.

Recall factor sets. The first features are factor variables, while the remainder are numerical.

In regards to the statistical information, the minimum, 1st quantile, median, and 3rd quantile are 0. However, mean is non-zero value. We interpret this as the Current Delinquencies feature have several zeros.

Analysis

Prosper Customer Analysis

** Motivation **

If we were a potential Prosper customer/borrower, what considerations should we factor in to be successful in paying off loan. I.e., If we were a borrower, what characteristics should I emulate to be well-off with Prosper (even considering I may want to borrow with them again, in the future)

Again, we are utilizing the loan_customer_perspective dataset. The following are the dataset variables available to us:

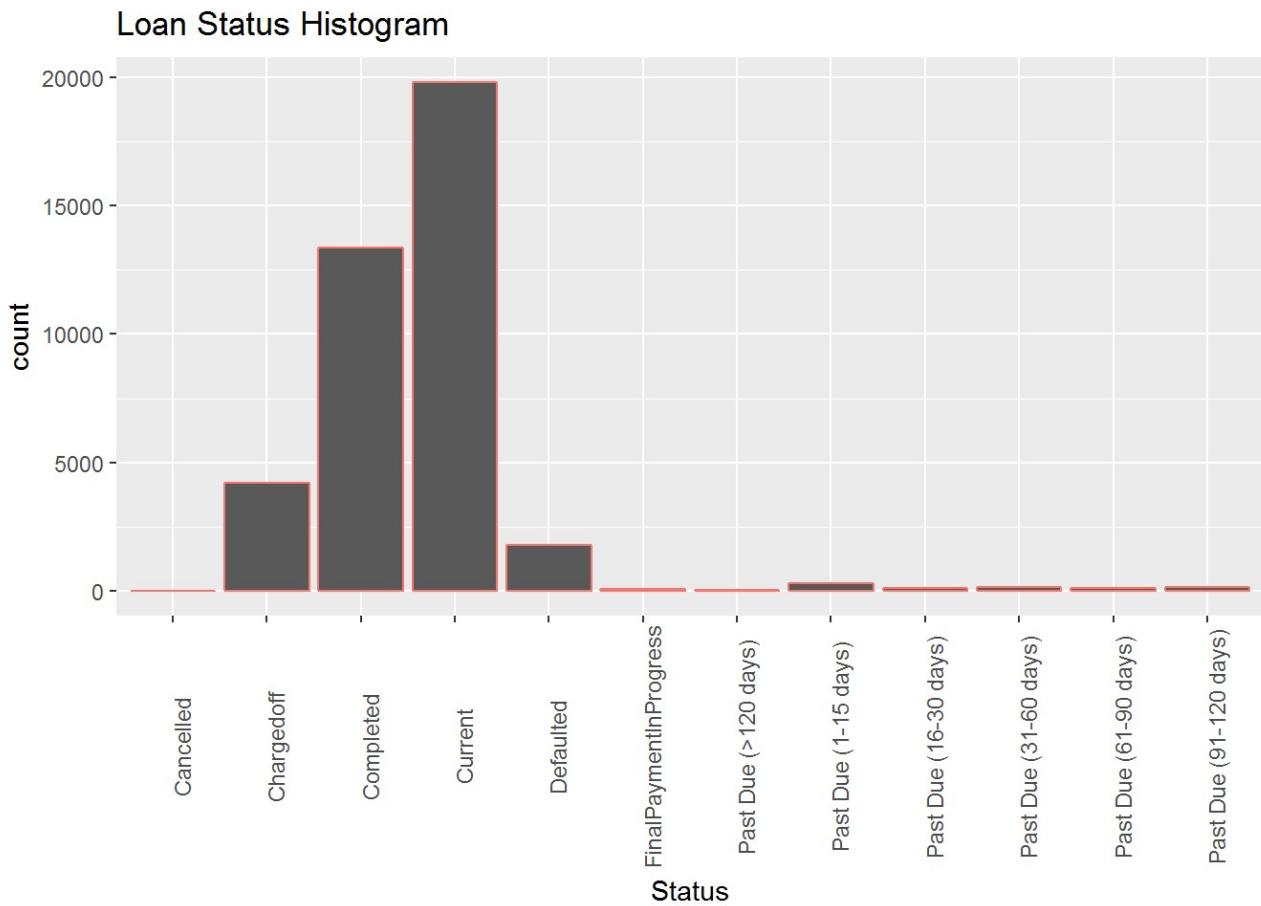
1. Credit Grade (Factor Variable)
2. Borrower Rate (Factor Variable)
3. Prosper Score (Num variable)
4. Loan Status (Num variable)
5. Estimated Loss (Num variable)
6. Current Delinquencies (int variable)

The size of our data set is 40000 rows x 6 columns.

```
## [1] "CreditGrade"          "LoanStatus"           "BorrowerRate"  
## [4] "EstimatedLoss"        "ProsperScore"         "CurrentDelinquencies"
```

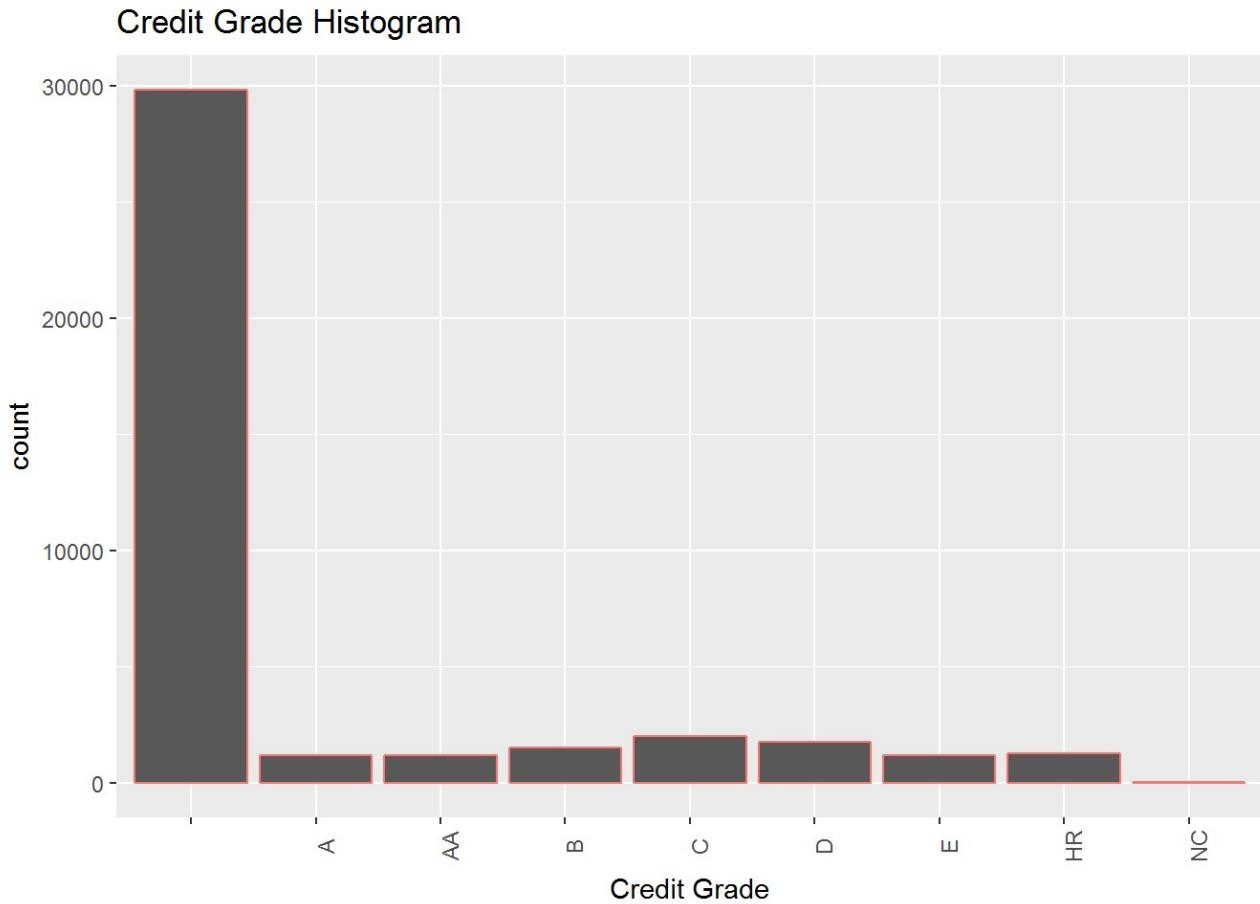
Histograms

Loan Status



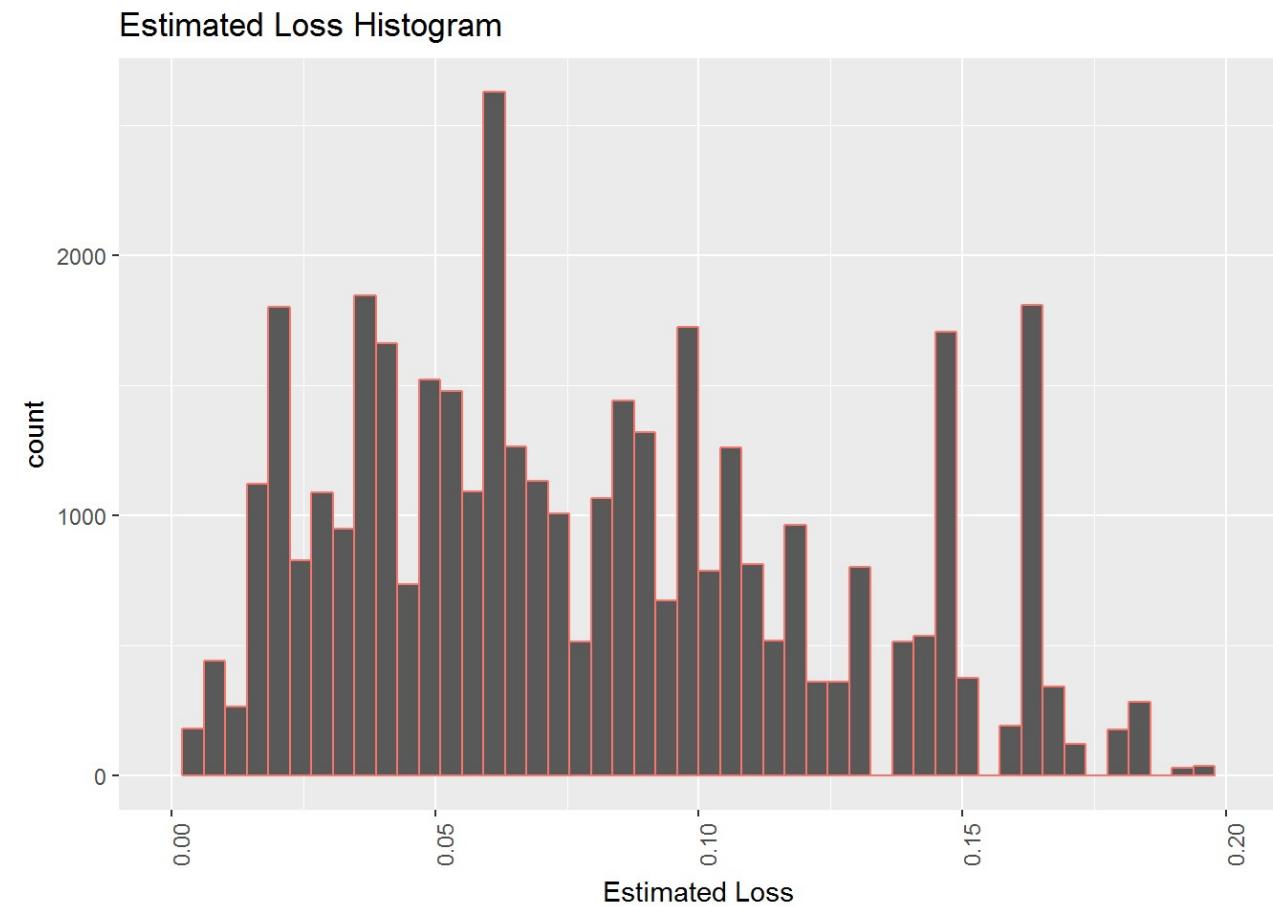
This is the count of the status of each Prosper customer loan. Of course, we observe higher completions, current, and charge offs than delinquent cases—that's good!

Credit Grade



We observe a large amount of empty values with the Credit Grade information. This is alarming. We consider a few cases. Was there data loss in moving this information? Did Prosper take higher initiative to focus on this information in 2014? Did we mess up during the data cleaning process?

Estimated Loss

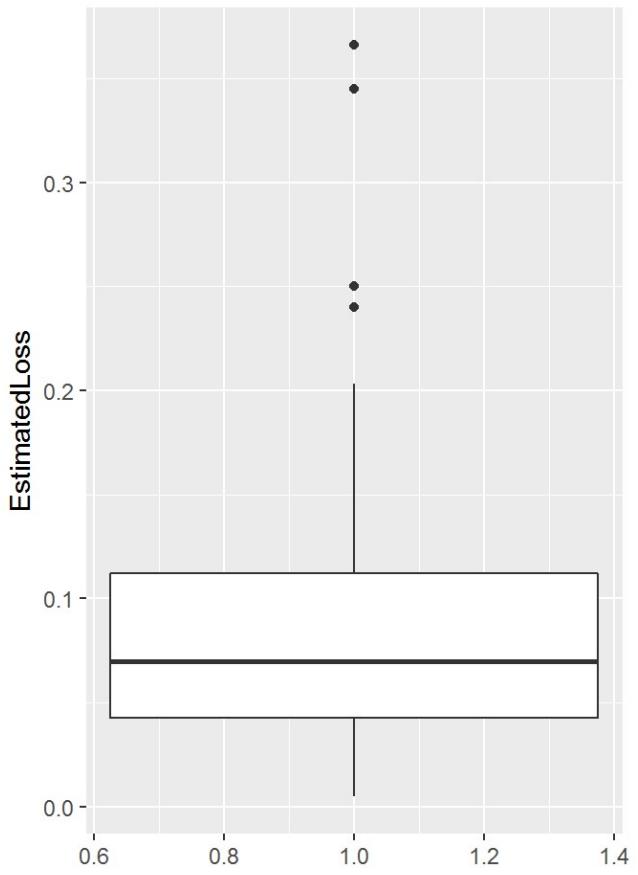
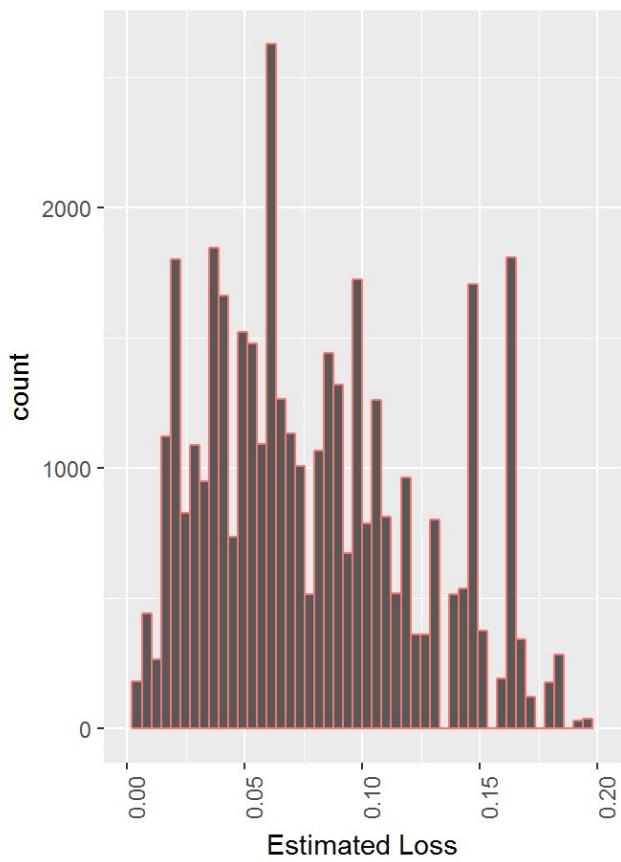


We observe our estimated loss of the original principle from chargeoffs ranges in the 10% region. From the following summary, we observe that the average estimate loss was actually 8.032%

```
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.00490 0.04240 0.06990 0.08032 0.11200 0.36600
```

The maximum value at which some estimated loss occurred was at 36.6%. This is alarming. However, we should observe if this is a typical ceiling scenario in loss from delinquency. As a customer, we don't want to default with a high loan. (As an investor, we do not want to see such loss!)

