Exploration of Prosper Loan Data

Christopher Ivanovich

June 6, 2017

```
library(tidyverse)
library(gridExtra)
library(choroplethr)
library(data.table)
library(GGally)
library(car)

setwd("c:/users/christopher/desktop/nanodegree/p4-eda/eda project")

variables = c("ProsperScore",
              "ProsperRating..Alpha.",
              "BorrowerAPR",
              "BorrowerRate",
              "LenderYield",
              "IsBorrowerHomeowner",
              "IncomeRange",
              "StatedMonthlyIncome",
              "DebtToIncomeRatio",
              "MonthlyLoanPayment",
              "LoanOriginalAmount",
              "AmountDelinquent",
              "BorrowerState",
              "LoanStatus",
              "CreditScoreRangeUpper",
              "CreditScoreRangeLower")

loans <- read.csv("prosperLoanData.csv", na.strings=c("",".","NA"))[variables]

attach(loans)
```

```
loans$AverageCredit <- round((loans$CreditScoreRangeUpper +
                              loans$CreditScoreRangeLower)/2)

loans <- loans[-15:-16]
```

In this report, we are concerned with a dataset from the Prosper microlending platform, through which investors fund individual loans outside of a commerical bank.

This dataset contains 113,937 observations and (initially) 15 variables chosen from the full set of 81. One of these variables, AverageCredit, has been created by averaging the Upper and Lower Credit Score variables.

```
str(loans)
```

```
## 'data.frame':    113937 obs. of  15 variables:
##  $ ProsperScore         : num  NA 7 NA 9 4 10 2 4 9 11 ...
##  $ ProsperRating..Alpha.: Factor w/ 7 levels "A","AA","B","C",..: NA 1 NA 1 5 3 6 4 2 2 ...
##  $ 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 ...
```

```
##  $ IsBorrowerHomeowner  : Factor w/ 2 levels "False","True": 2 1 1 2 2 2 1 1 2 2 ...
##  $ IncomeRange          : Factor w/ 8 levels "$0","$1-24,999",..: 4 5 7 4 3 3 4 4 4 4 ...
##  $ StatedMonthlyIncome  : num  3083 6125 2083 2875 9583 ...
##  $ DebtToIncomeRatio    : num  0.17 0.18 0.06 0.15 0.26 0.36 0.27 0.24 0.25 0.25 ...
##  $ MonthlyLoanPayment   : num  330 319 123 321 564 ...
##  $ LoanOriginalAmount   : int  9425 10000 3001 10000 15000 15000 3000 10000 10000 10000 ...
##  $ AmountDelinquent     : num  472 0 NA 10056 0 ...
##  $ BorrowerState        : Factor w/ 51 levels "AK","AL","AR",..: 6 6 11 11 24 33 17 5 15 15 ...
##  $ LoanStatus           : Factor w/ 12 levels "Cancelled","Chargedoff",..: 3 4 3 4 4 4 4 4 4 4 ...
##  $ AverageCredit        : num  650 690 490 810 690 750 690 710 830 830 ...
```

```
summary(loans)
```

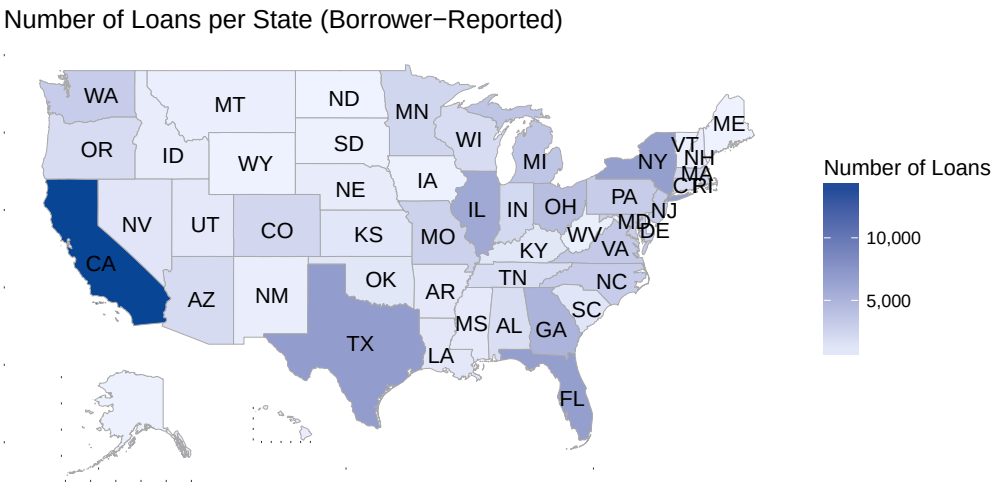
```
##   ProsperScore   ProsperRating..Alpha.  BorrowerAPR       BorrowerRate
##  Min.   : 1.00   C      :18345         Min.   :0.00653   Min.   :0.0000
##  1st Qu.: 4.00   B      :15581         1st Qu.:0.15629   1st Qu.:0.1340
##  Median : 6.00   A      :14551         Median :0.20976   Median :0.1840
##  Mean   : 5.95   D      :14274         Mean   :0.21883   Mean   :0.1928
##  3rd Qu.: 8.00   E      : 9795         3rd Qu.:0.28381   3rd Qu.:0.2500
##  Max.   :11.00   (Other):12307         Max.   :0.51229   Max.   :0.4975
##  NA's   :29084   NA's   :29084         NA's   :25
##   LenderYield      IsBorrowerHomeowner         IncomeRange
##  Min.   :-0.0100   False:56459         $25,000-49,999:32192
##  1st Qu.: 0.1242   True :57478         $50,000-74,999:31050
##  Median : 0.1730                       $100,000+     :17337
##  Mean   : 0.1827                       $75,000-99,999:16916
##  3rd Qu.: 0.2400                       Not displayed : 7741
##  Max.   : 0.4925                       $1-24,999     : 7274
##                                        (Other)       : 1427
##  StatedMonthlyIncome DebtToIncomeRatio MonthlyLoanPayment
##  Min.   :      0     Min.   : 0.000    Min.   :   0.0
##  1st Qu.:   3200     1st Qu.: 0.140    1st Qu.: 131.6
##  Median :   4667     Median : 0.220    Median : 217.7
##  Mean   :   5608     Mean   : 0.276    Mean   : 272.5
##  3rd Qu.:   6825     3rd Qu.: 0.320    3rd Qu.: 371.6
##  Max.   :1750003     Max.   :10.010    Max.   :2251.5
##                      NA's   :8554
##  LoanOriginalAmount AmountDelinquent   BorrowerState
##  Min.   : 1000      Min.   :     0.0   CA     :14717
##  1st Qu.: 4000      1st Qu.:     0.0   TX     : 6842
##  Median : 6500      Median :     0.0   NY     : 6729
##  Mean   : 8337      Mean   :   984.5   FL     : 6720
##  3rd Qu.:12000      3rd Qu.:     0.0   IL     : 5921
##  Max.   :35000      Max.   :463881.0   (Other):67493
##                     NA's   :7622       NA's   : 5515
##                  LoanStatus    AverageCredit
##  Current              :56576   Min.   : 10.0
##  Completed            :38074   1st Qu.:670.0
##  Chargedoff           :11992   Median :690.0
##  Defaulted            : 5018   Mean   :695.6
##  Past Due (1-15 days) :  806   3rd Qu.:730.0
##  Past Due (31-60 days):  363   Max.   :890.0
##  (Other)              : 1108   NA's   :591
```

Nota bene: Many fields in the source CSV file are missing values. When importing the data-set into R Studio,

I assign NA values to these empty fields.

