Exploratory Data Analysis by Benjamin Fundelits

Univariate Plots Section

Dimensions

```
## [1] 113937     81
```

The Dataset consists of 81 variables with almost 114,000 observations

Variable Names in Alphabetic Order

```
##  [1] "AmountDelinquent"
##  [2] "AvailableBankcardCredit"
##  [3] "BankcardUtilization"
##  [4] "BorrowerAPR"
##  [5] "BorrowerRate"
##  [6] "BorrowerState"
##  [7] "ClosedDate"
##  [8] "CreditGrade"
##  [9] "CreditScoreRangeLower"
## [10] "CreditScoreRangeUpper"
## [11] "CurrentCreditLines"
## [12] "CurrentDelinquencies"
## [13] "CurrentlyInGroup"
## [14] "DateCreditPulled"
## [15] "DebtToIncomeRatio"
## [16] "DelinquenciesLast7Years"
## [17] "EmploymentStatus"
## [18] "EmploymentStatusDuration"
## [19] "EstimatedEffectiveYield"
## [20] "EstimatedLoss"
## [21] "EstimatedReturn"
## [22] "FirstRecordedCreditLine"
## [23] "GroupKey"
## [24] "IncomeRange"
## [25] "IncomeVerifiable"
## [26] "InquiriesLast6Months"
## [27] "InvestmentFromFriendsAmount"
## [28] "InvestmentFromFriendsCount"
## [29] "Investors"
## [30] "IsBorrowerHomeowner"
## [31] "LenderYield"
## [32] "ListingCategory..numeric."
## [33] "ListingCreationDate"
## [34] "ListingKey"
## [35] "ListingNumber"
## [36] "LoanCurrentDaysDelinquent"
## [37] "LoanFirstDefaultedCycleNumber"
## [38] "LoanKey"
## [39] "LoanMonthsSinceOrigination"
## [40] "LoanNumber"
## [41] "LoanOriginalAmount"
## [42] "LoanOriginationDate"
## [43] "LoanOriginationQuarter"
## [44] "LoanStatus"
## [45] "LP_CollectionFees"
## [46] "LP_CustomerPayments"
## [47] "LP_CustomerPrincipalPayments"
## [48] "LP_GrossPrincipalLoss"
## [49] "LP_InterestandFees"
## [50] "LP_NetPrincipalLoss"
## [51] "LP_NonPrincipalRecoverypayments"
## [52] "LP_ServiceFees"
## [53] "MemberKey"
## [54] "MonthlyLoanPayment"
## [55] "Occupation"
## [56] "OnTimeProsperPayments"
## [57] "OpenCreditLines"
## [58] "OpenRevolvingAccounts"
## [59] "OpenRevolvingMonthlyPayment"
## [60] "PercentFunded"
## [61] "ProsperPaymentsLessThanOneMonthLate"
## [62] "ProsperPaymentsOneMonthPlusLate"
## [63] "ProsperPrincipalBorrowed"
## [64] "ProsperPrincipalOutstanding"
## [65] "ProsperRating..Alpha."
## [66] "ProsperRating..numeric."
## [67] "ProsperScore"
## [68] "PublicRecordsLast10Years"
## [69] "PublicRecordsLast12Months"
## [70] "Recommendations"
## [71] "RevolvingCreditBalance"
## [72] "ScorexChangeAtTimeOfListing"
## [73] "StatedMonthlyIncome"
## [74] "Term"
## [75] "TotalCreditLinespast7years"
## [76] "TotalInquiries"
## [77] "TotalProsperLoans"
## [78] "TotalProsperPaymentsBilled"
## [79] "TotalTrades"
## [80] "TradesNeverDelinquent..percentage."
## [81] "TradesOpenedLast6Months"
```

Structure

```
## 'data.frame':    113937 obs. of  81 variables:
##  $ ListingKey                         : Factor w/ 113066 levels
"00003546482094282EF90E5",..: 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 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 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","$1-
24,999",..: 4 5 7 4 3 3 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 ...
```

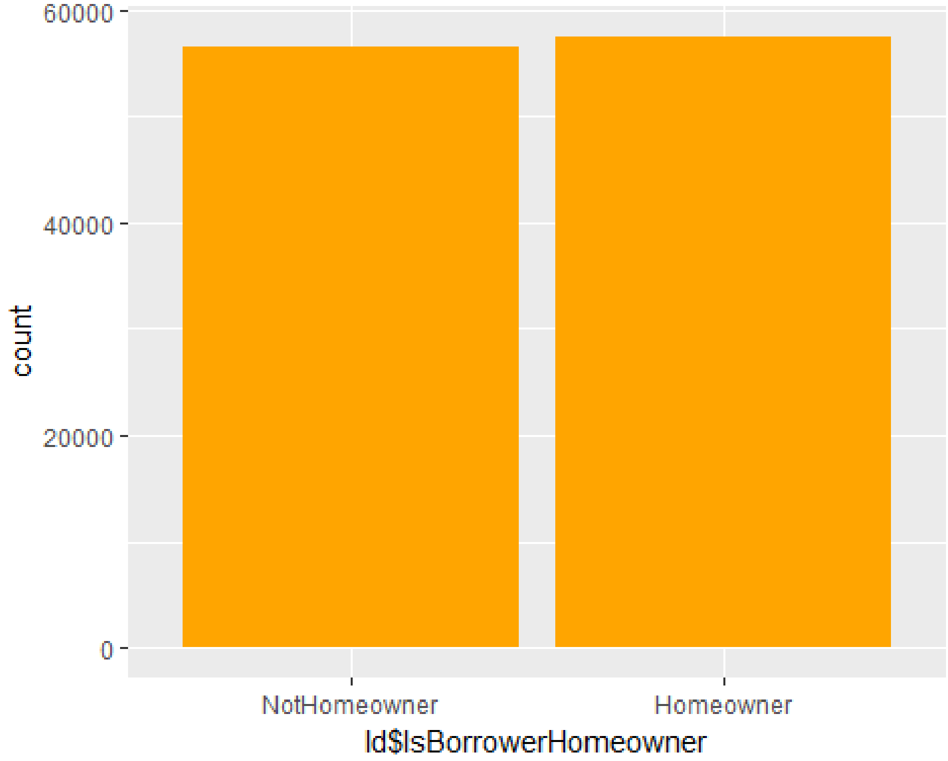
Information About Variables

```
##                    ListingKey     ListingNumber   
##  17A93590655669644DB4C06:     6   Min.   :      4  
##  349D3587495831350F0F648:     4   1st Qu.: 400919  
##  47C1359638497431975670B:     4   Median : 600554  
##  8474358854651984137201C:     4   Mean   : 627886  
##  DE8535960513435199406CE:     4   3rd Qu.: 892634  
##  04C13599434217079754AEE:     3   Max.   :1255725  
##  (Other)                :113912                    
##                     ListingCreationDate  CreditGrade         Term      
##  2013-10-02 17:20:16.550000000:     6          :84984   Min.   :12.00  
##  2013-08-28 20:31:41.107000000:     4   C      : 5649   1st Qu.:36.00  
##  2013-09-08 09:27:44.853000000:     4   D      : 5153   Median :36.00  
##  2013-12-06 05:43:13.830000000:     4   B      : 4389   Mean   :40.83  
##  2013-12-06 11:44:58.283000000:     4   AA     : 3509   3rd Qu.:36.00  
##  2013-08-21 07:25:22.360000000:     3   HR     : 3508   Max.   :60.00  
##  (Other)                      :113912   (Other): 6745                  
##                  LoanStatus                  ClosedDate   
##  Current              :56576                      :58848  
##  Completed           :38074   2014-03-04 00:00:00:  105
##  Chargedoff          :11992   2014-02-19 00:00:00:  100
##  Defaulted           : 5018   2014-02-11 00:00:00:   92
##  Past Due (1-15 days) :  806   2012-10-30 00:00:00:   81
##  Past Due (31-60 days):  363   2013-02-26 00:00:00:   78
##  (Other)              : 1108   (Other)            :54633
##   BorrowerAPR       BorrowerRate     LenderYield
##  Min.   :0.00653   Min.   :0.0000   Min.   :-0.0100
##  1st Qu.:0.15629   1st Qu.:0.1340   1st Qu.: 0.1242
##  Median :0.20976   Median :0.1840   Median : 0.1730
##  Mean   :0.21883   Mean   :0.1928   Mean   : 0.1827
##  3rd Qu.:0.28381   3rd Qu.:0.2500   3rd Qu.: 0.2400
##  Max.   :0.51229   Max.   :0.4975   Max.   : 0.4925
##  NA's   :25
##  EstimatedEffectiveYield EstimatedLoss   EstimatedReturn
##  Min.   :-0.183          Min.   :0.005   Min.   :-0.183
##  1st Qu.: 0.116          1st Qu.:0.042   1st Qu.: 0.074
##  Median : 0.162          Median :0.072   Median : 0.092
##  Mean   : 0.169          Mean   :0.080   Mean   : 0.096
##  3rd Qu.: 0.224          3rd Qu.:0.112   3rd Qu.: 0.117
##  Max.   : 0.320          Max.   :0.366   Max.   : 0.284
##  NA's   :29084           NA's   :29084   NA's   :29084
##  ProsperRating..numeric. ProsperRating..Alpha.  ProsperScore
##  Min.   :1.000                  :29084         Min.   : 1.00
##  1st Qu.:3.000           C      :18345         1st Qu.: 4.00
##  Median :4.000           B      :15581         Median : 6.00
##  Mean   :4.072           A      :14551         Mean   : 5.95
##  3rd Qu.:5.000           D      :14274         3rd Qu.: 8.00
##  Max.   :7.000           E      : 9795         Max.   :11.00
##  NA's   :29084           (Other):12307         NA's   :29084
##  ListingCategory..numeric. BorrowerState
##  Min.   : 0.000            CA     :14717
##  1st Qu.: 1.000            TX     : 6842
##  Median : 1.000            NY     : 6729
##  Mean   : 2.774            FL     : 6720
##  3rd Qu.: 3.000            IL     : 5921
##  Max.   :20.000                   : 5515
##                             (Other):67493
##                       Occupation         EmploymentStatus
##  Other                       :28617   Employed     :67322
##  Professional                :13628   Full-time    :26355
##  Computer Programmer         : 4478   Self-employed: 6134
##  Executive                   : 4311   Not available: 5347
##  Teacher                     : 3759   Other        : 3806
##  Administrative Assistant: 3688                : 2255
##  (Other)                 :55456   (Other)      : 2718
##  EmploymentStatusDuration IsBorrowerHomeowner CurrentlyInGroup
##  Min.   :  0.00           False:56459         False:101218
##  1st Qu.: 26.00           True :57478         True : 12719
##  Median : 67.00
##  Mean   : 96.07
##  3rd Qu.:137.00
##  Max.   :755.00
##  NA's   :7625
##                       GroupKey                 DateCreditPulled
##                         :100596   2013-12-23 09:38:12:     6
##  783C3371218786870A73D20:  1140   2013-11-21 09:09:41:     4
##  3D4D3366260257624AB272D:   916   2013-12-06 05:43:16:     4
##  6A3B336601725506917317E:   698   2014-01-14 20:17:49:     4
##  FEF83377364176536637E50:   611   2014-02-09 12:14:41:     4
##  C9643379247860156A00EC0:   342   2013-09-27 22:04:54:     3
##  (Other)                :  9634   (Other)            :113912
##  CreditScoreRangeLower CreditScoreRangeUpper
##  Min.   :  0.0         Min.   : 19.0
##  1st Qu.:660.0         1st Qu.:679.0
##  Median :680.0         Median :699.0
##  Mean   :685.6         Mean   :704.6
##  3rd Qu.:720.0         3rd Qu.:739.0
##  Max.   :880.0         Max.   :899.0
##  NA's   :591           NA's   :591
##         FirstRecordedCreditLine CurrentCreditLines OpenCreditLines
##                     :   697     Min.   : 0.00      Min.   : 0.00
##  1993-12-01 00:00:00:   185     1st Qu.: 7.00      1st Qu.: 6.00
##  1994-11-01 00:00:00:   178     Median :10.00      Median : 9.00
##  1995-11-01 00:00:00:   168     Mean   :10.32      Mean   : 9.26
##  1990-04-01 00:00:00:   161     3rd Qu.:13.00      3rd Qu.:12.00
##  1995-03-01 00:00:00:   159     Max.   :59.00      Max.   :54.00
##  (Other)            :112389     NA's   :7604       NA's   :7604
##  TotalCreditLinespast7years OpenRevolvingAccounts
##  Min.   :  2.00             Min.   : 0.00
##  1st Qu.: 17.00             1st Qu.: 4.00
##  Median : 25.00             Median : 6.00
##  Mean   : 26.75             Mean   : 6.97
##  3rd Qu.: 35.00             3rd Qu.: 9.00
##  Max.   :136.00             Max.   :51.00
##  NA's   :697
##  OpenRevolvingMonthlyPayment InquiriesLast6Months TotalInquiries
##  Min.   :    0.0             Min.   :  0.000      Min.   :  0.000
##  1st Qu.:  114.0             1st Qu.:  0.000      1st Qu.:  2.000
##  Median :  271.0             Median :  1.000      Median :  4.000
##  Mean   :  398.3             Mean   :  1.435      Mean   :  5.584
##  3rd Qu.:  525.0             3rd Qu.:  2.000      3rd Qu.:  7.000
##  Max.   :14985.0             Max.   :105.000      Max.   :379.000
##                              NA's   :697          NA's   :1159
##  CurrentDelinquencies AmountDelinquent   DelinquenciesLast7Years
##  Min.   : 0.0000      Min.   :     0.0   Min.   : 0.000
##  1st Qu.: 0.0000      1st Qu.:     0.0   1st Qu.: 0.000
##  Median : 0.0000      Median :     0.0   Median : 0.000
##  Mean   : 0.5921      Mean   :   984.5   Mean   : 4.155
##  3rd Qu.: 0.0000      3rd Qu.:     0.0   3rd Qu.: 3.000
##  Max.   :83.0000      Max.   :463881.0   Max.   :99.000
##  NA's   :697          NA's   :7622       NA's   :990
##  PublicRecordsLast10Years PublicRecordsLast12Months RevolvingCreditBalance
##  Min.   : 0.0000          Min.   : 0.000            Min.   :      0
##  1st Qu.: 0.0000          1st Qu.: 0.000            1st Qu.:   3121
##  Median : 0.0000          Median : 0.000            Median :   8549
##  Mean   : 0.3126          Mean   : 0.015            Mean   :  17599
##  3rd Qu.: 0.0000          3rd Qu.: 0.000            3rd Qu.:  19521
##  Max.   :38.0000          Max.   :20.000            Max.   :1435667
##  NA's   :697              NA's   :7604              NA's   :7604
##  BankcardUtilization AvailableBankcardCredit  TotalTrades
##  Min.   :0.000       Min.   :     0          Min.   :  0.00
##  1st Qu.:0.310       1st Qu.:   880          1st Qu.: 15.00
##  Median :0.600       Median :  4100          Median : 22.00
##  Mean   :0.561       Mean   : 11210          Mean   : 23.23
##  3rd Qu.:0.840       3rd Qu.: 13180          3rd Qu.: 30.00
##  Max.   :5.950       Max.   :646285          Max.   :126.00
##  NA's   :7604        NA's   :7544            NA's   :7544
##  TradesNeverDelinquent..percentage. TradesOpenedLast6Months
##  Min.   :0.000                      Min.   : 0.000
##  1st Qu.:0.820                      1st Qu.: 0.000
##  Median :0.940                      Median : 0.000
##  Mean   :0.886                      Mean   : 0.802
##  3rd Qu.:1.000                      3rd Qu.: 1.000
##  Max.   :1.000                      Max.   :20.000
##  NA's   :7544                       NA's   :7544
##  DebtToIncomeRatio         IncomeRange    IncomeVerifiable
##  Min.   : 0.000    $25,000-49,999:32192   False:  8669
##  1st Qu.: 0.140    $50,000-74,999:31050   True :105268
##  Median : 0.220    $100,000+     :17337
##  Mean   : 0.276    $75,000-99,999:16916
##  3rd Qu.: 0.320    Not displayed : 7741
##  Max.   :10.010    $1-24,999     : 7274
##  NA's   :8554      (Other)       : 1427
##  StatedMonthlyIncome                    LoanKey       TotalProsperLoans
##  Min.   :      0     CB1B37030986463208432A1:     6   Min.   :0.00
##  1st Qu.:   3200     2DEE3698211017519D7333F:     4   1st Qu.:1.00
##  Median :   4667     9F4B37043517554537C364C:     4   Median :1.00
##  Mean   :   5608     D895370150591392337ED6D:     4   Mean   :1.42
##  3rd Qu.:   6825     E6FB37073953690388BC56D:     4   3rd Qu.:2.00
##  Max.   :1750003     0D8F37036734373301ED419:     3   Max.   :8.00
##                      (Other)                :113912   NA's   :91852
##  TotalProsperPaymentsBilled OnTimeProsperPayments
##  Min.   :  0.00            Min.   :  0.00
##  1st Qu.:  9.00            1st Qu.:  9.00
##  Median : 16.00            Median : 15.00
##  Mean   : 22.93            Mean   : 22.27
##  3rd Qu.: 33.00            3rd Qu.: 32.00
##  Max.   :141.00            Max.   :141.00
##  NA's   :91852             NA's   :91852
##  ProsperPaymentsLessThanOneMonthLate ProsperPaymentsOneMonthPlusLate
##  Min.   : 0.00                       Min.   : 0.00
##  1st Qu.: 0.00                       1st Qu.: 0.00
##  Median : 0.00                       Median : 0.00
##  Mean   : 0.61                       Mean   : 0.05
##  3rd Qu.: 0.00                       3rd Qu.: 0.00
##  Max.   :42.00                       Max.   :21.00
##  NA's   :91852                       NA's   :91852
##  ProsperPrincipalBorrowed ProsperPrincipalOutstanding
##  Min.   :    0            Min.   :    0
##  1st Qu.: 3500            1st Qu.:    0
##  Median : 6000            Median : 1627
##  Mean   : 8472            Mean   : 2930
##  3rd Qu.:11000            3rd Qu.: 4127
##  Max.   :72499            Max.   :23451
##  NA's   :91852            NA's   :91852
##  ScorexChangeAtTimeOfListing LoanCurrentDaysDelinquent
##  Min.   :-209.00             Min.   :   0.0
##  1st Qu.: -35.00             1st Qu.:   0.0
##  Median :  -3.00             Median :   0.0
##  Mean   :  -3.22             Mean   : 152.8
##  3rd Qu.:  25.00             3rd Qu.:   0.0
##  Max.   : 286.00             Max.   :2704.0
##  NA's   :95009
##  LoanFirstDefaultedCycleNumber LoanMonthsSinceOrigination   LoanNumber
##  Min.   : 0.00                 Min.   :  0.0              Min.   :     1
##  1st Qu.: 9.00                 1st Qu.:  6.0              1st Qu.: 37332
##  Median :14.00                 Median : 21.0              Median : 68599
##  Mean   :16.27                 Mean   : 31.9              Mean   : 69444
##  3rd Qu.:22.00                 3rd Qu.: 65.0              3rd Qu.:101901
##  Max.   :44.00                 Max.   :100.0              Max.   :136486
##  NA's   :96985
##  LoanOriginalAmount          LoanOriginationDate LoanOriginationQuarter
##  Min.   : 1000      2014-01-22 00:00:   491   Q4 2013:14450
##  1st Qu.: 4000      2013-11-13 00:00:00:   490   Q1 2014:12172
##  Median : 6500      2014-02-19 00:00:00:   439   Q3 2013: 9180
##  Mean   : 8337      2013-10-16 00:00:00:   434   Q2 2013: 7099
##  3rd Qu.:12000      2014-01-28 00:00:00:   339   Q3 2012: 5632
##  Max.   :35000      2013-09-24 00:00:00:   316   Q2 2012: 5061
##                     (Other)            :111428   (Other):60343
##                    MemberKey      MonthlyLoanPayment LP_CustomerPayments
##  63CA34120866140639431C9:     9   Min.   :   0.0     Min.   :   -2.35
##  16083364744933457E57FB9:     8   1st Qu.: 131.6     1st Qu.: 1005.76
##  3A2F3380477699707C81385:     8   Median : 217.7     Median : 2583.83
##  4D9C3403302047712AD0CDD:     8   Mean   : 272.5     Mean   : 4183.08
##  739C338135235294782AE75:     8   3rd Qu.: 371.6     3rd Qu.: 5548.40
##  7E1733653050264822FAA3D:     8   Max.   :2251.5     Max.   :40702.39
##  (Other)                :113888
##  LP_CustomerPrincipalPayments LP_InterestandFees LP_ServiceFees
##  Min.   :    0.0              Min.   :   -2.35   Min.   :-664.87
##  1st Qu.:  500.9              1st Qu.:  274.87   1st Qu.: -73.18
##  Median : 1587.5              Median :  700.84   Median : -34.44
##  Mean   : 3105.5              Mean   : 1077.54   Mean   : -54.73
##  3rd Qu.: 4000.0              3rd Qu.: 1458.54   3rd Qu.: -13.92
##  Max.   :35000.0              Max.   :15617.03   Max.   :  32.06
##
##  LP_CollectionFees  LP_GrossPrincipalLoss LP_NetPrincipalLoss
##  Min.   :-9274.75   Min.   :  -94.2       Min.   : -954.5
##  1st Qu.:    0.00   1st Qu.:    0.0       1st Qu.:    0.0
##  Median :    0.00   Median :    0.0       Median :    0.0
##  Mean   :  -14.24   Mean   :  700.4       Mean   :  681.4
##  3rd Qu.:    0.00   3rd Qu.:    0.0       3rd Qu.:    0.0
##  Max.   :    0.00   Max.   :25000.0       Max.   :25000.0
##
##  LP_NonPrincipalRecoverypayments PercentFunded    Recommendations
##  Min.   :    0.00                Min.   :0.7000   Min.   : 0.00000
##  1st Qu.:    0.00                1st Qu.:1.0000   1st Qu.: 0.00000
##  Median :    0.00                Median :1.0000   Median : 0.00000
##  Mean   :   25.14                Mean   :0.9986   Mean   : 0.04803
##  3rd Qu.:    0.00                3rd Qu.:1.0000   3rd Qu.: 0.00000
##  Max.   :21117.90                Max.   :1.0125   Max.   :39.00000
##
##  InvestmentFromFriendsCount InvestmentFromFriendsAmount   Investors
##  Min.   : 0.00000           Min.   :    0.00            Min.   :   1.00
##  1st Qu.: 0.00000           1st Qu.:    0.00            1st Qu.:   2.00
##  Median : 0.00000           Median :    0.00            Median :  44.00
##  Mean   : 0.02346           Mean   :   16.55            Mean   :  80.48
##  3rd Qu.: 0.00000           3rd Qu.:    0.00            3rd Qu.: 115.00
##  Max.   :33.00000           Max.   :25000.00            Max.   :1189.00
##
```

Homeowners

```
# Set homeowner names from True & False to "Homeowner" & NotHomeowner"
levels(ld$IsBorrowerHomeowner)[levels(ld$IsBorrowerHomeowner) == 'True'] <-
'Homeowner'
levels(ld$IsBorrowerHomeowner)[levels(ld$IsBorrowerHomeowner) == "False"] <-
'NotHomeowner'

# Display Homeowners
ggplot(ld, aes(ld$IsBorrowerHomeowner)) +
  geom_histogram(stat = "count", fill = 'orange')
```

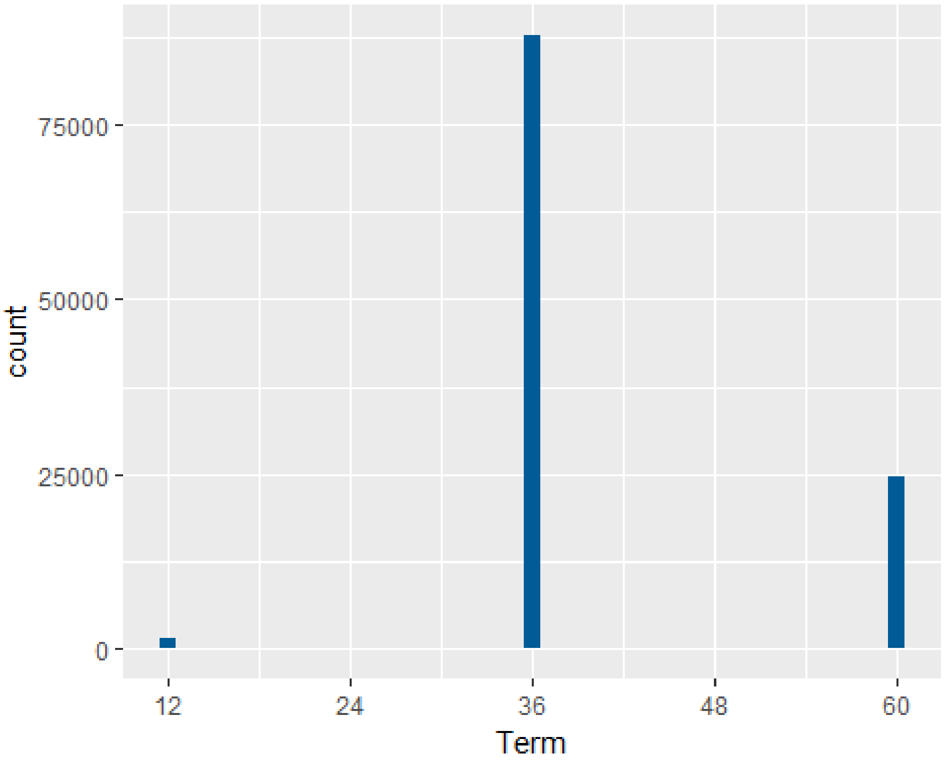


```
##
## NotHomeowner    Homeowner
##        56459        57478
```

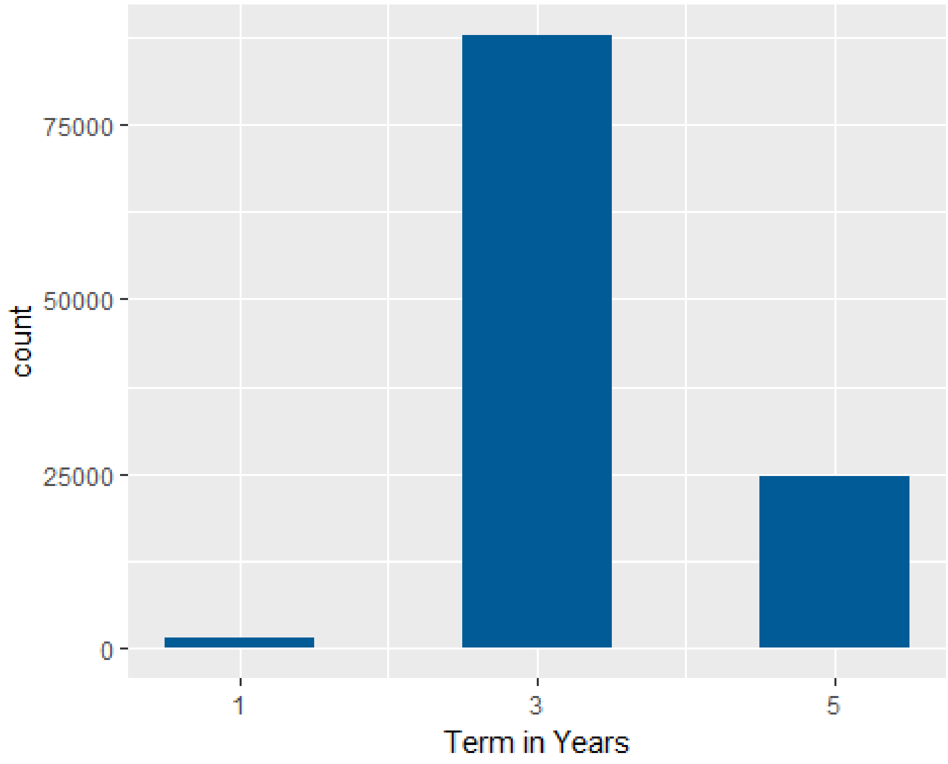
Homeowner and NotHomeowner counts are almost the same. The homeowner category cuts the dataset in half. Later we we will use that in our analysis.