Estimated Loss Histogram

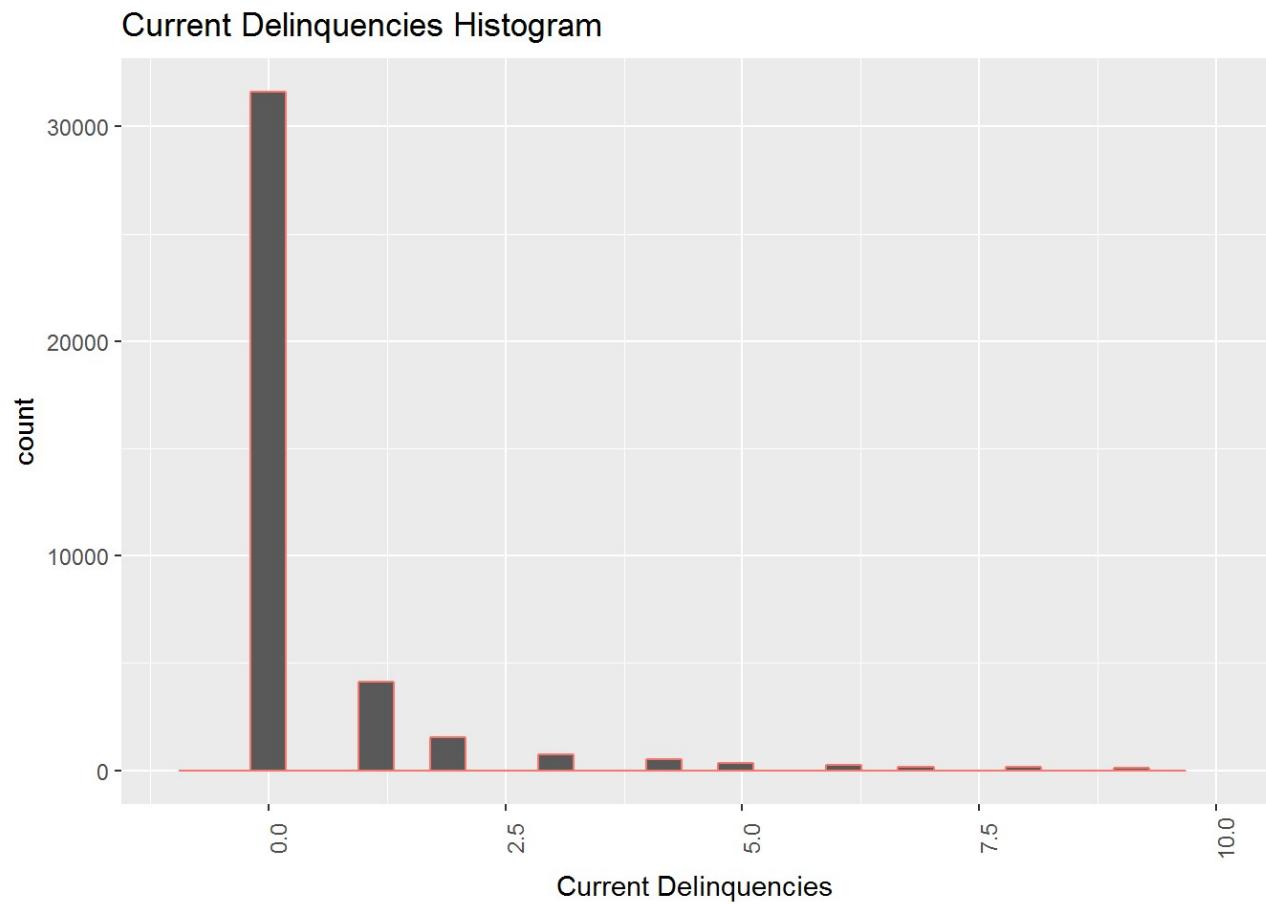


We observe that the significant 36.6% loss was, in fact, part of a cluster of outliers, as seen by the dots of the above right diagram. These outliers are abnormal cases.

It's good to note that at most 20%, the end of the vertical line on the right graph, of investment is lost. For a customers taking out a loan after July 2009, there likelyhood of defaulting is in 10-20 percent range.

However, this is no excuse from the investor perspective.

Current Delinquencies Histogram



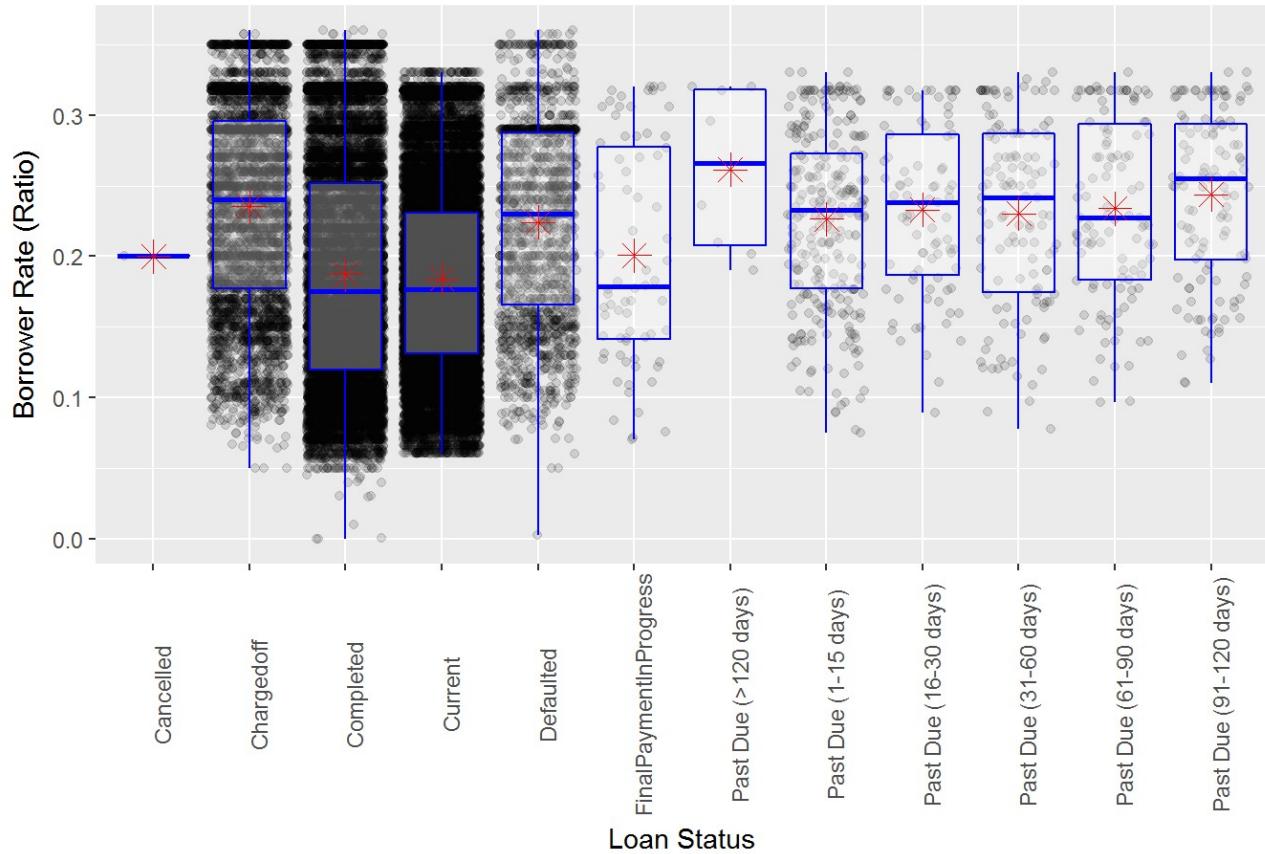
Similar to our Estimated loss and Loan Status, we observe a lack in delinquent customers in our sample of the Prosper customer information. This conclusion can be seen from the large amount of 0 valued current delinquencies.

Scatterplots

We now consider bi-variate cases. We display visualizations for these cases using scatterplots.

Loan Status vs Borrower Rate

Loan Status vs Borrower Rate Scatterplot



From the red stars, We observe most of the customer loan data centralized at the Borrower Rate of 0.25, 25% in most cases.

Though, we can just utilize the below statistical information to resolve this question.

```
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.0000  0.1349  0.1840  0.1933  0.2509  0.4800
```

So in fact, the average Borrower Rate for Prosper Loans is 0.193279

A little off, but close. This unexpected average could be weighed down by lower values. But what happens if we find the median borrower rate?

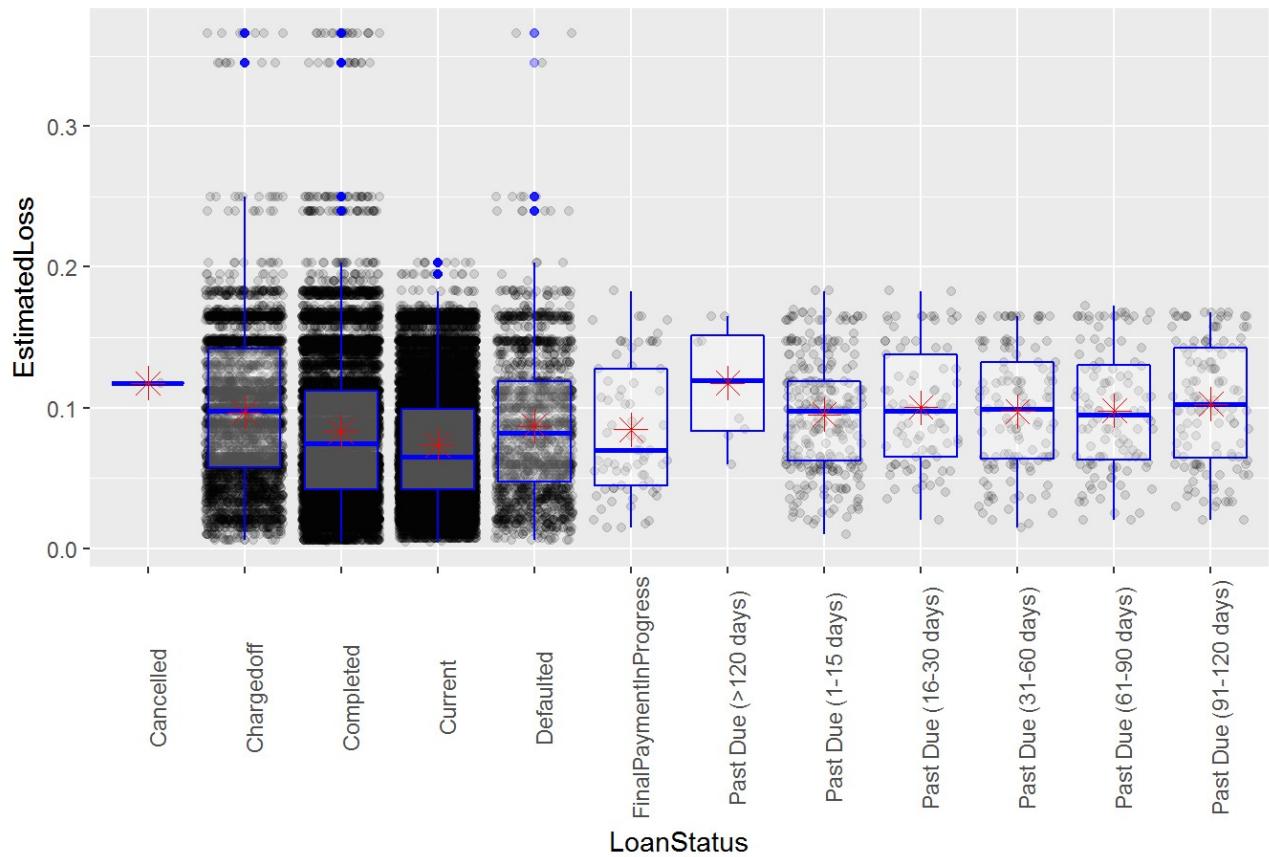
The median borrower rate 0.184We will reference this value as our central focal point of Borrower rate values.

Another observation is the concentration of dark colored data points in the non-past due and non-default cases. Again, this is because our data capture had no significant past due or default cases.

Note: There is a big concentration of chargeoff loans. One may ask why there is a major severity in this category. I would say that this is a recorded tally throughout Prosper's existense, and thus is why we see a large amount of charge offs, yet see a lack of Prosper customers being past due/defaulted.