A First Question: Where Are the Borrowers?

```
#we make a data.table for easy joining with the state data set
state_totals <- data.table(group_by(loans, BorrowerState) %>%
                                dplyr::summarize(n()))

state_totals <- state_totals[!is.na(state_totals$BorrowerState),]
#removing the NA row

state_totals <- state_totals[-8,]
#removing DC, which is too difficult to convert

names(state_totals)[1:2] <- c("state.abb","value")
# joining a state name column on a common name for the abb column

state_totals <- inner_join(state_totals, data.table(state.name, state.abb))
names(state_totals)[3] <- c("region")
state_totals$region <- tolower(state_totals$region)

#Make the map
state_choropleth(state_totals,
                 title = "Number of Loans per State (Borrower-Reported)",
                 legend = "Number of Loans", num_colors = 1)
```



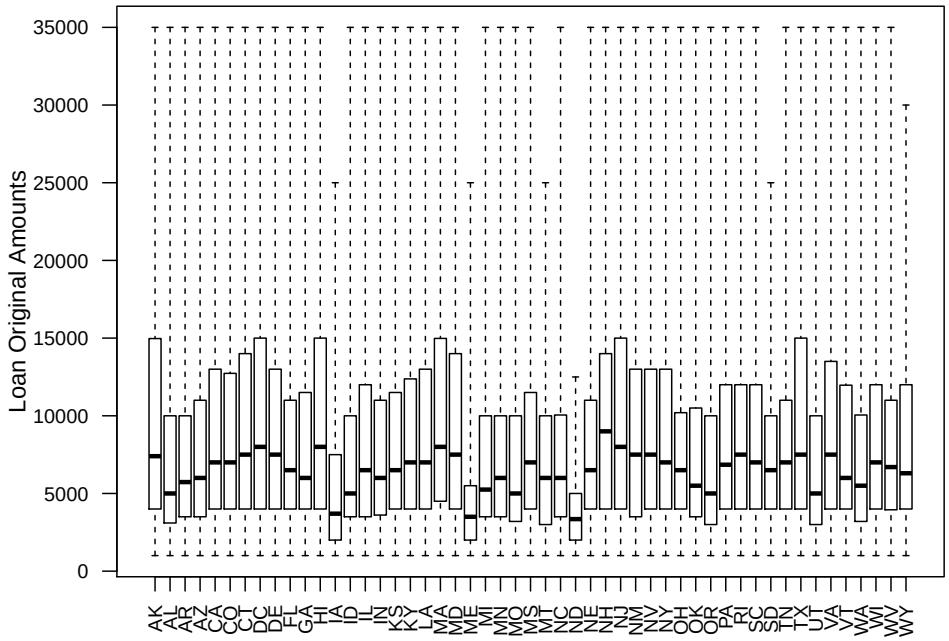
```
#because the chloroplethr package masks dplyr, we unload it

unloadNamespace(choroplethr)
```

Clearly, California leads the way, at least among this set of borrowers for which state values were listed. This may have something to do with Prosper's popularity among the Silicon Valley set as a microlending platform, since it is rooted in the startup culture of that place, or it may have more to do with state populations, as other stand-out states include Texas, Illinois, and New York.

Distribution of Original Loans by State

Let's also consider the distribution of loans across the full set of states (this time including DC):

```
par(las=2, mar = c(8, 4, 2, 2), cex.axis = 0.8)
boxplot(LoanOriginalAmount ~ BorrowerState, loans, range=0, ylab =
          "Loan Original Amounts")
```




```
#Code adapted from an example in Roger Peng's "The Art of Data Science", pg. 48.
```

There were many empty values for borrowers' states of residency, which somewhat surprises–it is hard to imagine one could take out a loan from Prosper without providing at least this modicum of location data.

This begs the question, how do NA-values vary across the different fields of the data set? Let's find out!

```
na_count <-sapply(loans, function(y) sum(length(which(is.na(y)))))

na_counts <- data.frame(round(na_count/nrow(loans), 4))
```

```
names(na_counts)[1] <- "Percent of Loans Missing Value"
na_counts
```

```
##                       Percent of Loans Missing Value
## ProsperScore                                  0.2553
## ProsperRating..Alpha.                         0.2553
## BorrowerAPR                                   0.0002
## BorrowerRate                                  0.0000
## LenderYield                                   0.0000
## IsBorrowerHomeowner                           0.0000
## IncomeRange                                   0.0000
## StatedMonthlyIncome                           0.0000
## DebtToIncomeRatio                             0.0751
## MonthlyLoanPayment                            0.0000
## LoanOriginalAmount                            0.0000
## AmountDelinquent                              0.0669
## BorrowerState                                 0.0484
## LoanStatus                                    0.0000
## AverageCredit                                 0.0052
```

```
#NA counting lambda code courtesty of Stack Overflow:
# https://stackoverflow.com/questions/24027605/
# determine-the-number-of-na-values-in-a-column
```

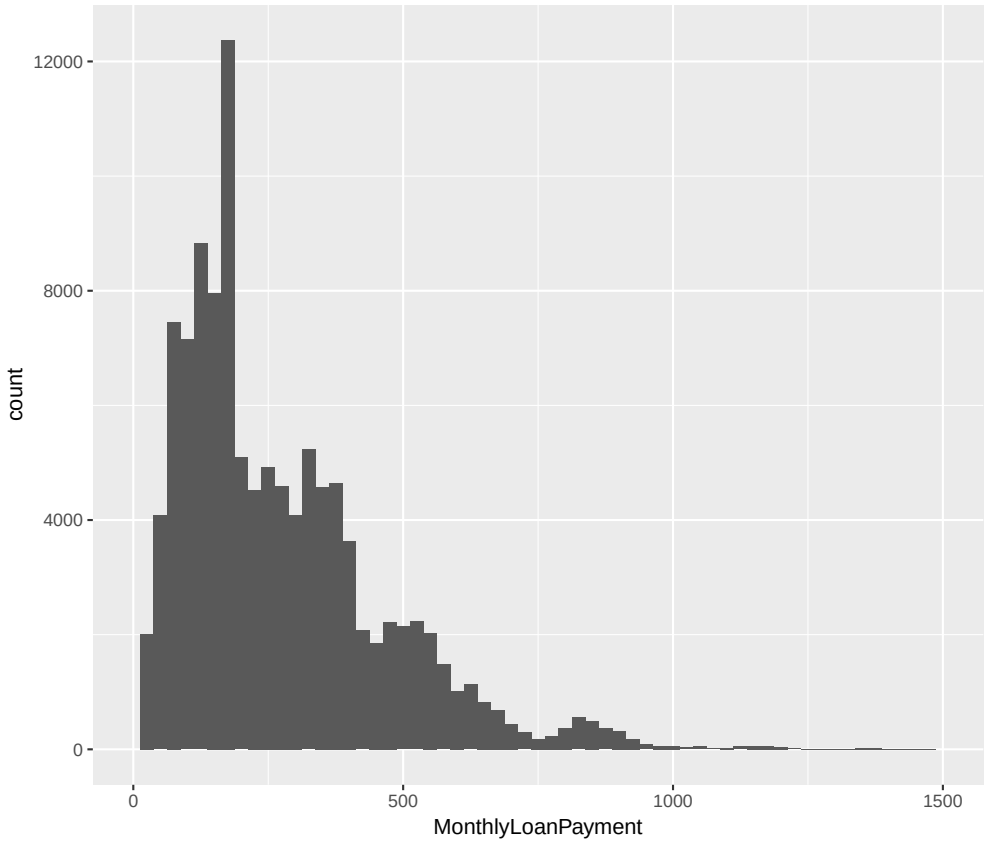
These values are proportions of the total number of observations, and we note the large proportion of entries missing Prosper Scores and Ratings.