The length of the Loan Expressed in Months

```
# DIsplay the legths of loan payments in months
ggplot(aes(x = Term), data = ld)+
  geom_histogram(binwidth = 1, fill = I('#005b96')) +
  scale_x_continuous(breaks = seq(0,60,12))
```



```
# Display the legths of loan payments in years
ggplot(aes(x = Term/12), data = ld)+
  geom_histogram(binwidth = 1, fill = I('#005b96')) +
  scale_x_continuous(breaks = seq(1,5,2)) +
  xlab("Term in Years")
```

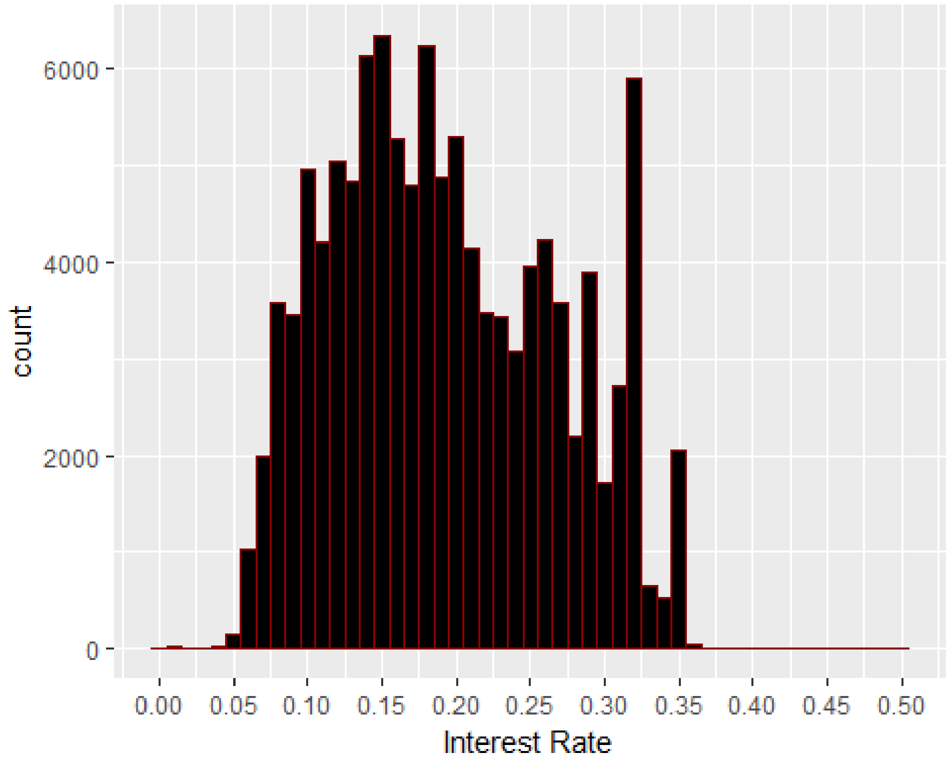


```
## 
##    12    36    60 
##  1614 87778 24545
```

All the borrowing terms are either 12, 36 or 60 months. Most of the terms are 3 years (90,000), many of them are 5 years (25,000) and a few are 1 year (1,500).

Borrower's Interest Rate

```
# Display interest rate
ggplot(aes(BorrowerRate), data =ld) +
  geom_histogram(binwidth = .01, fill = 'black', color = 'darkred') +
  scale_x_continuous(breaks = seq(0,.5,.05)) +
  xlab('Interest Rate')
```



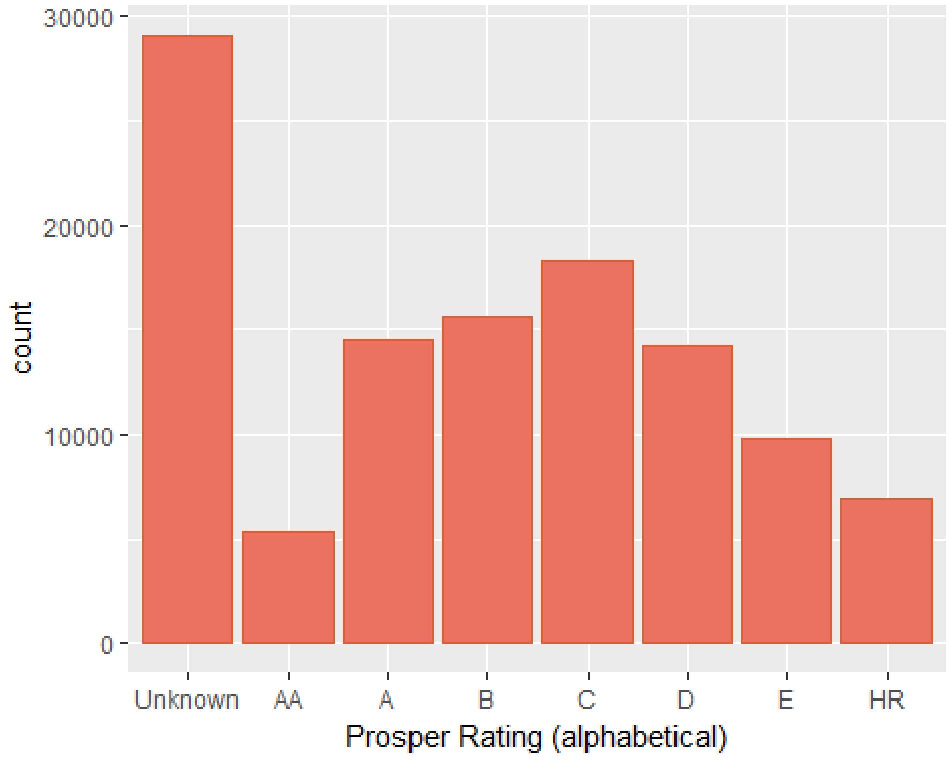
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.1340  0.1840  0.1928  0.2500  0.4975
```

The interest rates are between 0 and 0.5 Most of the loans have interest rates between 0.05 and 0.35.

Alphabetical Prosper rating

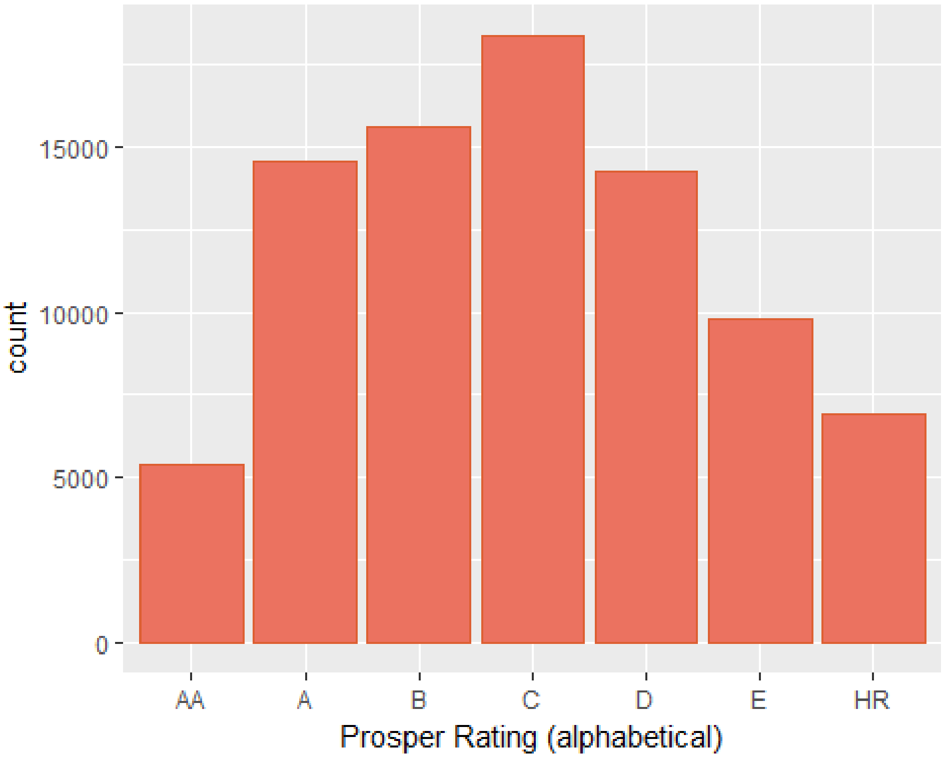
```
# Order ProsperRating(alpha) levels from the best rating to the lowest.
ld$ProsperRating..Alpha. <- ordered(ld$ProsperRating..Alpha., levels = c("",
"AA", "A", "B", "C", "D", "E", "HR"))

# Display Prosper ratings
ggplot(aes(ProsperRating..Alpha.), data = ld) +
  geom_bar(fill = I('#EB7260'), color = I( '#DD5F32')) +
  xlab('Prosper Rating (alphabetical)')
```



We have no prosper rating available for about 30,000 loans. This number of the unknown rating category is even bigger than the number of the largest known rating category. Let's get rid of the unknown bin to get a better look at the remaining ones.

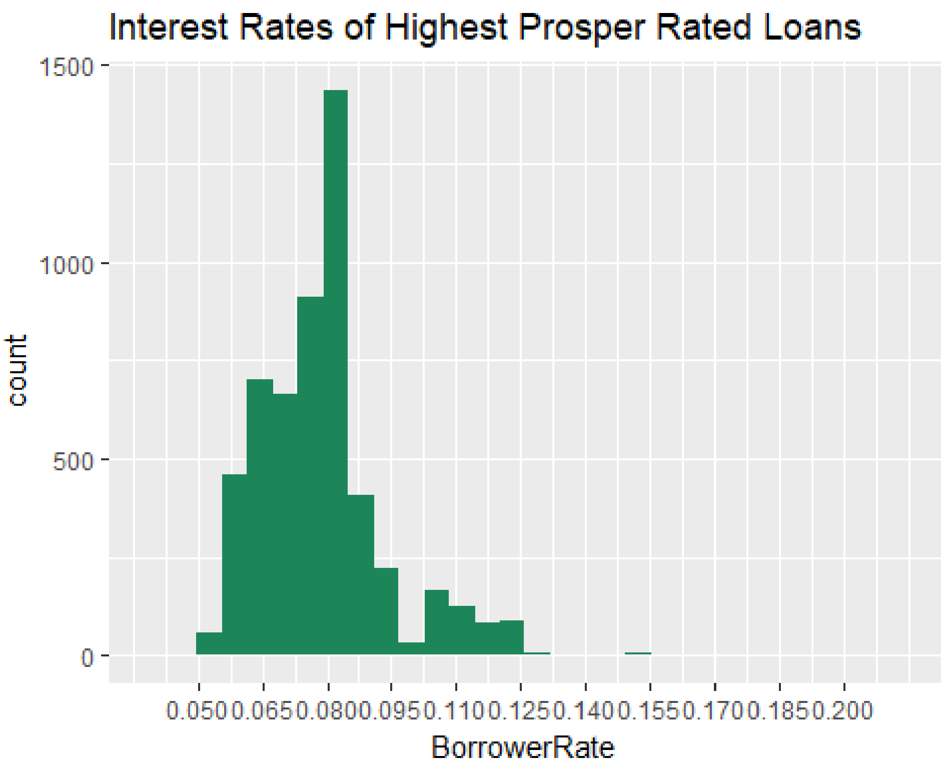
```r
# Display prosper ratings without unknown data.
ggplot(aes(ProsperRating..Alpha.), data = ld) +
  geom_bar(fill = I('#EB7260'), color = I( '#DD5F32'))+
  scale_x_discrete(limits = c("AA", "A", "B", "C", "D", "E", "HR")) +
  xlab('Prosper Rating (alphabetical)')
```



As we can see, as we go towards to the middle ranked (C) rating from both sides, the count goes higher too. The x-axis is ordered from the highest rating (AA) to the lowest (HR).

Interest Rate Difference Between the Best Rated and the Worst Rated Borrowers

```
# Best (AA) rated loans
ggplot(aes(BorrowerRate), data = subset(ld, ProsperRating..Alpha. == 'AA')) +
  geom_histogram(fill = I('#1d8659')) +
  scale_x_continuous(breaks = seq(0.05,0.2,0.015))+
  ggtitle('Interest Rates of Highest Prosper Rated Loans')
```



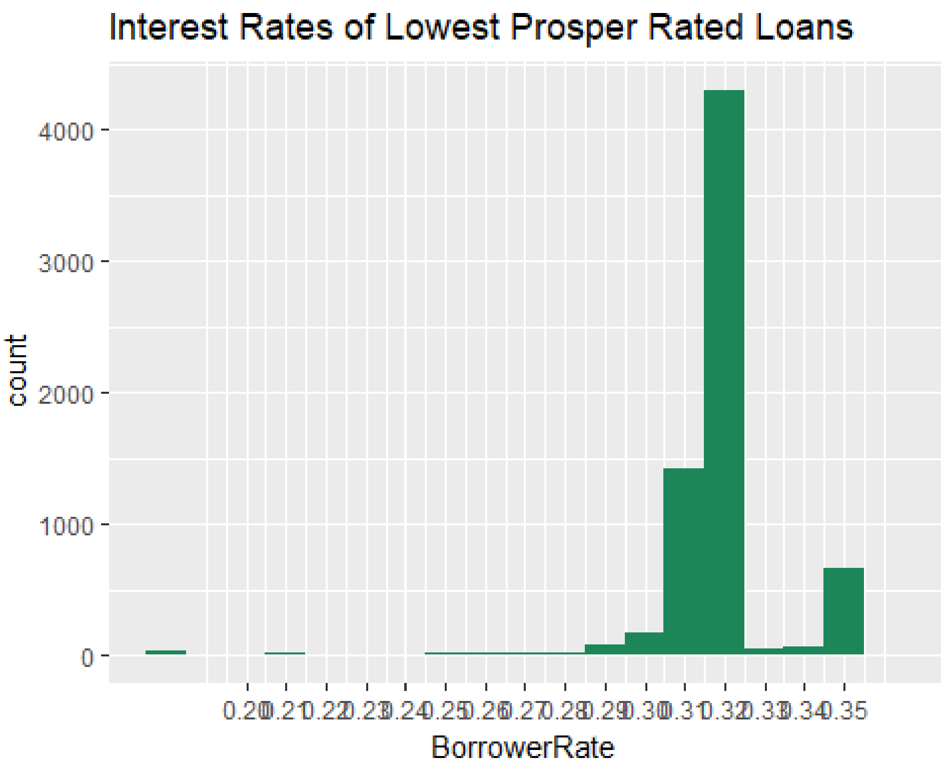
```
## [1] 5372   81

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.04000 0.06990 0.07790 0.07912 0.08450 0.21000
```

The highest class in prosper rating is skewed to the right. A borrower with an “AA” rating has a better chance to have a lower interest rate than the mean of its own group’s distribution. Most borrowers are between 0.055 and 0.09.

There are 5,400 people in the “AA” class.

```
# Worst (HR) rated loans
ggplot(aes(BorrowerRate), data = subset(ld, ProsperRating..Alpha. == 'HR')) +
  geom_histogram(fill = I('#1d8659'), binwidth = .01) +
  scale_x_continuous(breaks = seq(0.20,0.35,0.01)) +
  ggtitle('Interest Rates of Lowest Prosper Rated Loans')
```



```
## [1] 6935   81

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1779  0.3134  0.3177  0.3173  0.3177  0.3600
```

This is a left skewed distribution with most borrowers between 0.3 and 0.325. This is a skinnier dataset. It means that the investors don't really differentiate between the people in this group. If someone is a member of this worst rated group, he gets almost the same interest rate like the others in there. In this group, borrowers have to pay significantly higher interest rates. The difference between the medians of these two ranked groups is very huge. The worst ranked class has four times bigger median interest rate than the best ranked class does.

How do the values spread on this "HR" distribution? My assumption is that there will be only a few rates that have big counts. Let's see.

```
## 
## 0.1779 0.1789 0.1794 0.1799   0.18 0.1805 0.1806  0.181 0.1818 0.1823 
##      2      1      1      1      4      1      1      1      4      2 
## 0.1827 0.1829 0.1845 0.1875 0.1895   0.19 0.1925  0.195 0.1988  0.199 
##      5     13      2      1      3      5      1      1      1      1 
##    0.2 0.2003 0.2085 0.2095   0.21   0.22 0.2234 0.2285 0.2295   0.23 
##      9      1      1      3      9      4      1      1      3      5 
## 0.2322  0.238   0.24 0.2417  0.245 0.2487 0.2495   0.25  0.255 0.2574 
##      1      1      4      1      2      1      1     12      1      1 
## 0.2588 0.2595   0.26  0.265 0.2682 0.2695   0.27 0.2724  0.275 0.2785 
##      1      4     12      1      1      1     10      1      2      2
##  0.279 0.2799   0.28 0.2841 0.2845  0.285 0.2872 0.2874 0.2888  0.289
##      1      2      9      1      2      4      1      7      1      1
##   0.29 0.2903 0.2924 0.2943  0.295 0.2955  0.297 0.2975  0.298  0.299
##     12      1     62      1      1      1      1      2      1      1
## 0.2994 0.2995 0.2998 0.2999    0.3 0.3009  0.301 0.3025  0.303  0.304
##      2      2      1    131     16      4      1      1      1      1
## 0.3049  0.305 0.3059 0.3075 0.3079 0.3089 0.3093 0.3094 0.3095 0.3097
##      1      1    113     26      3      1      1      1      4      1
## 0.3099   0.31  0.311 0.3121 0.3125 0.3127 0.3134 0.3174 0.3175 0.3177
##      4     55      1      1    268    218    722      1      1   3672
## 0.3179 0.3195 0.3199   0.32  0.321 0.3239  0.324 0.3248 0.3249  0.325
##      1      1    604     17      1      1      1      2      2      2
## 0.3267 0.3269  0.327 0.3275  0.328 0.3283 0.3285 0.3289  0.329 0.3295
##      1      1      2      3      2      1      1      1      3      1
## 0.3297 0.3299   0.33 0.3323  0.333 0.3334 0.3335 0.3345  0.335 0.3375
##      1      1     21      1      1      1      1      1      5      1
##  0.338 0.3384 0.3385 0.3387  0.339 0.3391 0.3395 0.3399   0.34 0.3411
##      1      1      1      1      2      1      2      3     30      1
## 0.3418 0.3423 0.3424 0.3425 0.3428 0.3433  0.344 0.3445  0.345 0.3459
##      1      1      2      1      1      1      3      2      8      1
##  0.346 0.3475  0.348 0.3484 0.3485  0.349 0.3494 0.3495 0.3498 0.3499
##      1      2      2      1      6      1      2     12      3      7
##   0.35   0.36
##    633      4
```

As i thought, there are a few interest rates concerned with almost everyone in this group. For example, 0.3177 interest rate is what almost half of this group pays. Let's do some clean up, and remove all the rates from the table that have less than 15 counts:

```
subset(ld.prosperRating_by_worst, ld.prosperRating_by_worst >= 15)

## 
## 0.2924 0.2999    0.3 0.3059 0.3075   0.31 0.3125 0.3127 0.3134 0.3177
##     62    131     16    113     26     55    268    218    722   3672
## 0.3199   0.32   0.33   0.34   0.35
##    604     17     21     30    633
```

The remaining set of data has 15 unique values! Only 15 separated interest rates have around 6500 counts in sum. And half of has less than 75.

Just to compare with the best rated class, I removed all values containing less than 15 counts:

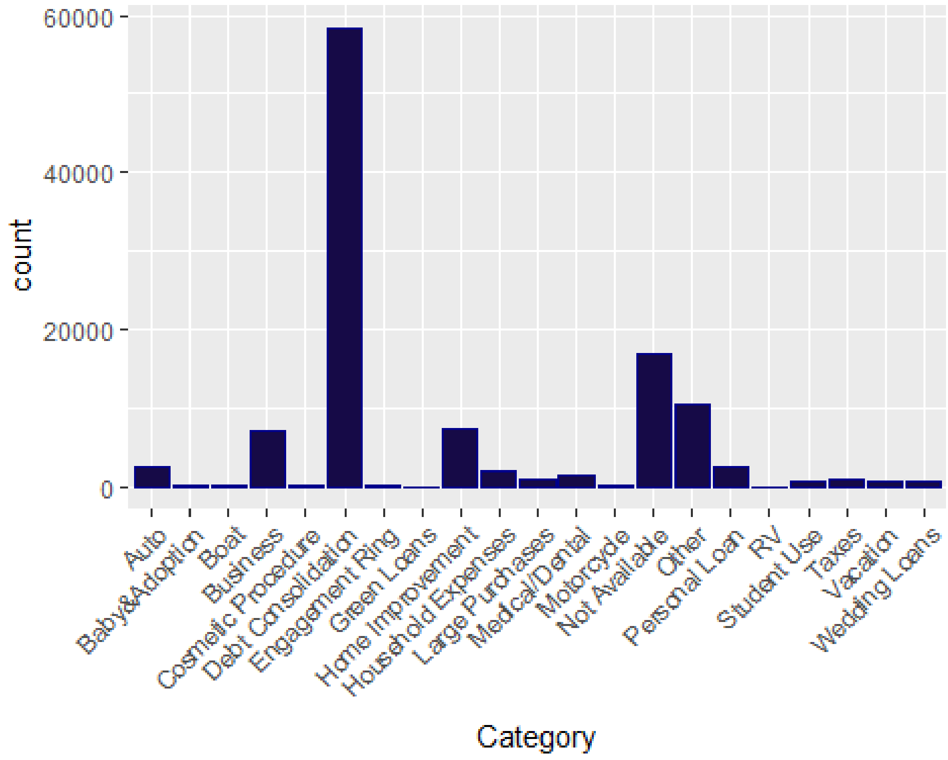
```
#make a table with the best rated ProsperRating unique value counts
ld.prosperRating_AA_table <- table(ld.by_alpha_AA$BorrowerRate)

## 
## 0.0499 0.0565 0.0599 0.0604 0.0605  0.061 0.0625 0.0629 0.0649 0.0655
##     45     81     46     27    235     26     28     85    160    101
## 0.0659 0.0666 0.0699   0.07  0.071 0.0715 0.0716  0.072 0.0724 0.0749
##    253     45    171     29    140     19    200     33     38    141
## 0.0759 0.0765 0.0766 0.0769 0.0779 0.0785 0.0789 0.0799   0.08 0.0804
##    125     23    125    350     39     26     36    219     57     23
## 0.0809 0.0814 0.0819 0.0825  0.083 0.0839 0.0845 0.0849 0.0854 0.0864
##    413     31    130     57     57    216     24    182     37     52
## 0.0869 0.0899  0.093 0.0945  0.096 0.0961    0.1  0.103 0.1042  0.105
##    222     76     64     21     19     57     15     16     20     35
## 0.1071 0.1076 0.1085   0.11 0.1101  0.115 0.1154 0.1199 0.1208
##     15     62     39     23     32     15     22     34     82
```

As we can see the distribution is much more separated, there are a lot more unique numbers in the best rated group. Therefore, there are more variables that can influence your interest rate in this distribution. So the prosper rate itself is not enough to tell within a small range what will be your interest rate. But in the lowest rated distribution it seems like it is rare, to get another rate than that very skinny range of values where 90% of the borrowers can be found.

Category

```
#Display Loan Categories
ggplot(aes(ListingCategory..numeric.), data = ld) +
  geom_bar(fill = I('#160A47'), color = 'darkblue') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1)) +
  xlab('Category')
```



Because this is a categorical, unordered distribution, we can not tell the shape of the distribution, because it does not make any sense. We can always transpose their bins to play with the shape of the distribution. As seeing this bar chart we can see that there are a lot of unknown data, around 17,000. I'll remove this from the chart, because it does not give any useful information for me right now. And there is a very high column which is ruining the chart, called "Debt Consolidation". I have to transform this chart somehow to get a better look at the lower data. Logarithm would be a good choice because the wide number range, but first let's look at the counts.

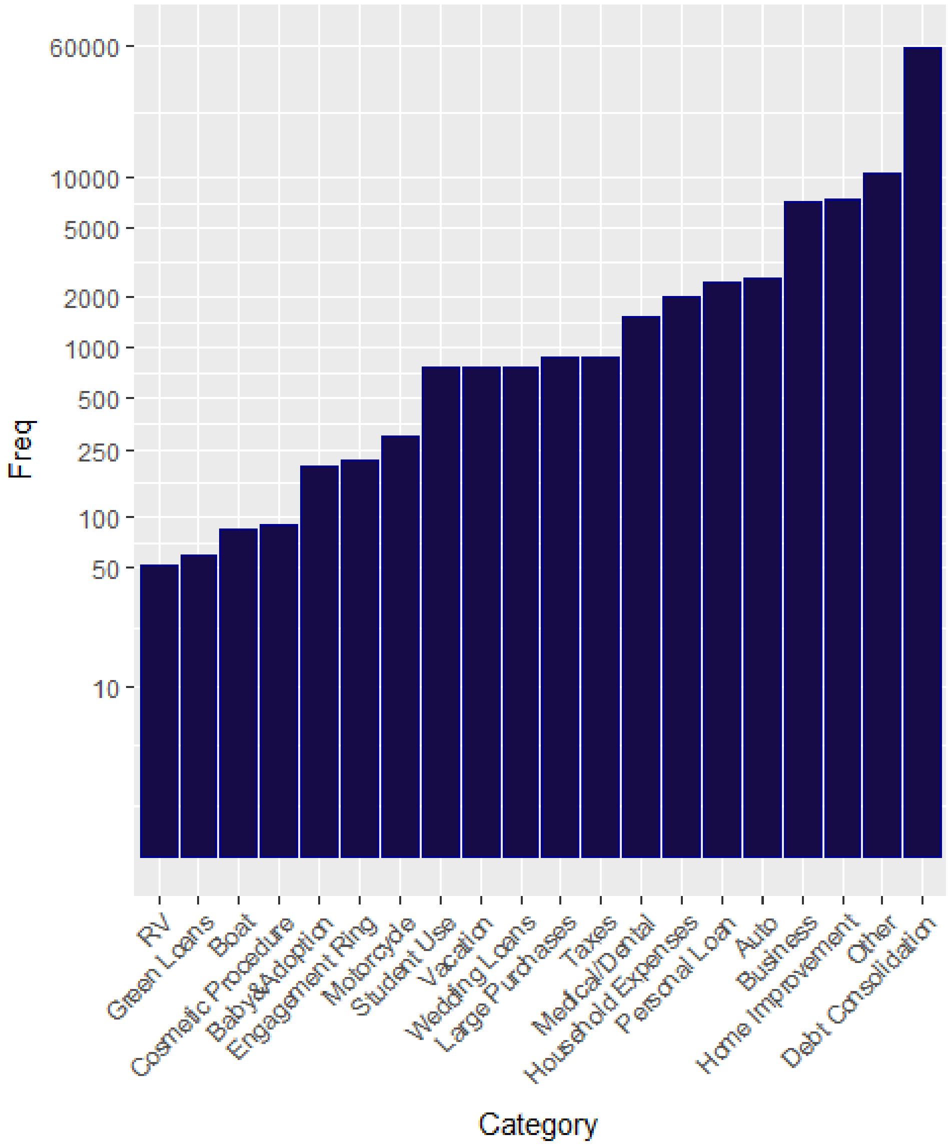
```
## 
##               Auto      Baby&Adoption               Boat 
##               2572                199                 85 
##           Business Cosmetic Procedure Debt Consolidation 
##               7189                 91              58308 
##    Engagement Ring        Green Loans   Home Improvement 
##                217                 59               7433 
## Household Expenses    Large Purchases     Medical/Dental 
##               1996                876               1522 
##         Motorcycle      Not Available              Other 
##                304              16965              10494 
##      Personal Loan                 RV        Student Use 
##               2395                 52                756 
##              Taxes           Vacation      Wedding Loans 
##                885                768                771
```

These category counts are separated in high range. There are a few counted data like "Green Loans", and high counted, like "Home Improvement". First let's apply a logarithm transformation, because the square root will not give a good view in this wide number range.

We can apply the logarithm transformation, because every number is bigger than one! The problem with 0 values is they have no logarithm. And logarithm one is zero, so if we have a 1 counted category, it will display 0. This is why we have to be sure of this. We always have to check this condition.

So let's apply a transformation and reorder the X-axis to have a better looking shape of the data. And make the visualization a little bit bigger for a better view

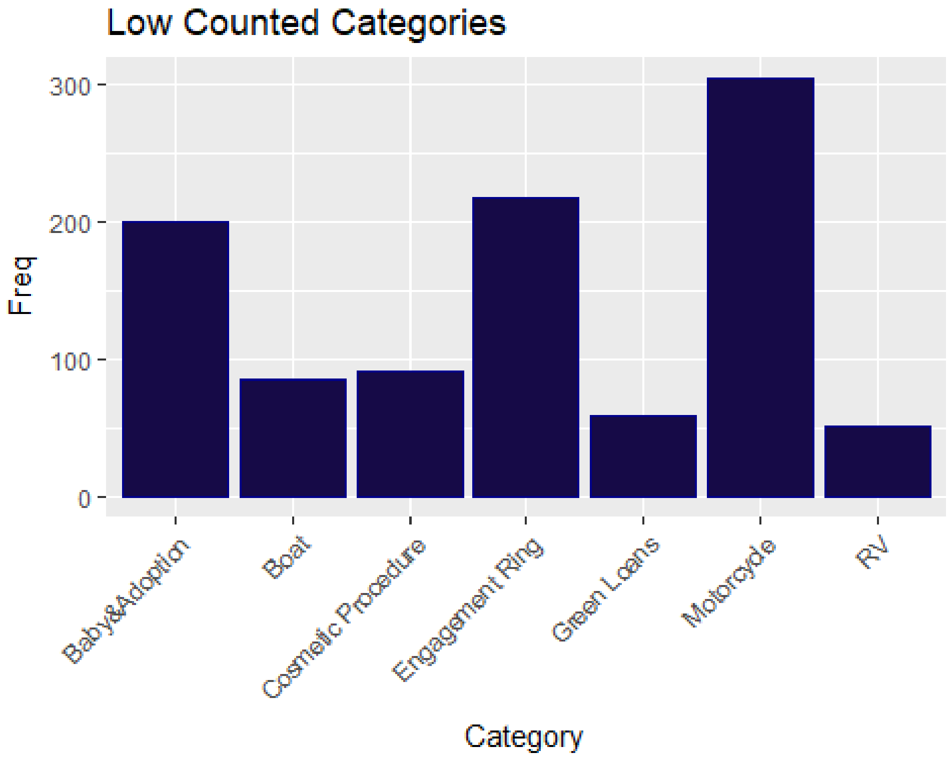
```r
#Display Loan Categories without unknow data with reordered bins
ggplot(aes(x = reorder(Category , Freq), y = Freq),
       data = subset(listing_category_df, Category != 'Not Available')) +
  geom_bar(stat = 'identity' ,fill = I('#160A47'), color = I('darkblue')) +
  scale_y_log10(breaks = c(0,10,50,100,250,500,1000,2000,5000,10000,60000)) +
  coord_cartesian(ylim = c(1,60000)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))  +
  xlab('Category')
```



Perfect! Now we see all the data, all together. Logarithm transformation made a good job here, I won't apply another rescale here. Watching this chart I can see that we can split these categories into 3 groups to better display the count differences between the groups. I'm going to give names to these groups for an easier reference in the future if it's needed, and also because we get a better, more precise groups.

1) Low Counted (RV-Motorcycle)
2) Medium Counted (Student Use - Auto)
3) High Counted (Business - Debt Consolidation)

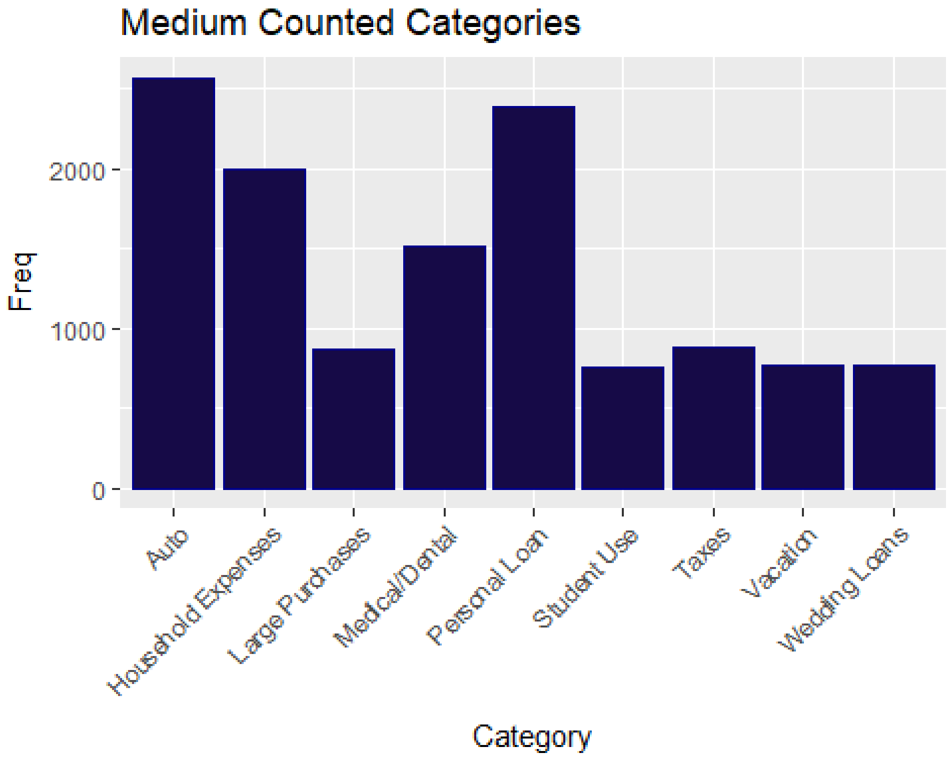
```
# Display low counted categories (RV-Motorcycle)
ggplot(low_counted_group.df, aes(Category, Freq)) +
  geom_bar(stat = 'Identity' ,fill = I('#160A47'), color = I('darkblue')) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1)) +
  ggtitle('Low Counted Categories')
```