Loan Status vs Estimated Loss

Loan Status vs Estimated Loss Scatterplot



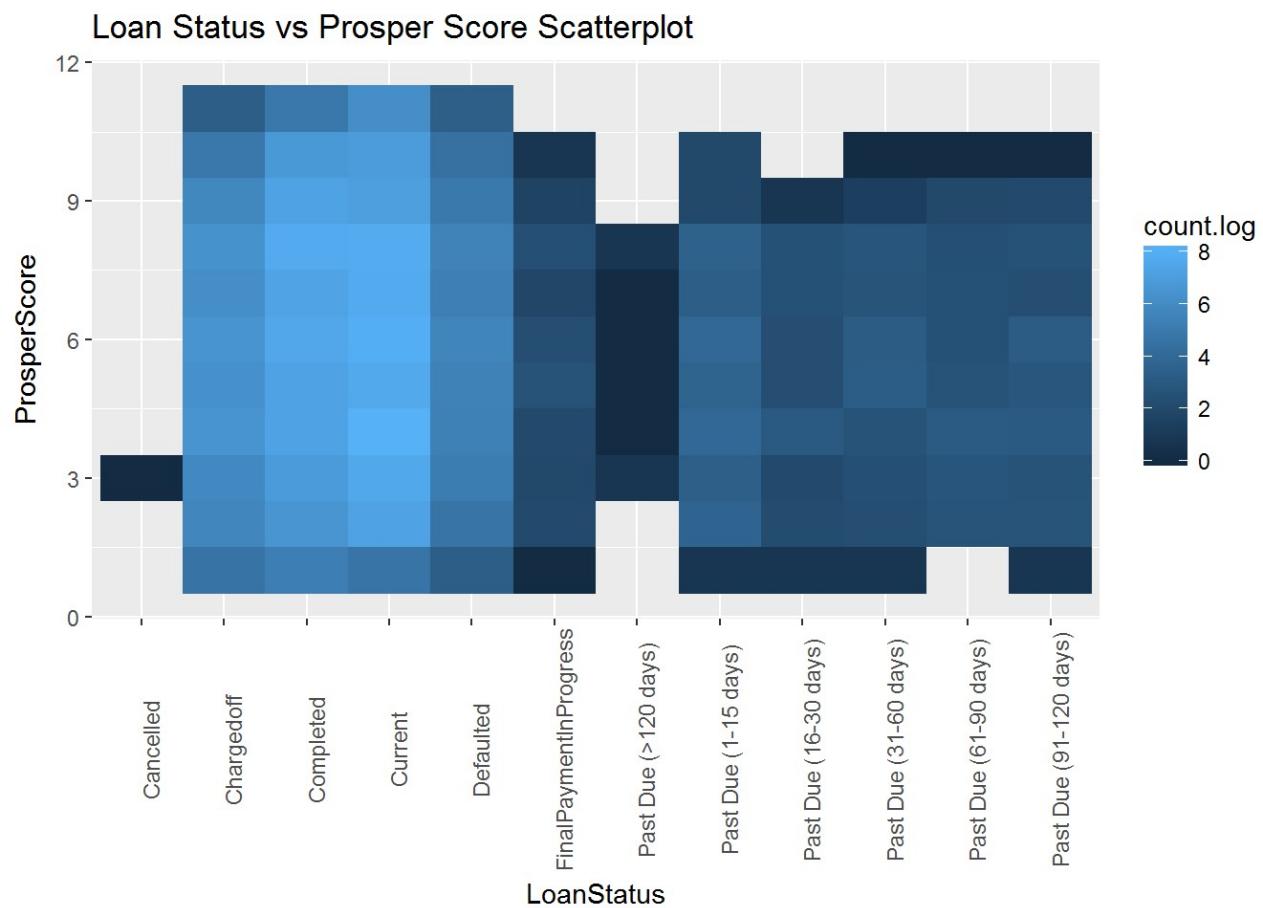
Again, There is a large concentration of dark colored data points in the non-past due and non-default cases. Again, this is because our data capture had no significant past due or default cases.

Note: There is a big concentration of chargeoff loans. One may ask why there is a major severity in this category. I would say that this is a recorded tally throughout Prosper's existense, and thus is why we see a large amount of charge offs, yet see a lack of Prosper customers being past due/defaulted.

Second Note: Notice there is a darker concentration for the outliers in Charged Off and Complete categories, compared to the Borrow Rate vs Loan Status information. I am not leading in to draw connections to the two scenarios. However, I am addressing that this situation has something interesting. When an estimated loss outlier occurs, it ranges into the 30% Estimated Loss region.

Loan Status vs Prosper Score

```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

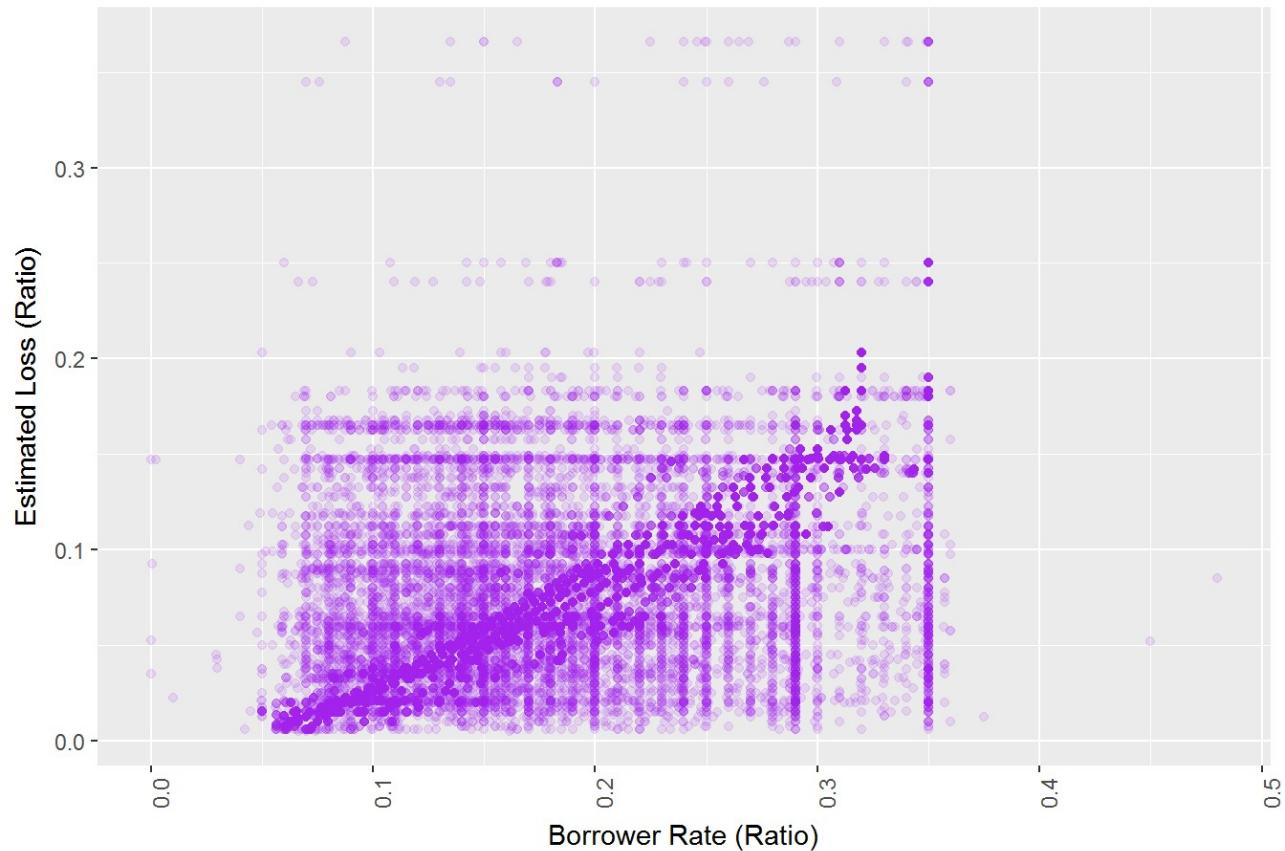


One would anticipate Prosper Scores would be higher if a loan status was in good standing (current, completed, etc.) However, we observe a dark concentration in Charged off, completed, and current categories in the entire Prosper Score range, and not just the top 10% of scores.

Moreover, we observe that charge off has quite a few Prosper Scores of 7 and up. This reason could, again, lie in the fact that this category is an overall collection of Charge Off information, containing false positives.

Borrower Rate vs Estimated Loss

Borrower Rate vs Estimated Loss Scatterplot



Notice as the borrower rate increase, the estimated loss percentage increases as well. One may believe that there is some association between borrower rate and estimated loss.

We utilize the following regression model to determine if there exists some significant connection between borrower rate and estimated loss. \$ EstimatedLoss = 0.4389099{(BorrowerRate)} + -0.0045159\$

```
loan_customer_perspective.mdl <- lm(EstimatedLoss ~ BorrowerRate,
                                     data = loan_customer_perspective)
summary(loan_customer_perspective.mdl)
```