These values are key, public-facing predictors of the riskiness of a loan–their absence from this dataset may be suggestive of cherry-picking or tampering with the data to foil just such an analysis as this one (I emphasize that this is only a off-the-cuff hypothesis, but it is worth considering, given how central these values are for lenders on the platform, as it is the only metric lenders can see that is not voluntary and unverified).

Univariate Plots

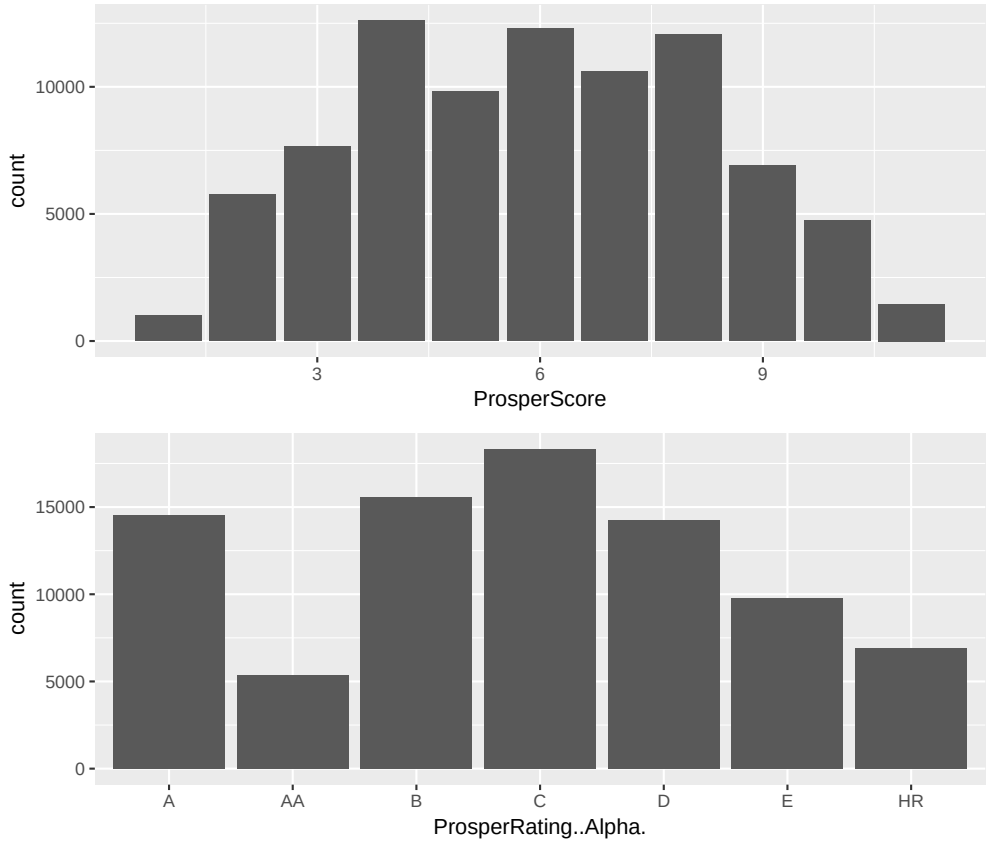
Monthly Loan Payments

```
ggplot(data = loans, aes(x=MonthlyLoanPayment)) +
  geom_histogram(binwidth = 25) + xlim(0, 1500)
```



Prosper's Assigned Borrower Scores and Ratings

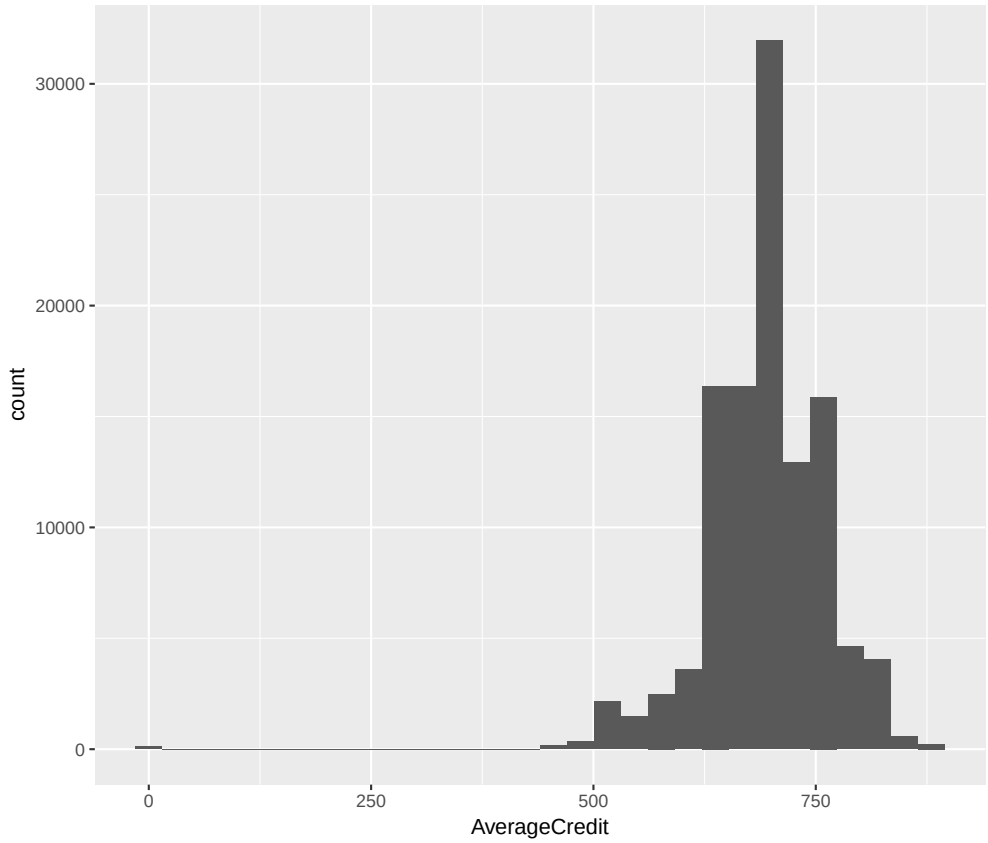
```
q1 <- qplot(data=loans, x=ProsperScore, geom = "bar")
q2 <- qplot(data=subset(loans, !is.na(ProsperRating..Alpha.)),
            x=ProsperRating..Alpha., geom="bar")
grid.arrange(q1, q2, ncol=1)
```