```
##                Category Freq
## 2       Baby&Adoption  199
## 3                Boat   85
## 5  Cosmetic Procedure   91
## 7     Engagement Ring  217
## 8         Green Loans   59
## 13         Motorcycle  304
## 17                 RV   52
```

These categories are very unpopular choices for borrowers.

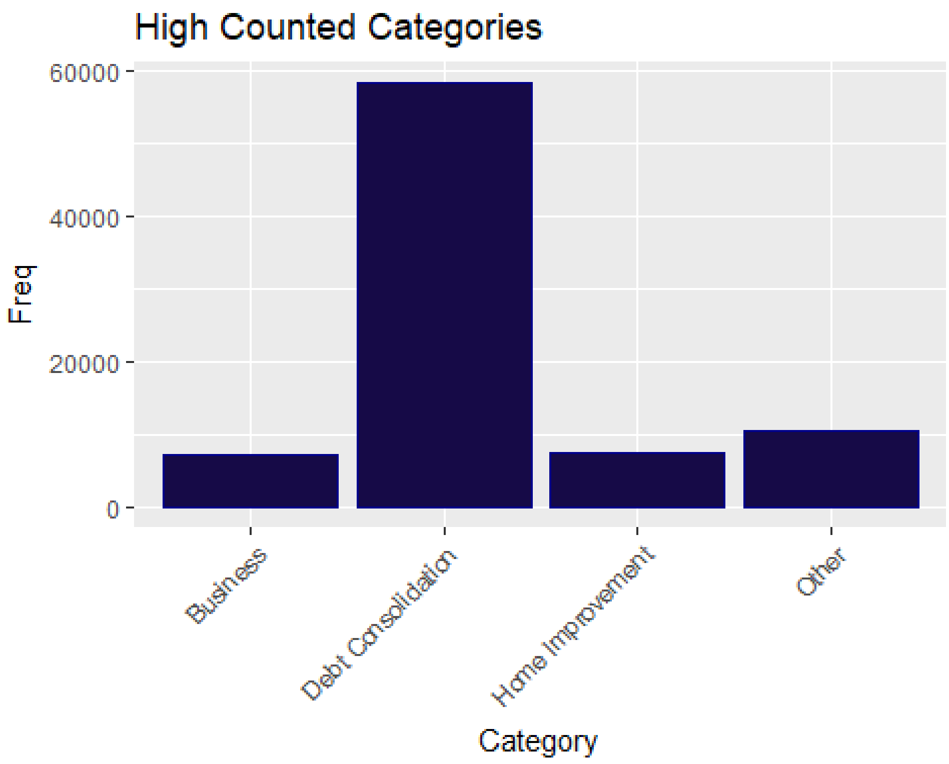
```
# Display medium counted categories (Student Use - Auto)
ggplot(medium_counted_group.df, aes(Category, Freq)) +
  geom_bar(stat = 'Identity',fill = I('#160A47'), color = I('darkblue'))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1)) +
  ggtitle('Medium Counted Categories')
```



```
##              Category Freq
## 1                Auto 2572
## 10 Household Expenses 1996
## 11     Large Purchases  876
## 12      Medical/Dental 1522
## 16       Personal Loan 2395
## 18         Student Use  756
## 19               Taxes  885
## 20            Vacation  768
## 21       Wedding Loans  771
```

These are the categories that borrowers will most likely choose if they are not picking from the High Counted Categories.

```
# DIsplay high counted categories (Business - Debt Consolidation)
ggplot(high_counted_group.df, aes(Category, Freq)) +
  geom_bar(stat = 'Identity',fill = I('#160A47'), color = I('darkblue')) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
  ggtitle('High Counted Categories')
```



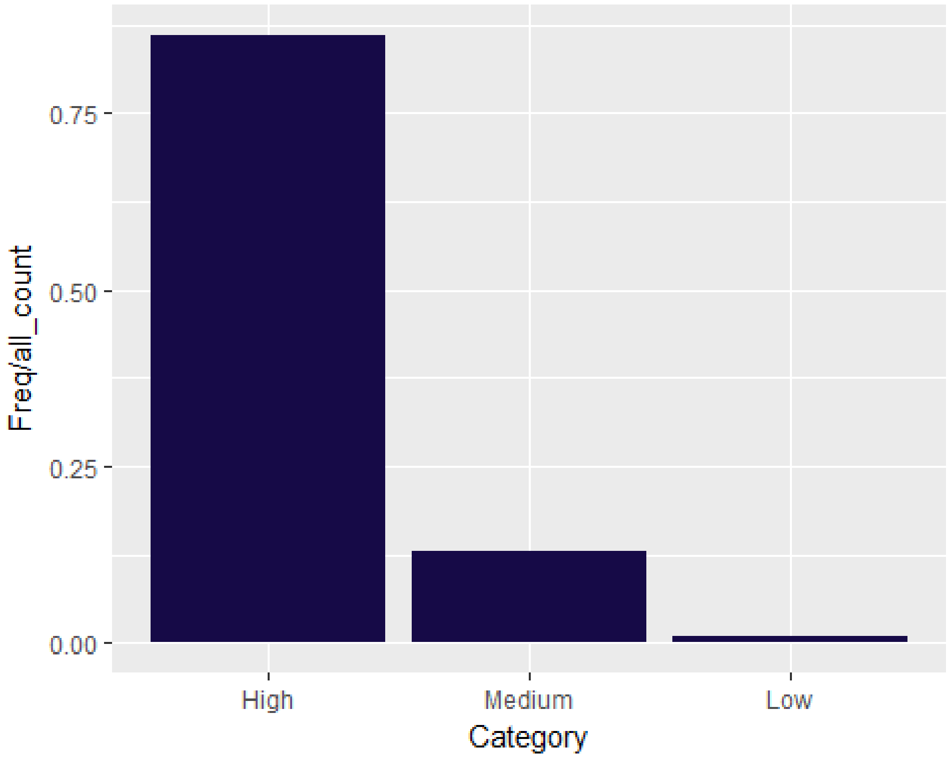
```
##              Category  Freq
## 4            Business  7189
## 6  Debt Consolidation 58308
## 9    Home Improvement  7433
## 15              Other 10494
```

The borrowers are most likely to pick a category from these 4.

What are the low-, medium- and high rated proportions of the categories? Of course I will remove the unknown data from the algorithm. It is very important, because if I would count them in, the sum of the proportions will not be 1.

Let’s see the proportions.

```
# Category proportion
ggplot(category_sums, aes(x =reorder(Category, - Freq), y =Freq/all_count)) +
  geom_bar(stat = 'Identity', fill = I('#160A47')) +
  xlab('Category')
```

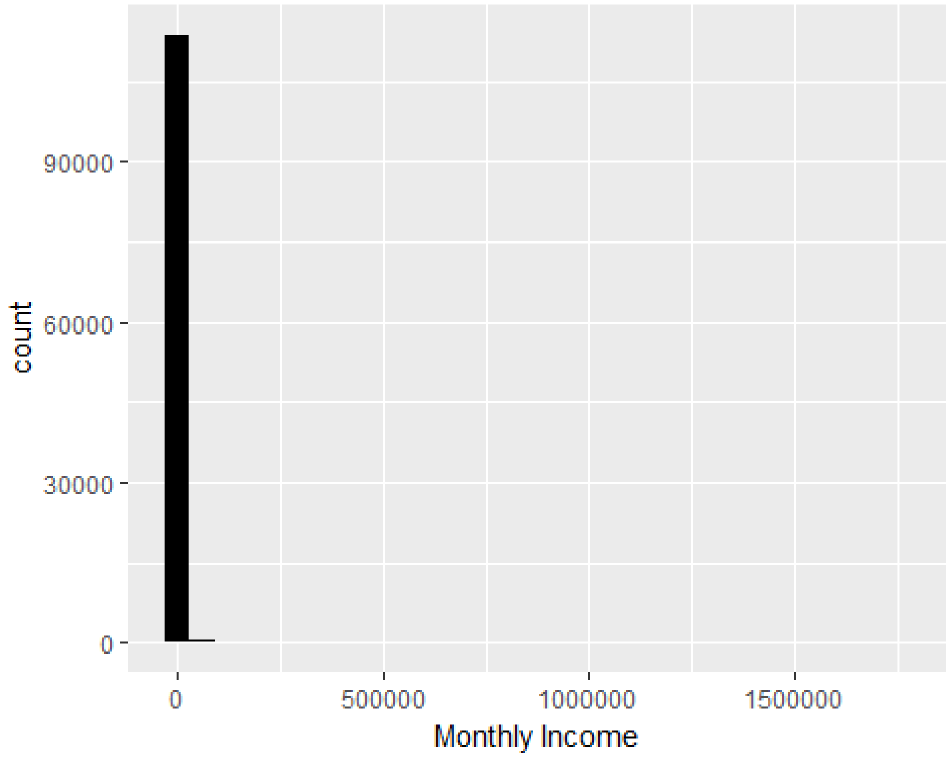


```
## [1] 0.01038444 0.12932599 0.86028957
```

Just as I thought. “High” is very far from the rest of the data, roughly seven times the size of the “Medium”. The “Low” almost disappears next to the other categorical groups, but it still has around 1% slice from the loan data set. So every 100th borrower choose from those categories that the “Low” group contains. So there is 1% chance to choose from the “Low” groups which consists of 7 categories from the 20. Small, right? Third of the categories with sum of 1%. Almost nothing.

Monthly Income

```
# Display monthly income
ggplot(aes(StatedMonthlyIncome), data = ld) +
  geom_histogram(fill = I('#000000')) +
  xlab('Monthly Income')
```

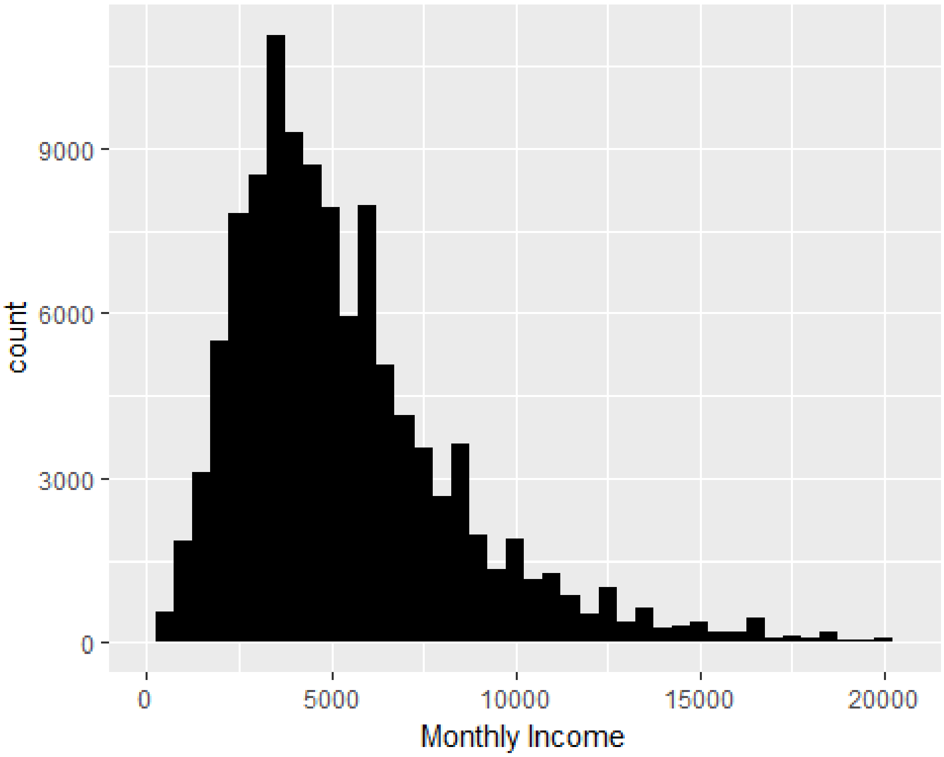


Well, there is one peak on the left but nothing more in the whole system. But why is there a 1,500,000 monthly income in the x axis? Does someone really has that much income? What is the scale for these income numbers?

```
## [1]       0 1750003
```

The maximum monthly income is 1,750,003. This completely ruins our chart. Is this real? Can it be a measurement mistake? Maybe a billionaire has this much income. Anyway, I am not going to care about this data. I am going to add a maximum scale, a 99 % quantile. This will hopefully make a lot skinnier scale, to see the other counts.

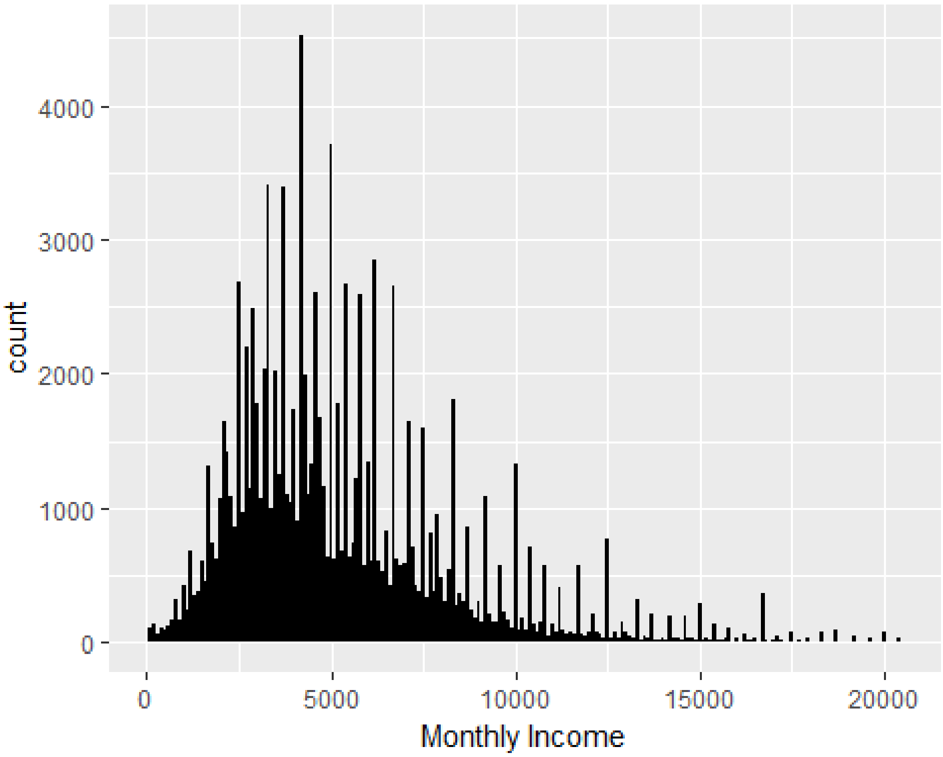
```r
# Display monthly income with a 99% quantile
ggplot(aes(StatedMonthlyIncome), data = ld) +
  geom_histogram(binwidth = 500, fill = I('#000000')) +
  scale_x_continuous(limits = c(0, quantile(ld$StatedMonthlyIncome, .99))) +
  xlab('Monthly Income')
```



Here it is! Looks better. This is a positively skewed distribution just as everyone expected it.

I expect that there are incomes that has way more counts than others. Like the very rounded numbers, for example 5000, 2500, 1000, et cetera.

```
# Display monthly income with a 99% quantile and smaller bin numbers
ggplot(aes(x = StatedMonthlyIncome), data = ld) +
  geom_histogram(binwidth = 100, fill = I('#000000')) +
  scale_x_continuous(limits = c(0, quantile(ld$StatedMonthlyIncome, .99))) +
  xlab('Monthly Income')
```

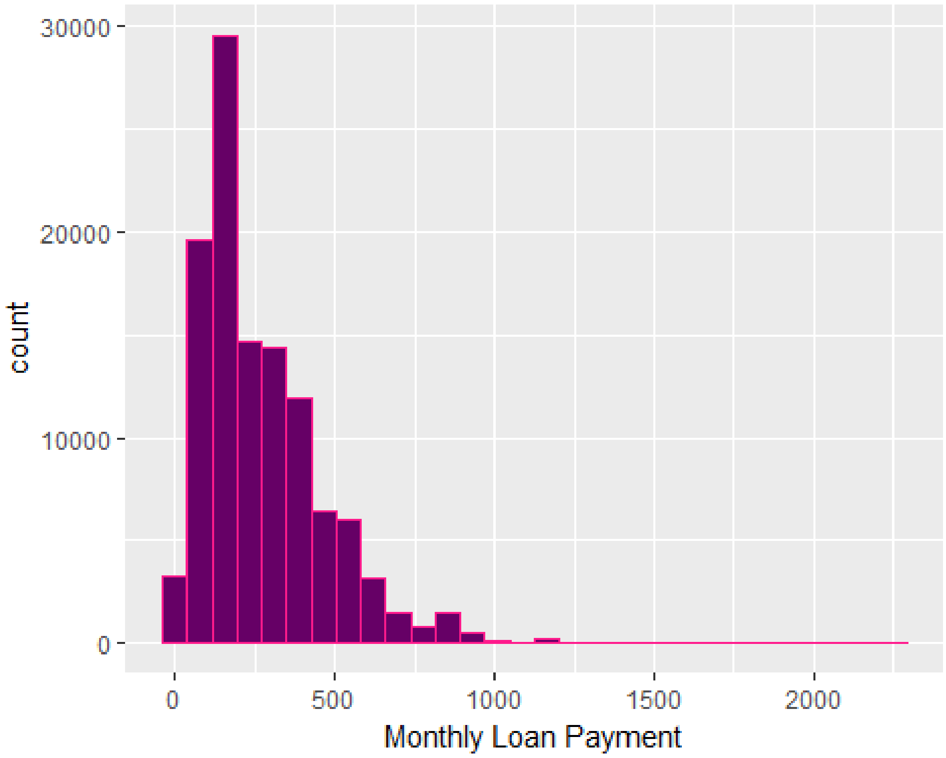


```
##
##        2500 3333.333333        3750 4166.666667 4583.333333        5000
##        2256        2917        2428        3526        2211        3389
## 5416.666667 5833.333333        6250 6666.666667
##        2374        2319        2276        2162
```

As I expected. I Listed those values where the borrower count is more than 2,000. I think values like 2500, 3750, 5000, 6250 comes from when the employee and the employer make a deal about his gross payment. They will not say that, let it be 4,996 or 4,984. They will make it 5,000. The rest comes from the net payment. For example they make it net 2000, but maybe before taxes may have been 3333 or 4166. It makes sense. The same system works on the gross and on the net income.

Monthly Loan Payment

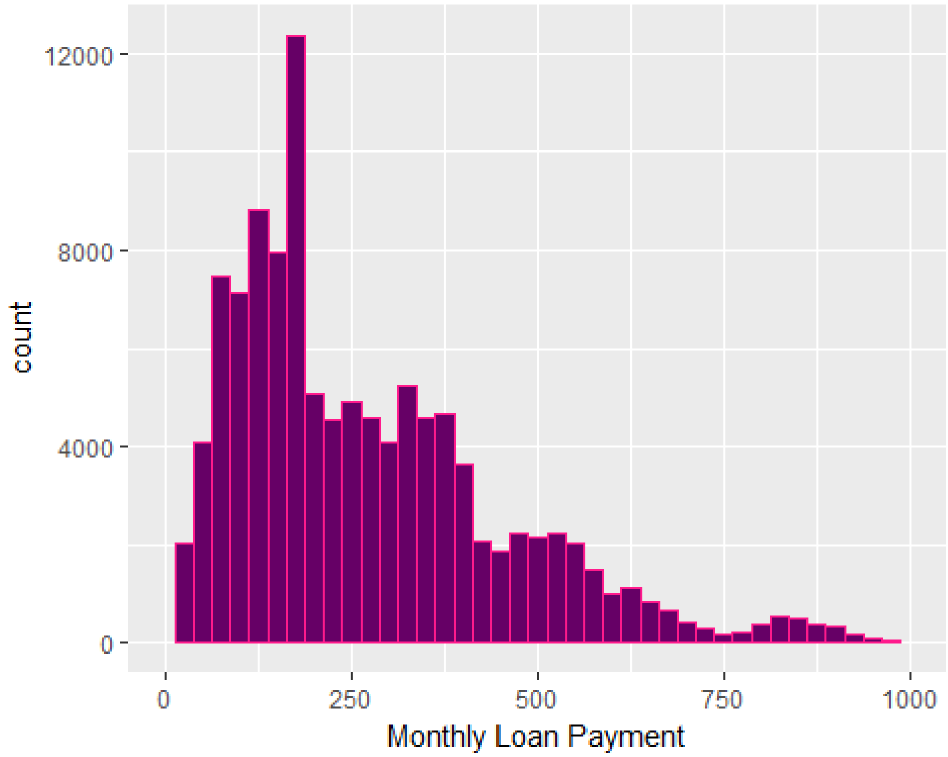
```
#Display monthly loan payment
ggplot(aes(MonthlyLoanPayment), data = ld) +
  geom_histogram(color = I('#ff1a8c'), fill = I('#660066')) +
  xlab('Monthly Loan Payment')
```



```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   131.6   217.7   272.5   371.6  2251.5
```

Positively skewed dataset. Nothing special. People rarely pay more than 1,000 in a month. Most of the data is below 220.Half of the data is between 131 and 372. I am going to scale the x axis from 0 to 1,000.

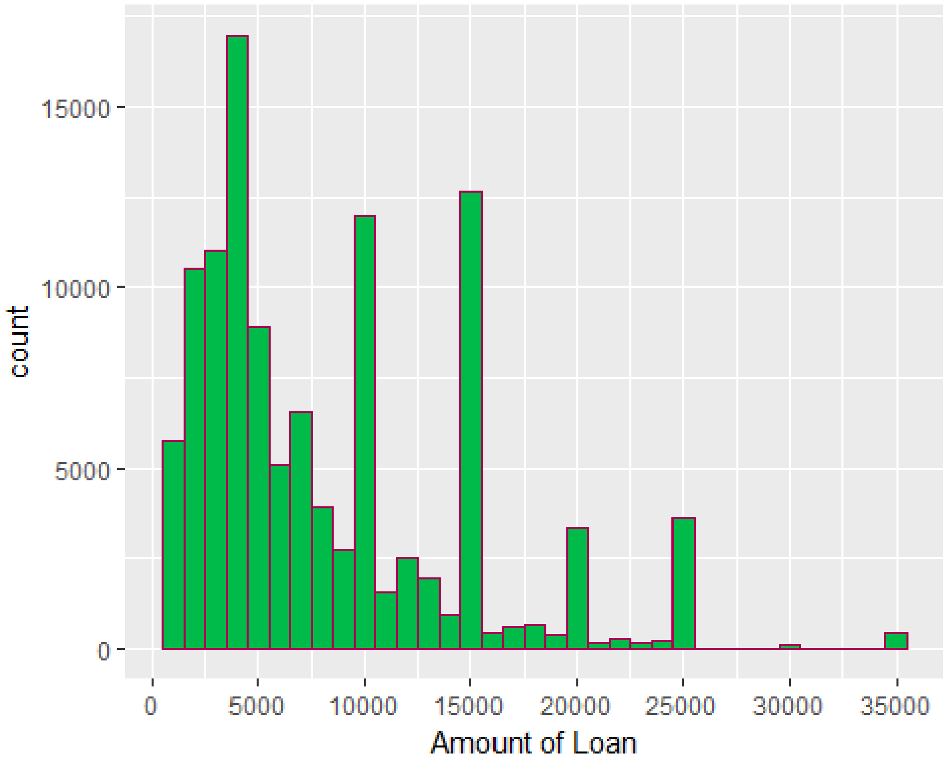
```
#Display monthly loan payment & Rescale to 1,000
ggplot(aes(x = MonthlyLoanPayment), data = ld) +
  geom_histogram(color = I('#ff1a8c'), fill = I('#660066'), binwidth = 25) +
  scale_x_continuous(breaks = seq(0,1000,50), limits = c(0,1000)) +
  xlim(0,1000) +
  xlab('Monthly Loan Payment')
```



From 0 to 175 increasing, and then decreasing till 750. There is a little peak around 850, and then it is going to be almost nothing. After 400 it suddenly falls to the half of the previous height. From the statistical numbers, half of the people is at 217, so most people stand before 220. But from the shape of the chart it is maybe better to say that the monthly amount of payment is popular until 400.

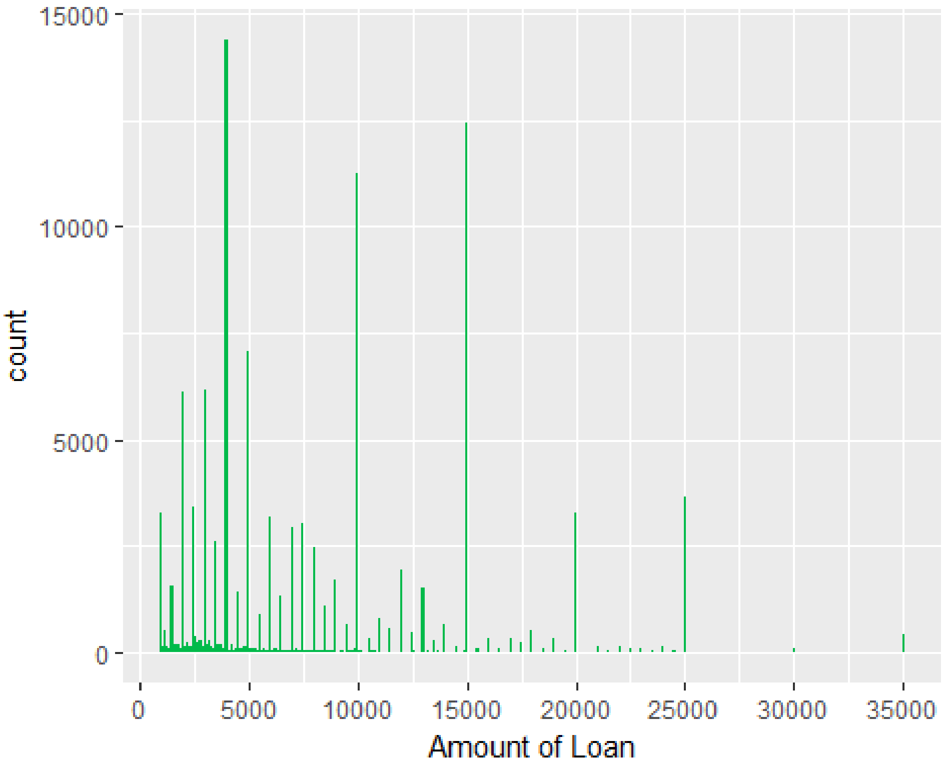
Borrowed Amount of Loan

```
# Display borrowed loan
ggplot(aes(LoanOriginalAmount), data = ld) +
  geom_histogram(fill = I('#00ba4a'), color =I('#aa0052'), binwidth = 1000) +
  scale_x_continuous(breaks = seq(0,35000,5000)) +
  xlab('Amount of Loan')
```



The shape of this distribution is positively skewed again. I explained previously the reason for the growing bins from the rounded numbers. People like to borrow monstly rounded numbers. Like 5,000, 10,000, 15,000, et cetera. To prove this, let's change the bin's width to a skinnier one.