```

## 
## Call:
## lm(formula = EstimatedLoss ~ BorrowerRate, data = loan_customer_perspective)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.14768 -0.01409 -0.00473  0.01119  0.33211 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.0045159  0.0004614 -9.788 <2e-16 ***
## BorrowerRate  0.4389099  0.0022271 197.074 <2e-16 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.03321 on 39998 degrees of freedom
## Multiple R-squared:  0.4926, Adjusted R-squared:  0.4926 
## F-statistic: 3.884e+04 on 1 and 39998 DF,  p-value: < 2.2e-16

```

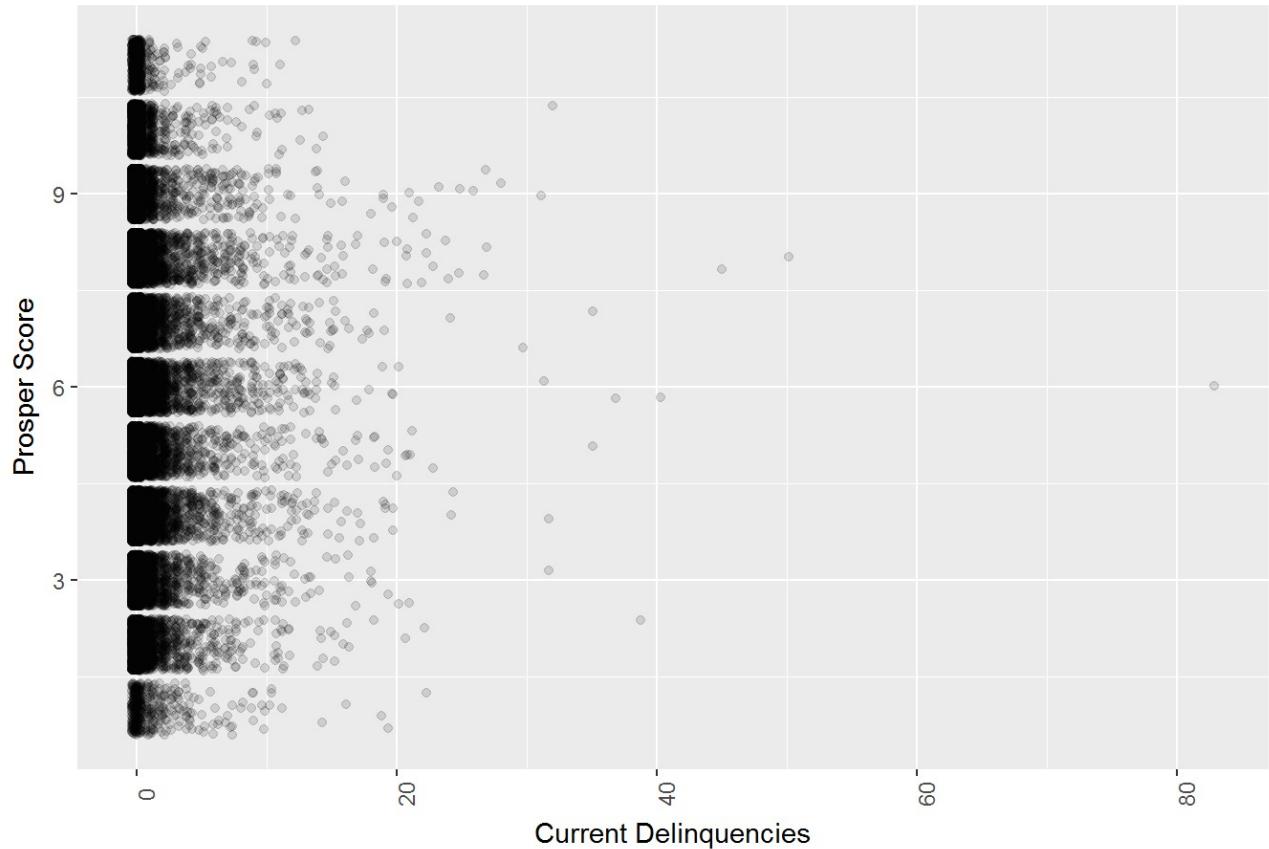
From the summary information from above, we determine that this model is significant under the assumption there is a 0.001% chance of the model being random.

Note: Let's assume some connection between two variables is due to chance, say 5%. If our variables were statistically significant, two stars would appear to the far right of the table of variables above. However, we see three stars. If we look at the Signif. codes, we see that three stars is associated with 0.001. Hence, we utilize 0.001 as our α value in this situation.

Additional Note: However, we should still be skeptical of our model. Though there is a high Adjusted R² value of .4926 and significance between the two variables, we need to ensure no bias, error, or even lurking variables affect this model.

Current Delinquencies vs Prosper Score

Current Delinquencies vs Prosper Score Scatterplot

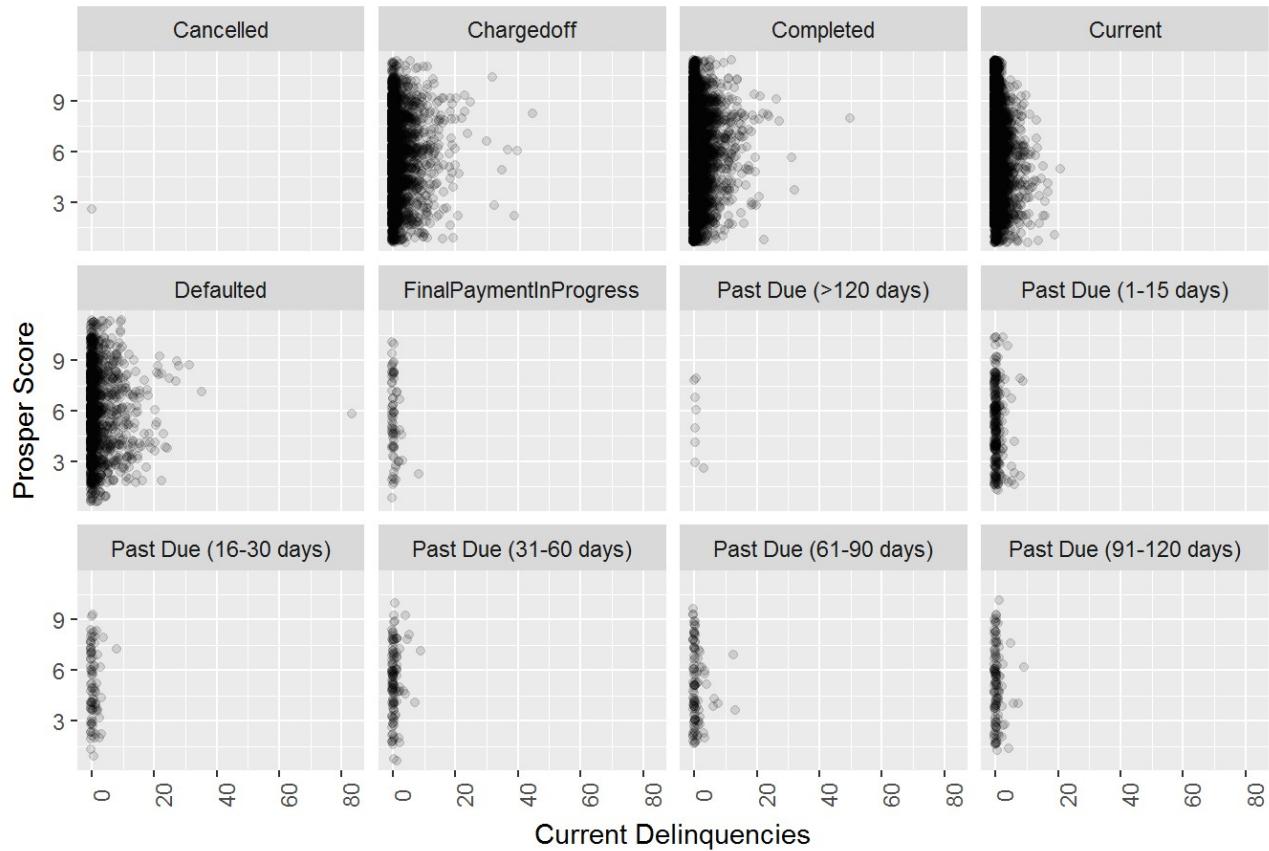


Between Current Delinquencies and Prosper Score, we do not see a connection. And this should be true. Delinquencies happen by chance, and not by some explicit finding/observation in our data. > Note: However, there may be a case where some connection between Current Delinquencies can be predicted, from associating it with the remaining 70+ variables in the original Prosper Loan dataset.

Current Delinquencies vs Prosper Score by Loan Status

However, let's determine some association by factoring in the Loan Status feature. Below is a similar graph from the previous visualization. But, the only difference is factoring in the Prosper customer loan status.

Current Delinquencies vs Prosper Score by Loan Status Scatterplot



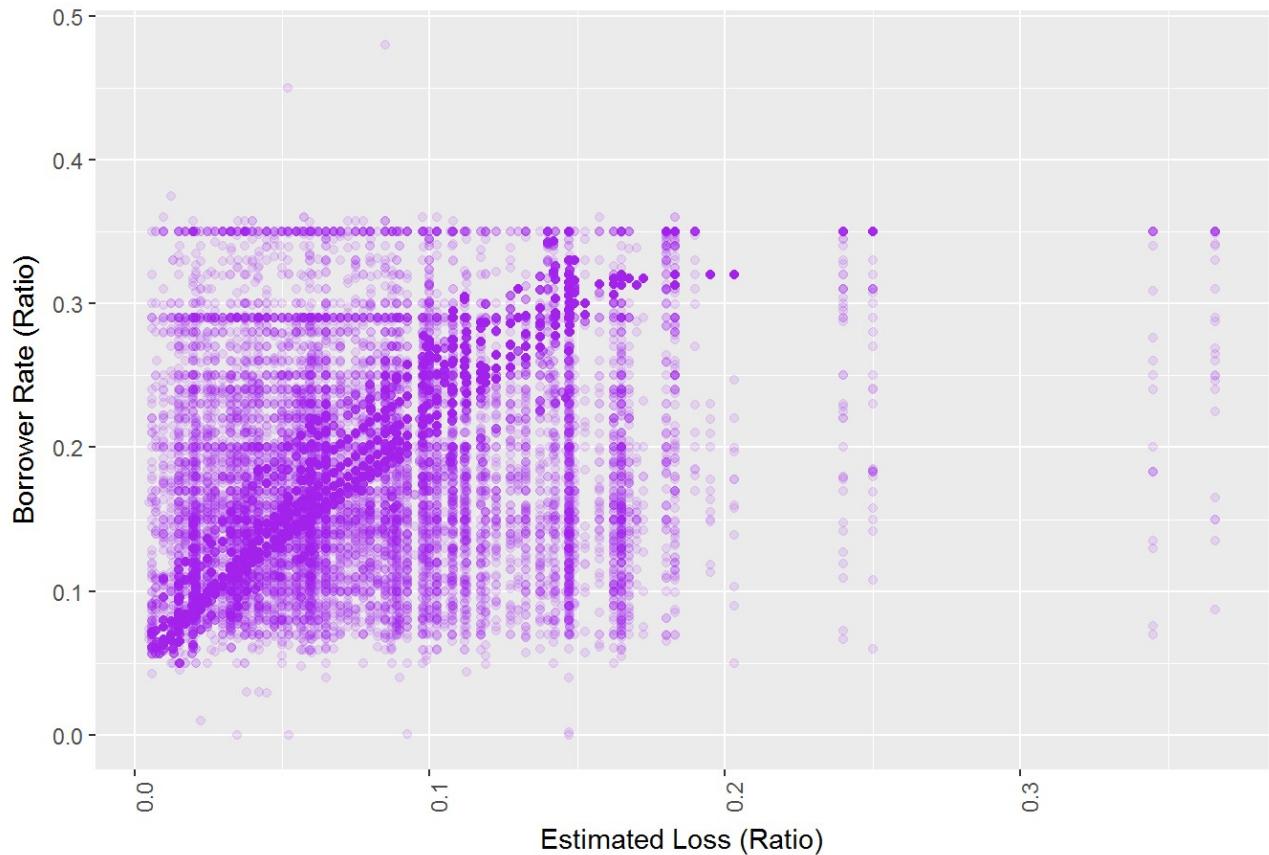
There appears to be no significant findings from considering another variable. I.e., The above visualization is just a similar understanding to that of the original example.

It's good to know that we tried, right?

Estimated Loss vs Borrower Rate Colored by Prosper Score

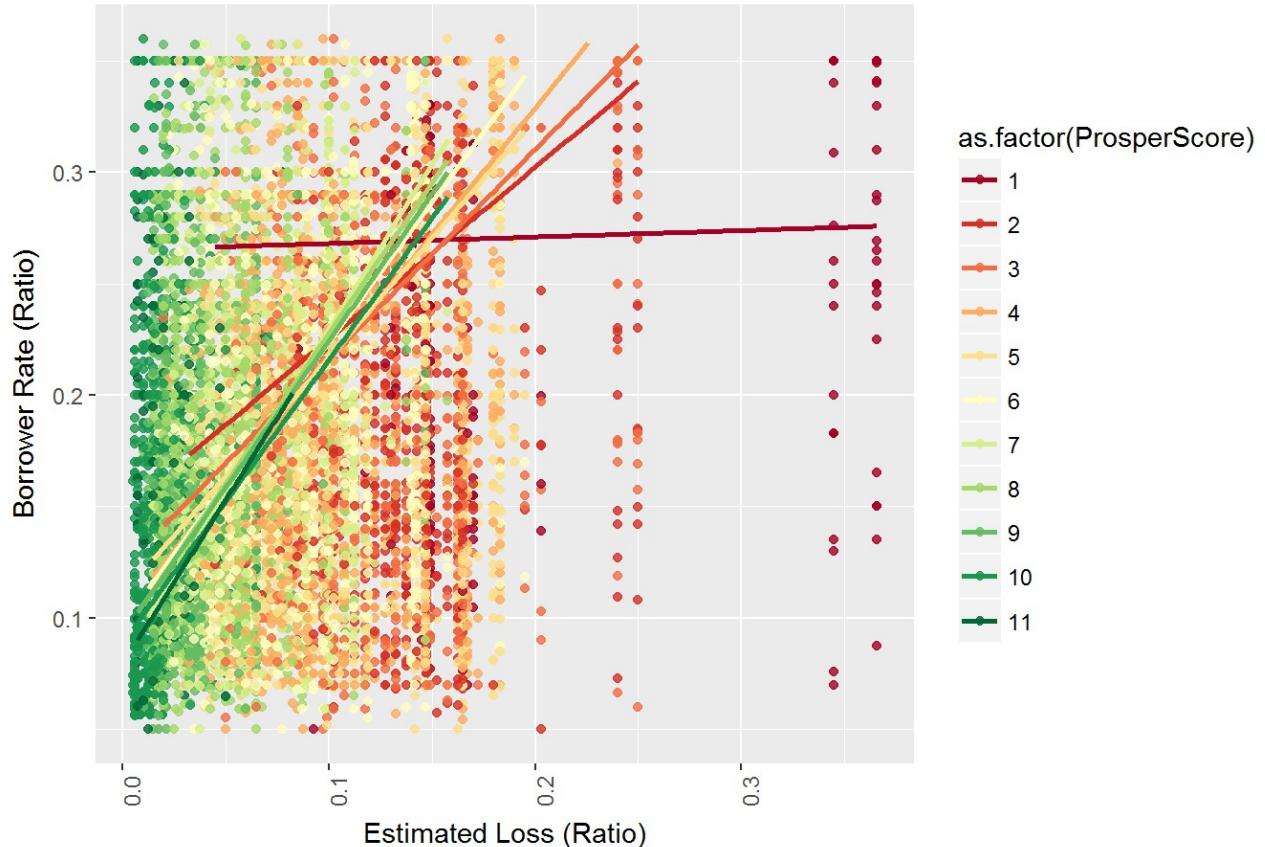
Now we switch things up. We consider the estimated loss vs borrower rate visualization from before. The following is a reminder of the scatterplot:

Estimated Loss vs Borrower Rate Scatterplot



This time, we factor in Prosper Score in our visualization, like so:

Estimated Loss vs Borrower Rate Colored by Prosper Score Scatterplot



Again, notice as the borrower rate increase, the estimated loss percentage increases as well. This time, as we increase our estimated loss, meanwhile observe borrower rate increasing too, we see that Prosper scores tend to decrease as the degree of estimated loss occurs.

This is trivial. In hindsight, we categorize people that negatively effect us to be “bad” people; it’s human nature.

To this opinion, we extend the previous linear regression from before by factoring in Prosper Score.

$$\$ \text{EstimatedLoss} = 0.3076\{\text{(BorrowerRate)}\} - 0.008560(\text{ProsperScore}) + 0.07183\$$$

```
loan_customer_perspective.md2 <- lm(EstimatedLoss ~ BorrowerRate + ProsperScore,
                                     data = loan_customer_perspective)
summary(loan_customer_perspective.md2)
```

```

## 
## Call:
## lm(formula = EstimatedLoss ~ BorrowerRate + ProsperScore, data = loan_customer_perspective)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.11747 -0.01607 -0.00202  0.01236  0.27581 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 7.183e-02 7.180e-04 100.0   <2e-16 ***
## BorrowerRate 3.076e-01 2.148e-03 143.2   <2e-16 ***
## ProsperScore -8.560e-03 6.761e-05 -126.6   <2e-16 ***  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.02806 on 39997 degrees of freedom
## Multiple R-squared:  0.6378, Adjusted R-squared:  0.6378 
## F-statistic: 3.522e+04 on 2 and 39997 DF, p-value: < 2.2e-16

```

From the summary information from above, we determine that this model is significant under the assumption there is a 0.001% chance of the model being random.

Note: Let's assume some connection between two variables is due to chance, say 5%. If our variables were statistically significant, two stars would appear to the far right of the table of variables above. However, we see three stars. If we look at the Signif. codes, we see that three stars is associated with 0.001. Hence, we utilize 0.001 as our α value in this situation.

Prosper Investor Analysis

** Motivation ** Now, we look into the Prosper Investor perspective. Usually Investors invest in some type of portfolio of Prosper customer loans to seek investment return.

With modern day data-driven companies like Prosper, we suspect that the investor acknowledges some default rate in their investment, Prosper service fees taken out, and some return is made as well.

However, what information can we extract from the Prosper data “df” such that an investor knows what financial factors to consider in his/her investment.

Note: The above conclusion can be made from such financial institutions like Lending Club (<https://www.lendingclub.com/investing/alternative-assets/why-lending-club>)

Again, the Prosper investor subset “loan_investor_perspective” will contain the following variables:

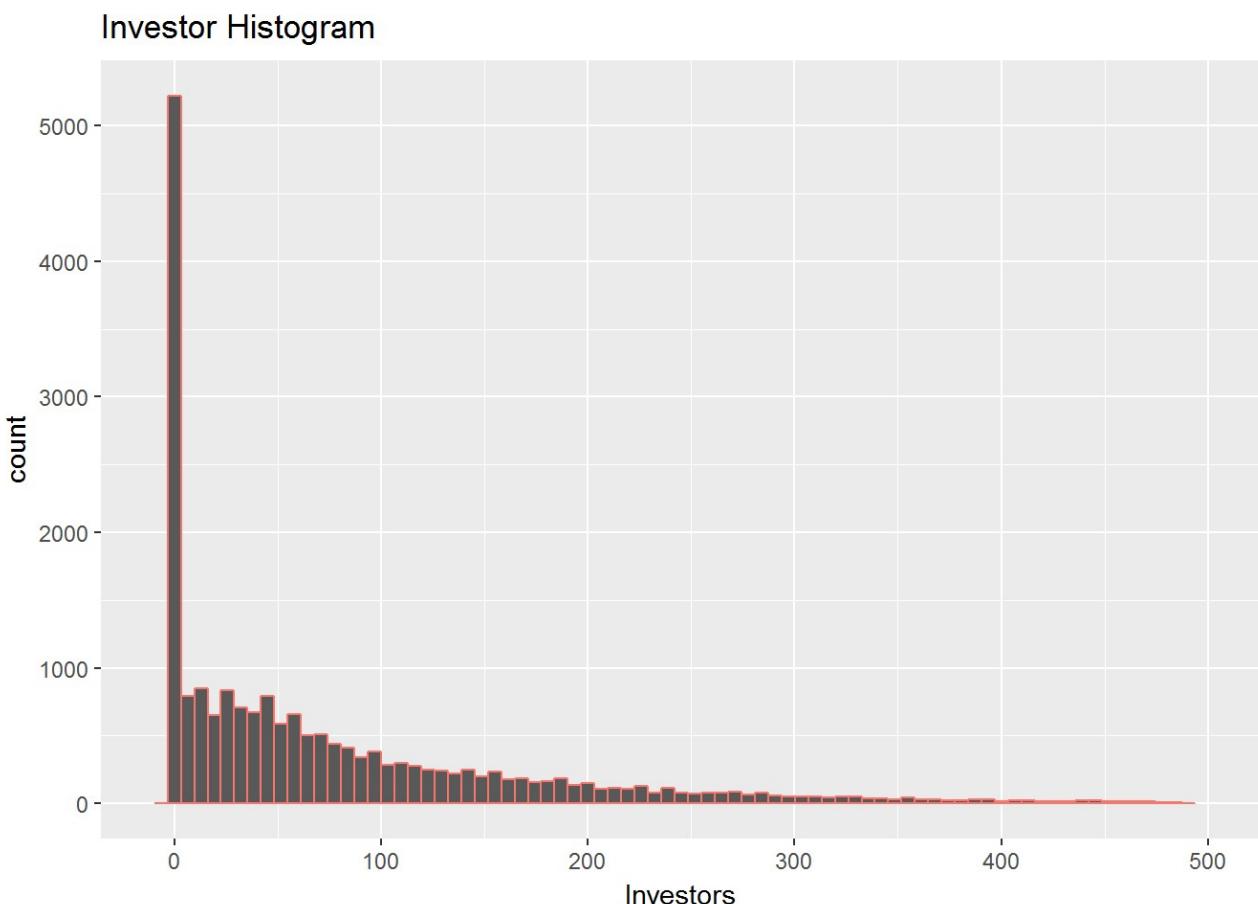
1. Employment Status
2. Bank Card Utilization
3. Investors
4. LP_Collection Fees
5. Prosper Score
6. IsBorrowerHomeowner
7. CurrentDelinquencies