Further Univariate Plots with Some Analysis

Average Credit Scores

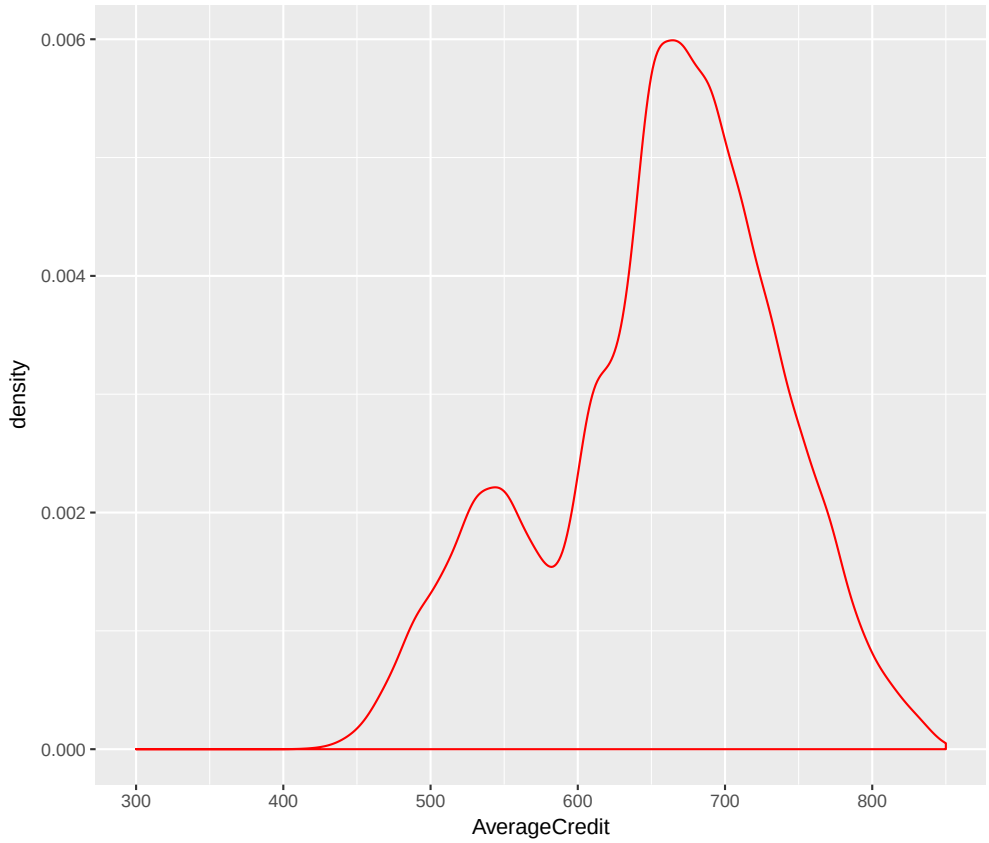
```
ggplot(data=loans, aes(x=AverageCredit)) +
  geom_histogram()
```



As far as FICO (the most common type of) scores go, the current range of possible scores is 300-850. That alone tells us that the values outside this range are likely in error. So, we remove them by setting the limits that we know to be feasible.

Average Credit for Loans in Delinquency

```
delinquency_types <- levels(loans$LoanStatus)[c(5, 7:12)]
late_or_defaulted_loans <- filter(loans, LoanStatus %in% delinquency_types)

ggplot(data=late_or_defaulted_loans, aes(x=AverageCredit)) +
  geom_density(color="red") +
  xlim(300, 850)
```



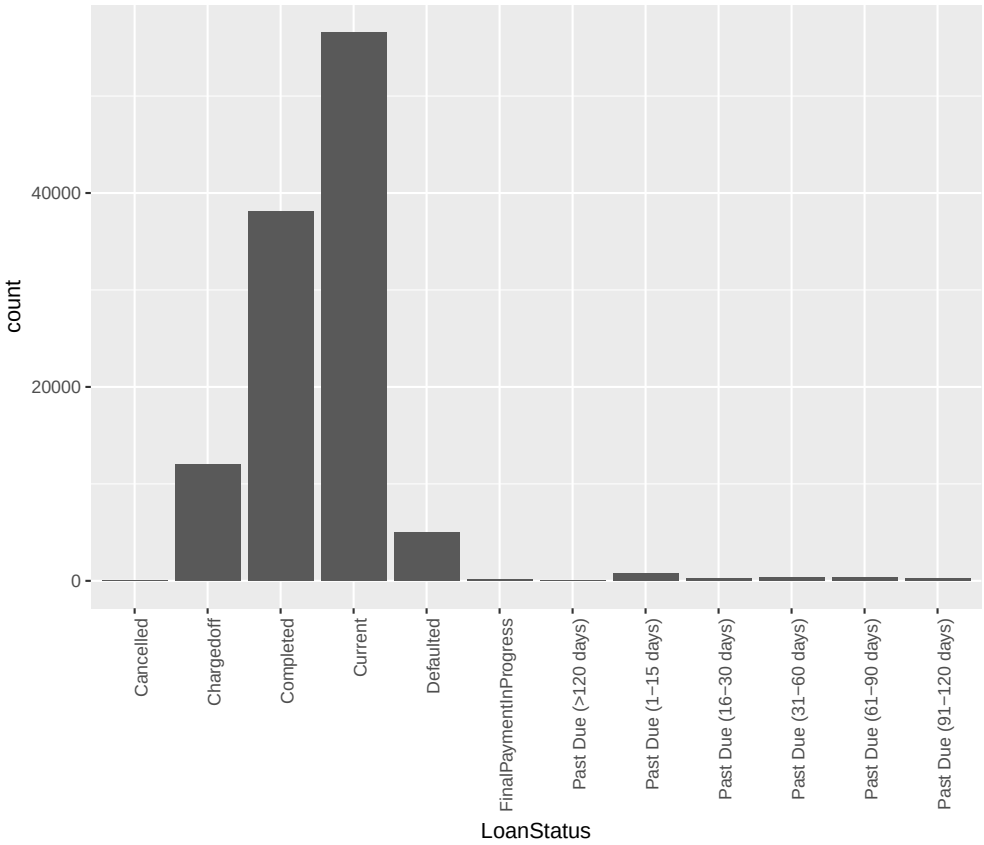
This density graph is limited to only those loans that are in some sort of delinquency, which in the data set are labelled as one of the following:

`"Defaulted", "Past Due (>120 days)", "Past Due (1-15 days)", "Past Due (16-30 days)", "Past Due (31-60 days)", "Past Due (61-90 days)", "Past Due (91-120 days)"`

Compared to the full set of Credit scores, we see much more activity at the lower ends, unsurprisingly.

Loan Statuses

```
ggplot(data=loans, aes(x=LoanStatus)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



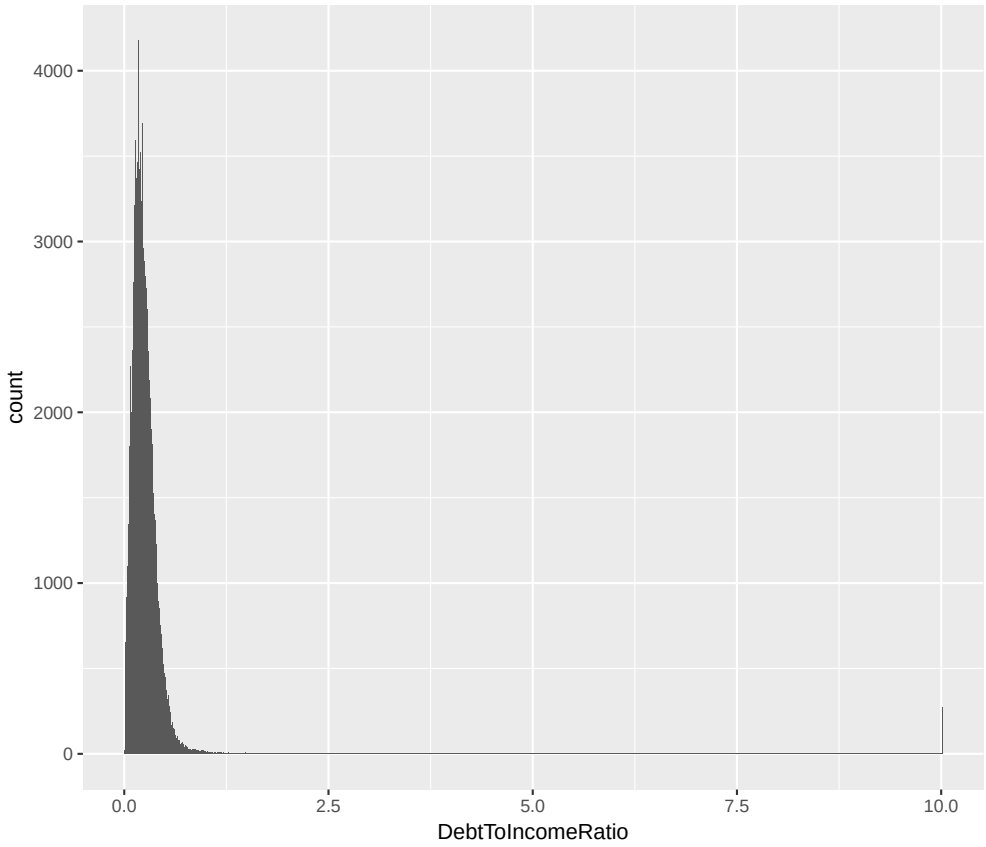
In the density chart above, you may have noticed the extremely small range of 0.000 to 0.006, which is the proportion of delinquent loans by credit score for the whole dataset.

Following up on that here, we see very few loans that are actually past due in this data set. Wikipedia informs me that "Chargedoff" means a creditor has written off the loan as unlikely to be paid (usually after 6 months of non-payment), so there is something incongruous about having 17,000 loans (about 14% of all loans) or so charged off or defaulted, and an almost total lack of Past Due loans–one would think becoming Past Due is a necessary first step to defaulting on a loan or having it charged off, so there should be a larger proportion of past due loans here.

Given how few "past due" loans there are, we will remove them from future plots involving Loan Status.

Debt to Income Ratio