```
# Display borrowed loan and 100 bin width
ggplot(aes(LoanOriginalAmount), data = ld) +
  geom_histogram(fill = I('#00ba4a'), binwidth = 100) +
  scale_x_continuous(breaks = seq(0,35000,5000)) +
  xlab('Amount of Loan')
```



There it is! As I expected, they grow out from those numbers. I made a mistake, the first one is not 5,000 but rather 4,000. I think those sequentially grow out, the sequential number is 1,000 or 500. Let's prove these statements with real numbers! I will list values with counts greater than or equal to 2,000.

```
ld.by_monthly_payment.tb <- table(ld$LoanOriginalAmount)

##
##  1000  2000  2500  3000  3500  4000  5000  6000  7000  7500  8000 10000
##  3206  6067  2992  5749  2567 14333  6990  2869  2949  2975  2442 11106
## 15000 20000 25000
## 12407  3291  3630
```

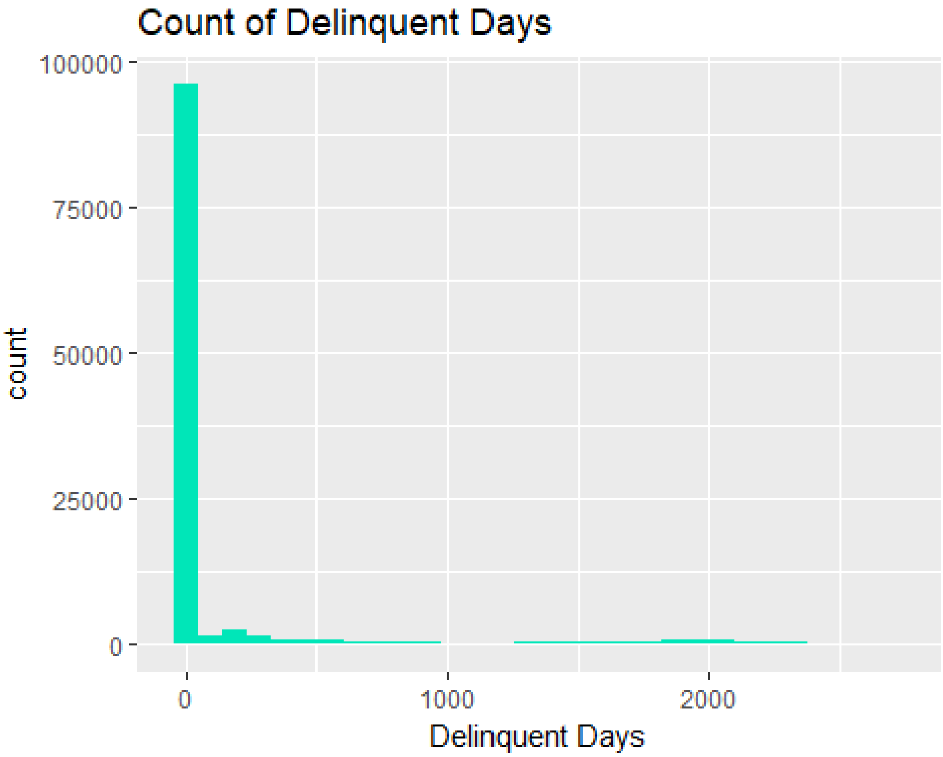
So what we can tell is, people borrow very rounded numbers. From 1,000 to 5,000 the biggest step size is 1,000, the smallest is 500. 4,000 borrower count is way more than 5,000. Maybe they think 5,000 is too much money to borrow, let's borrow only 4,000. Even if they need more, they just can't risk not being able to pay it back. Very few people borrow 4,500, maybe because it is too close to 5000 or 4000 and people that, it is that close to 4-5,000, maybe they should go for the 4,000 or the 5,000. And they just rethink that 5,000 is too much, 4,500 is not that far from it, and maybe it is much too. So let's go back to 4,000, it is safer.

From 5,000 to 10,000 step size is a 1,000. They are not going for the X,500. They like rounded numbers. 9,000 is too low. Who borrow 10,000 possibly have more money, so they can go for the 10,000. It is a nicer number. 7,500 is also a nice number, from 10,000 to

35,000 the step size is 5,000. Anyone who borrow from that range won't just go for, let's say 32,000, they will go for 30,000 or 35,000.

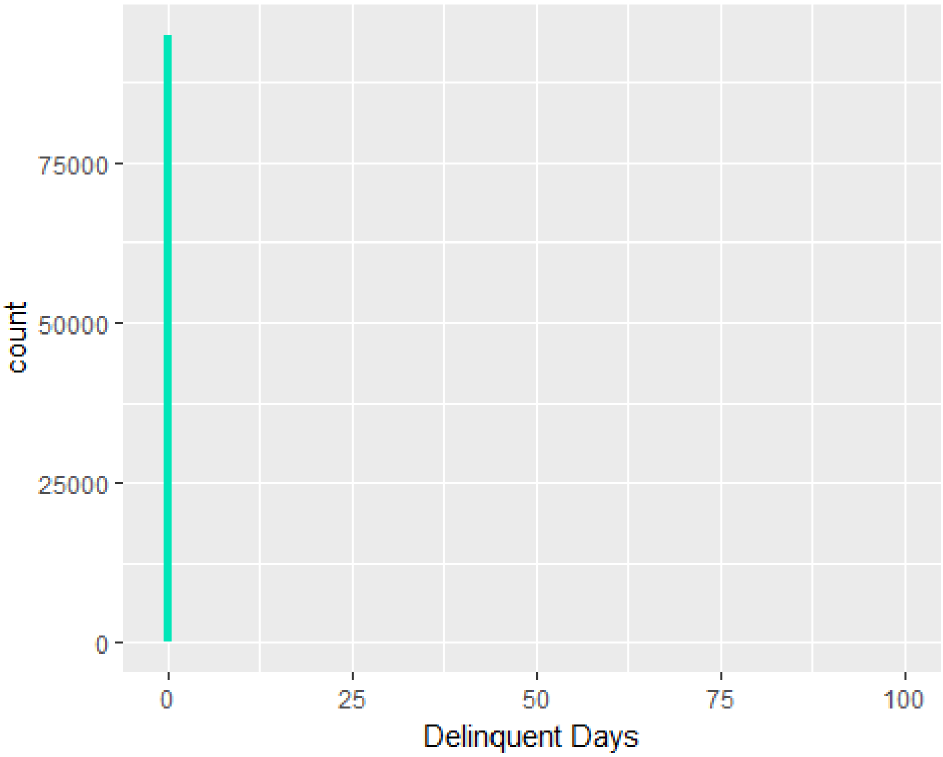
Current Delinquent Days

```r
# Display delinquent days
ggplot(aes(ld$LoanCurrentDaysDelinquent), data = ld) +
  geom_histogram( fill =I('#00e6b8')) +
  ggtitle("Count of Delinquent Days") +
  xlab("Delinquent Days")
```



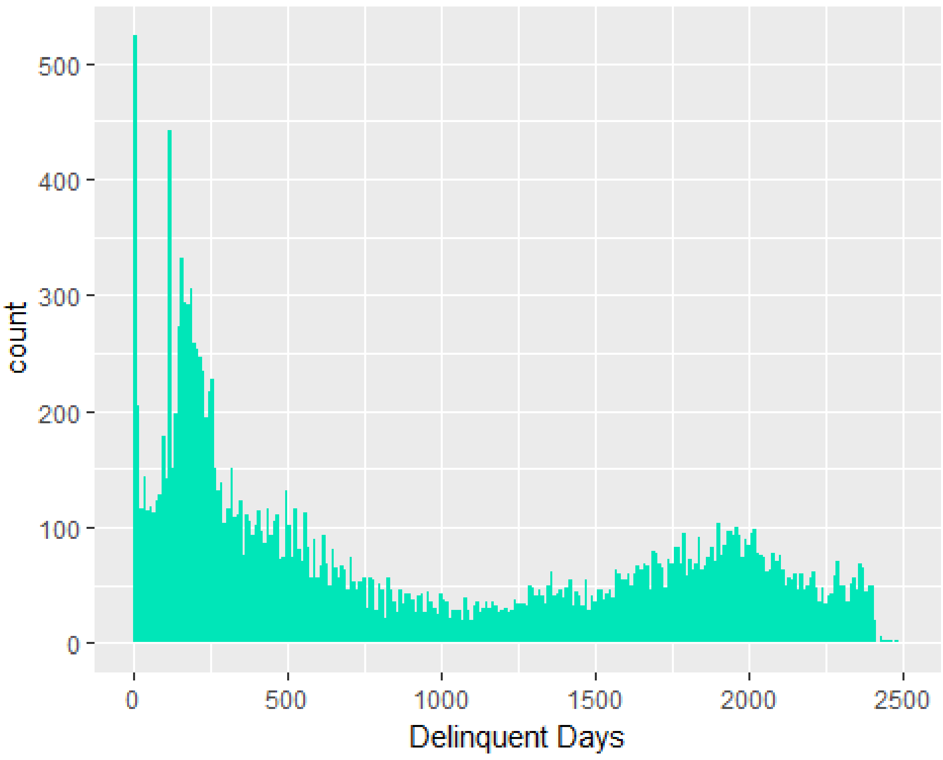
100,000 counted column at the start destroys our chart. Let's examine what it is.

```r
# Display delinquent days and rescale
ggplot(aes(ld$LoanCurrentDaysDelinquent), data = ld) +
  geom_histogram(binwidth = 1,  fill =I('#00e6b8')) +
  scale_x_continuous(limits = c(-1,100)) +
  xlab("Delinquent Days")
```



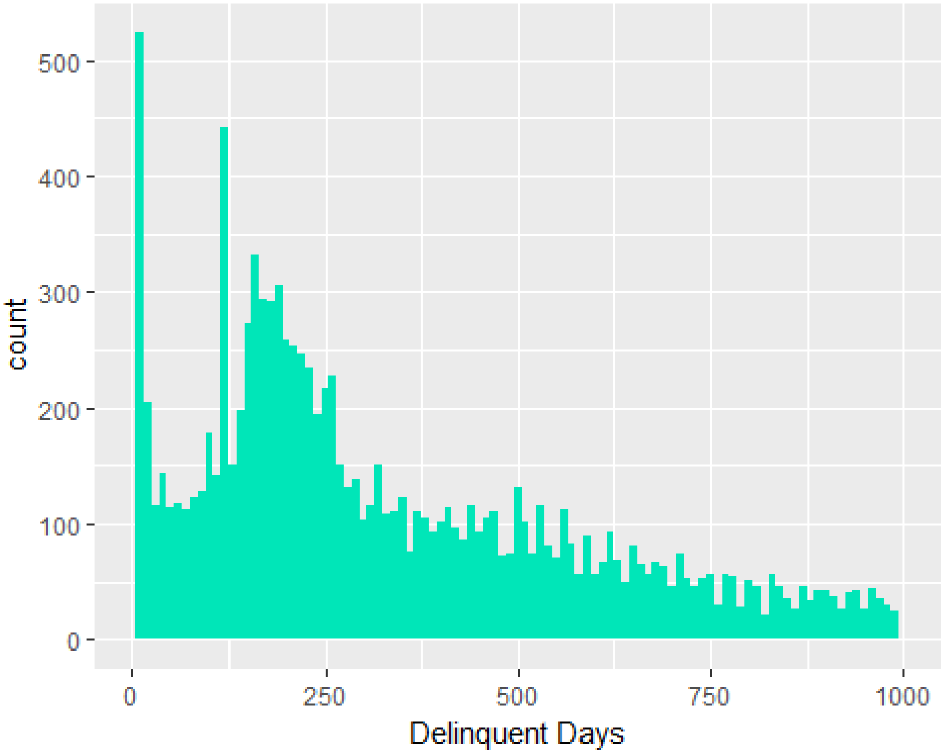
It is a zero. from 110,000 people, 90,000 pays his debt in time. Very good. We should not bother ourselves, just delete it from the chart to see the others.

```
# Display delinquent days without zero
ggplot(aes(ld$LoanCurrentDaysDelinquent), data = ld) +
  geom_histogram(binwidth = 10,  fill =I('#00e6b8')) +
  scale_x_continuous(limits = c(1,2500)) +
  xlab("Delinquent Days")
```



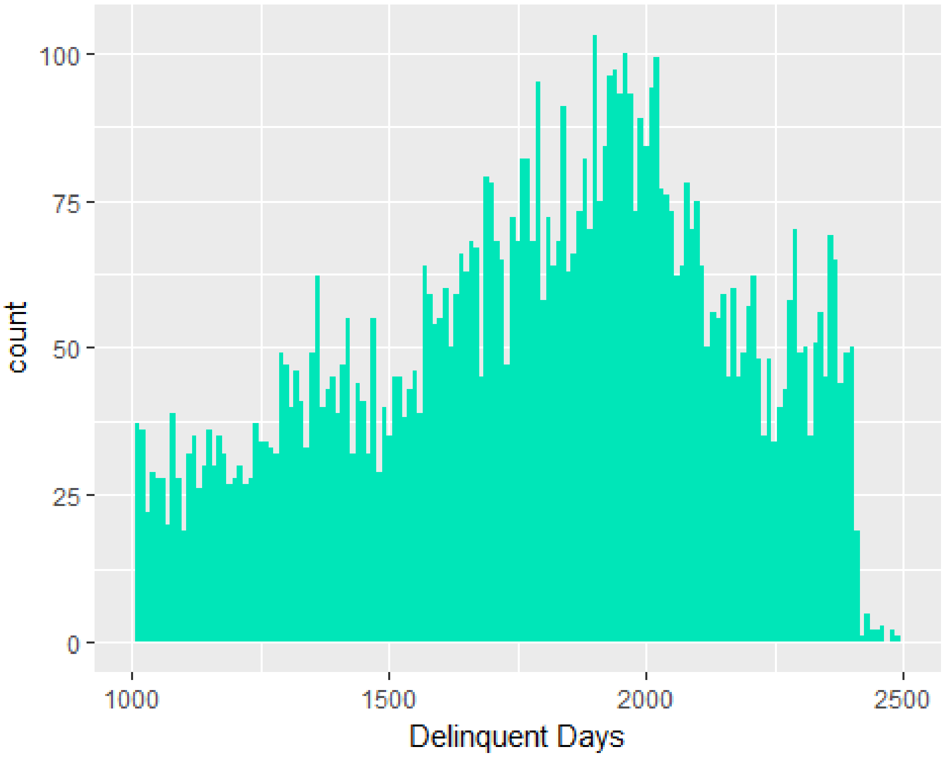
There are 2 peaks in the data. One at the beginning, and one around 2,000. We can split our data set into two parts. [1;1000] &]1000;2500].

```
# Display delinquent days from 1 to 1,000
ggplot(aes(ld$LoanCurrentDaysDelinquent), data = ld) +
  geom_histogram(binwidth = 10,  fill =I('#00e6b8')) +
  scale_x_continuous(limits = c(1,1000)) +
  xlab("Delinquent Days")
```



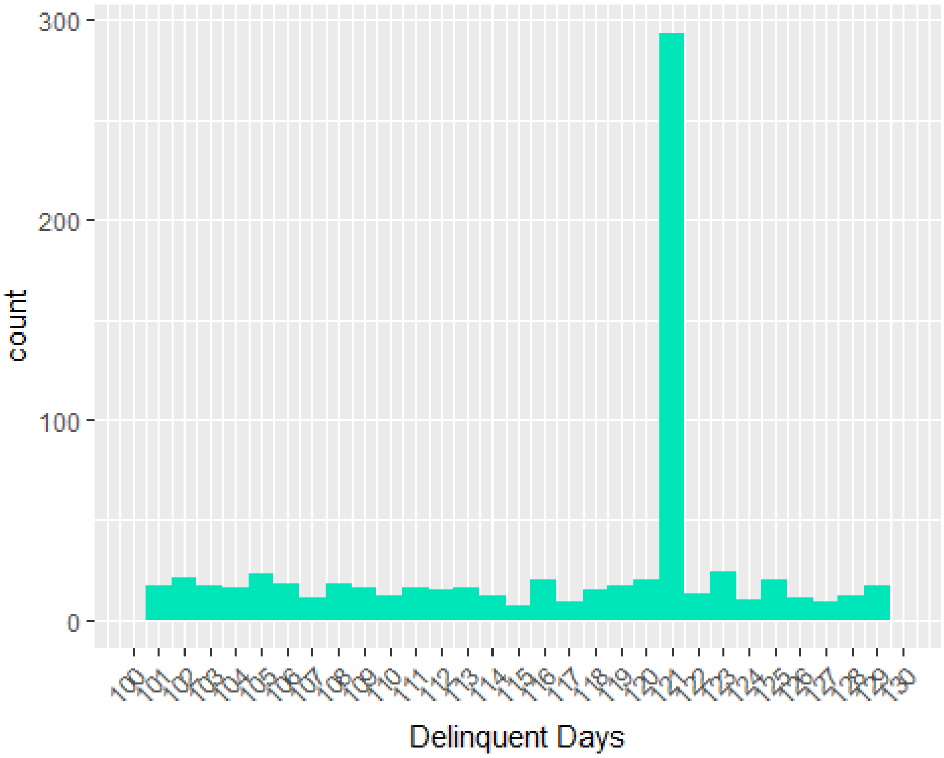
There are some who late for a few days, after that it goes back to normal. At ~125 there is one outstanding bin.

```
# Display delinquent days from 1,001 to 2,500
ggplot(aes(ld$LoanCurrentDaysDelinquent), data = ld) +
  geom_histogram(binwidth = 10,  fill =I('#00e6b8')) +
  scale_x_continuous(limits = c(1001,2500)) +
  xlab("Delinquent Days")
```



Something bothers me, are these outstanding bins sequentially repeated? This can be seen in the first part and in the second part I'll examine this later, but first let's find out what is that strange bin at ~125.

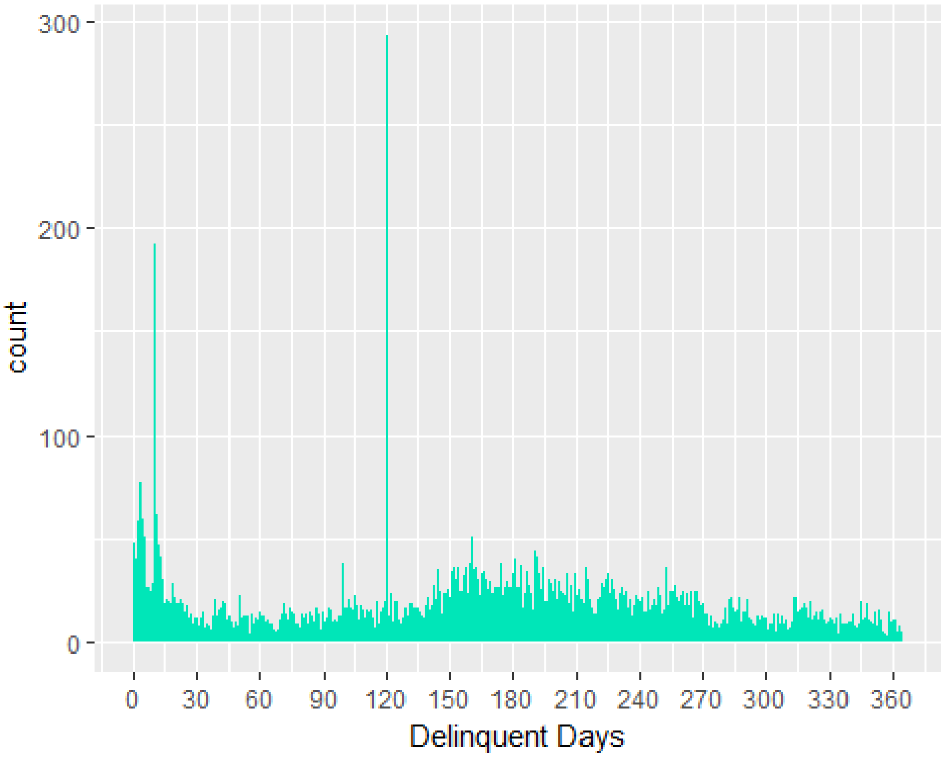
```
# Display delinquent days, investigate outstanding bin at around 125
ggplot(aes(ld$LoanCurrentDaysDelinquent), data = ld) +
  geom_histogram(binwidth = 1,  fill =I('#00e6b8')) +
  scale_x_continuous(limits = c(100,130), breaks = seq(100,130,1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1)) +
  xlab("Delinquent Days")
```



It is at 121. What is it? It is a quarter of a year. Maybe after 1 quarter and 1 day they have to pay late charges. So people who don’t really have money, wait until the last day.

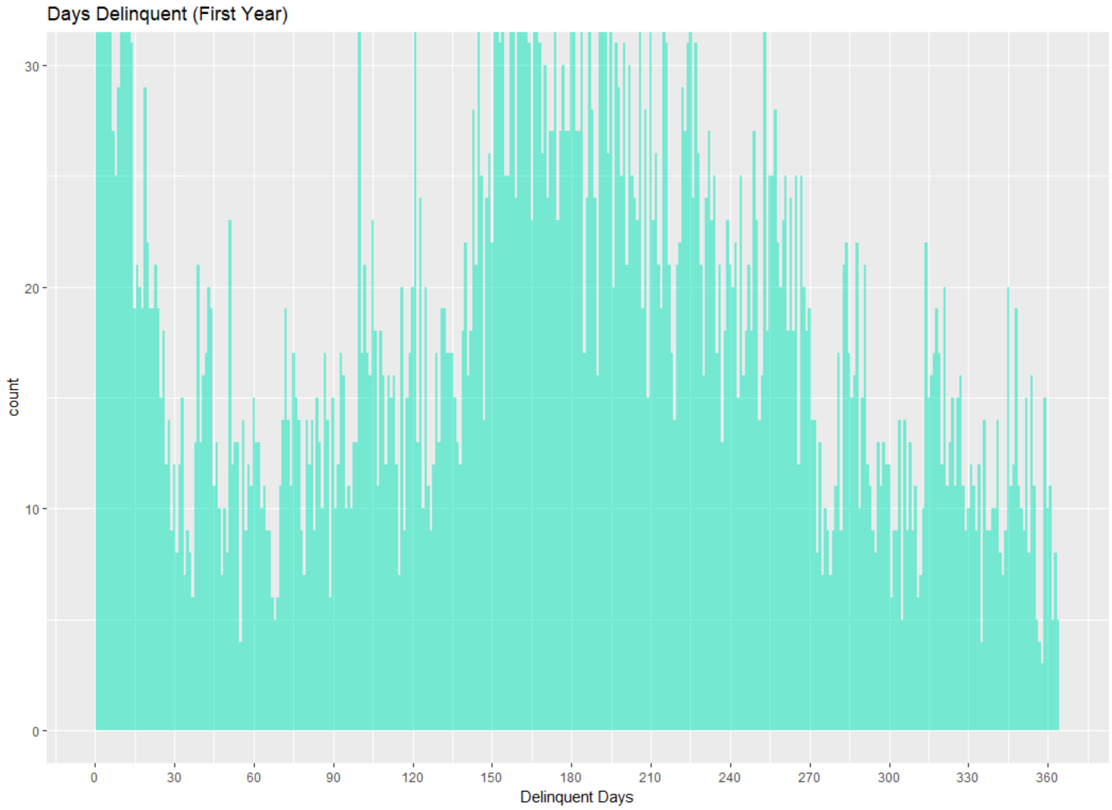
And now let’s examine the sequence.

```
# Display delinquent days
#Investigate outstanding bin sequency
#First year without 0
ggplot(aes(x =LoanCurrentDaysDelinquent),
       data = subset(ld, LoanCurrentDaysDelinquent >0)) +
  geom_histogram(binwidth =1, alpha = 1, fill =I('#00e6b8')) +
  scale_x_continuous(breaks = seq(0,365,30), limits = c(0,365)) +
  xlab("Delinquent Days")
```



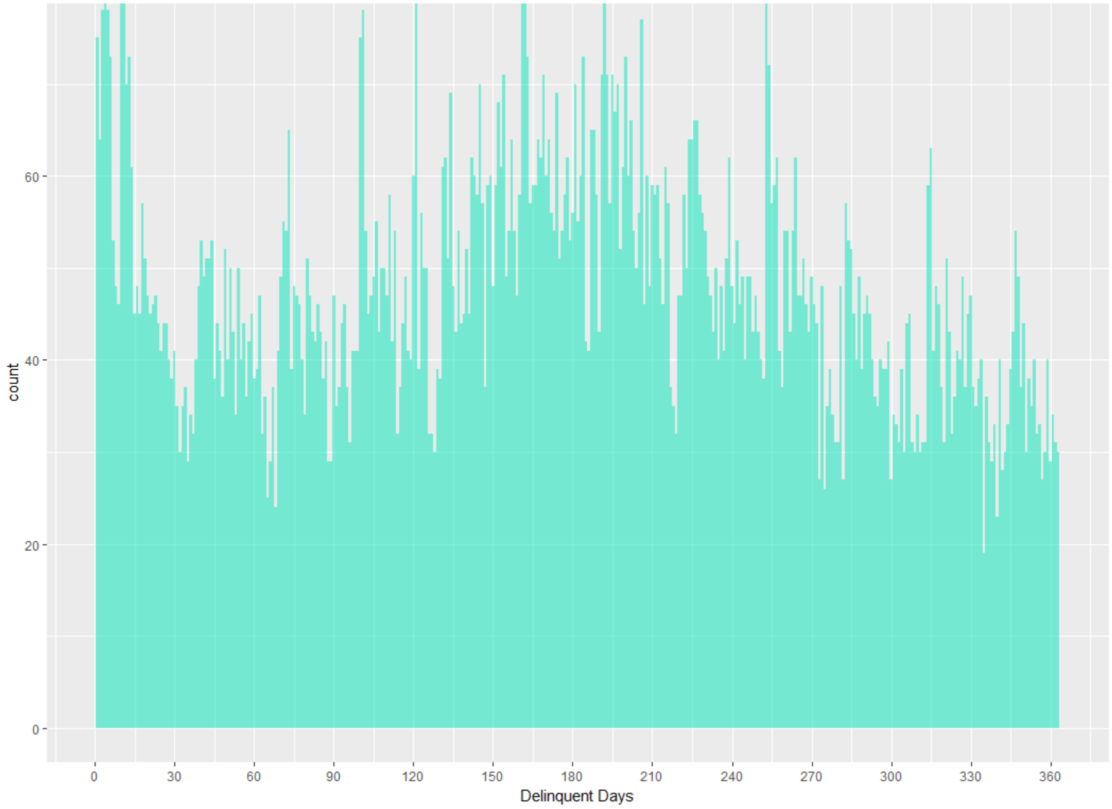
Not a really good view, value 121 is too high. Let's zoom in. And it would be better if I would change the chart's opacity value lower to better pair the downsizes to the x axis and also make the chart wider for a more precise examination.

```
# Display delinquent days
#Investigate outstanding bin sequency
#First year without 0
# Zoom in
ggplot(aes(x =LoanCurrentDaysDelinquent), data = ld) +
  geom_histogram(binwidth =1, alpha =.5, fill =I('#00e6b8')) +
  scale_x_continuous(breaks = seq(0,365,30), limits = c(0,365)) +
  coord_cartesian(ylim = c(0,30)) +
  ggtitle('Days Delinquent (First Year)') +
  xlab("Delinquent Days")
```



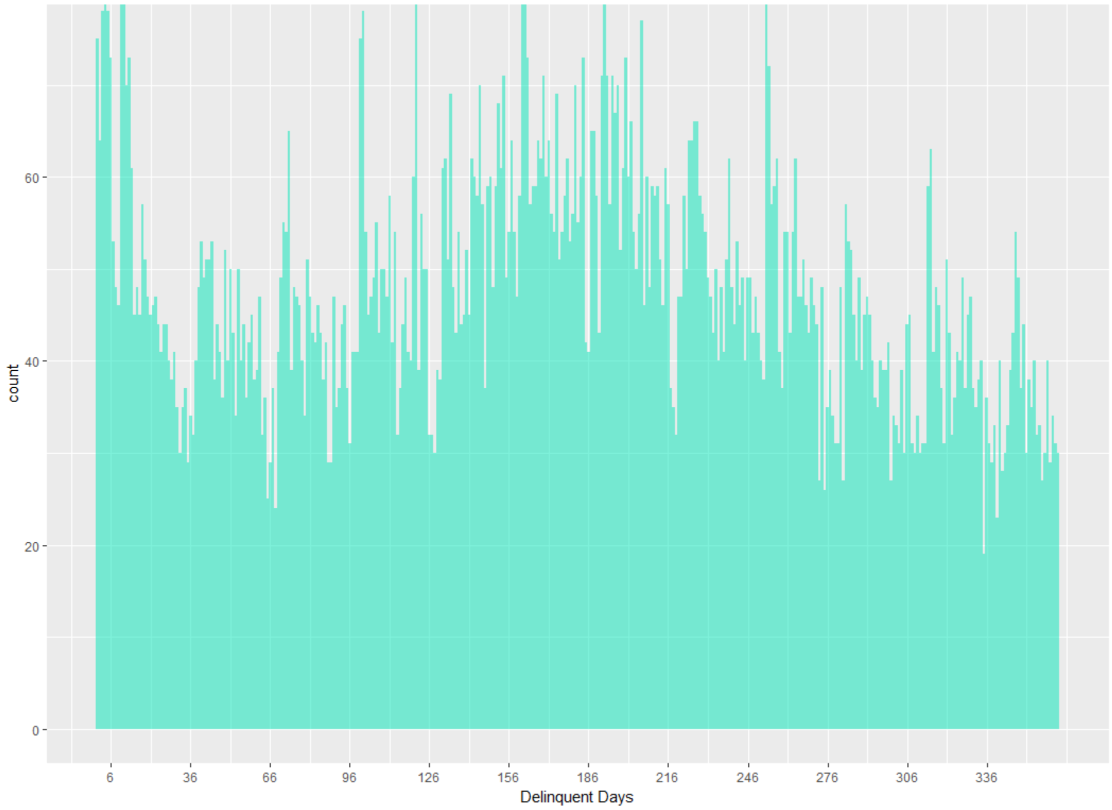
It is hard to recognize exactly where the data goes down on the x axis. I do not see any strong sequential in the downsizes. Maybe there is some, but I cannot tell. Let’s examine this from 0 to 364 interval in the first six years by apply a division with remainder. Maybe it will refine the chart “curves”.