```
names(loan_investor_perspective)
```

```
## [1] "ProsperScore"           "EmploymentStatus"      "IsBorrowerHomeowner"  
## [4] "BankcardUtilization"    "Investors"            "LP_CollectionFees"  
## [7] "CurrentDelinquencies"
```

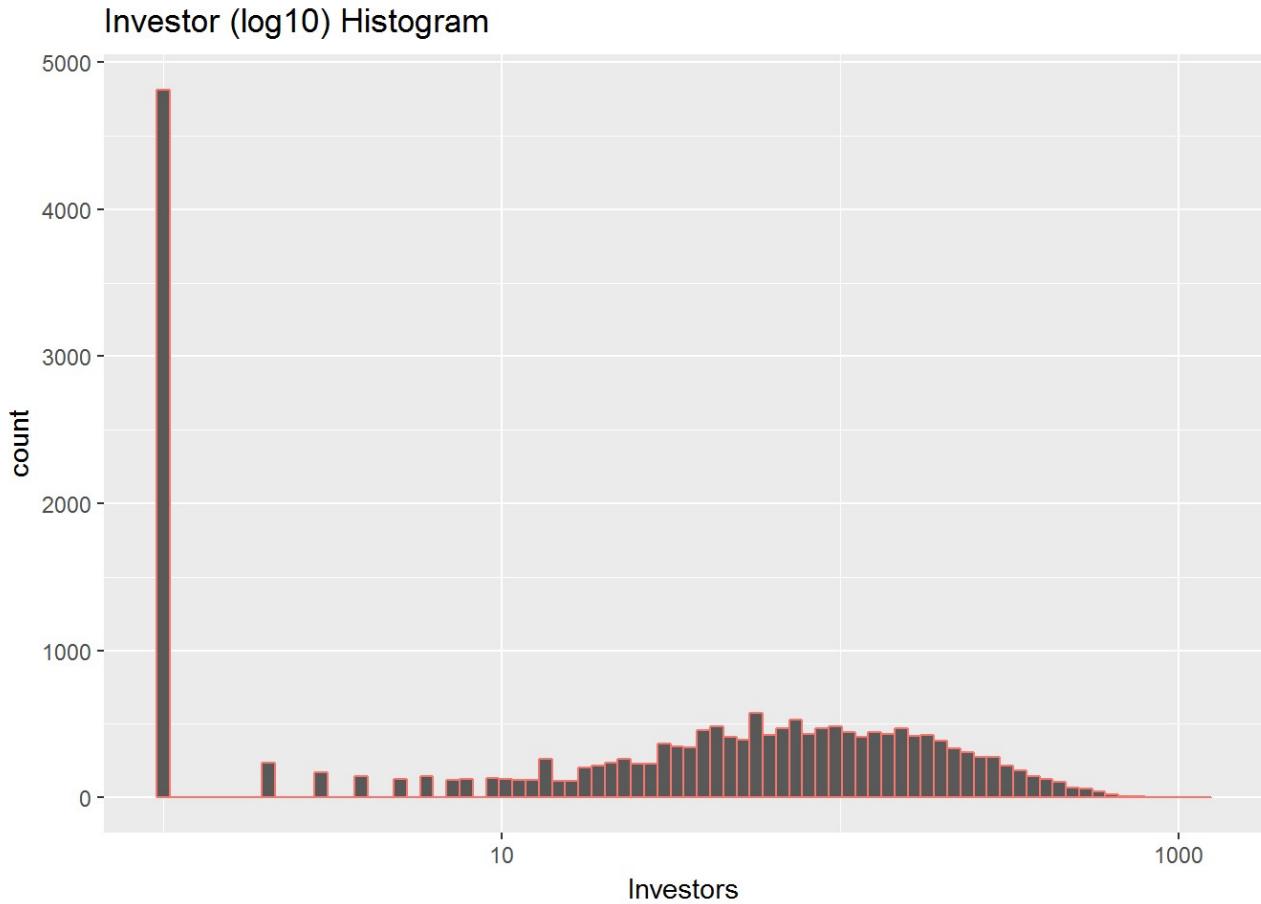
Histograms

Investors



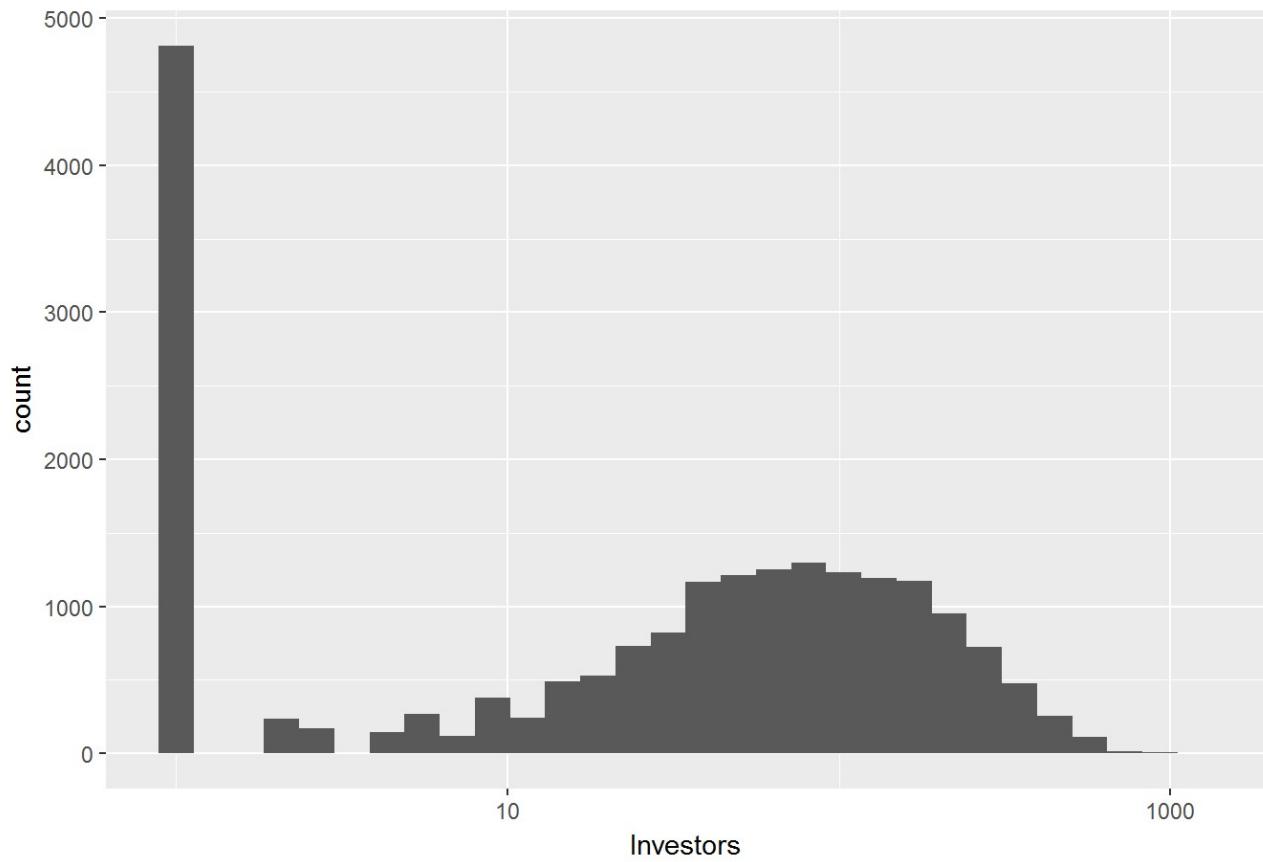
Most of the investor contributions to a Prosper loan lie near 100. The case for this is empty values "" being entered into our data. If we were to observe this in full, we might anticipate a more normalized histogram.

Taking this to a different scale, specifically the log scale, we observe the following:



We confirm the huge 0 value spiked of investors from above. Disregarding this observation, we also observe that there is a normal count of investors near the middle region of the range 10-1000, as seen below

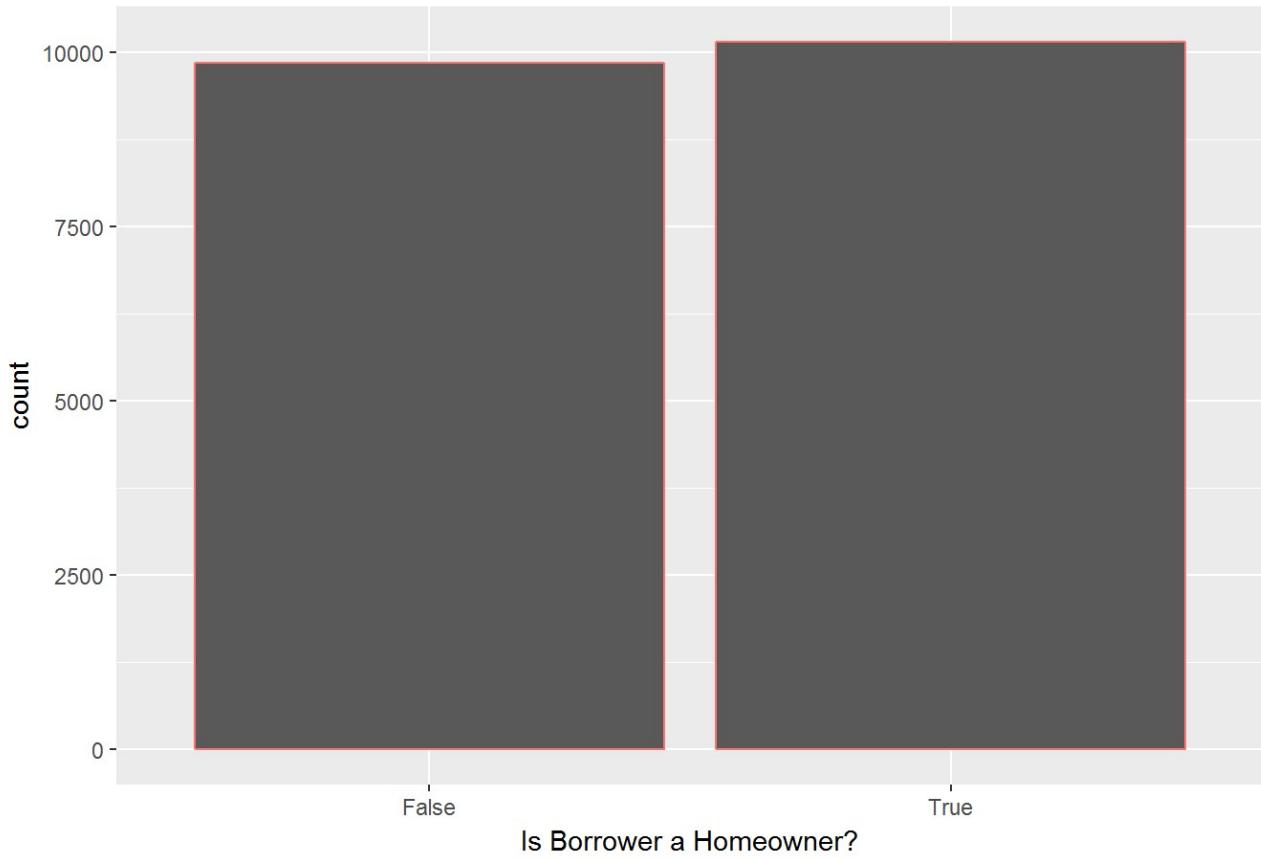
Investor (log10) Histogram



Is Borrower a Homeowner?

Typically homeowners have some pre-existing loan with a financial institution. Is there a case for Prosper customers where investors would favor non-homeowners or homeowner?

Borrower Homeowner? Histogram

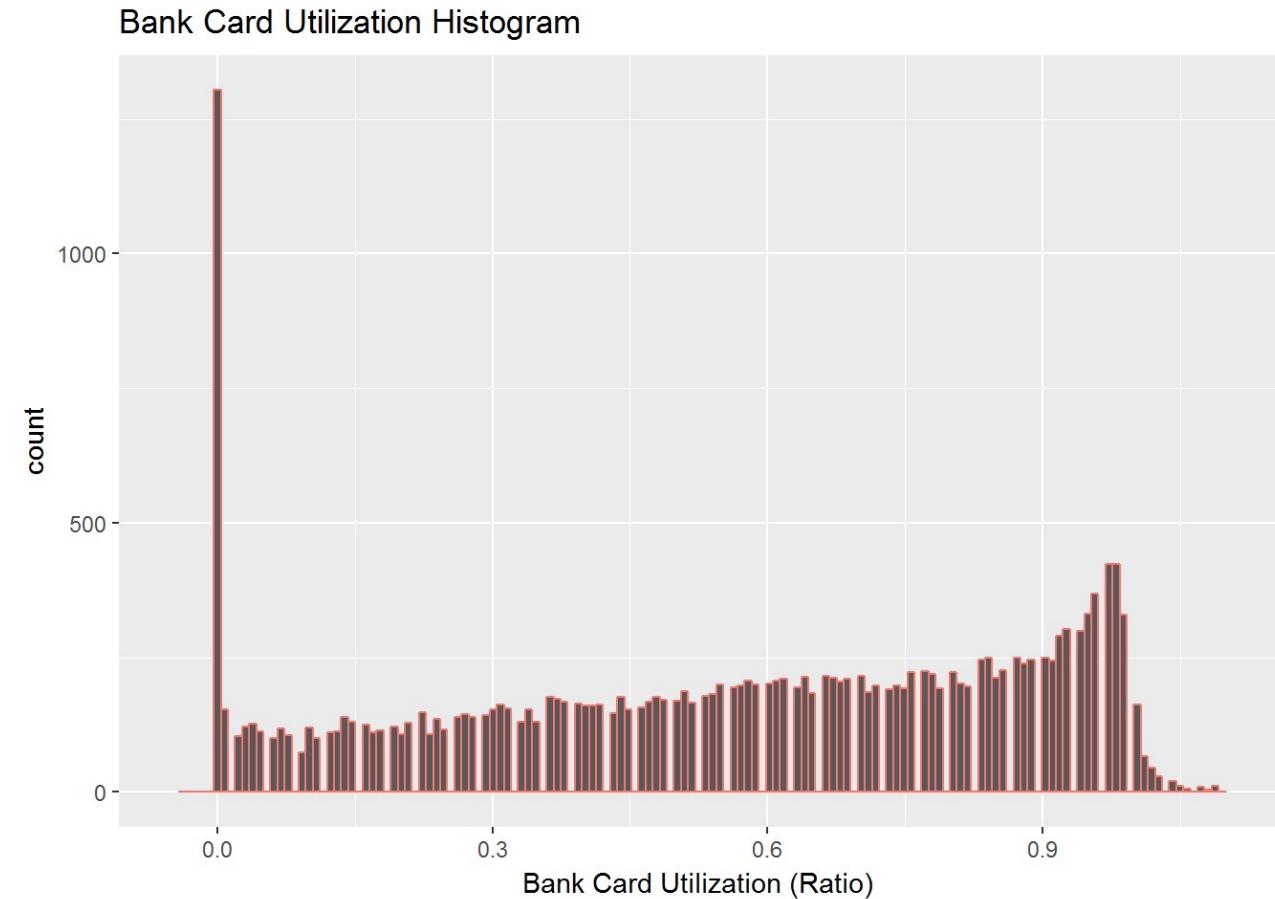


We observe there is no bias towards being or not being a homeowner, when attempting to borrow from Prosper. As a customer, this is good.

Moreover, we observe that Prosper investors are investing into more randomized packages.

Note: Of course, due to discrimination regulations in financial institutions, we should suspect that there exists no bias in this case. Maybe if the regulations were not imposed, would we see the opposite scenario?

Bank Card Utilization



We observe a large amount of non-bank card utilizations records. That is alarming. However, maybe a borrower does not have a bank card, and they are simply borrowing with the intent of not opening a card or even not being permitted to hold a card.

If Prosper has a mission to get into the riskier market of underprivileged person under financial constraints, we can conclude why there exists a large amount of non-bank card users.

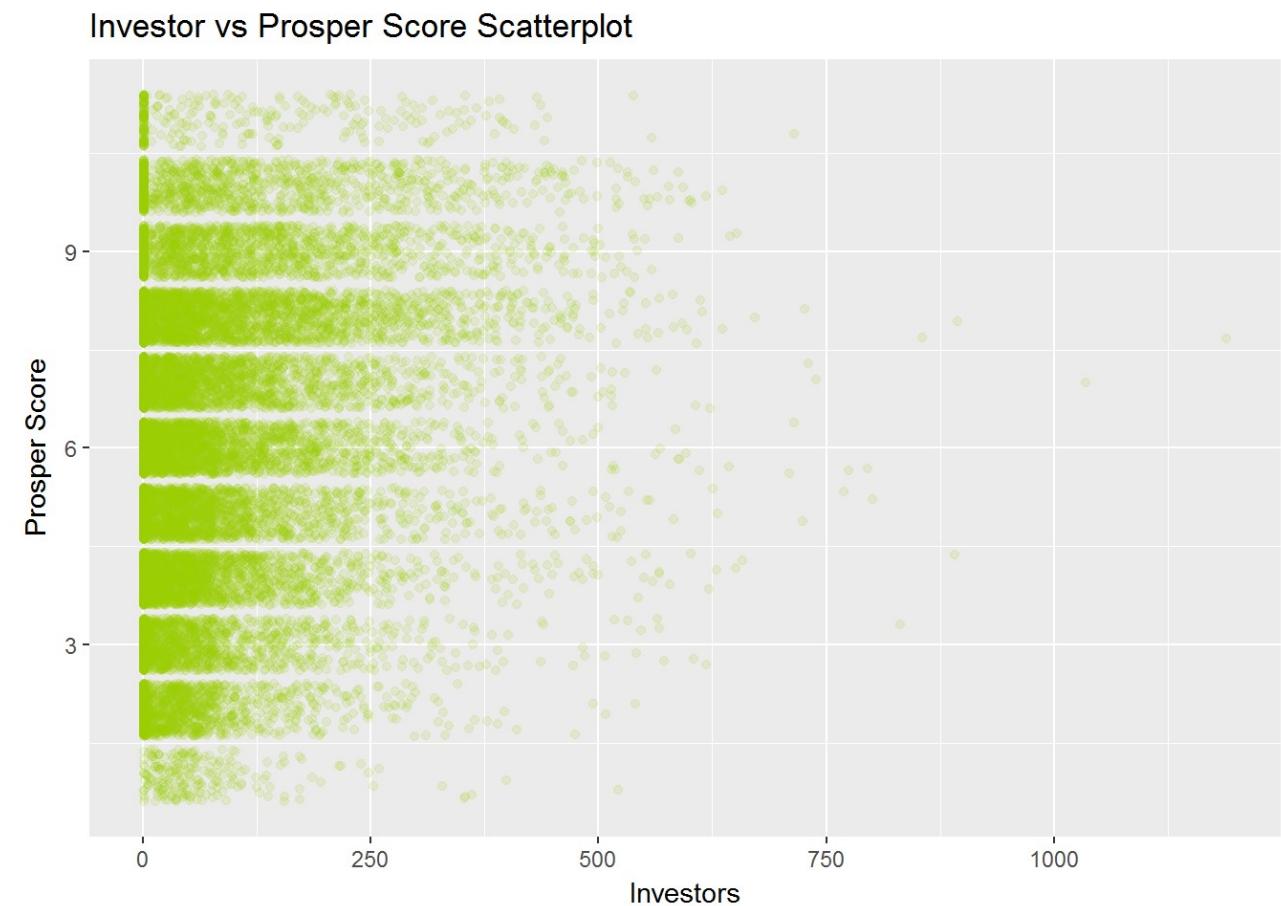
Another observation is just the ever so climbing bank card utilization as we look along the x-axis.

Lastly, noticed the spikes in our observation. We first thought some cyclic pattern occurred from some unknown variable or just assume "seasonality." However, let's consider another thought.

It could be the case that reports from some financial institutions realease customer information in a more friendly manner. I.e., maybe the input of customer bank card utilization is more on a list of options compared to typing in the full bank card utilization rate.

Scatterplots

Investors vs Prosper Score



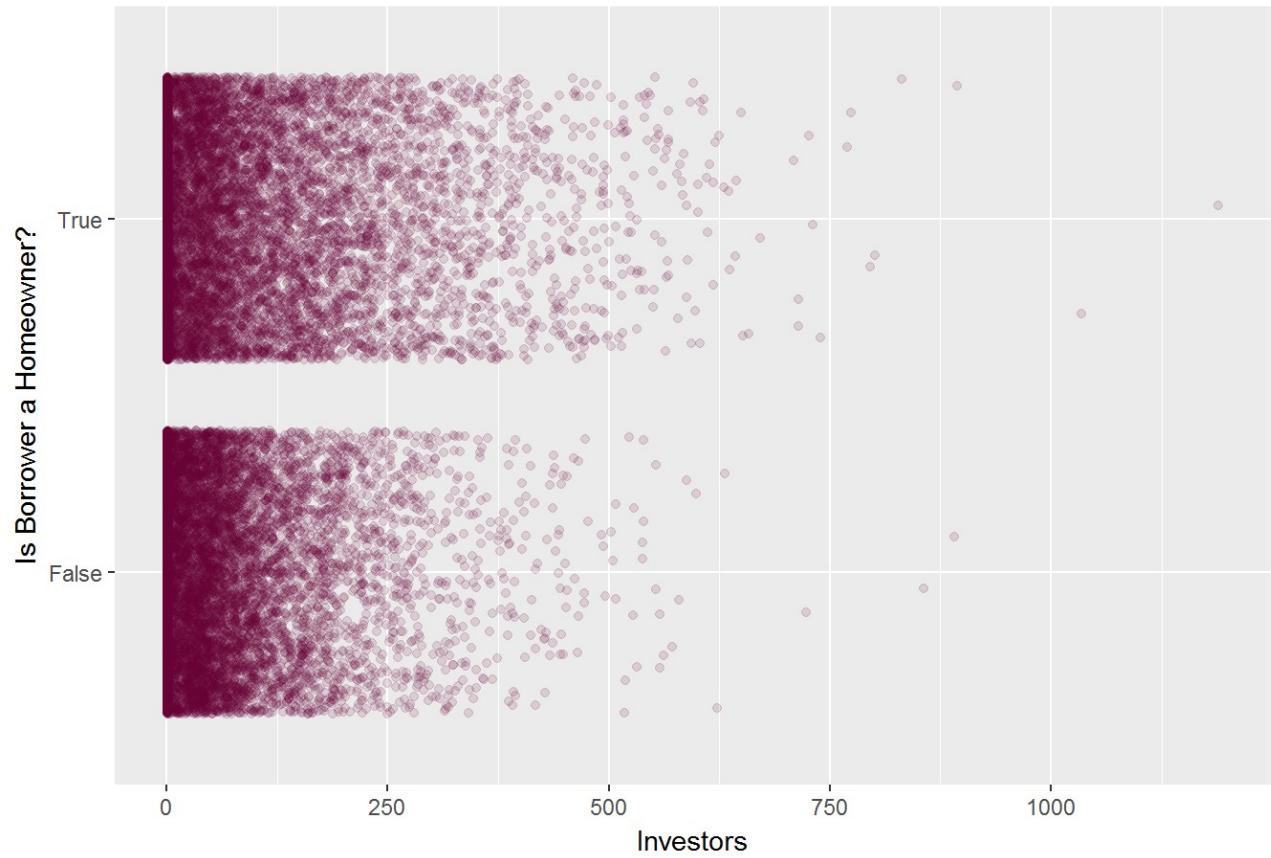
Again, we observe a normal distribution among Prosper investors and what they invest in. I.e., we are noticing that investors are no selecting loans that guarantee success.

Let's look at another Investor observation.

Investors vs Is Borrower a Homeowner

If I was an investor, am I able decipher and choose loans based on if a borrower is a homeowner?

Investor vs Is Borrower a Homeowner Scatterplot

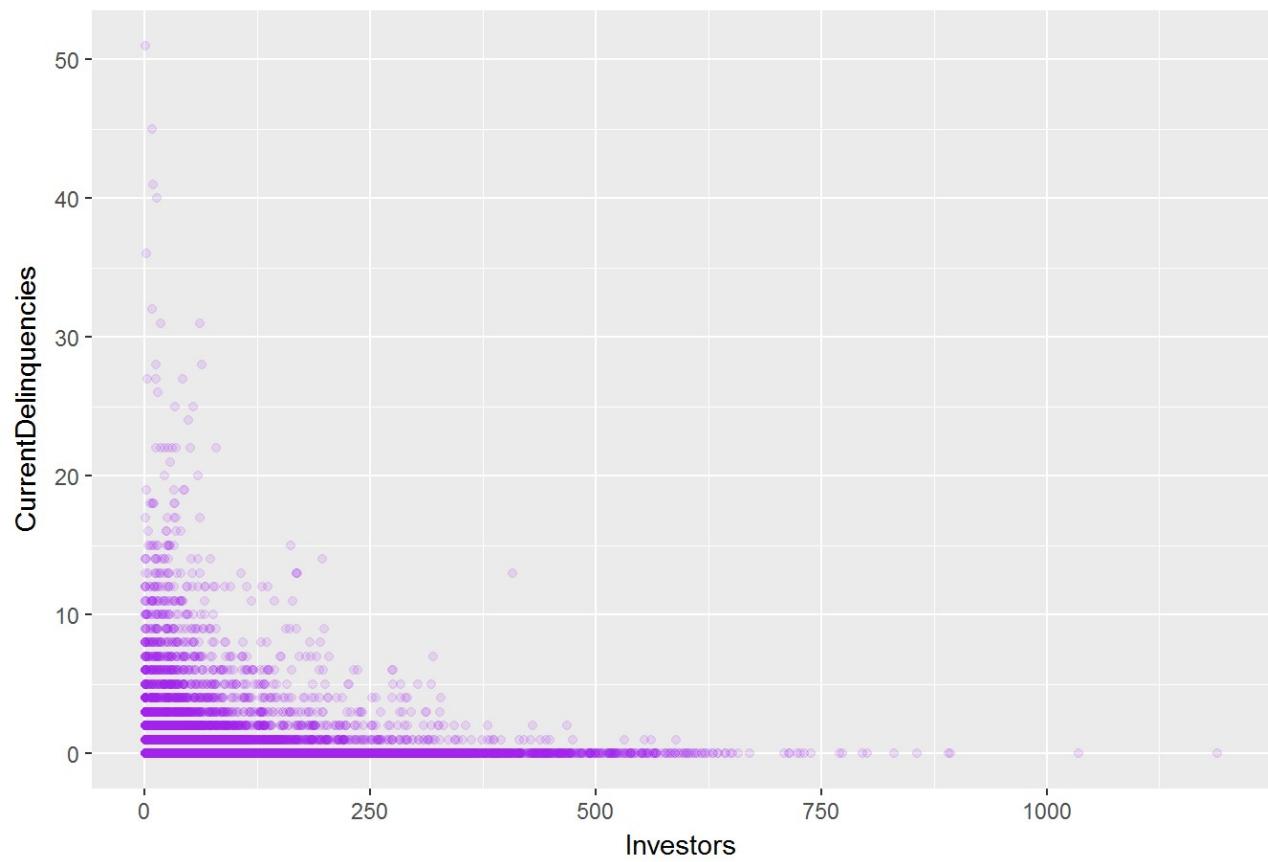


Again, we observe an investor does not choose borrower loans. There is no significant finding that an investor decisively invests into borrowers based on being a homeowner.

Investors vs Current Delinquencies

What is the case if an investor wants to avoid borrowers with delinquencies?

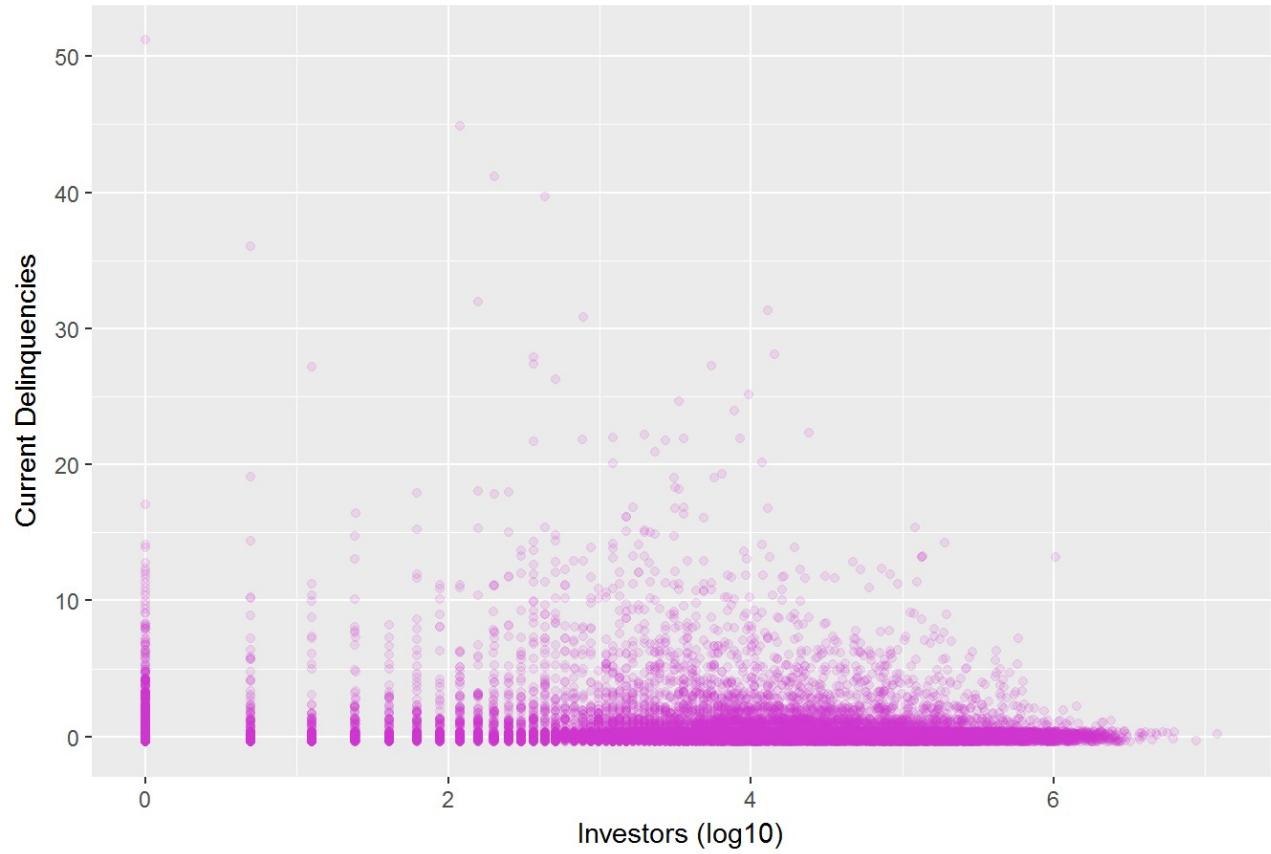
Investor vs Current Delinquencies Scatterplot



We observe that most investors end up investing into Prosper loans with some type of delinquency. We re-scale the visualization before concluding that investors, based on the graph, prefer Prosper loans with current delinquencies.

Investors (log10) vs Current Delinquencies

Investor (log10) vs Current Delinquencies Scatterplot

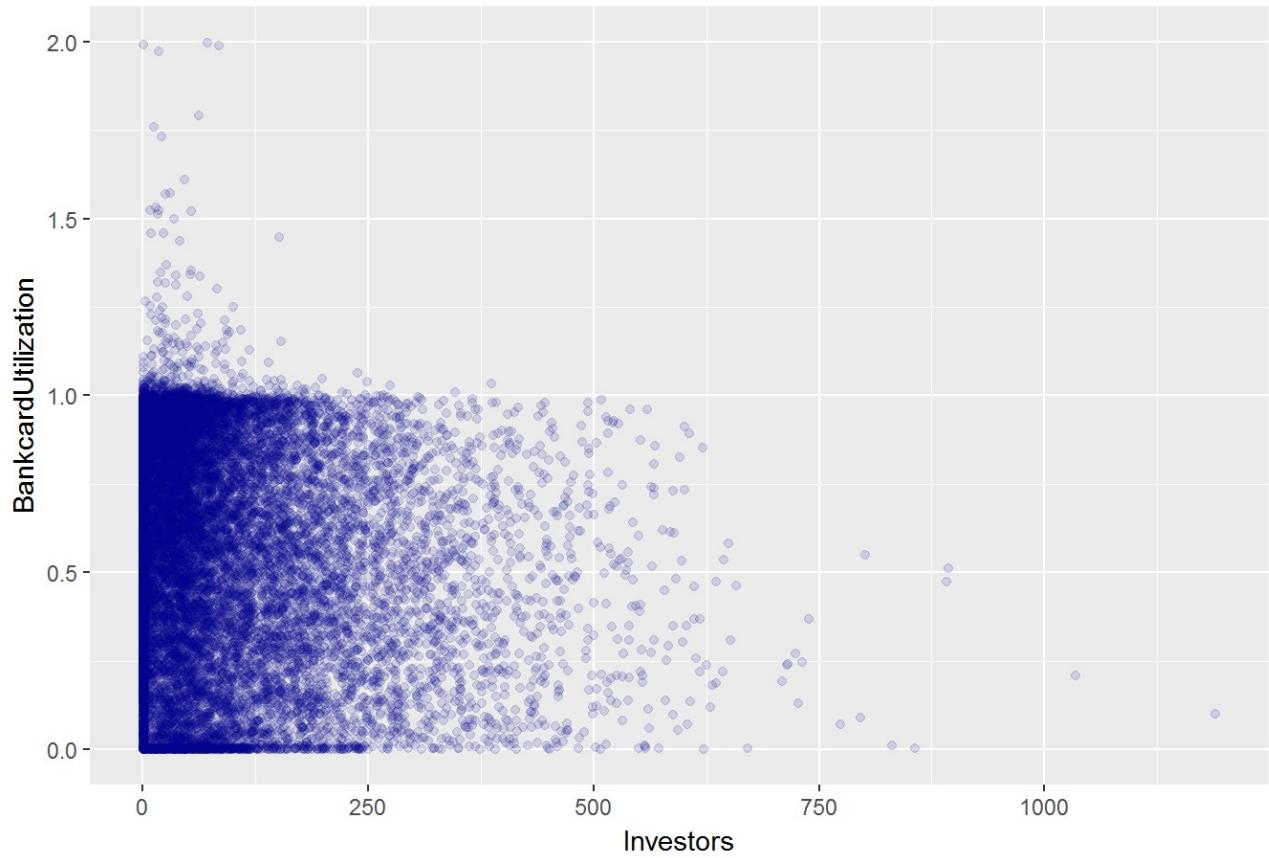


We observe that most investor contributions do not have a significant outcome of low delinquency situations. Of course, there is a more likely possibility investing into Prosper loans, and having no delinquent borrowers in return.

Investors vs Bank Card Utilization

We are interested in seeing if investors would be more inclined to investing into loans where a borrower has bank card usage.

Investor vs Bank Card Utilization Scatterplot



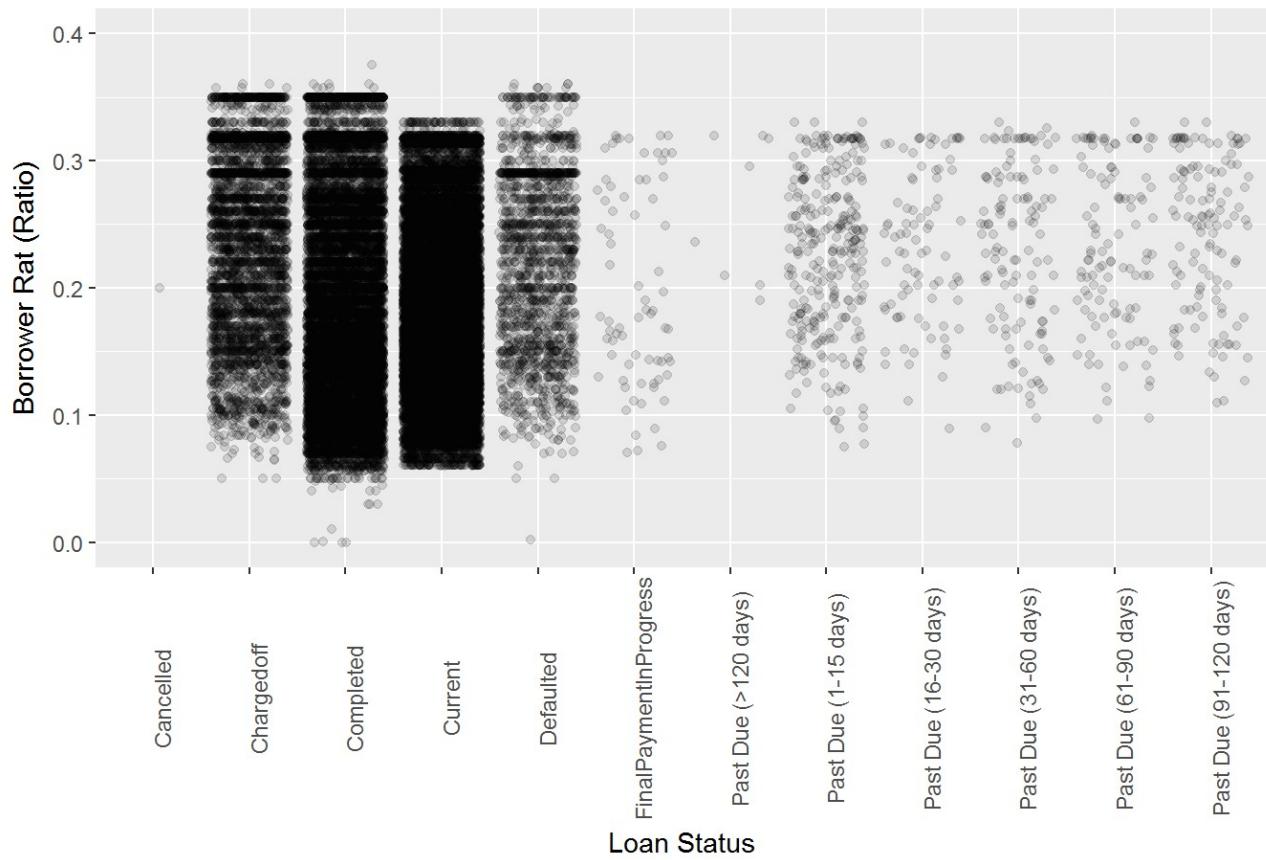
We observe some rectangular cluster in between 0 and 1. Thereafter, there just seem to be several outliers. This 0-1 range is the percentage of usage for borrowers. Because there is no visual pattern being displayed, we observe that, again, investors are not strategically investing.