```
ggplot(data=loans, aes(x=DebtToIncomeRatio)) +
  geom_histogram(binwidth=.01)
```



We note the mode around a value of 0.50, as well as the distant outliers in which borrowers on Prosper apparently have 10.01-times as much debt as income–one wonders what their Prosper scores are, and how many investors they could get with such a scary ratio of debt to income...

```
##   ProsperScore   AverageCredit   ProsperRating..Alpha.
##  Min.   :1.000   Min.   :470.0   HR     : 19
##  1st Qu.:3.000   1st Qu.:630.0   E      : 11
##  Median :4.500   Median :690.0   B      :  6
##  Mean   :4.478   Mean   :678.8   D      :  5
##  3rd Qu.:6.000   3rd Qu.:730.0   C      :  3
##  Max.   :8.000   Max.   :870.0   (Other):  2
##  NA's   :226                     NA's   :226
##          IncomeRange  StatedMonthlyIncome DebtToIncomeRatio
##  $1-24,999     :187   Min.   :    0.000   Min.   :10.01
##  Not displayed : 59   1st Qu.:    0.083   1st Qu.:10.01
##  Not employed  : 24   Median :    0.083   Median :10.01
##  $100,000+     :  1   Mean   :  133.299   Mean   :10.01
##  $50,000-74,999:  1   3rd Qu.:    0.396   3rd Qu.:10.01
##  $0            :  0   Max.   :17083.333   Max.   :10.01
##  (Other)       :  0
##  MonthlyLoanPayment LoanOriginalAmount
##  Min.   :   0.0     Min.   : 1000
##  1st Qu.: 107.2     1st Qu.: 3000
##  Median : 209.5     Median : 6000
##  Mean   : 301.9     Mean   : 8296
##  3rd Qu.: 388.6     3rd Qu.:10000
##  Max.   :1047.6     Max.   :25000
```

##

Quite a surprising amount of variation in this group of

```
nrow(loans[which(loans$DebtToIncomeRatio>=10.01),])
```

```
## [1] 272
```

loans. Given the lack of intermediate debt:income ratios, and this variation, it is possible that these extreme outliers are entry errors of some sort, but if so, they are not few in number.

We note that the maximum loan in this set is $25,000, which falls short of the largest loans offered by Prosper ($35,000).

A quick call to the `table` function, to take a look at the income ranges of those with such extreme debt-burdens, reveals:

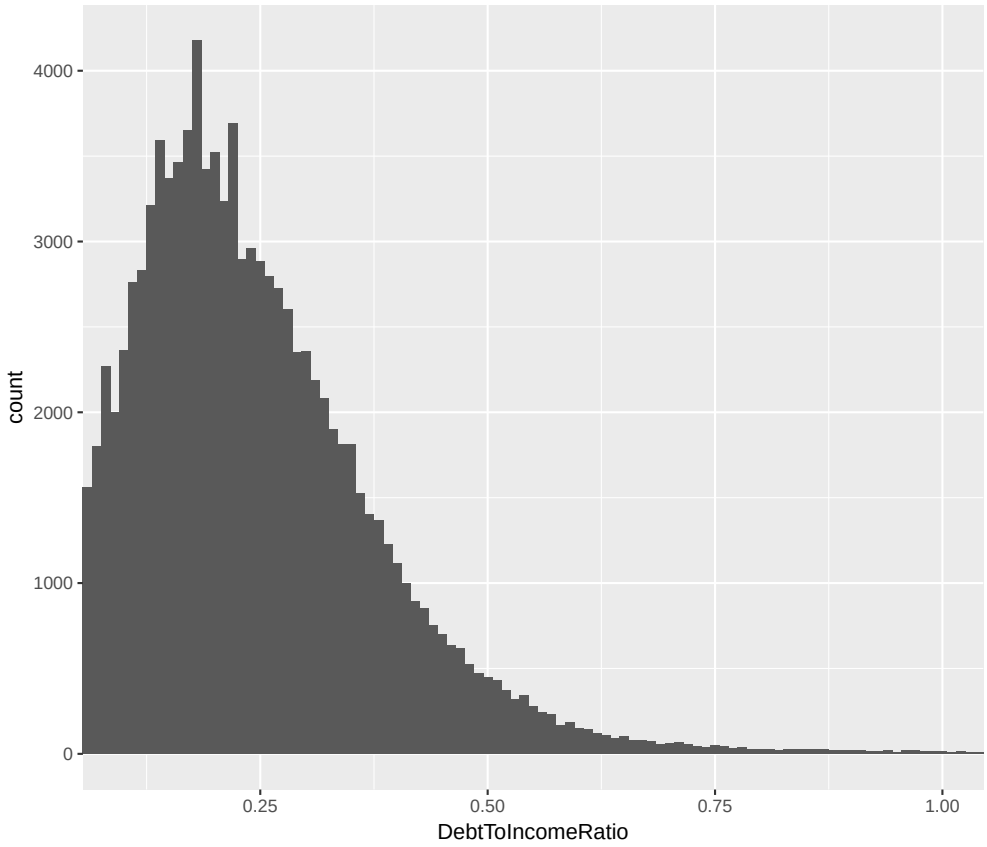
```
##
##      $1-24,999  Not displayed   Not employed      $100,000+ $50,000-74,999
##            187             59             24              1              1
##             $0 $25,000-49,999 $75,000-99,999
##              0              0              0
```

These results likely scuttle a pet theory I had that a significant number of these highly indebted people are upper-class individuals using Prosper as a source of speculative capital. It looks like the majority are quite low-income, or else lying about their income, since so many have reported zero income: The mean stated income is only ~$133 USD per month.

Debt to Income Histogram, non-special cases

In our **Debt:Income histogram**, the vast majority of the data is clustered within a very narrow range of about (0. 0.8), so we cut out the outliers by setting reasonable limits of 0.1 and 1.0:

```
ggplot(data=loans, aes(x=DebtToIncomeRatio)) +
  geom_histogram(binwidth=.01) +
  coord_cartesian(xlim=c(0.1, 1.0))
```

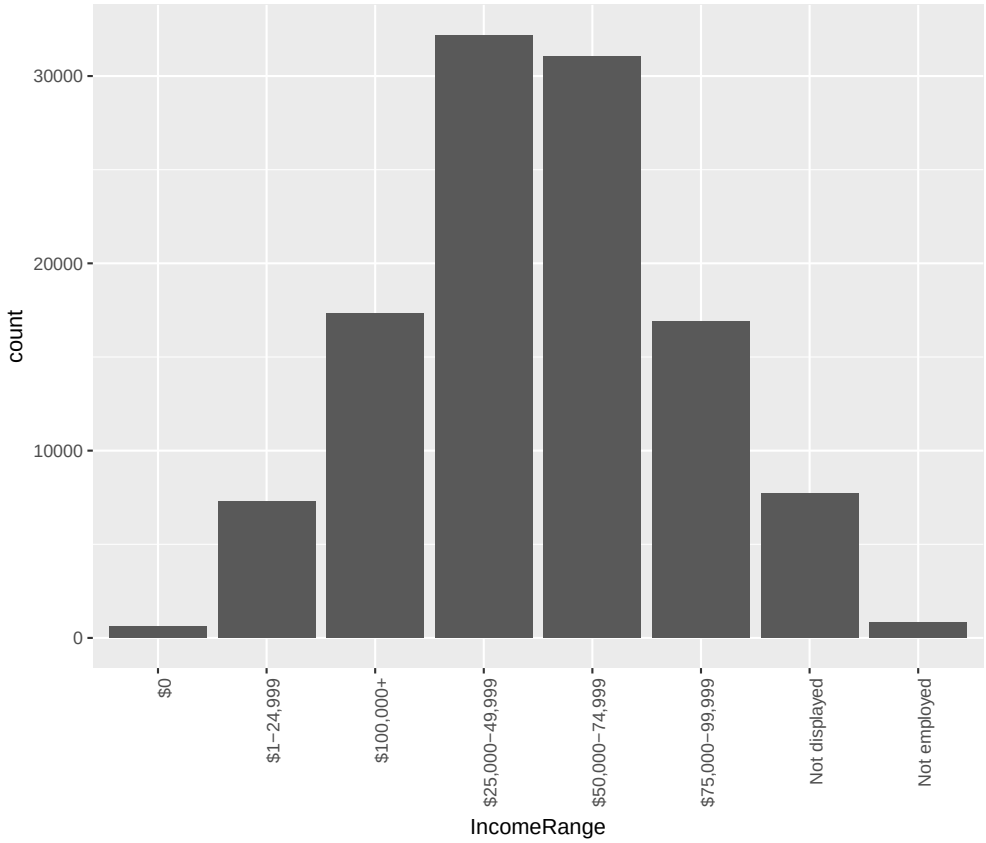


Here we see a normal, positively-skewed distribution. Our earlier summary data for **DebtToIncomeRatio** tells us that the mean value is 0.276.

Income Range

For our next plot, we look at **Income Range**:

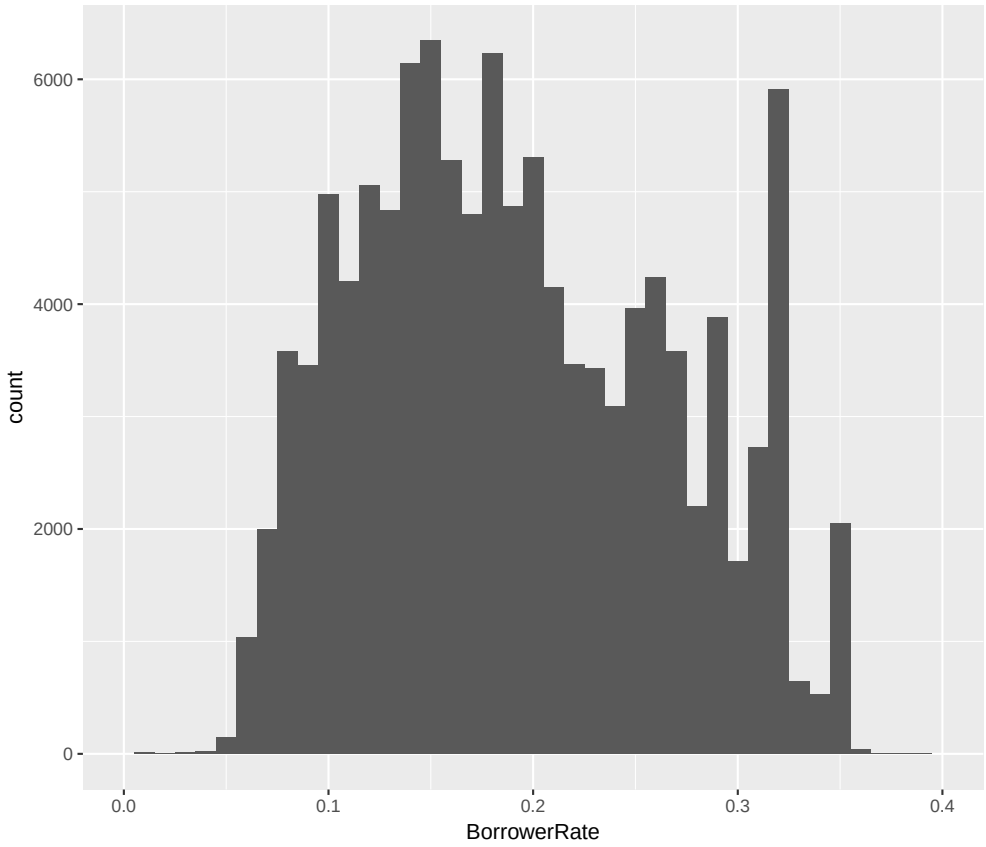
```
ggplot(data=loans, aes(x=IncomeRange)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



This barchart looks wonderfully normal as well, though the granularity of such wide intervals is probably quite low.

Next, we look at **BorrowerRate**, which is the interest rate assigned to the borrower by Prosper:

```
ggplot(data=loans) +
  geom_histogram(aes(x=BorrowerRate), binwidth=.01) +
  xlim(c(0.0, 0.4))
```



These are percentage-rates, and we see a nearly normal distribution with an interesting secondary peak around 32% interest, which is quite a high rate. This makes me wonder what the Prosper scores are for these borrowers, that so many of them would be dealt such a punishing interest rate.

How many borrowers are in this rate bracket, and what do their other variables look like? We subset for borrowers with rates in the open interval of (0.31, 0.32), which contains 7427 loan data entries, and we take a summary of a selection of variables:

```
poor_saps <- subset(loans, BorrowerRate >= 0.31 & BorrowerRate <=0.32)
summary(poor_saps[c("ProsperRating..Alpha.",
                    "StatedMonthlyIncome",
                    "BorrowerRate",
                    "LenderYield",
                    "AverageCredit")])
```

```
##  ProsperRating..Alpha. StatedMonthlyIncome  BorrowerRate
##  HR     :5562          Min.   :      0     Min.   :0.3100
##  E      :1654          1st Qu.:   2500     1st Qu.:0.3160
##  D      :  35          Median :   3875     Median :0.3177
##  C      :   3          Mean   :   4992     Mean   :0.3171
##  B      :   1          3rd Qu.:   5833     3rd Qu.:0.3177
##  (Other):   0          Max.   :1750003     Max.   :0.3200
##  NA's   : 172
##   LenderYield     AverageCredit
##  Min.   :0.2900   Min.   :530.0
##  1st Qu.:0.3060   1st Qu.:670.0
##  Median :0.3077   Median :690.0
##  Mean   :0.3071   Mean   :689.8
```

```
##  3rd Qu.:0.3077   3rd Qu.:710.0
##  Max.   :0.3100   Max.   :870.0
##                   NA's   :5
```