```
# Display delinquent days
#Investigate outstanding bin sequency
#6 years (division with remainder)
# Zoom in
ggplot(aes(x =LoanCurrentDaysDelinquent%% 365, y=..count.., fill =..count..),
       data = subset(ld, LoanCurrentDaysDelinquent > 0 &
                        LoanCurrentDaysDelinquent< 2190)) +
  geom_histogram(binwidth = 1, alpha =.5, fill =I('#00e6b8')) +
  scale_x_continuous(limits = c(0,364), breaks = seq(0,364,30)) +
  coord_cartesian(ylim = c(0,75)) +
  xlab("Delinquent Days")
```



Yes, it made the decreases more clear. What i see is that there are decreases after a couple of days from the 30 multiplications. I want to be more precise and show them better by changing the x axis scale labels.

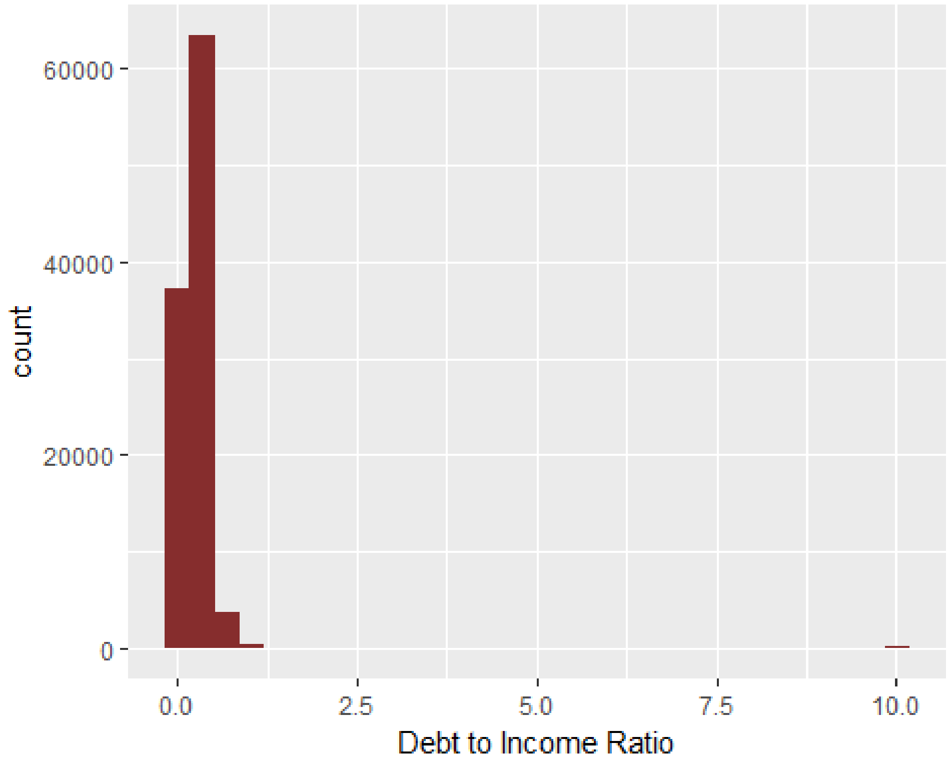
```
# Display delinquent days
#Investigate outstanding bin sequency
#First year without 0
# Zoom in
# Rescale x-axis
ggplot(aes(x =LoanCurrentDaysDelinquent%% 365, y=..count.., fill =..count..),
       data = subset(ld, LoanCurrentDaysDelinquent > 0 &
                       LoanCurrentDaysDelinquent< 2190)) +
  geom_histogram(binwidth = 1, alpha =.5, fill =I('#00e6b8')) +
  scale_x_continuous(limits = c(0,364), breaks = seq(6,364,30)) +
  coord_cartesian(ylim = c(0,75)) +
  xlab("Delinquent Days")
```



Here it is. We can see the downsizes at the x-axis breaks.

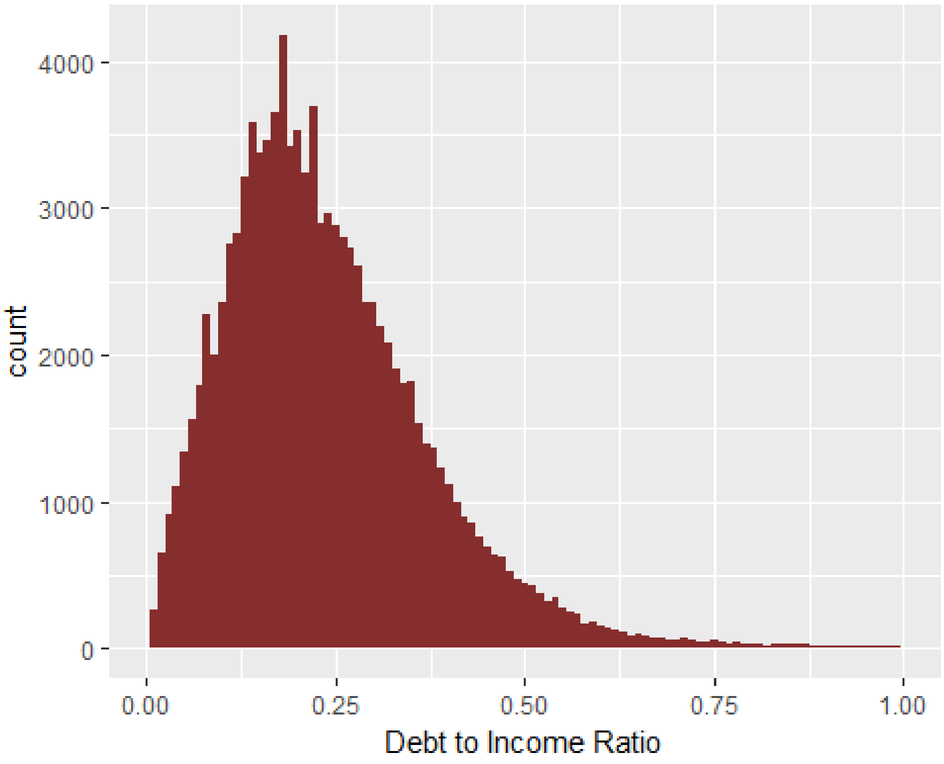
Debt to Income Ratio

```
# Display Debt to Income Ratio
ggplot(aes(ld$DebtToIncomeRatio), data = ld)+
  geom_histogram(fill = I('#862d2d')) +
  xlab('Debt to Income Ratio')
```



Wow! someone has 10 times more debt than his income. Thanks for him, we have to rescale the x axis again. Let's do it.

```
# Display Debt to Income Ratio
# Limit x-axis
ggplot(aes(ld$DebtToIncomeRatio), data = ld)+
  geom_histogram(fill = I('#862d2d'), binwidth = .01) +
  scale_x_continuous(limits = c(0,1), breaks = seq(-.25,1,.25)) +
  xlab('Debt to Income Ratio')
```



```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   0.140   0.220   0.276   0.320  10.010    8554 
```

What do we have here? A positively skewed distribution. Most of the data is between 0 and 0.5. Median is 0.22, so half of the people pay less than or equal to 22% of their monthly income. That is a very acceptable rate.

Univariate Analysis

What is the structure of your dataset?

There are 113,837 observations with 82 variables. I am not investigating all of the variables, just the important ones from them, like prosper rate, interest rate, “is borrower a homeowner” et cetera. There are categorical, numerical variables. And also date formats, but I will not examine them.

What is/are the main feature(s) of interest in your dataset?

The main features in this data set are the interest rates, prosper rating and the borrowed loan. I’m sure that prosper rating has an influence to the interest rate. And maybe some other variables, like the amount of the borrowed loan have too.

What other features in the dataset do you think will help support your investigation into your feature(s) of interest?

My assumption is, the “term” and the “is borrower a homeowner” will help me later.

Did you create any new variables from existing variables in the dataset?

I changed the “IsBorrowerHomeowner” factors from True and False to “Homeowner” and “NotHomeowner”. Because it is easier to plot.

Of the features you investigated, were there any unusual distributions? Did you perform any operations on the data to tidy, adjust, or change the form of the data? If so, why did you do this?

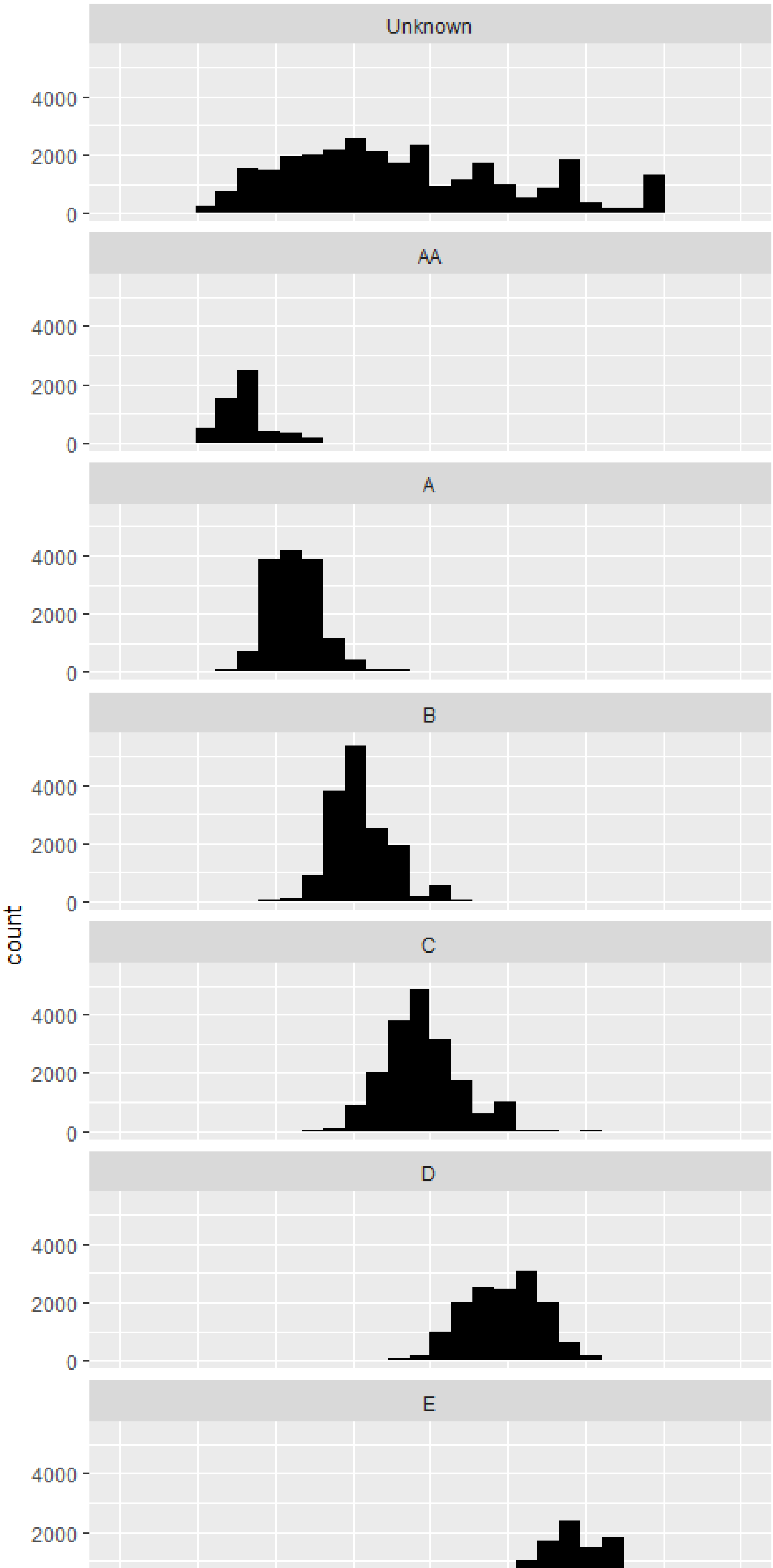
No, I did not.

Bivariate Plots Section

Bivariate Analysis

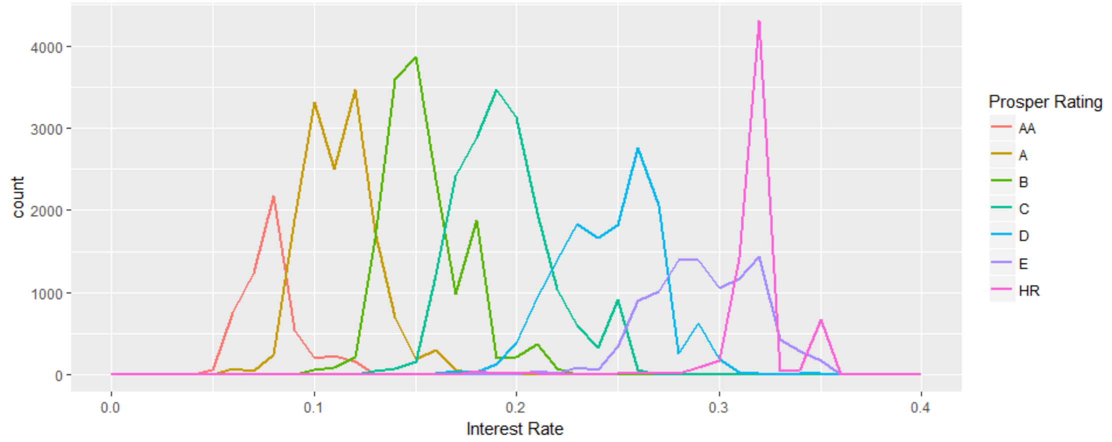
Connection Between Interest Rate & Prosper Rating

```r
# Interest rate & Prosper ratings
ggplot(aes(BorrowerRate), data = ld) +
  geom_histogram(fill = 'black') +
  facet_wrap(~ProsperRating..Alpha., ncol = 1) +
  scale_x_continuous(limits = c(0,.4)) +
  xlab('Interest Rate')
```


Unknown
AA
A
B
C
D
E
count
4000
2000
0

As watching this plot, we can see how the distribution positions move to the right as we go down to a lower rated category. Let’s compare them in one plot without the unknown data.

```
# Interest rate & Prosper ratings
ggplot(aes(BorrowerRate),
       data = subset(ld, ld$ProsperRating..Alpha. != "Unknown")) +
  geom_freqpoly(aes(color = ProsperRating..Alpha.), size = 1, binwidth = .01)
+
  scale_x_continuous(limits = c(0,.4)) +
  xlab('Interest Rate')+
  labs(color ="Prosper Rating")
```

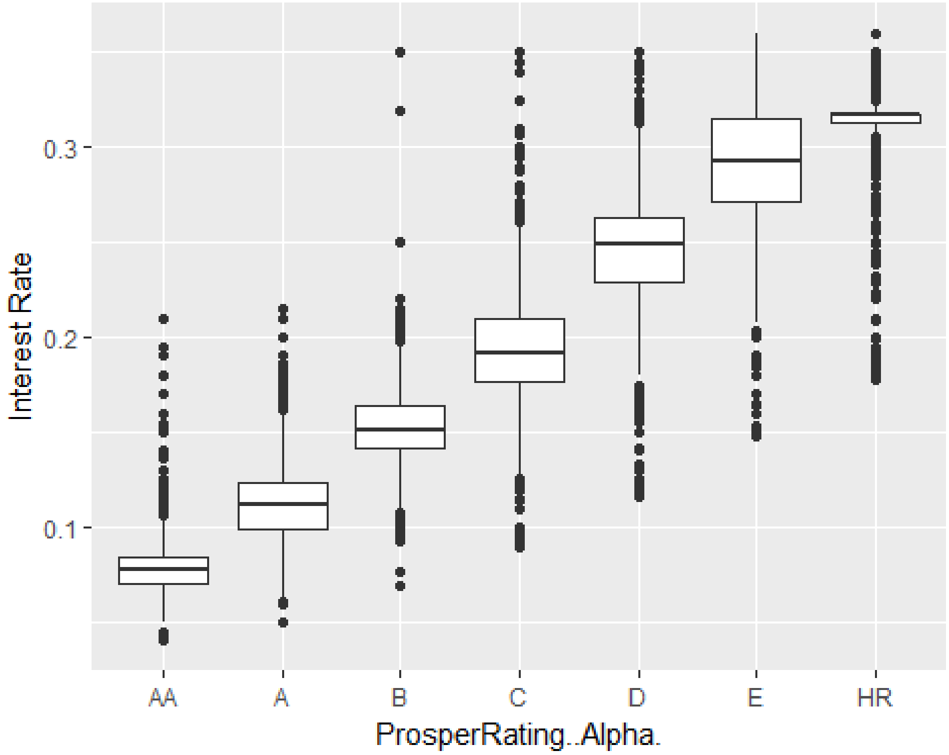


I also removed the Unknown data so it won’t be crossing every single distribution and ruin the sight.

```
## ld$ProsperRating..Alpha.: Unknown
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.1269  0.1700  0.1833  0.2364  0.4975 
## -------------------------------------------------------- 
## ld$ProsperRating..Alpha.: AA
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.04000 0.06990 0.07790 0.07912 0.08450 0.21000 
## -------------------------------------------------------- 
## ld$ProsperRating..Alpha.: A
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0498  0.0990  0.1119  0.1129  0.1239  0.2150 
## -------------------------------------------------------- 
## ld$ProsperRating..Alpha.: B
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0693  0.1414  0.1509  0.1545  0.1639  0.3500 
## -------------------------------------------------------- 
## ld$ProsperRating..Alpha.: C
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0895  0.1765  0.1914  0.1944  0.2099  0.3500 
## -------------------------------------------------------- 
## ld$ProsperRating..Alpha.: D
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1157  0.2287  0.2492  0.2464  0.2625  0.3500
## --------------------------------------------------------
## ld$ProsperRating..Alpha.: E
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1479  0.2712  0.2925  0.2933  0.3149  0.3600
## --------------------------------------------------------
## ld$ProsperRating..Alpha.: HR
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1779  0.3134  0.3177  0.3173  0.3177  0.3600
```

If we get a better Prosper rating, the lower the Interest rates will be. There is an interesting thing about these ratings. From B to HR the maximum interest rate is almost equal. But the ‘AA’ and ‘A’ maximum interest rate is way lower than the others. HR category is a very thin distribution. Q1 is 0.3134, Q3 is 0.3177. So 50% of the data is between this two numbers, with range of only 0.0043. Let’s show this statistical data in a plot.

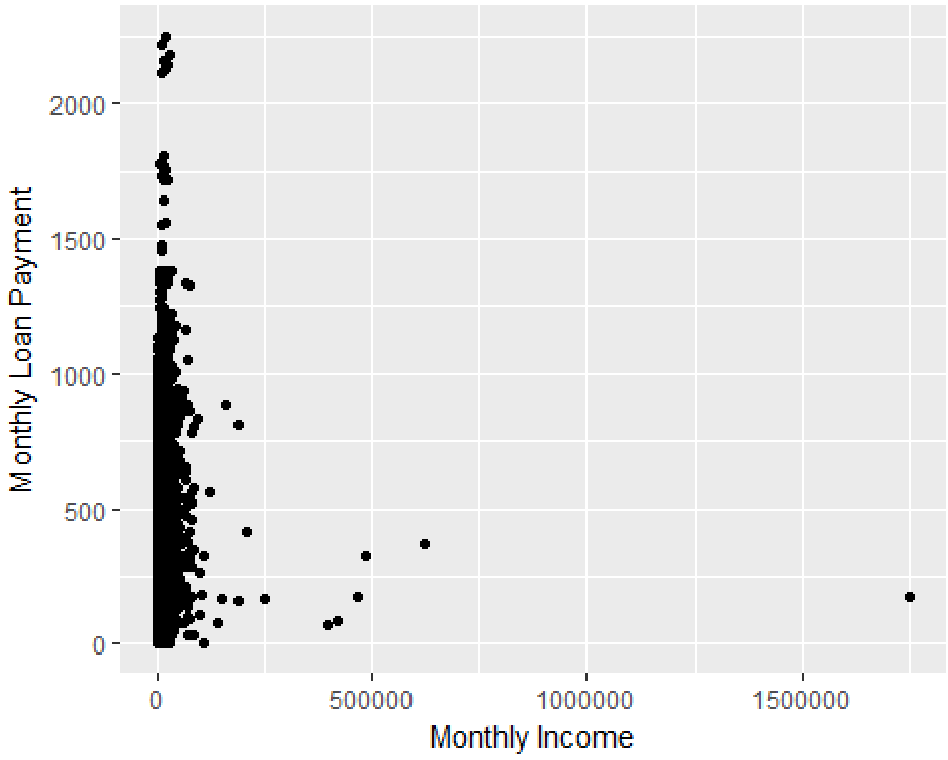
```
# Interest rate & Proper rating
ggplot(subset(ld, ld$ProsperRating..Alpha. != "Unknown"),
       aes(ProsperRating..Alpha., BorrowerRate)) +
  geom_boxplot() +
  ylab('Interest Rate')
```



This plot describes the tendency very well. Median, Q1 and Q3 grow every time. More outlier goes up than down until D where it changes this routine into its opposite direction. This shows us very well, how thin the HR category is.

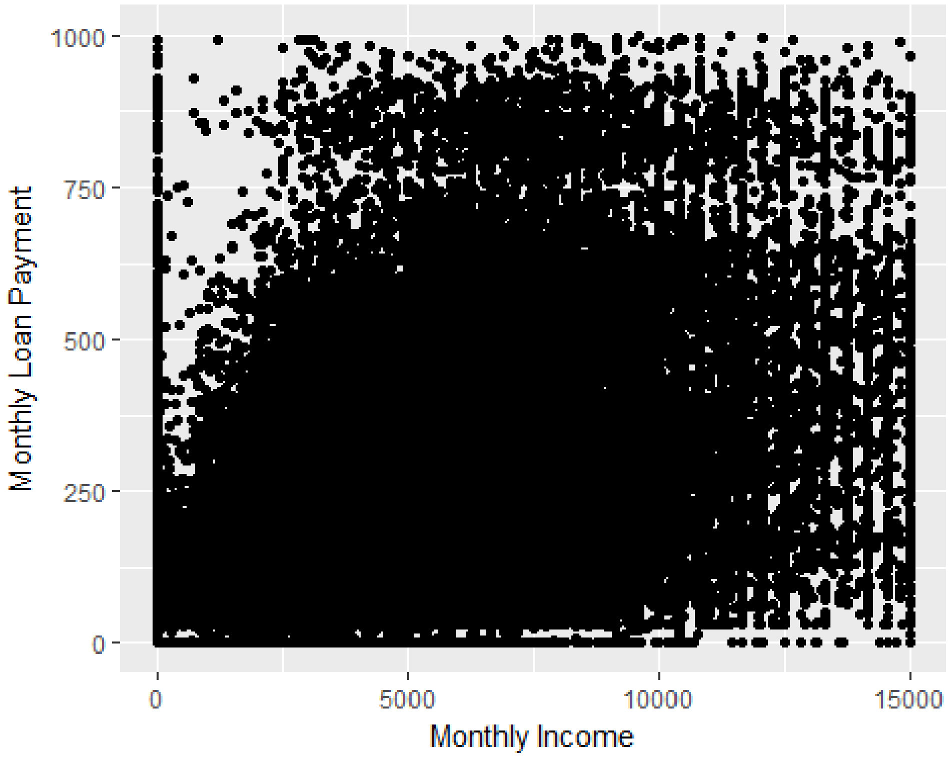
Monthly Income Vs Loan Payment

```
# Monthly Income & monthly Loan Payment
ggplot(ld, aes(ld$StatedMonthlyIncome, ld$MonthlyLoanPayment)) +
  geom_point() +
  ylab("Monthly Loan Payment") +
  xlab("Monthly Income")
```



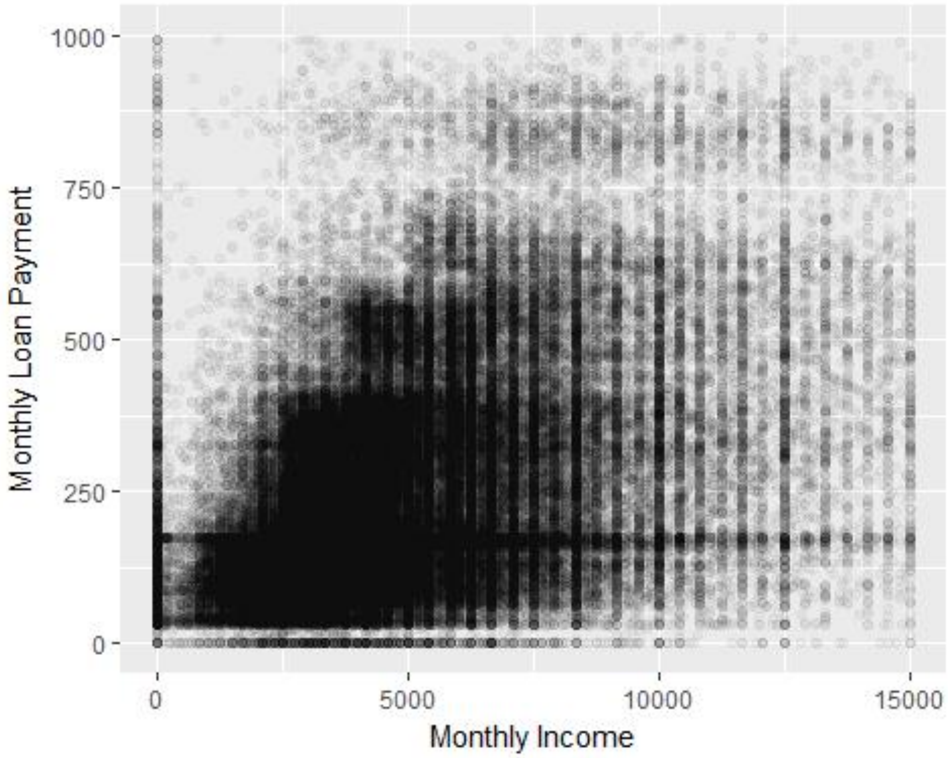
There are few values really far from others. Let’s add a limit to the x-axis, to see that group on the left.

```
# Monthly Income & monthly Loan Payment
# Limit x axis
ggplot(ld, aes(ld$StatedMonthlyIncome, ld$MonthlyLoanPayment)) +
  geom_point() +
  scale_x_continuous(limits = c(0,15000)) +
  ylim(c(0,1000)) +
  ylab("Monthly Loan Payment") +
  xlab("Monthly Income")
```



Interesting. Why is there 2 groups on the plot(one big in the middle and one upwards)? Maybe later i will get an answer for this by adding some other variables to the plot. But first to get a better idea about the values, change the alpha level.