Summary

We observe from the overall perspective that there is

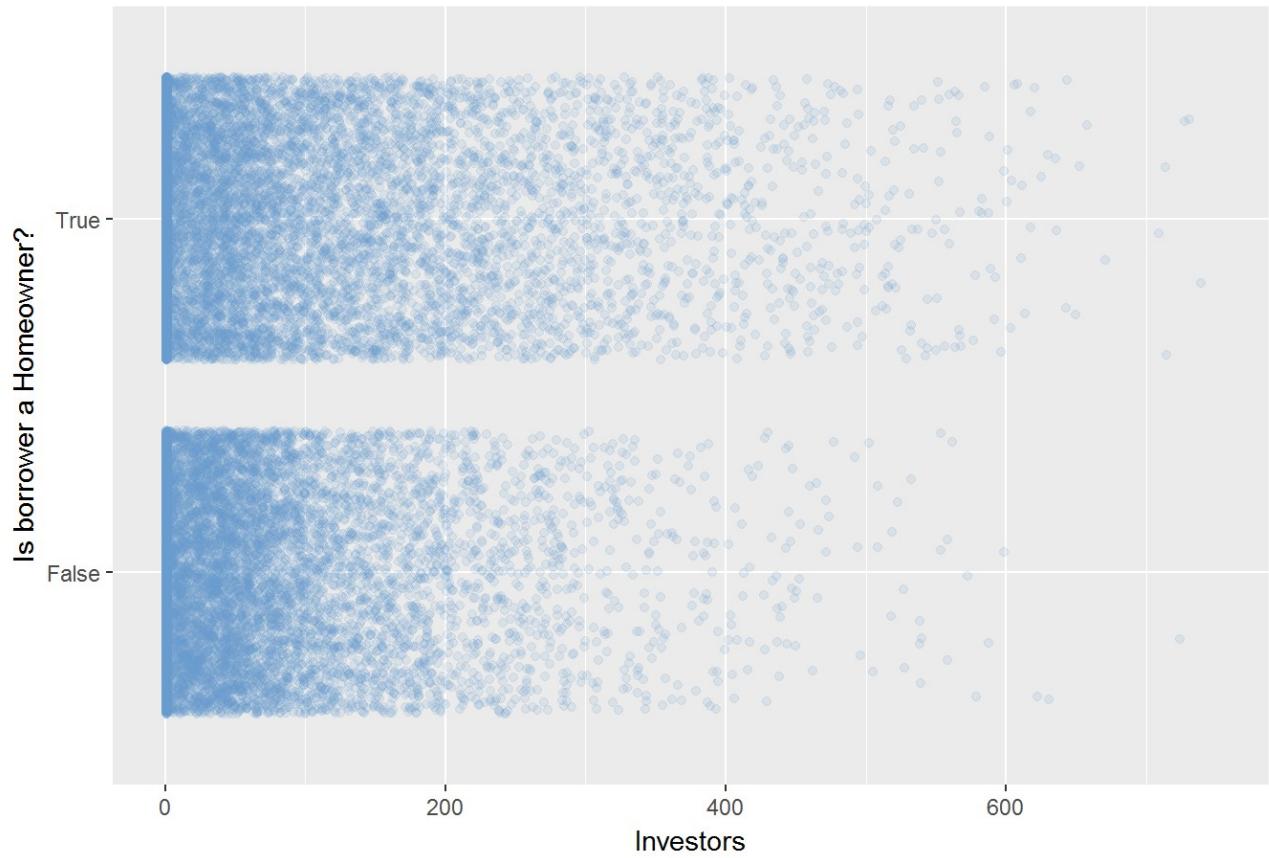
no concrete leaning to avoid bad Prosper loans. As a Prosper borrower, we see from the following image that there exists investors investing in delinquent loans. As an investor, they can now observe they are undergoing risk in investing into loans.

Loan Status vs Borrower Rate Scatterplot



Even in the case of Prosper Score and being a homeowner, we observe no investor bias towards some spectrum, as seen below

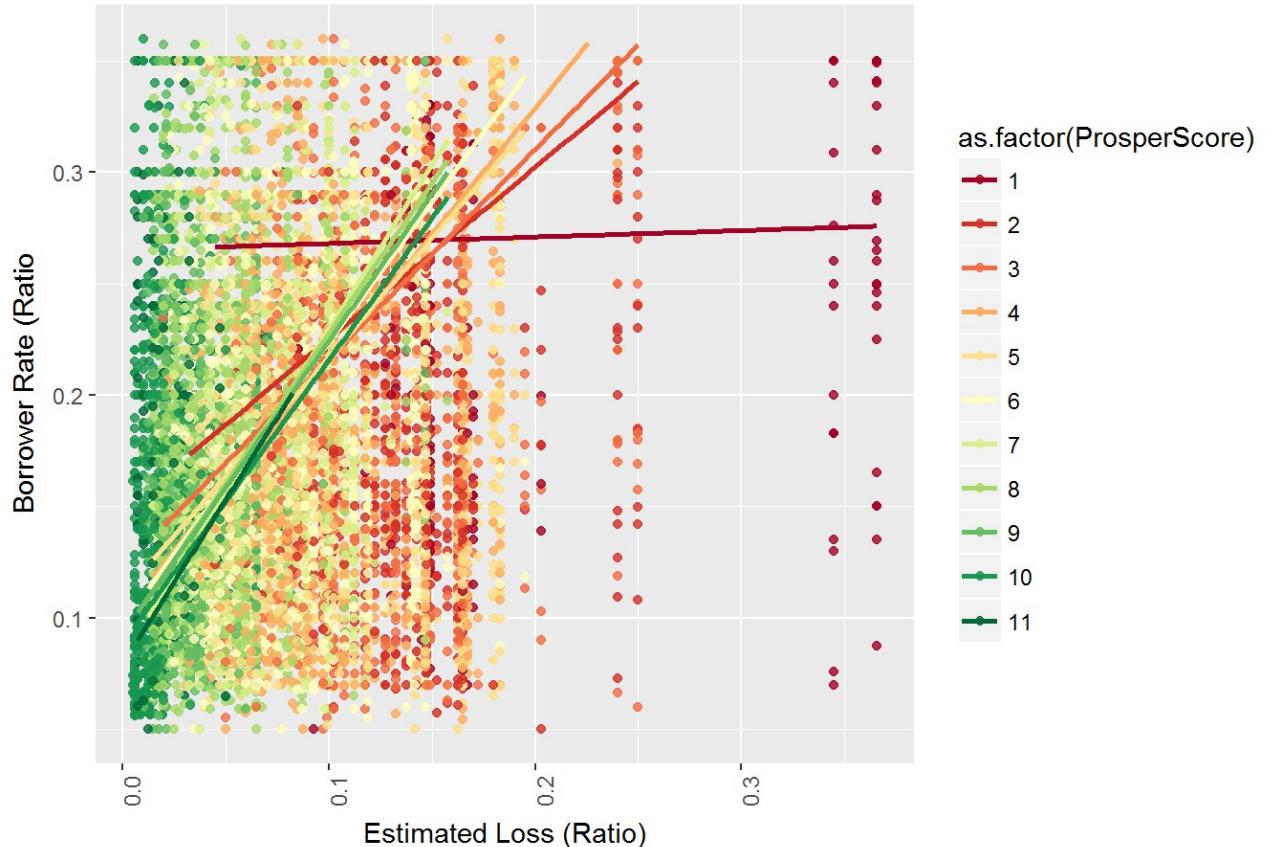
Investor vs Is Borrower a Homeowner Scatterplot



Out of curiosity, is there at least some abnormal pattern within Prosper lending operations?

The below graph is an indication that there exists no abnormal pattern. Moreover, we see that there is a pattern between estimated loss and borrower rate. Of course, that would be suspected in Prosper operations. We would like to track the bad accounts/borrowers. However, Prosper does not suspiciously categorize individuals from other discriminative occurs.

Estimated Loss vs Borrower Rate Colored by Prosper Score Scatterplot



Meanwhile observe borrower rate increasing too, we see that Prosper scores tend to decrease as the degree of estimated loss occurs.

This is trivial. In hindsight, we categorize people that negatively effect us to be “bad” people; it’s human nature. We categorize persons on some scale of their consequential patterns against the Prosper lending operations.

Conclusion

We conclude that there is no abnormal pattern between Prosper borrowers and Prosper operations. If I was a potential Prosper customer, I would like to take out a loan given the fact of the lack of bias in what type of customers they want.

As an investor, we saw that they invest randomly. That is, the investor does not choose what type of loans they invest in. And that is normal.

If you saw the **Lending Club** video earlier, you would see that investors agree to invest to some package of borrower. This package has calculated risk of 10%. Of course, a financial institution abides to that with specialized insights through data.

So, we observe an investor chooses to invest, but not which borrower loans to invest in.

The Prosper operations work according to banking regulations. Moreover, we observe classification and statistical findings about how to be a good Prosper customer and how the Prosper operations benefit prospective investors.

Reflections

During this process, I encountered quite a few thoughts in regards how I can improve either the documentation or analysis.

First, the difficulties I initially ran into the analysis was computing power, a gargantuan amount of variables, and just logistics of resolving empty entries. Though I ran into these issues, I feel I succeeded in resolving NA values, partitioning the data for sake of computing power, and just performing more than just a data visualization analysis.

This analysis could be enriched with more statistical findings and just proper resolution of filling in blank-spaced entries. Moreover, the addition to feature engineering would have been beneficial in the initial process.

Ending some of my thoughts, I feel I could have:

1. Not induce entries into NA value entries so quickly
2. Exploration into why there were 16 null values remaining, after our data-filling process
3. Subsetted the original “df” data in a more strategic manner

Resources

Data Dictionary

1. **ListingKey:** Unique key for each listing, same value as the ‘key’ used in the listing object in the API.
2. **ListingNumber:** The number that uniquely identifies the listing to the public as displayed on the website.
3. **ListingCreationDate:** The date the listing was created.
4. **CreditGrade:** The Credit rating that was assigned at the time the listing went live. Applicable for listings pre-2009 period and will only be populated for those listings.
5. **Term:** The length of the loan expressed in months.
6. **LoanStatus:** The current status of the loan: Cancelled, Chargedoff, Completed, Current, Defaulted, FinalPaymentInProgress, PastDue. The PastDue status will be accompanied by a delinquency bucket.
7. **ClosedDate:** Closed date is applicable for Cancelled, Completed, Chargedoff and Defaulted loan statuses.
8. **BorrowerAPR:** The Borrower’s Annual Percentage Rate (APR) for the loan.

9. **BorrowerRate:** The Borrower's interest rate for this loan.
10. **LenderYield:** The Lender yield on the loan. Lender yield is equal to the interest rate on the loan less the servicing fee.
11. **EstimatedEffectiveYield:** Effective yield is equal to the borrower interest rate
 - minus the servicing fee rate