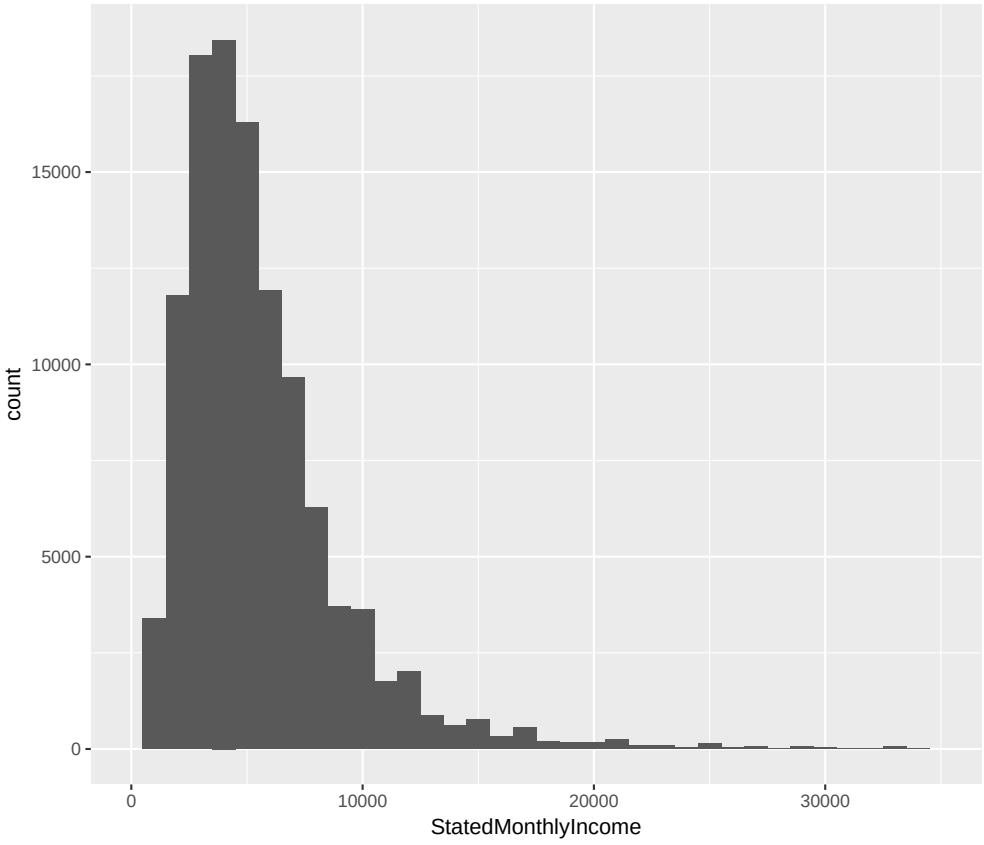
As we see, most of these people have the worst possible Prosper Rating of "HR", i.e. high-risk, with on-average higher debt:income ratios, but a surprisingly high monthly income mean and median. We note that the max stated income value in our subset matches that in the summary of the full dataset, and clearly this high-earner is distorting the mean. Yet, the median monthly income remains solidly high, even for this HR subset, which might help explain why the **LenderYield** parameters do not deviate substantially from the interest rate.

From the perspective of Prosper, it would make sense to only offer these high-risk, high interest rate loans to individuals with high incomes.

The distribution of the credit scores is a bit of a mystery–we would expect somewhat uniformly low values for such high interest rates and terrible ratings. We also note that there are only 5 NA values for credit scores in this entire data set–this suggests that Prosper deems these values as of great importance, and they may be the best predictor of loan size, interest rate, and other fields (to be explored in the multivariate section further below).

For now, let's take a look at the distribution of monthly incomes with the self-professed millionaire outlier removed, to see if anything else emerges.

```
ggplot(data=loans) +
  geom_histogram(aes(x=StatedMonthlyIncome), binwidth=1000) +
  xlim(c(0, 35000))
```



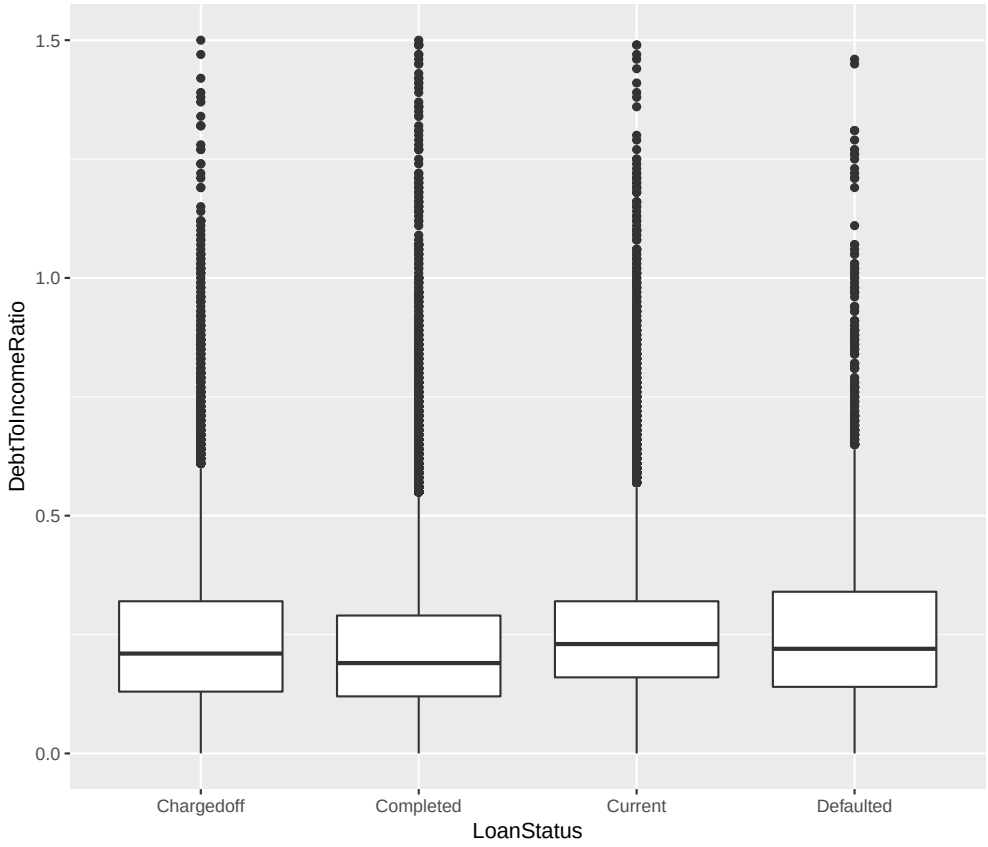
It looks like the bulk of the borrowers here inhabit the American middle-class income range, and now the intervals chosen by the data curator in the IncomeRange variable make more sense.

We could log10 transform our counts to flatten the peaks a bit and better visualize the right tail of this positively skewed distribution, but no good reason to do so comes to mind just yet. Given the non-trivial amounts of high-earners here, though, one wonders what their occupations and other indicators of wealth and financial activity might look like (or are they simply exaggerating?).