```
# Monthly Income & monthly Loan Payment
# Limit x axis
# Lower alpha level
ggplot(ld, aes(ld$StatedMonthlyIncome, ld$MonthlyLoanPayment)) +
  geom_point(alpha = 1/30) +
  scale_x_continuous(limits = c(0,15000)) +
  ylim(c(0,1000)) +
  ylab("Monthly Loan Payment") +
  xlab("Monthly Income")
```

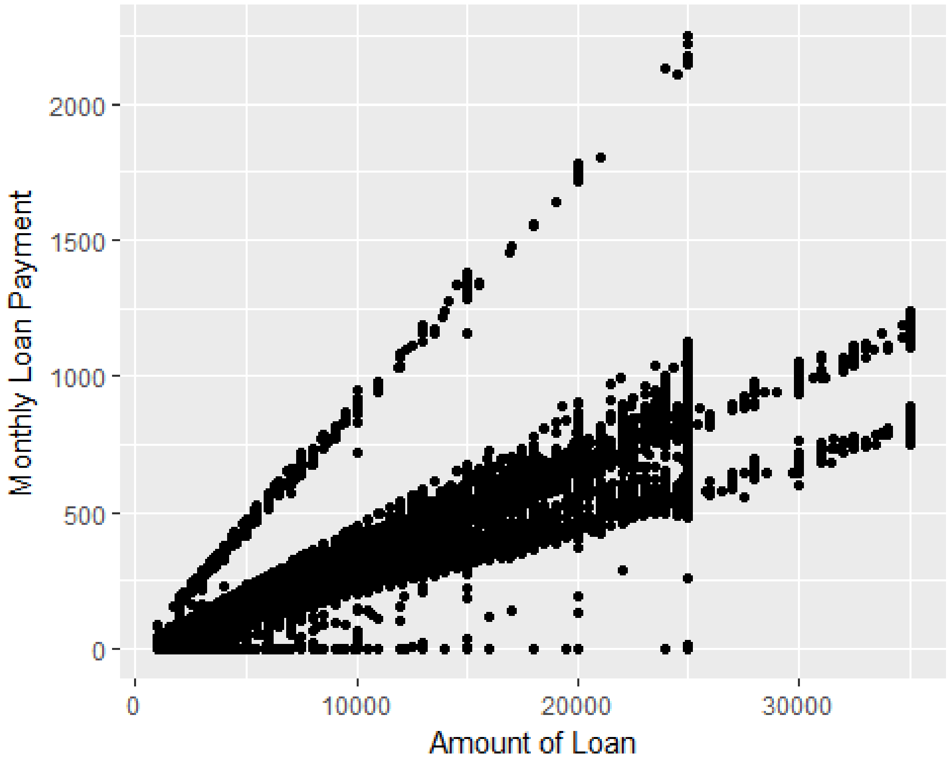


There is a stairway looking growth between Income and Debt Payment. As income grows, occasionally debt payment grows too. There is not a strong relationship in the plot, let's see what the correlation is.

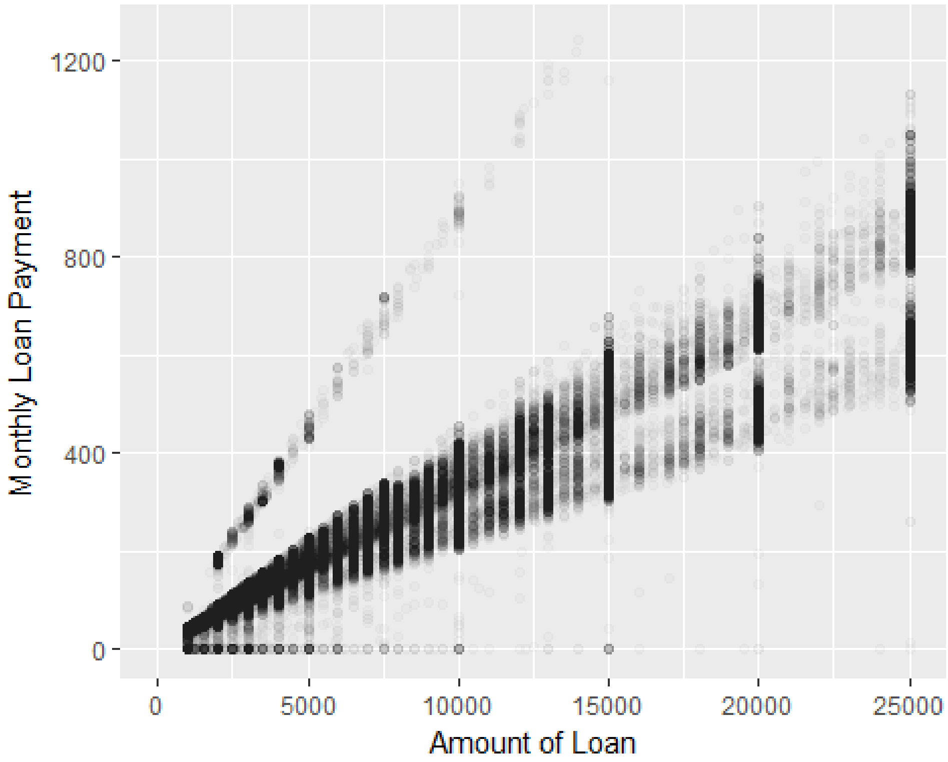
```
##
##  Pearson's product-moment correlation
##
## data:  ld$StatedMonthlyIncome and ld$MonthlyLoanPayment
## t = 67.764, df = 113940, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1912423 0.2024055
## sample estimates:
##       cor
## 0.1968303
```

Borrowed Money and Monthly Loan Payment

```
#  Borrowed Money and Monthly Loan Payment
ggplot(ld, aes(ld$LoanOriginalAmount, ld$MonthlyLoanPayment)) +
  geom_point() +
  ylab("Monthly Loan Payment") +
  xlab("Amount of Loan")
```



```
# Borrowed Money and Monthly Loan Payment
# change alpha level
ggplot(ld, aes(ld$LoanOriginalAmount, ld$MonthlyLoanPayment)) +
  geom_point(alpha =1/60) +
  xlim(c(0,25000)) +
  ylim(c(0,1250)) +
  ylab("Monthly Loan Payment") +
  xlab("Amount of Loan")
```



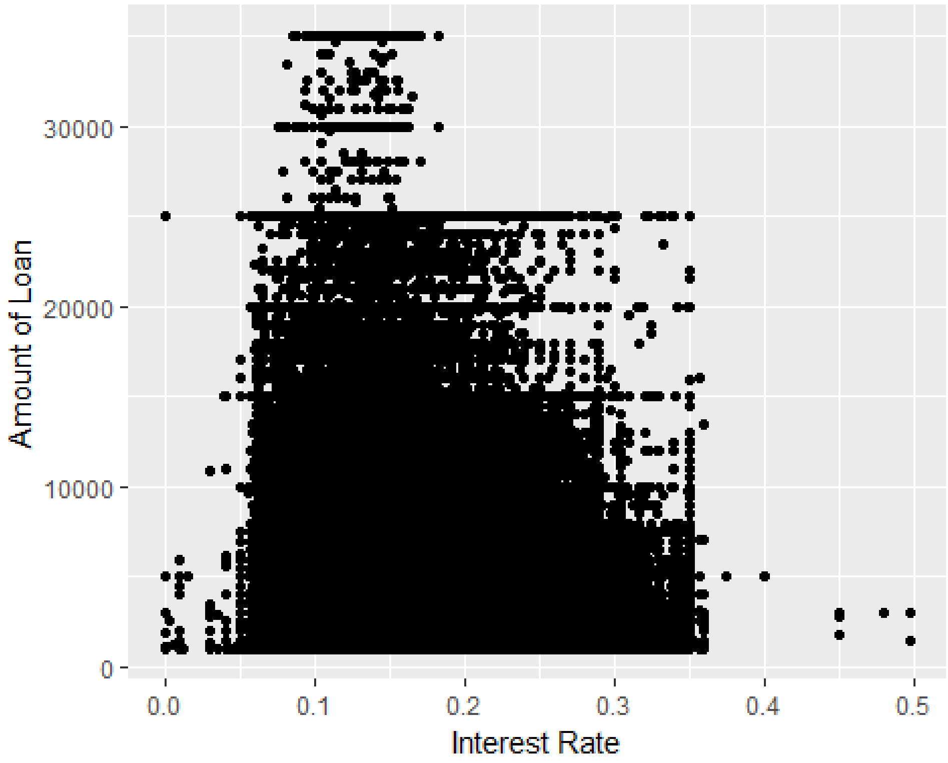
I see 3 trend lines in the plot. Two of them just little differ from each other, the third one is way higher. I remember, there was 3 terms, telling how much time it takes to pay back the loan. 1,3,5 years. I'm pretty sure these 3 trend lines are connected to these terms. Later, I am going to examine this idea by adding the "Term" variable to the plot.

```
## 
##  Pearson's product-moment correlation
## 
## data:  ld$LoanOriginalAmount and ld$MonthlyLoanPayment
## t = 867.82, df = 113940, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9312165 0.9327426
## sample estimates:
##       cor 
## 0.9319837
```

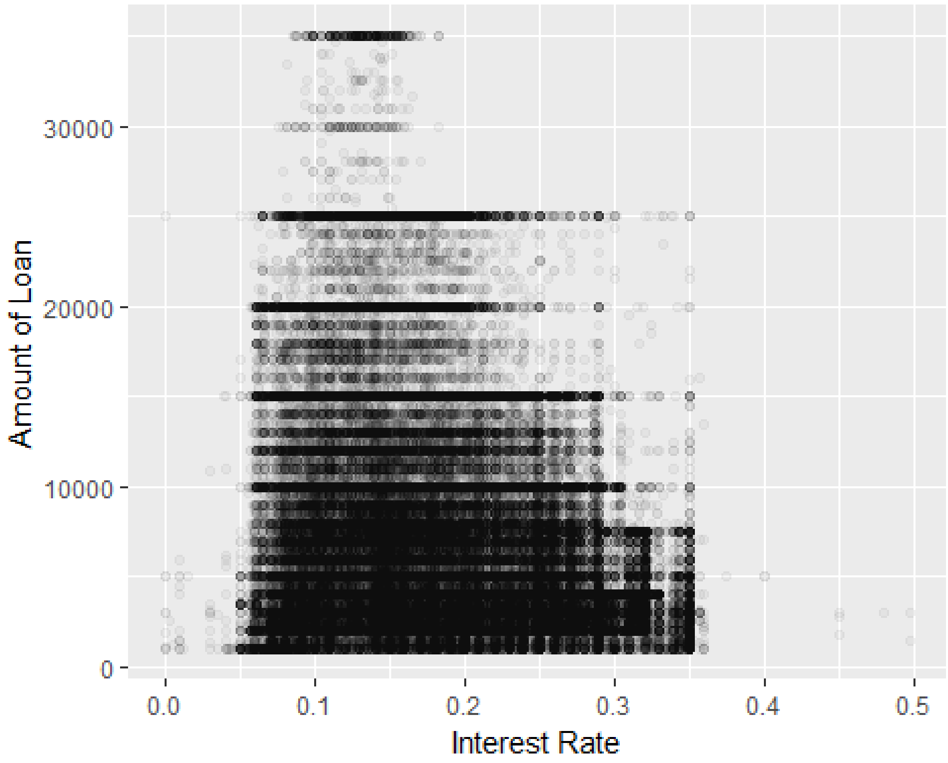
There is a really strong relationship between the two variables. As the borrowed amount of loan grows, the monthly payment grows too.

Interest Rate and Borrowed Money

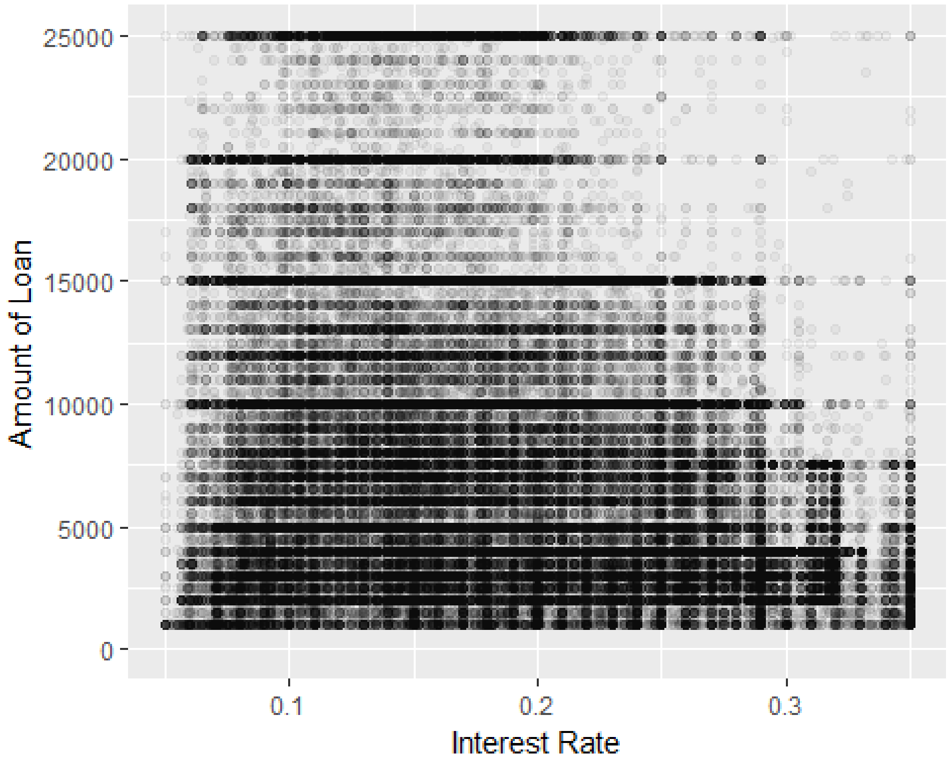
```
# Interest Rate and Borrowed Money
ggplot(ld, aes(ld$BorrowerRate, ld$LoanOriginalAmount)) +
  geom_point() +
  ylab("Amount of Loan") +
  xlab("Interest Rate")
```



```r
# Interest Rate and Borrowed Money
# Change alpha level
ggplot(ld, aes(ld$BorrowerRate, ld$LoanOriginalAmount)) +
  geom_point(alpha = 1/30) +
  ylab("Amount of Loan") +
  xlab("Interest Rate")
```

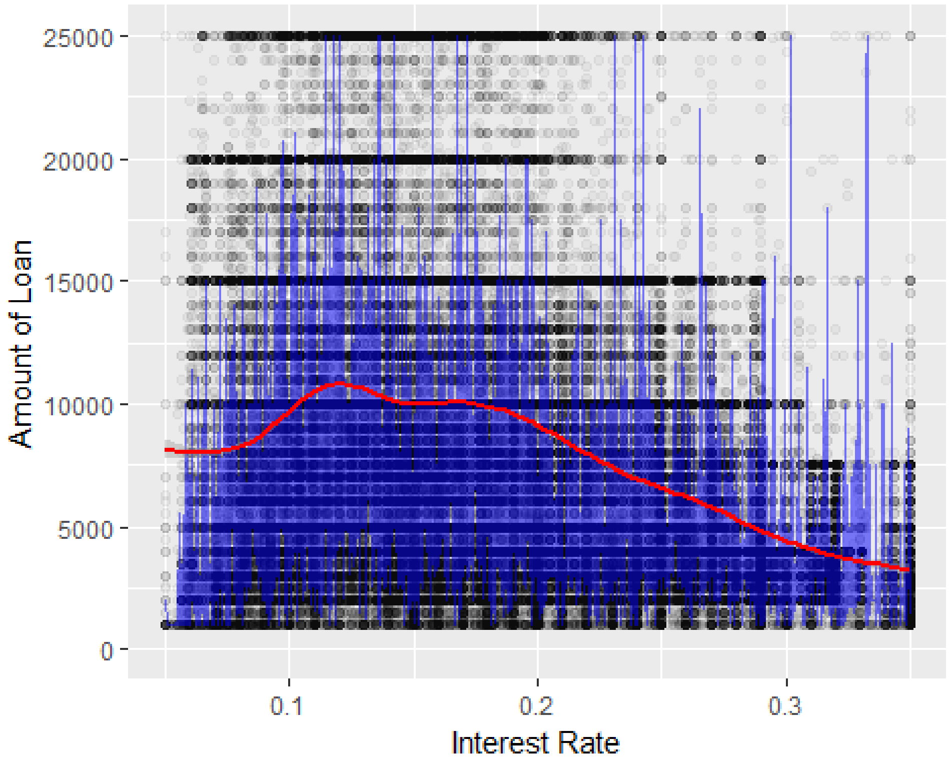


```r
# Interest Rate and Borrowed Money
# Change alpha level
# Limit x
ggplot(ld, aes(ld$BorrowerRate, ld$LoanOriginalAmount)) +
  geom_point(alpha = 1/25) +
  xlim(c(0.05,0.35)) +
  ylim(c(0,25000)) +
  ylab("Amount of Loan") +
  xlab("Interest Rate")
```



People get money in a high range of interest rate. This plot will be interesting when I'll add the prosper rating variable. We already know that, borrowers in a lower rated prosper going to get loan with bigger interest rate. I can barely see the main shape of the dataset, let's add the median to the dataset.

```
# Interest Rate and Borrowed Money
# Change alpha level
# Limits x
# Statistical curves
ggplot(ld, aes(ld$BorrowerRate, ld$LoanOriginalAmount)) +
  geom_point(alpha = 1/25) +
  xlim(c(0.05,0.35)) +
  ylim(c(0,25000)) +
  geom_line(stat = 'summary', fun.y = median, color = "blue", alpha = 0.5) +
  geom_smooth(color = 'red') +
  ylab("Amount of Loan") +
  xlab("Interest Rate")
```



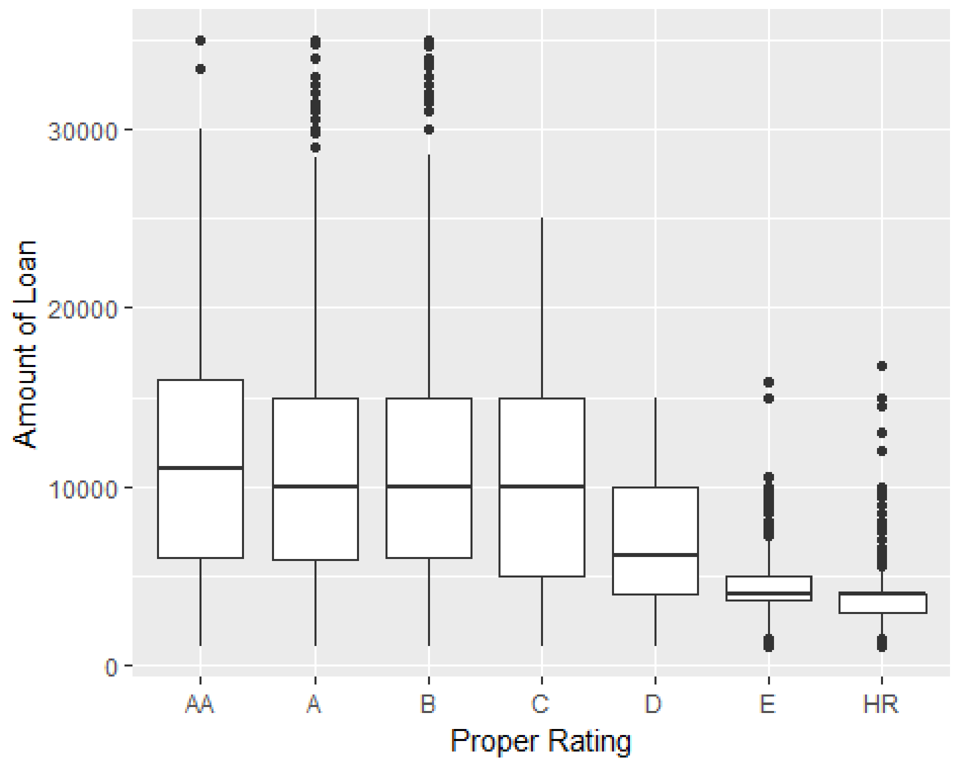
Blue line is the median. Red line removes the "noises" from the dataset, so it is not jumping like the median's line.

```
##
##  Pearson's product-moment correlation
##
## data:  ld$BorrowerRate and ld$LoanOriginalAmount
## t = -117.58, df = 113940, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3341283 -0.3237719
## sample estimates:
##        cor
## -0.3289599
```

There is a small, but meaningful correlation between the variables. As Amount of the loan grows, the Interest Rate decreases.

Prosper Rating and Borrowed Loan

```
# Prosper Rating and Borrowed Loan
# Without unknown prosper rating
ggplot(subset(ld, ld$ProsperRating..Alpha. != 'Unknown'),
aes(ProsperRating..Alpha., LoanOriginalAmount)) +
  geom_boxplot() +
  ylab("Amount of Loan") +
  xlab("Proper Rating")
```



```
## ld_prosper_rationg_without_unknown$ProsperRating..Alpha.: AA
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000    6000   10940   11584   16000   35000
## --------------------------------------------------------
## ld_prosper_rationg_without_unknown$ProsperRating..Alpha.: A
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000    5850   10000   11460   15000   35000
## --------------------------------------------------------
## ld_prosper_rationg_without_unknown$ProsperRating..Alpha.: B
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000    6000   10000   11622   15000   35000
## --------------------------------------------------------
## ld_prosper_rationg_without_unknown$ProsperRating..Alpha.: C
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000    5000   10000   10392   15000   25000
## --------------------------------------------------------
## ld_prosper_rationg_without_unknown$ProsperRating..Alpha.: D
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000    4000    6100    7083   10000   15000
## --------------------------------------------------------
## ld_prosper_rationg_without_unknown$ProsperRating..Alpha.: E
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000    3600    4000    4586    5000   15900
## --------------------------------------------------------
## ld_prosper_rationg_without_unknown$ProsperRating..Alpha.: HR
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000    3000    4000    3463    4000   16800
```

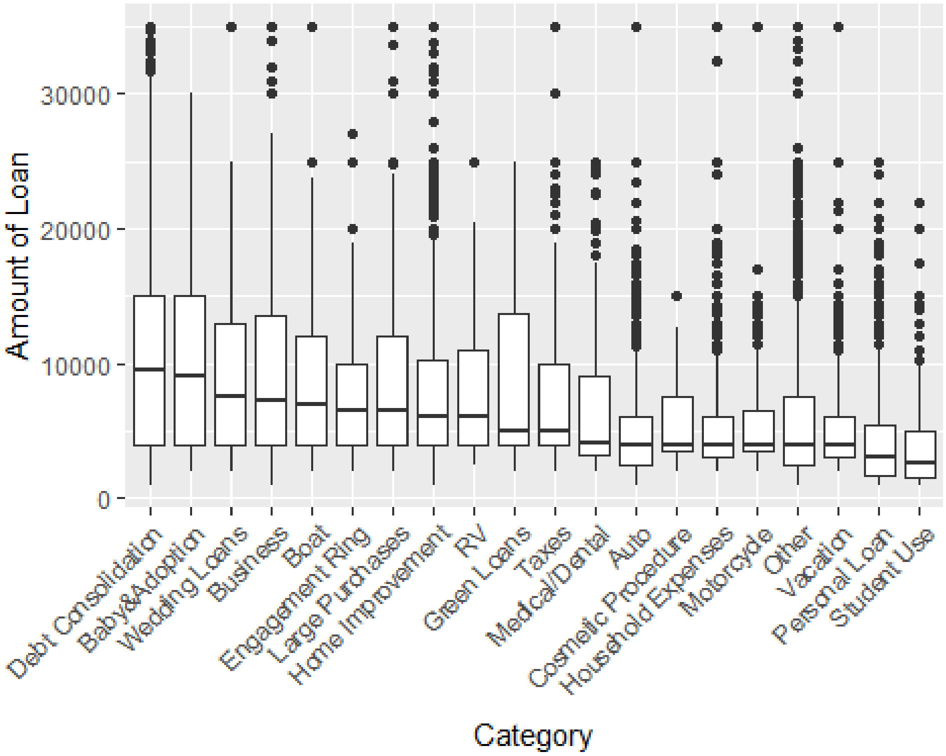
People with better prosper rating borrowed more money. It is logical, because someone who has more money, can borrow more, because they can pay back more every month.

There is no big difference between AA, A, B, C in Q1, median and Q3. The difference is seen at the values outside of this area. A,B has significantly more outlier values than the other

D is like a bridge between AA-C & E-HR ratings. D rating's Q3 is around C rating's median, and Q1 is around E rating's median. E and HR needed way less money than other ratings.

Borrowed Loan by Categories

```
# Borrowed loan # Categories
ggplot(subset(ld, ld$ListingCategory..numeric. != 'Not Available'),
       aes(reorder(ListingCategory..numeric., -LoanOriginalAmount, median),
           LoanOriginalAmount)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1)) +
  ylab("Amount of Loan") +
  xlab("Category")
```

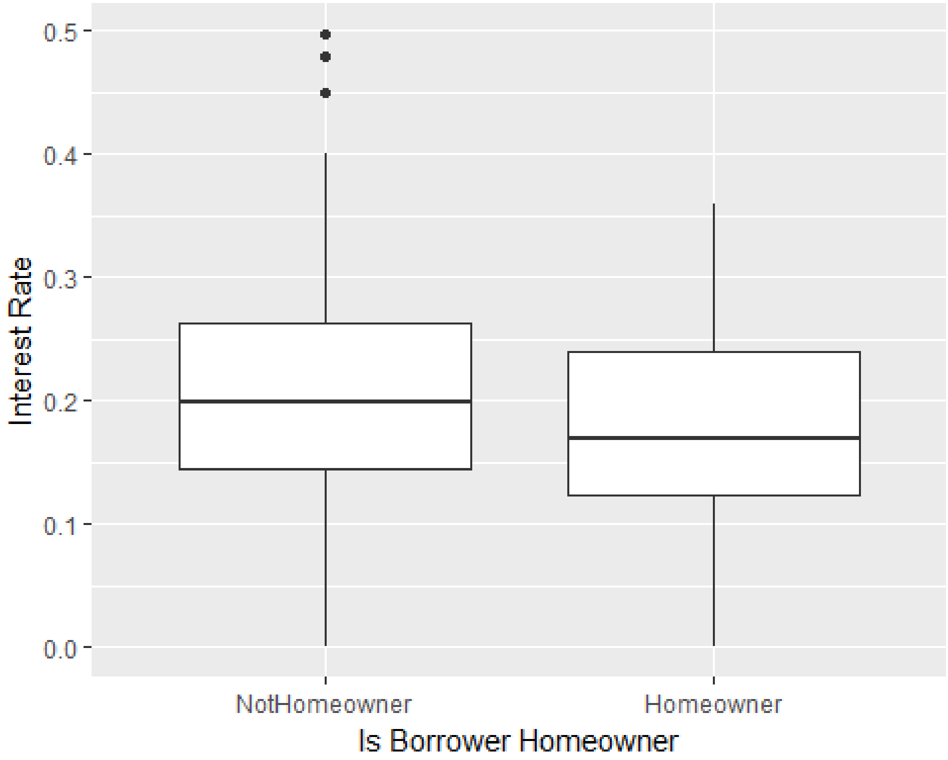


As we can see, outliers are always above the Q3. I have sorted the boxplots by their medians. There is no huge difference between Q1s for a while from Debt Consolidation to Taxes. In the first categories, there is a higher range in the borrowed money.

Baby$Adobtion takes the second place, but previously we saw that, there is only a few people in that category.

Interest Rate and Homeowner's connection

```
# Interest Rate & Homeowner
ggplot(ld, (aes(ld$IsBorrowerHomeowner, ld$BorrowerRate))) +
  geom_boxplot() +
  ylab("Interest Rate") +
  xlab("Is Borrower Homeowner")
```

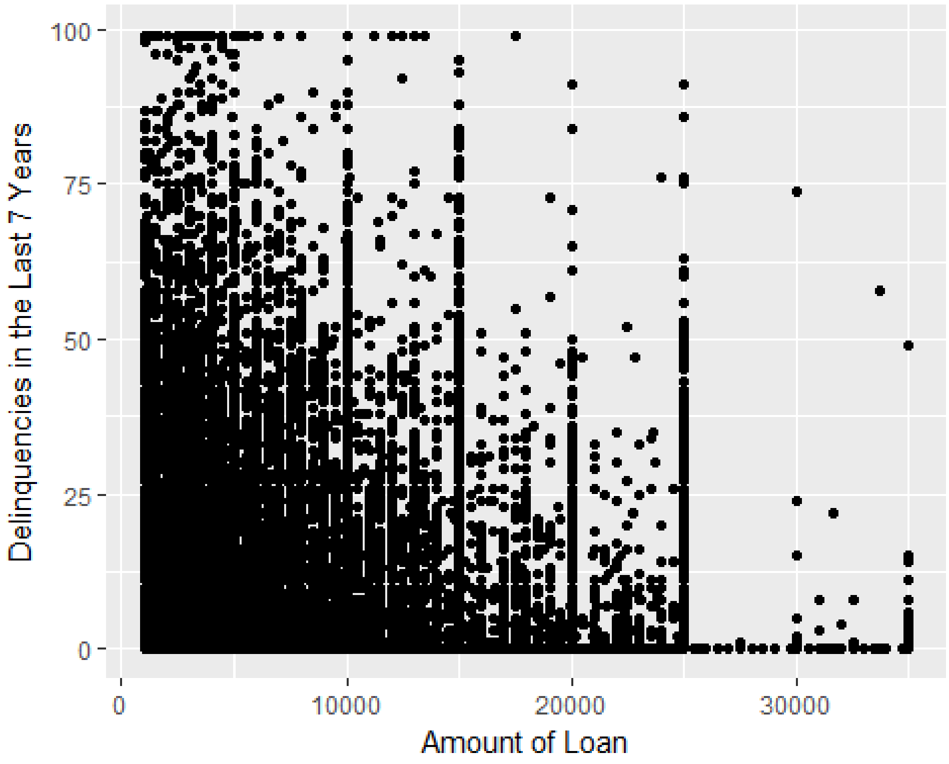


```
## ld$IsBorrowerHomeowner: NotHomeowner
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.1449  0.1980  0.2029  0.2624  0.4975 
## -------------------------------------------------------- 
## ld$IsBorrowerHomeowner: Homeowner
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.1239  0.1700  0.1828  0.2394  0.3600
```

As seen, the homeowner borrowers have lower interest rates. Homeowner median is 0.17, while the not homeowner is almost 0.2.

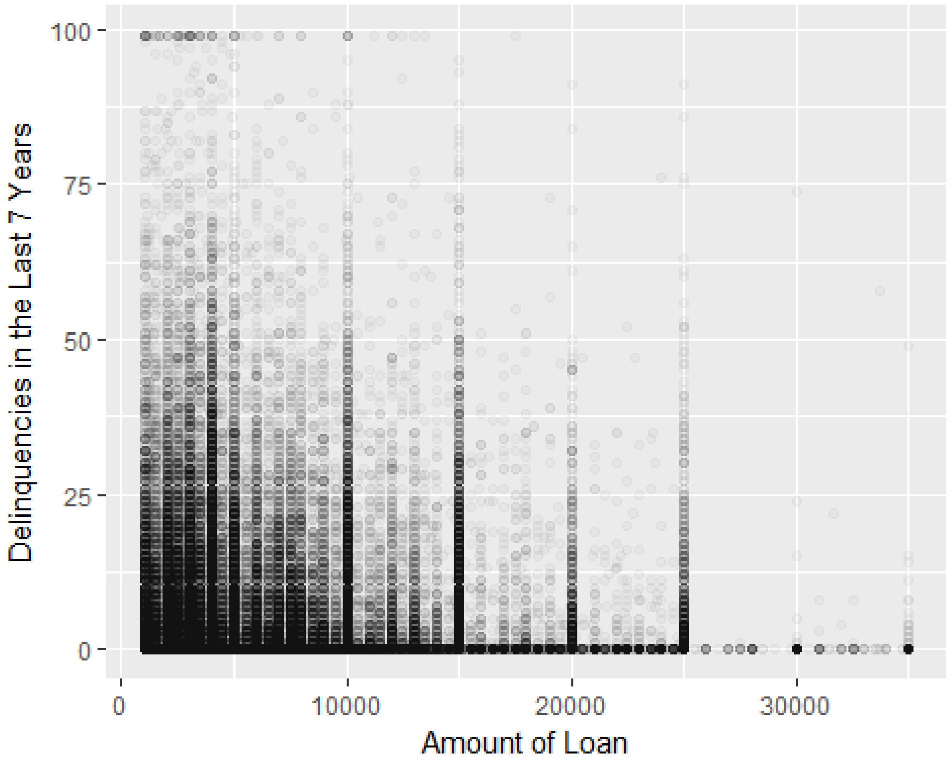
Times of Delinquencies and Amount of Loan

```
# Times of Delinquencies & Amount of Loan
ggplot(ld,aes(x =ld$LoanOriginalAmount, y = ld$DelinquenciesLast7Years)) +
  geom_point() +
  ylab("Delinquencies in the Last 7 Years") +
  xlab("Amount of Loan")
```



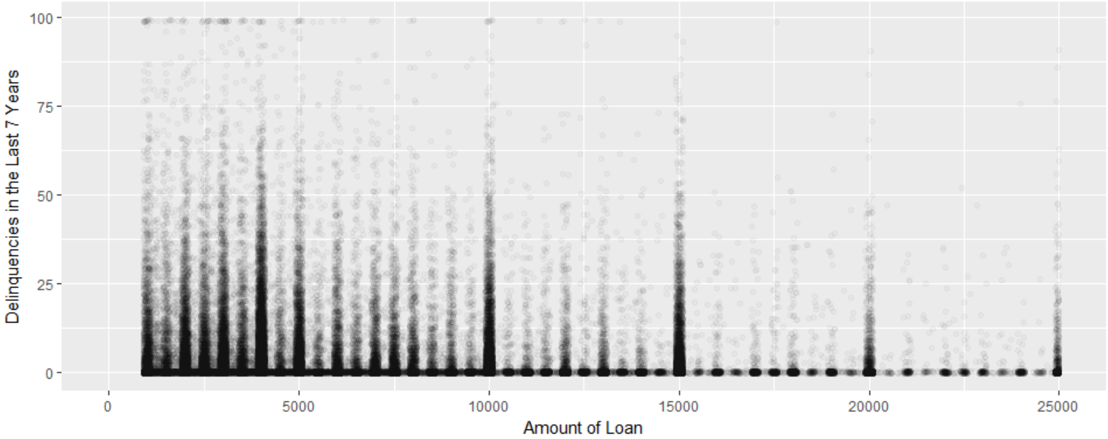
Too much point on the plot. Let’s change the alpha level.