 - minus estimated uncollected interest on charge-offs
 - plus estimated collected late fees. Applicable for loans originated after July 2009.
12. **EstimatedLoss:** Estimated loss is the estimated principal loss on charge-offs. Applicable for loans originated after July 2009.
13. **EstimatedReturn:** The estimated return assigned to the listing at the time it was created. Estimated return is the difference between the Estimated Effective Yield and the Estimated Loss Rate. Applicable for loans originated after July 2009.
14. **ProsperRating (numeric):** The Prosper Rating assigned at the time the listing was created: 0 - N/A, 1 - HR, 2 - E, 3 - D, 4 - C, 5 - B, 6 - A, 7 - AA. Applicable for loans originated after July 2009.
15. **ProsperRating (Alpha):** The Prosper Rating assigned at the time the listing was created between AA - HR. Applicable for loans originated after July 2009.
16. **ProsperScore:** A custom risk score built using historical Prosper data. The score ranges from 1-10, with 10 being the best, or lowest risk score. Applicable for loans originated after July 2009.
17. **ListingCategory:** The category of the listing that the borrower selected when posting their listing: 0 - Not Available, 1 - Debt Consolidation, 2 - Home Improvement, 3 - Business, 4 - Personal Loan, 5 - Student Use, 6 - Auto, 7- Other, 8 - Baby&Adoption, 9 - Boat, 10 - Cosmetic Procedure, 11 - Engagement Ring, 12 - Green Loans, 13 - Household Expenses, 14 - Large Purchases, 15 - Medical/Dental, 16 - Motorcycle, 17 - RV, 18 - Taxes, 19 - Vacation, 20 - Wedding Loans
18. **BorrowerState:** The two letter abbreviation of the state of the address of the borrower at the time the Listing was created.
19. **Occupation:** The Occupation selected by the Borrower at the time they created the listing.
20. **EmploymentStatus:** The employment status of the borrower at the time they posted the listing.
21. **EmploymentStatusDuration:** The length in months of the employment status at the time the listing was created.
22. **IsBorrowerHomeowner:** A Borrower will be classified as a homeowner if they have a mortgage on their credit profile or provide documentation confirming they are a homeowner.
23. **CurrentlyInGroup:** Specifies whether or not the Borrower was in a group at the time the listing was created.
24. **GroupKey:** The Key of the group in which the Borrower is a member of. Value will be null if the borrower does not have a group affiliation.
25. **DateCreditPulled:** The date the credit profile was pulled.

26. **CreditScoreRangeLower:** The lower value representing the range of the borrower's credit score as provided by a consumer credit rating agency.
27. **CreditScoreRangeUpper:** The upper value representing the range of the borrower's credit score as provided by a consumer credit rating agency.
28. **FirstRecordedCreditLine:** The date the first credit line was opened.
29. **CurrentCreditLines:** Number of current credit lines at the time the credit profile was pulled.
30. **OpenCreditLines:** Number of open credit lines at the time the credit profile was pulled.
31. **TotalCreditLinespast7years:** Number of credit lines in the past seven years at the time the credit profile was pulled.
32. **OpenRevolvingAccounts:** Number of open revolving accounts at the time the credit profile was pulled.
33. **OpenRevolvingMonthlyPayment:** Monthly payment on revolving accounts at the time the credit profile was pulled.
34. **InquiriesLast6Months:** Number of inquiries in the past six months at the time the credit profile was pulled.
35. **TotalInquiries:** Total number of inquiries at the time the credit profile was pulled.
36. **CurrentDelinquencies:** Number of accounts delinquent at the time the credit profile was pulled.
37. **AmountDelinquent:** Dollars delinquent at the time the credit profile was pulled.
38. **DelinquenciesLast7Years:** Number of delinquencies in the past 7 years at the time the credit profile was pulled.
39. **PublicRecordsLast10Years:** Number of public records in the past 10 years at the time the credit profile was pulled.
40. **PublicRecordsLast12Months:** Number of public records in the past 12 months at the time the credit profile was pulled.
41. **RevolvingCreditBalance:** Dollars of revolving credit at the time the credit profile was pulled.
42. **BankcardUtilization:** The percentage of available revolving credit that is utilized at the time the credit profile was pulled.
43. **AvailableBankcardCredit:** The total available credit via bank card at the time the credit profile was pulled.
44. **TotalTrades:** Number of trade lines ever opened at the time the credit profile was pulled.