```
ggplot(ld,aes(x =ld$LoanOriginalAmount, y = ld$DelinquenciesLast7Years)) +
  geom_point(alpha = 1/35) +
  ylab("Delinquencies in the Last 7 Years") +
  xlab("Amount of Loan")
```



Much better. But the points are too small. Let’s change the x limits.

```
ggplot(ld,aes(x =ld$LoanOriginalAmount, y = ld$DelinquenciesLast7Years)) +
  geom_point(alpha = 1/35, position = position_jitter(width = 100)) +
  xlim(c(0,25000)) +
  ylab("Delinquencies in the Last 7 Years") +
  xlab("Amount of Loan")
```



There are way more counts than the others at around 2,000, 3000, 4000, 5000, 10,000 et cetera. Likely because at those values there are more people who borrowed money. I’ll add those counts to the plot later, to see the connection.

Talk about some of the relationships you observed in this part of the investigation. How did the feature(s) of interest vary with other features in the dataset?

I Have found a relationship between interest rate and prosper rating. Higher ranked prosper rated people have less interest rates.

There is also a connection between the amount of the loan and the monthly payment, a 0.93 correlation coefficient. And it seems like there is also a connection with the term variable.

If the prosper rating is “HR” we can easily tell what the interest rate will be with a high percentage.

Did you observe any interesting relationships between the other features (not the main feature(s) of interest)?

No, it seems like the variables are connected with the featured ones.

What was the strongest relationship you found?

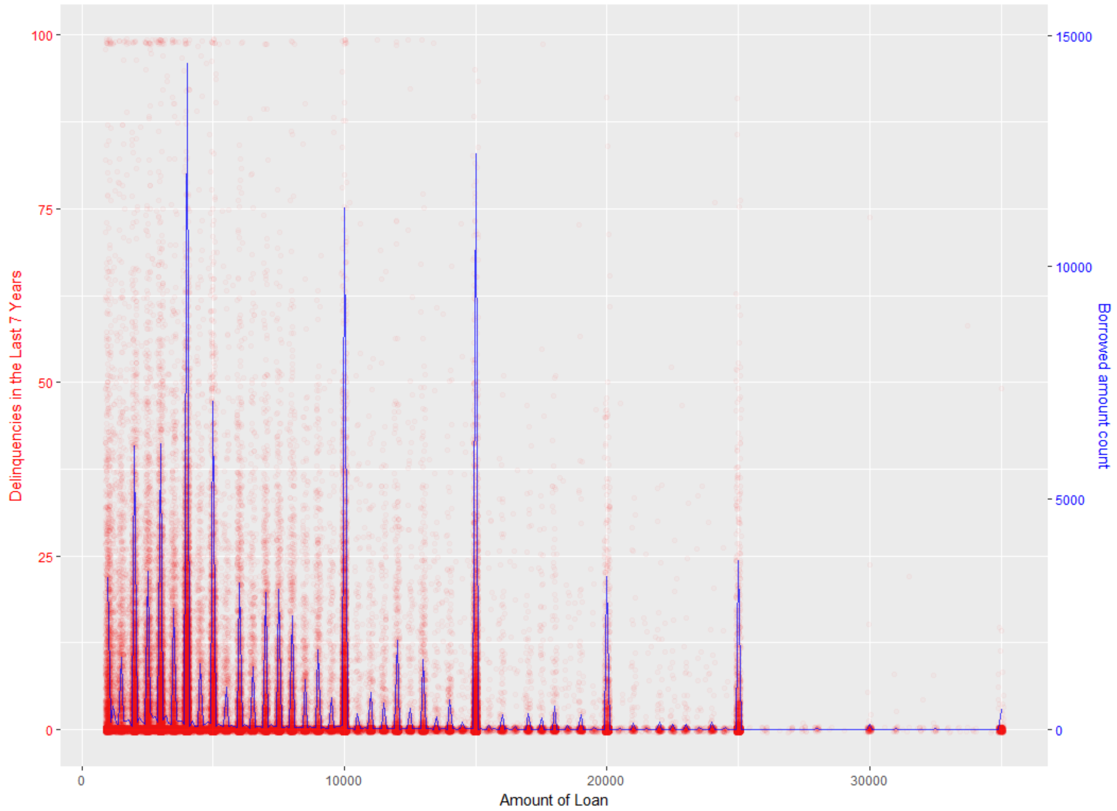
The amount of the loan and the monthly payment. There is a really big relationship.

Multivariate Plots Section

Multivariate Analysis

Borrowed loan with Delinquencies in the Last 7 Years and Borrower Counts.

```
# Amount_and_deliquencies plot with borrowed amount counts
ggplot(ld, aes(ld$LoanOriginalAmount, ld$DelinquenciesLast7Years)) +
  geom_point(alpha = 1/35, color = 'red',
             position = position_jitter(width = 100)) +
  geom_line(aes(ld$LoanOriginalAmount, ..count../150),
            color = 'blue', stat = 'bin', binwidth = 100, alpha = 0.75) +
  scale_y_continuous(sec.axis = sec_axis(~. * 150 ,
                                         name = 'Borrowed amount count')) +
  theme(axis.text.y = element_text(color = 'red')) +
  theme(axis.text.y.right = element_text(color = 'blue')) +
  theme(axis.title.y = element_text(color = 'red')) +
  theme(axis.title.y.right = element_text(color = 'blue')) +
  ylab("Delinquencies in the Last 7 Years") +
  xlab("Amount of Loan")
```

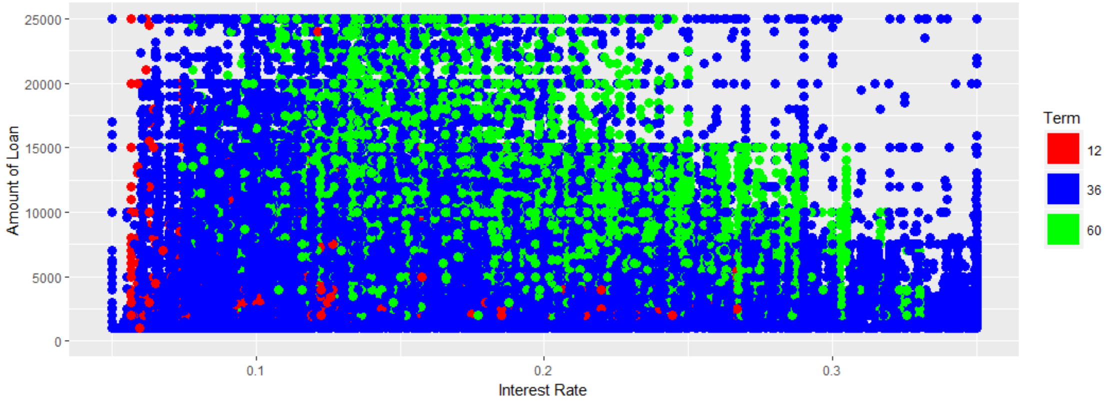


As I thought, the "Borrowed Amount Count" perfectly fits onto the relationship between the loan amount and delinquencies in the last 7 years. This correlates to the connection between the other two.

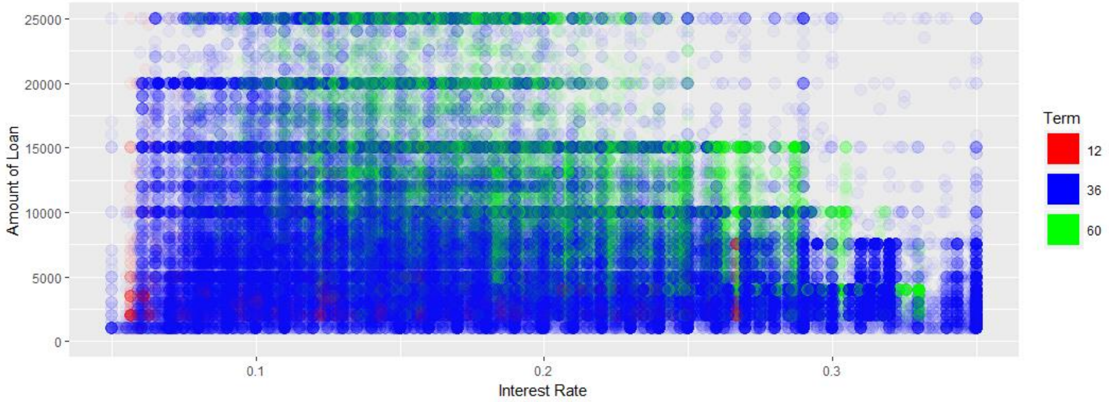
Interest rate, Borrowed Money and Other Categorical Variables

In the second plots I added an 1/25 alpha level.

```
# Interest Rate and Borrowed Money & Term Length
ggplot(ld, aes(ld$BorrowerRate, ld$LoanOriginalAmount,
               color = factor(ld$Term)),) +
  geom_point(size = 3) +
  xlim(c(0.05,0.35)) +
  ylim(c(0,25000)) +
  guides(col = guide_legend(override.aes =
                              list(shape = 15, size = 10, alpha = 1))) +
  scale_color_manual(values = term_colors) +
  ylab("Amount of Loan") +
  xlab("Interest Rate") +
  labs(color = "Term")
```

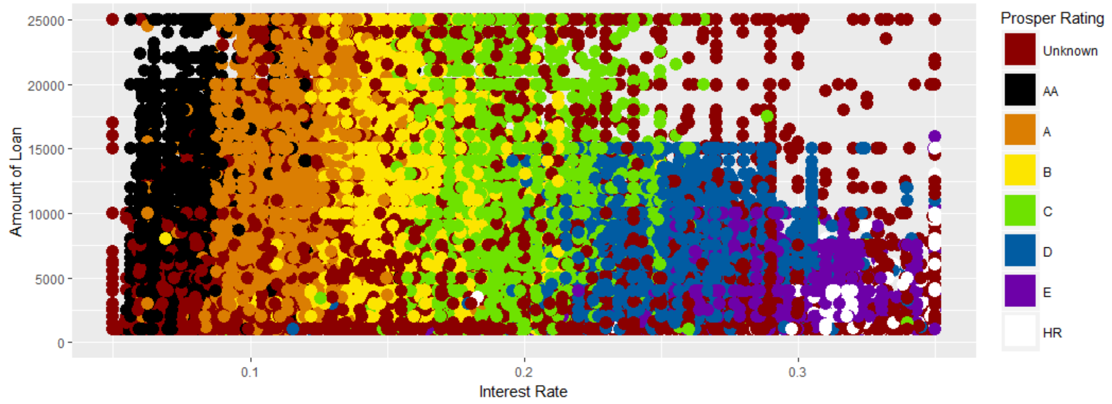


```r
# Interest Rate and Borrowed Money & Term Length
ggplot(ld, aes(ld$BorrowerRate, ld$LoanOriginalAmount,
               color = factor(ld$Term)),) +
  geom_point(size = 4, alpha = 1/25) +
  xlim(c(0.05,0.35)) +
  ylim(c(0,25000)) +
  guides(col = guide_legend(override.aes =
                              list(shape = 15, size = 10, alpha = 1))) +
  scale_color_manual(values = term_colors) +
  ylab("Amount of Loan") +
  xlab("Interest Rate") +
  labs(color = "Term")
```

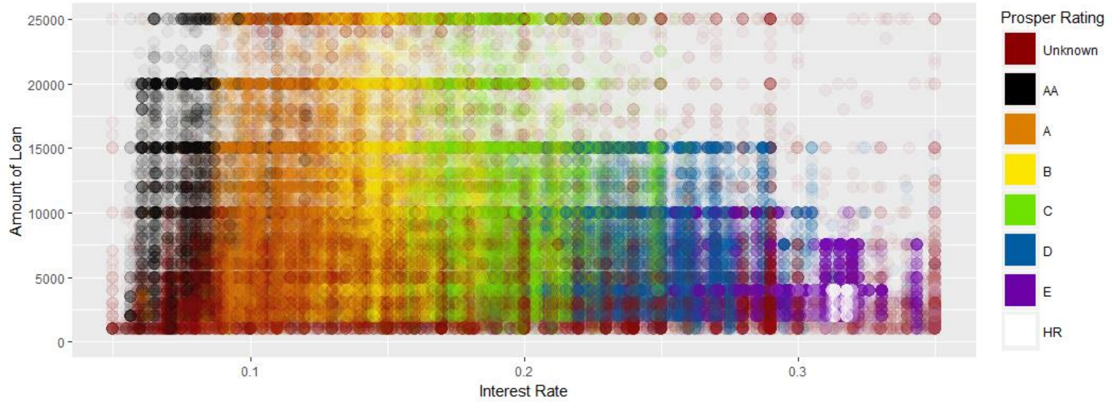


Borrowers with longer payment term and bigger borrowed money often have the same interest rates as the shorter term borrowers with lower loans.

```r
# Interest Rate and Borrowed Money & prosper rating
ggplot(ld, aes(ld$BorrowerRate, ld$LoanOriginalAmount,
               color = ld$ProsperRating..Alpha.)) +
  geom_point( size = 4) +
  xlim(c(0.05,0.35)) +
  ylim(c(0,25000)) +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                                size = 10, alpha = 1))) +
  scale_color_manual(values = prosper_rating_colors ) +
  ylab("Amount of Loan") +
  xlab("Interest Rate") +
  labs(color = "Prosper Rating")
```

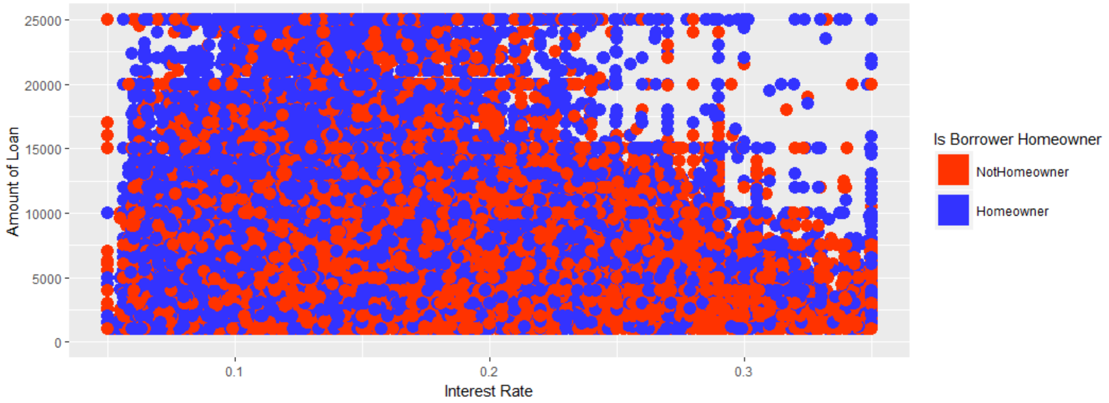


```
# Interest Rate and Borrowed Money & prosper rating
ggplot(ld, aes(ld$BorrowerRate, ld$LoanOriginalAmount,
               color = ld$ProsperRating..Alpha.)) +
  geom_point( size = 4, alpha = 1/25) +
  xlim(c(0.05,0.35)) +
  ylim(c(0,25000)) +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                                size = 10, alpha = 1))) +
  scale_color_manual(values = prosper_rating_colors ) +
  ylab("Amount of Loan") +
  xlab("Interest Rate") +
  labs(color = "Prosper Rating")
```

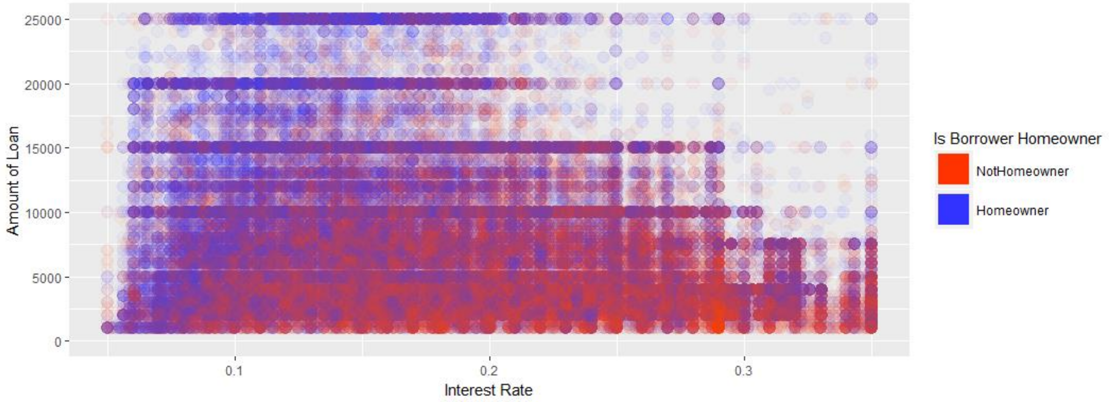


Interesting. Usually it does not really matter how much money do you borrow, because the prosper rating will be the key when they calculate the interest rate at better prosper ratings. In lower prosper ratings it counts more until the HR category.

```
# Interest Rate and Borrowed Money & Is Homeowner
ggplot(ld, aes(ld$BorrowerRate, ld$LoanOriginalAmount,
               color = ld$IsBorrowerHomeowner)) +
  geom_point(size = 4) +
  xlim(c(0.05,0.35)) +
  ylim(c(0,25000)) +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                                size = 10, alpha = 1))) +
  scale_color_manual(values = is_homeowner_colors) +
  ylab("Amount of Loan") +
  xlab("Interest Rate") +
  labs(color = "Is Borrower Homeowner")
```



```
# Interest Rate and Borrowed Money & Is Homeowner
ggplot(ld, aes(ld$BorrowerRate, ld$LoanOriginalAmount,
               color = ld$IsBorrowerHomeowner)) +
  geom_point(size = 4, alpha = 1/25) +
  xlim(c(0.05,0.35)) +
  ylim(c(0,25000)) +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                                size = 10, alpha = 1))) +
  scale_color_manual(values = is_homeowner_colors) +
  ylab("Amount of Loan") +
  xlab("Interest Rate") +
  labs(color = "Is Borrower Homeowner")
```

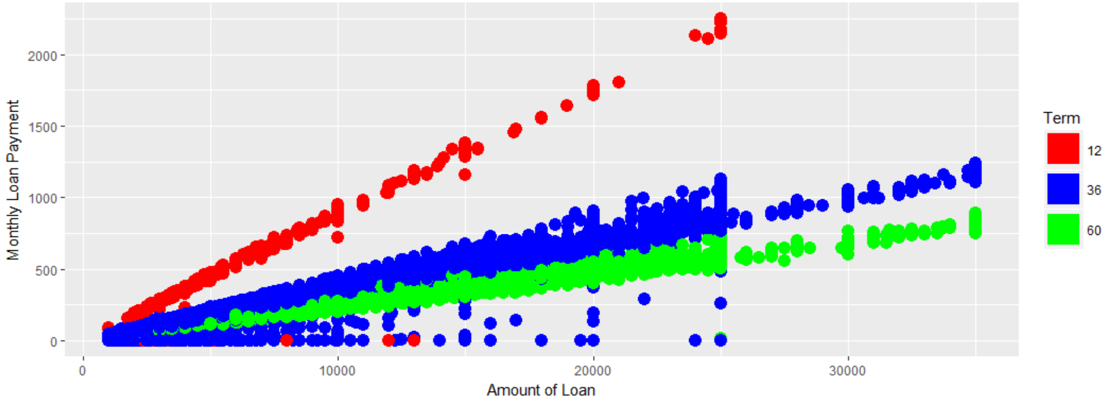


Homeowners are grouping in the left upper corner (lower interest rate, bigger loan), meanwhile the rest of the people are going to the right lower corner (bigger interest rate, lower loan).

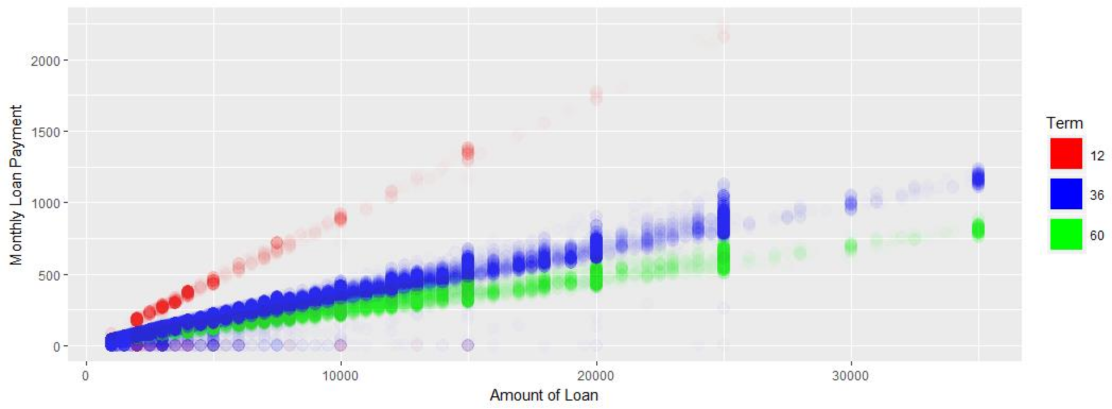
Amount of Loan, Monthly Loan Payment and Other Categorical Variables

In the second plots I added an 1/75 alpha level.

```
#Amount of Loan & Monthly Loan Payment & Term
ggplot(ld, aes(ld$LoanOriginalAmount, ld$MonthlyLoanPayment,
               color = factor(ld$Term))) +
  geom_point(size = 4) +
  scale_color_manual(values = term_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Amount of Loan") +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                                size = 10, alpha = 1))) +
  labs(color = "Term")
```

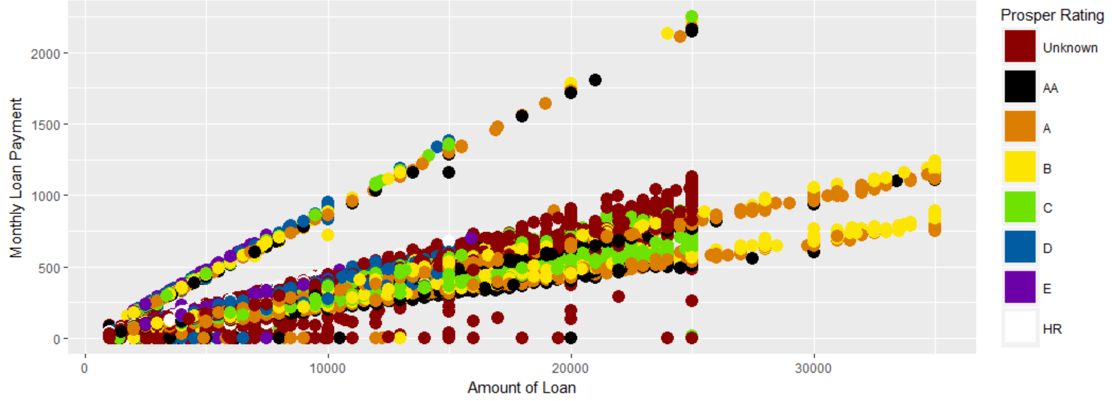


```
#Amount of Loan & Monthly Loan Payment & Term
ggplot(ld, aes(ld$LoanOriginalAmount, ld$MonthlyLoanPayment,
               color = factor(ld$Term))) +
  geom_point(size = 4, alpha = 1/75) +
  scale_color_manual(values = term_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Amount of Loan") +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                                size = 10, alpha = 1))) +
  labs(color = "Term")
```

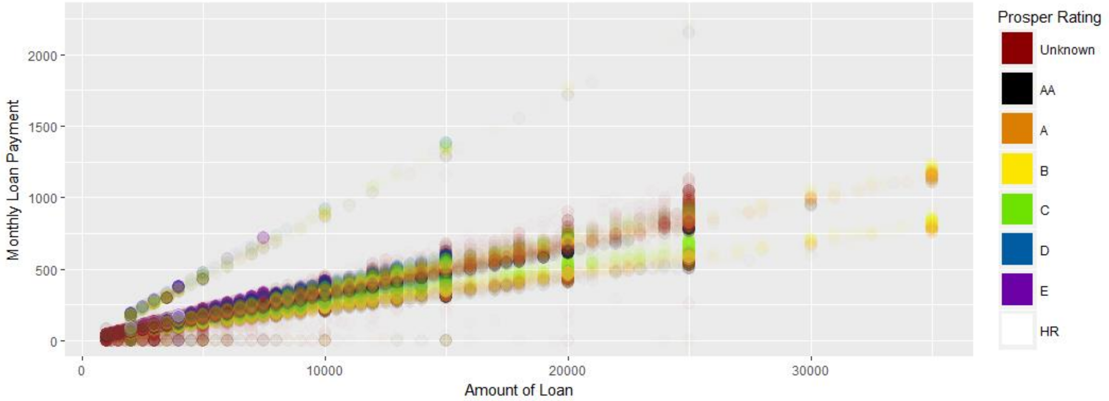


We know that, there are 12, 36 and 60 months long terms. It is trivial who borrowed money for a longer period paid less every month. But there are some interesting points. How can two 1 year long points be at the bottom of the Y axis at around 12,000?

```
#Amount of Loan & Monthly Loan Payment & Propser Rating
ggplot(ld, aes(ld$LoanOriginalAmount, ld$MonthlyLoanPayment,
               color = ld$ProsperRating..Alpha.)) +
  geom_point(size = 4) +
   scale_color_manual(values = prosper_rating_colors ) +
  ylab("Monthly Loan Payment") +
  xlab("Amount of Loan") +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                                size = 10, alpha = 1))) +
  labs(color = "Prosper Rating")
```



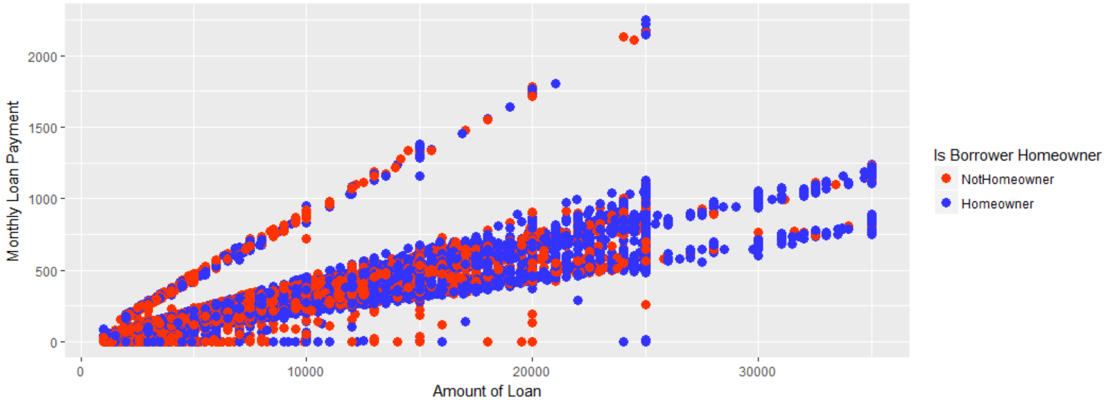
```
#Amount of Loan & Monthly Loan Payment & Propser Rating
ggplot(ld, aes(ld$LoanOriginalAmount, ld$MonthlyLoanPayment,
               color = ld$ProsperRating..Alpha.)) +
  geom_point(size = 4, alpha = 1/75) +
   scale_color_manual(values = prosper_rating_colors ) +
  ylab("Monthly Loan Payment") +
  xlab("Amount of Loan") +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                                size = 10, alpha = 1))) +
  labs(color = "Prosper Rating")
```



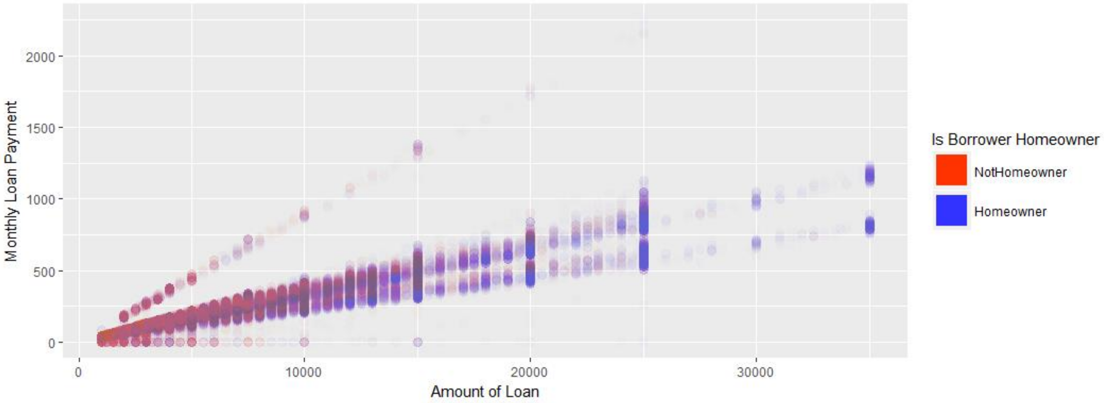
We still don't know the answer, because we don't have information about the points, next to them (red ones) we see that, they are "A" and "B" rated points. But without knowing the neighbor points, it is hard to tell. But maybe because they have good prosper ratings with a special offer. Like pay back in a year, with a low monthly payment. And if he gets money soon, he can pay back the whole loan then.

From this plot it is seen that most of the missing data comes from the 3 years long term.

```
#Amount of Loan & Monthly Loan Payment & IsHomeowner
ggplot(ld, aes(ld$LoanOriginalAmount, ld$MonthlyLoanPayment,
               color = ld$IsBorrowerHomeowner)) +
  geom_point(size = 3)  +
  scale_color_manual(values = is_homeowner_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Amount of Loan") +
  labs(color = "Is Borrower Homeowner")
```



```
#Amount of Loan & Monthly Loan Payment & IsHomeowner
ggplot(ld, aes(ld$LoanOriginalAmount, ld$MonthlyLoanPayment,
               color = ld$IsBorrowerHomeowner)) +
  geom_point(size = 3, alpha = 1/75)  +
  scale_color_manual(values = is_homeowner_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Amount of Loan") +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                              size = 10, alpha = 1))) +
  labs(color = "Is Borrower Homeowner")
```

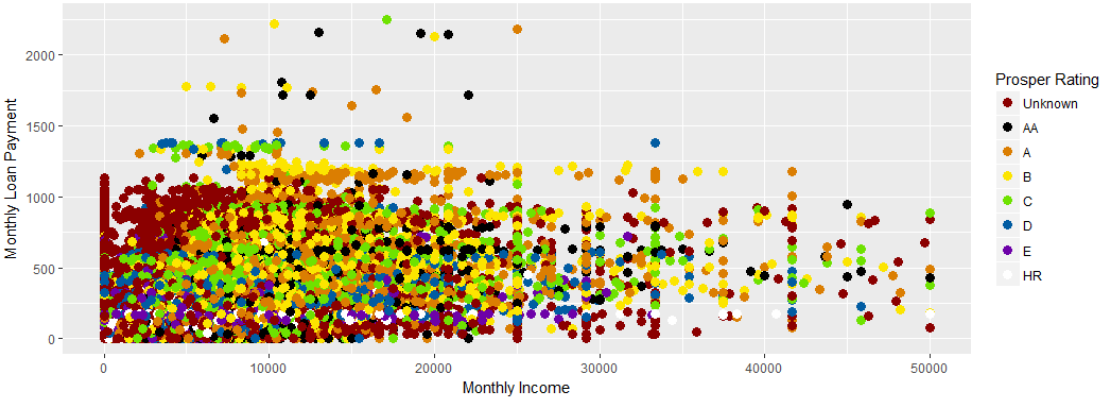


Homeowners are more likely to go for a 3 or a 5 years long payment terms. People who don’t own a home borrow less money than homeowners.

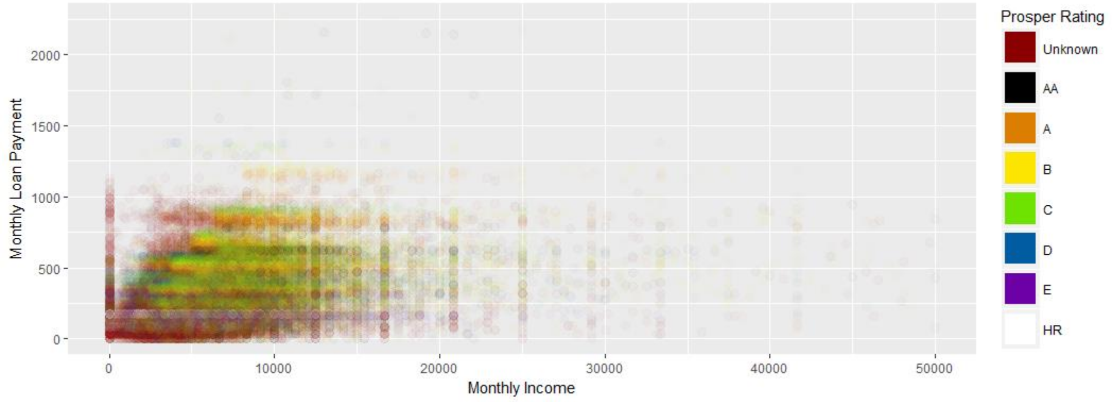
Monthly Income & Monthly Loan Payment & Other Categorical Variables

In the second plots I added an 1/50 alpha level.

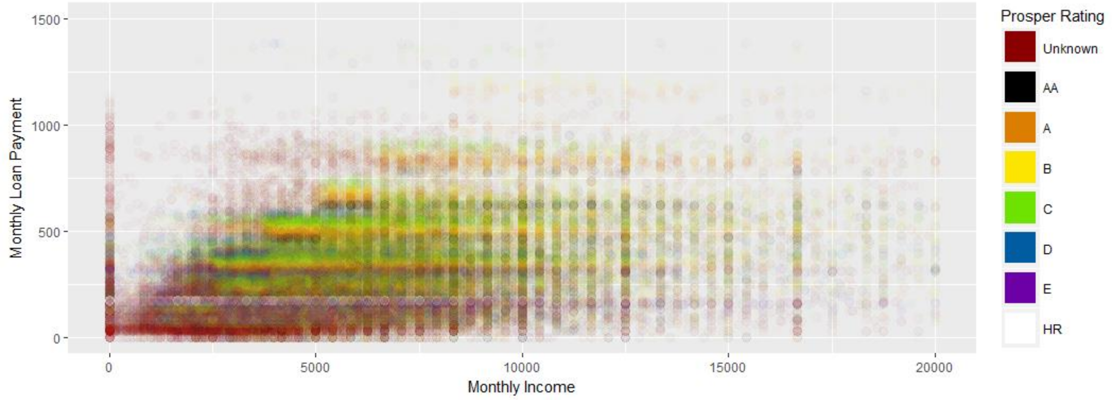
```
# Monthly Income & Monthly Loan Payment & Prosper Rating
ggplot(ld, aes(ld$StatedMonthlyIncome, ld$MonthlyLoanPayment,
               color = ld$ProsperRating..Alpha.)) +
  geom_point(size = 3) +
  xlim(c(0,50000)) +
  scale_colour_manual(values = prosper_rating_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Monthly Income") +
  labs(color = "Prosper Rating")
```



```
# Monthly Income & Monthly Loan Payment & Prosper Rating
ggplot(ld, aes(ld$StatedMonthlyIncome, ld$MonthlyLoanPayment,
               color = ld$ProsperRating..Alpha.)) +
  geom_point(size = 3, alpha = 1/50) +
  xlim(c(0,50000)) +
  scale_colour_manual(values = prosper_rating_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Monthly Income") +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                                size = 10, alpha = 1))) +
  labs(color = "Prosper Rating")
```

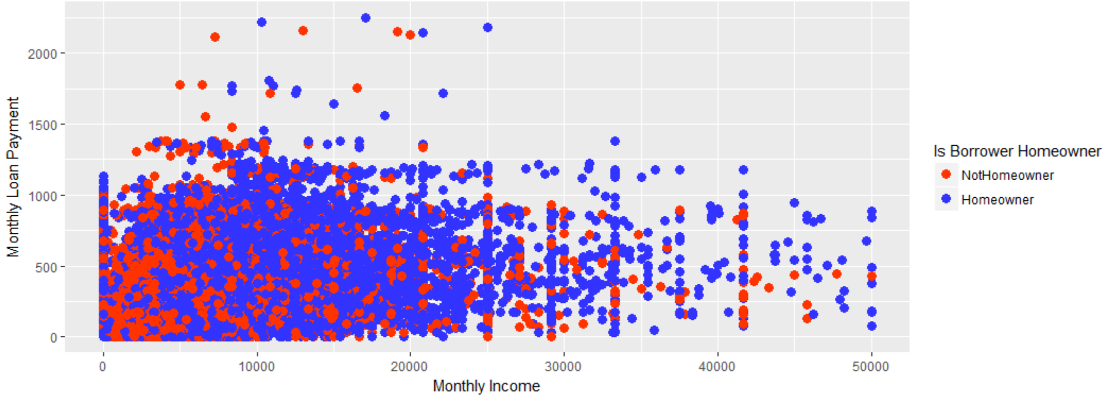


```
# Monthly Income & Monthly Loan Payment & Prosper Rating
ggplot(ld, aes(ld$StatedMonthlyIncome, ld$MonthlyLoanPayment,
               color = ld$ProsperRating..Alpha.)) +
  geom_point(size = 3, alpha = 1/50) +
  xlim(c(0,20000)) +
  ylim(0,1500) +
  scale_colour_manual(values = prosper_rating_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Monthly Income") +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                                size = 10, alpha = 1))) +
  labs(color = "Prosper Rating")
```

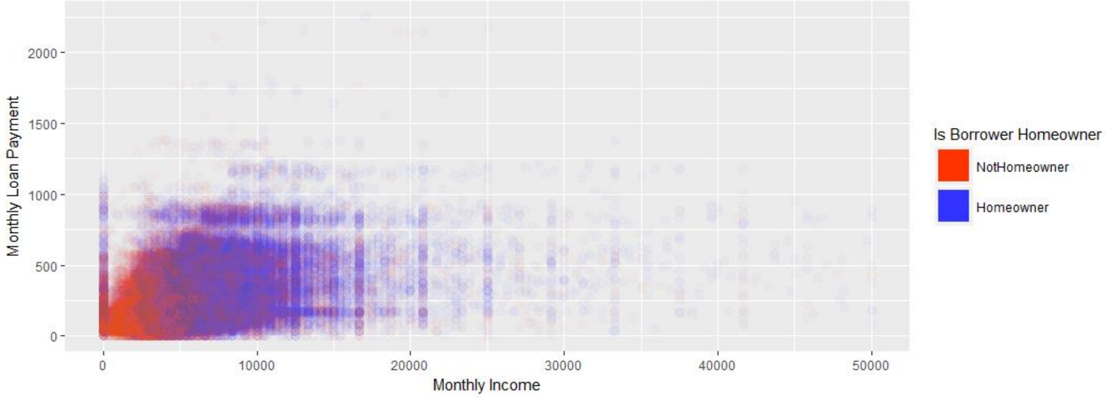


Another interesting plot. The best rated people are in the middle of the group. “HR” category has almost the same monthly loan payment for everyone. “A” and “B” categories have wider and higher ranges than others.

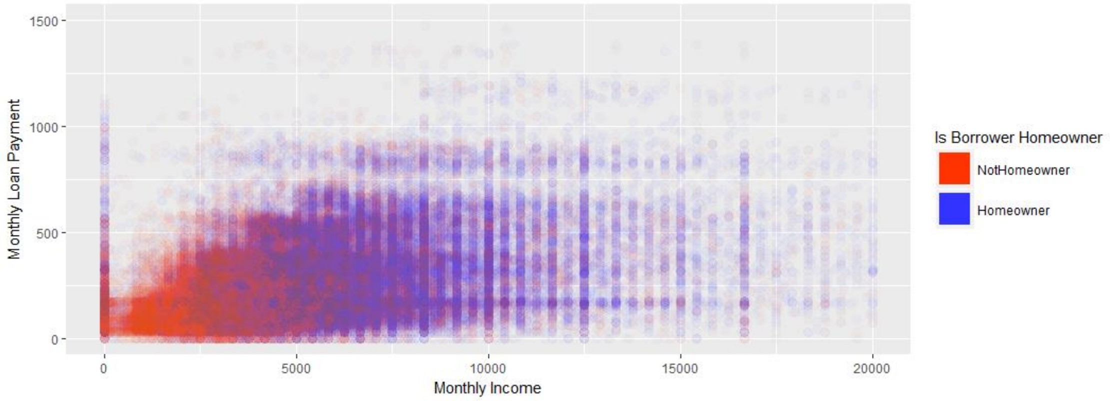
```
# Monthly Income & Monthly Loan Payment & Is Borrower Homeowner
ggplot(ld, aes(ld$StatedMonthlyIncome, ld$MonthlyLoanPayment,
                 color = ld$IsBorrowerHomeowner)) +
  geom_point(size = 3) +
  xlim(c(0,50000)) +
  scale_colour_manual(values = is_homeowner_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Monthly Income") +
  labs(color = "Is Borrower Homeowner")
```



```
# Monthly Income & Monthly Loan Payment & Is Borrower Homeowner
ggplot(ld, aes(ld$StatedMonthlyIncome, ld$MonthlyLoanPayment,
                 color = ld$IsBorrowerHomeowner)) +
  geom_point(size = 3, alpha = 1/50) +
  xlim(c(0,50000)) +
  scale_colour_manual(values = is_homeowner_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Monthly Income") +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                                 size = 10, alpha = 1))) +
  labs(color = "Is Borrower Homeowner")
```

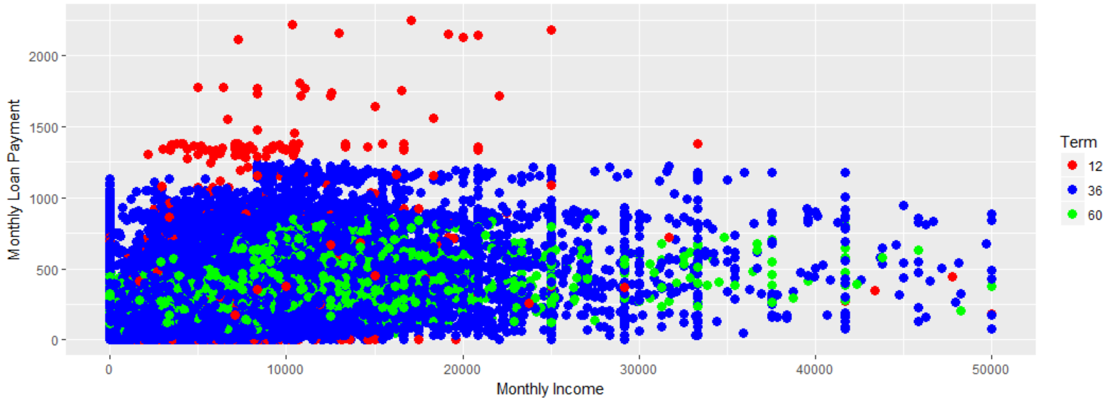


```
# Monthly Income & Monthly Loan Payment & Is Borrower Homeowner
ggplot(ld, aes(ld$StatedMonthlyIncome, ld$MonthlyLoanPayment,
                 color = ld$IsBorrowerHomeowner)) +
  geom_point(size = 3, alpha = 1/50) +
  xlim(c(0,20000)) +
  ylim(0,1500) +
  scale_colour_manual(values = is_homeowner_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Monthly Income") +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                              size = 10, alpha = 1))) +
  labs(color = "Is Borrower Homeowner")
```

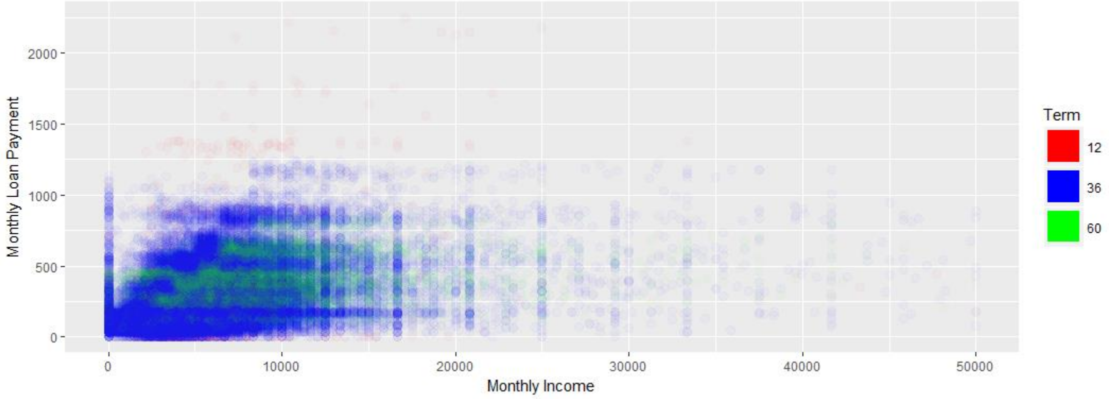


Someone who does not own a home is more likely to go for a smaller loan with a lower monthly payment. But home owners are not interested in that, just only a few of them.

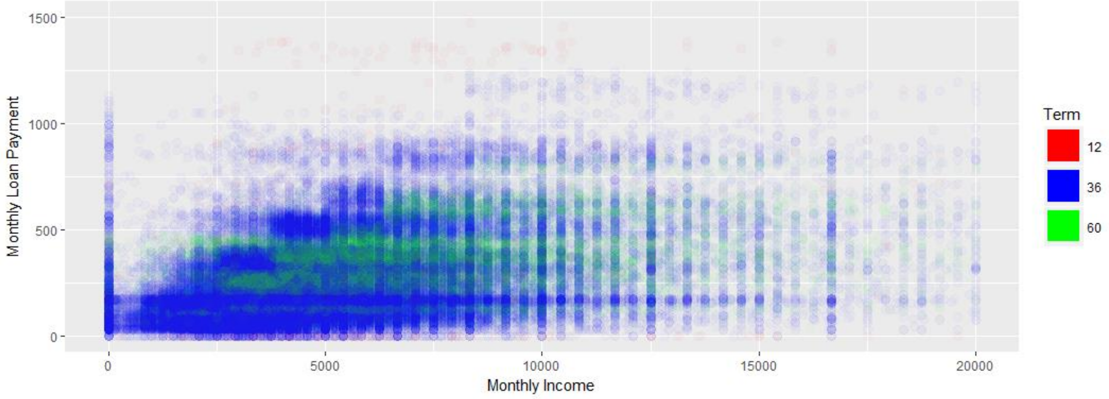
```
# Monthly Income & Monthly Loan Payment & Term
ggplot(ld, aes(ld$StatedMonthlyIncome, ld$MonthlyLoanPayment,
               color = factor(ld$Term))) +
  geom_point(size = 3) +
  xlim(c(0,50000)) +
  scale_color_manual(values = term_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Monthly Income") +
  labs(color = "Term")
```



```
# Monthly Income & Monthly Loan Payment & Term
ggplot(ld, aes(ld$StatedMonthlyIncome, ld$MonthlyLoanPayment,
               color = factor(ld$Term))) +
  geom_point(size = 3, alpha = 1/50) +
  xlim(c(0,50000)) +
  scale_color_manual(values = term_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Monthly Income") +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                            size = 10, alpha = 1))) +
  labs(color = "Term")
```



```
# Monthly Income & Monthly Loan Payment & Term
ggplot(ld, aes(ld$StatedMonthlyIncome, ld$MonthlyLoanPayment,
               color = factor(ld$Term))) +
  geom_point(size = 3, alpha = 1/50) +
  xlim(c(0,20000)) +
  ylim(0,1500) +
  scale_color_manual(values = term_colors) +
  ylab("Monthly Loan Payment") +
  xlab("Monthly Income") +
  guides(col = guide_legend(override.aes = list(shape = 15,
                                            size = 10, alpha = 1))) +
  labs(color = "Term")
```



1-year long term rules the upper side of the plot. They borrow for higher monthly loan payment. But they are not likely to borrow a big loan. 3-year long term is spreading at the middle. They don’t pay too much or too little in a month. 5 years long term is almost everywhere. Except at the upper side of the plot. There are only 1 yearlong terms.

Talk about some of the relationships you observed in this part of the investigation. Were there features that strengthened each other in terms of looking at your feature(s) of interest?

The “Borrowed Amount Count” fits onto the relationship between the loan amount and delinquencies in the last 7 years.

We can see the layers in the connection of interest rate and loan amount if we add the prosper rating categories. It definitely split the dataset by its categories.

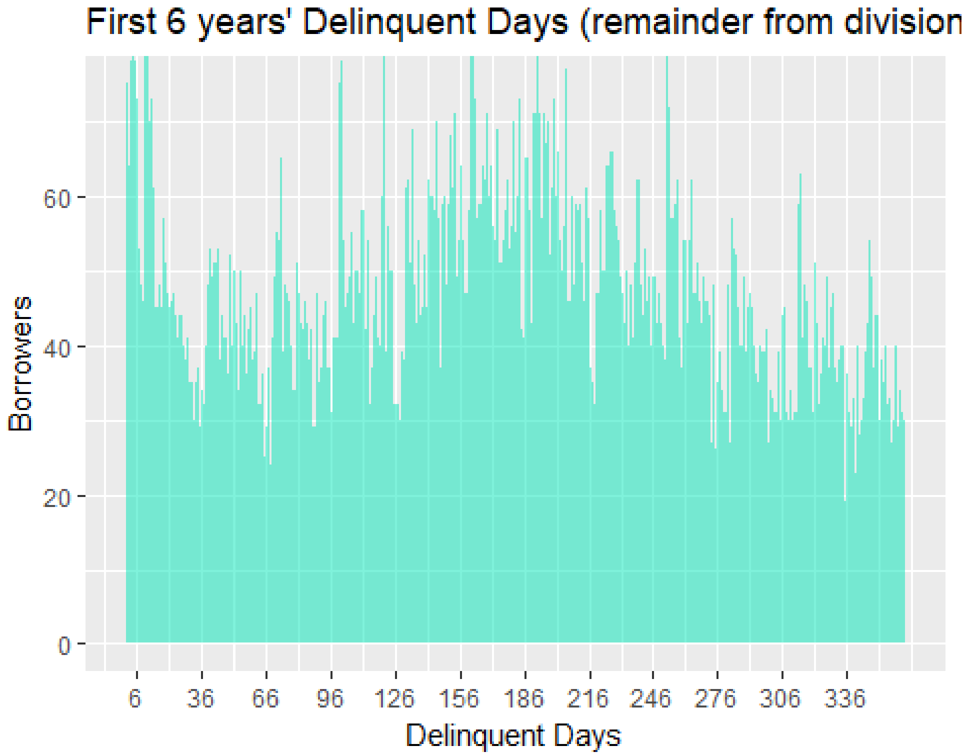
There is a really strong relationship between the monthly payment, amount of loan and the term. There are 3 trend lines separated by the term variable.

Were there any interesting or surprising interactions between features?

The plot where we can see the connections between these 3 variables: Monthly Income, Monthly Loan Payment, and Prosper Rating. I would have not expected that how the categorical variable is distributed on the chart. It surprised me. ——

Final Plots and Summary

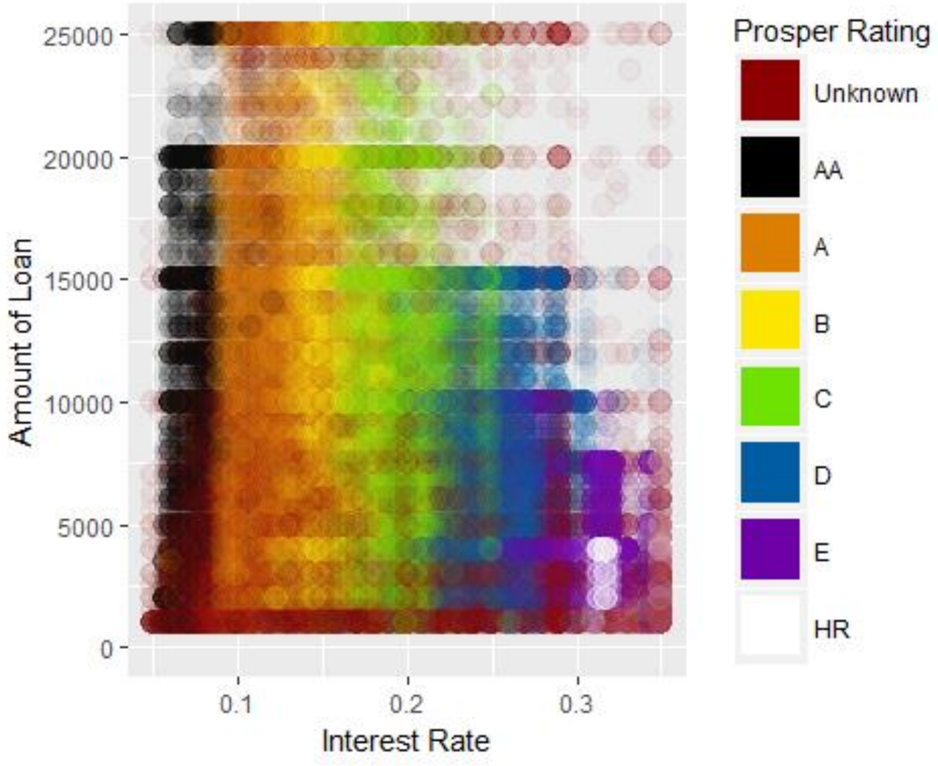
Plot One



Description One

This is definitely one of the 3 plots that I would summarize in the end. And here it is why. I thought about this plot's structure a lot. How should I find the sequences? I have a lot of work in this. I found that from 6 days to 336 days with a 30 step we can see the changes. Less people have delinquency at those times than next to them. It was hard to find.

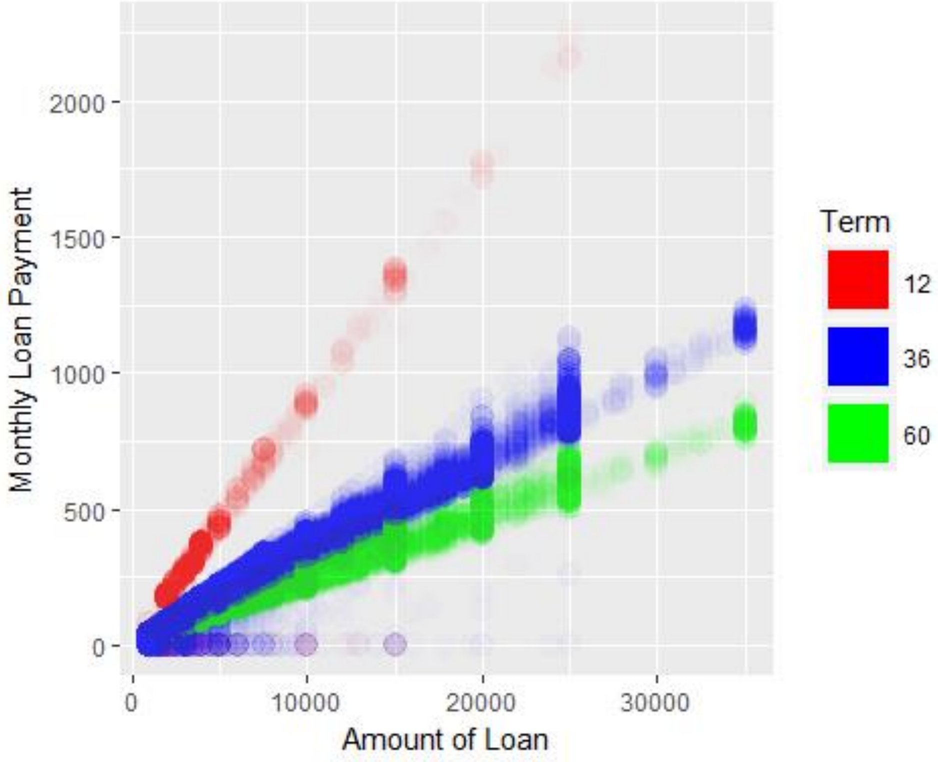
Plot Two



Description Two

Here we can see that how the prosper rating splits vertically the plot. As we get a worse rating, we get higher interest rate. And the borrowed money does not really have an influence on this, except in the HR category.

Plot Three



Description Three

This is a good descriptive plot. When you look at it, you just know what is happening. You see the 3 trend lines connected with the term. It says it all. The shortest term has the highest monthly payment as we expected. Nothing strange.

Reflection

I have investigated the relationships between many variables. But this is still a small part of the 81. But at least I think these are the most important variables. In the future we can investigate more of course, but this gives us a lot of information.

It was hard to get the sequences in the delinquencies plot. I thought a lot about it. Playing with the colors, the plots themselves et cetera. It is good to see now. And there were easier plots, like prosper ratings with interest rates. It is trivial that if you are in a bad rated group, you will have bigger interest rate.

There were some information which would have been good to know before hand. Because only from the dataset we can’t figure out everything. We can only guess.

It would be interesting to explore some date type variables in the future.

Sources

https://eu.udacity.com/ https://stackoverflow.com/