45. **TradesNeverDelinquent:** Number of trades that have never been delinquent at the time the credit profile was pulled.
46. **TradesOpenedLast6Months:** Number of trades opened in the last 6 months at the time the credit profile was pulled.

47. **DebtToIncomeRatio:** The debt to income ratio of the borrower at the time the credit profile was pulled. This value is Null if the debt to income ratio is not available. This value is capped at 10.01 (any debt to income ratio larger than 1000% will be returned as 1001%).
48. **IncomeRange:** The income range of the borrower at the time the listing was created.
49. **IncomeVerifiable:** The borrower indicated they have the required documentation to support their income.
50. **StatedMonthlyIncome:** The monthly income the borrower stated at the time the listing was created.
51. **LoanKey:** Unique key for each loan. This is the same key that is used in the API.
52. **TotalProsperLoans:** Number of Prosper loans the borrower at the time they created this listing. This value will be null if the borrower had no prior loans.
53. **TotalProsperPaymentsBilled:** Number of on time payments the borrower made on Prosper loans at the time they created this listing. This value will be null if the borrower had no prior loans.
54. **OnTimeProsperPayments:** Number of on time payments the borrower had made on Prosper loans at the time they created this listing. This value will be null if the borrower has no prior loans.
55. **ProsperPaymentsLessThanOneMonthLate:** Number of payments the borrower made on Prosper loans that were less than one month late at the time they created this listing. This value will be null if the borrower had no prior loans.
56. **ProsperPaymentsOneMonthPlusLate:** Number of payments the borrower made on Prosper loans that were greater than one month late at the time they created this listing. This value will be null if the borrower had no prior loans.
57. **ProsperPrincipalBorrowed:** Total principal borrowed on Prosper loans at the time the listing was created. This value will be null if the borrower had no prior loans.
58. **ProsperPrincipalOutstanding:** Principal outstanding on Prosper loans at the time the listing was created. This value will be null if the borrower had no prior loans.
59. **ScorexChangeAtTimeOfListing:** Borrower's credit score change at the time the credit profile was pulled. This will be the change relative to the borrower's last Prosper loan. This value will be null if the borrower had no prior loans.
60. **LoanCurrentDaysDelinquent:** The number of days delinquent.
61. **LoanFirstDefaultedCycleNumber:** The cycle the loan was charged off. If the loan has not charged off the value will be null.
62. **LoanMonthsSinceOrigination:** Number of months since the loan originated.
63. **LoanNumber:** Unique numeric value associated with the loan.
64. **LoanOriginalAmount:** The origination amount of the loan.
65. **LoanOriginationDate:** The date the loan was originated.

66. **LoanOriginationQuarter:** The quarter in which the loan was originated.
67. **MemberKey:** The unique key that is associated with the borrower. This is the same identifier that is used in the API member object.
68. **MonthlyLoanPayment** The scheduled monthly loan payment.
69. **LP_CustomerPayments:** Pre charge-off cumulative gross payments made by the borrower on the loan. If the loan has charged off, this value will exclude any recoveries.
70. **LP_CustomerPrincipalPayments:** Pre charge-off cumulative principal payments made by the borrower on the loan. If the loan has charged off, this value will exclude any recoveries.
71. **LP_InterestandFees:** Pre charge-off cumulative interest and fees paid by the borrower. If the loan has charged off, this value will exclude any recoveries.
72. **LP_ServiceFees:** Cumulative service fees paid by the investors who have invested in the loan.
73. **LP_CollectionFees:** Cumulative collection fees paid by the investors who have invested in the loan.
74. **LP_GrossPrincipalLoss:** The gross charged off amount of the loan.
75. **LP_NetPrincipalLoss:** The principal that remains uncollected after any recoveries.
76. **LP_NonPrincipalRecoverypayments:** The interest and fee component of any recovery payments. The current payment policy applies payments in the following order: Fees, interest, principal.
77. **PercentFunded:** Percent the listing was funded. Recommendations Number of recommendations the borrower had at the time the listing was created.
78. **InvestmentFromFriendsCount:** Number of friends that made an investment in the loan.
79. **InvestmentFromFriendsAmount:** Dollar amount of investments that were made by friends.
80. **Investors:** The number of investors that funded the loan.