Bivariate Plots with Analysis

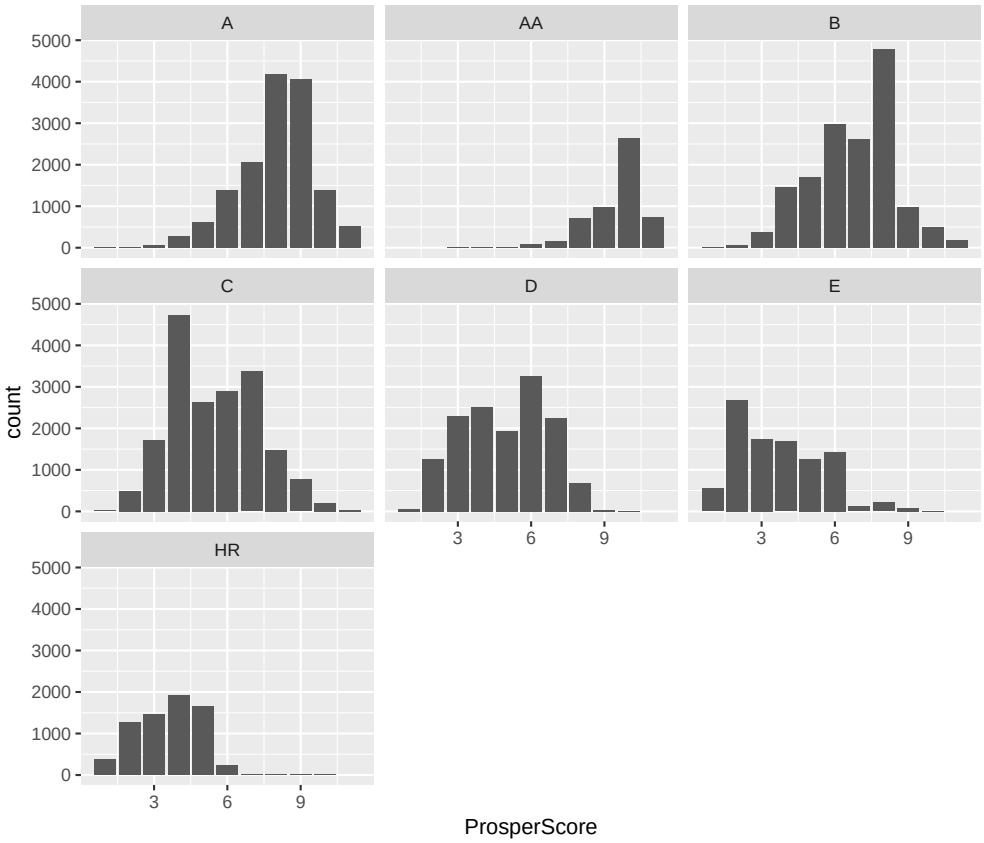
Debt to Income Ratio and Loan Status

```
valid_statuses <- c("Completed", "Current", "Chargedoff", "Defaulted")
loans_with_status <- subset(loans, LoanStatus %in% valid_statuses)
ggplot(data=loans_with_status,
       aes(x = LoanStatus, y = DebtToIncomeRatio))+
  geom_boxplot() +
  scale_y_continuous(limits = c(0.0, 1.5))
```



Prosper Scores versus Prosper Ratings

```
ggplot(data=subset(loans, !is.na(ProsperRating..Alpha.))) +
  geom_histogram(aes(x=ProsperScore), stat="count") +
  facet_wrap(~ ProsperRating..Alpha.)
```



This histogram counts up Prosper Scores, separated by so-called Prosper Ratings.

At first glance, this appears quite strange–Why is there so much divergence between the two scoring methods? Prosper is pretty mum on the definition of the **ProsperRating** (i.e, ratings of AA-HR), beyond calling it a proprietary scoring method for evaluating risk, with AA being the best, HR being the worst.

ProsperScores (not to be confused with Prosper's (Numeric) Ratings) are also defined as a risk-measuring tool in the range:

1 -> 11, i.e., Riskiest -> Safest

If these methods have any kind of parity, we would expect a very high density of 10-11 Scores to be in the "AA" rating graph above, and we would only expect to see scores of 1-2 under the "HR" facet of our graph.

The power of the `table` function allows us to quickly tabulate the frequencies of ProsperScore vs. ProsperRating:

```
##              ProsperRating..Alpha.
## ProsperScore    A   AA    B    C    D    E   HR
##            1     2    0    3   21   49  552  365
##            2     6    0   61  490 1255 2692 1262
##            3    53    4  367 1711 2295 1750 1462
##            4   285   14 1451 4738 2506 1681 1920
##            5   608   11 1695 2642 1940 1262 1655
##            6  1392   94 2968 2900 3256 1438  230
##            7  2050  164 2624 3370 2251  118   20
##            8  4170  718 4781 1482  679  214    9
##            9  4070  983  968  770   36   75    9
##            10 1394 2645  492  196    7   13    3
```

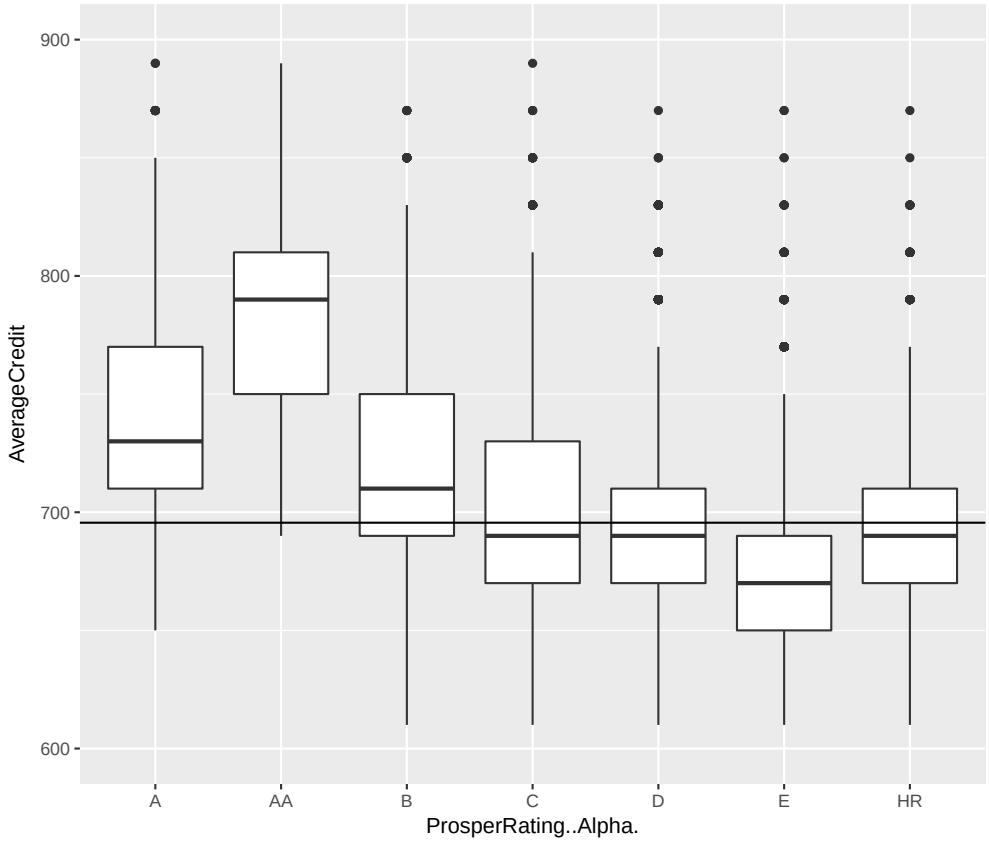
```
##           11  521  739  171   25    0    0    0
```

We see that there does appear to be some consistency in the frequencies between these two methods, but that the relationship appears weaker than we might hope or expect. That is, aside from the large spread of the data, the modal letter Ratings appear to correspond to particular numerical Scores, though with oddities: scores of 10-11 peak at the best rating of AA, a score of 9 peaks at A and 8 peaks at B, but 6 peaks at D while 4-5 peak at C.

This divergence tells us that there are some significant differences in how risk gets assigned to using these two different "proprietary" representations.

Average Credit Against Other Fields

```
ggplot(aes(x=ProsperRating..Alpha., y=AverageCredit),
       data = subset(loans, !is.na(ProsperRating..Alpha.))) +
  geom_boxplot(na.rm = T) + ylim(c(600,900)) +
  geom_hline(yintercept = mean(loans$AverageCredit, na.rm = T))
```



We see some interesting results here: The median credit scores for loans rated lower than B are all quite closely clustered, with the exception of E, which is oddly lower and with greater variation than the riskiest category of HR.

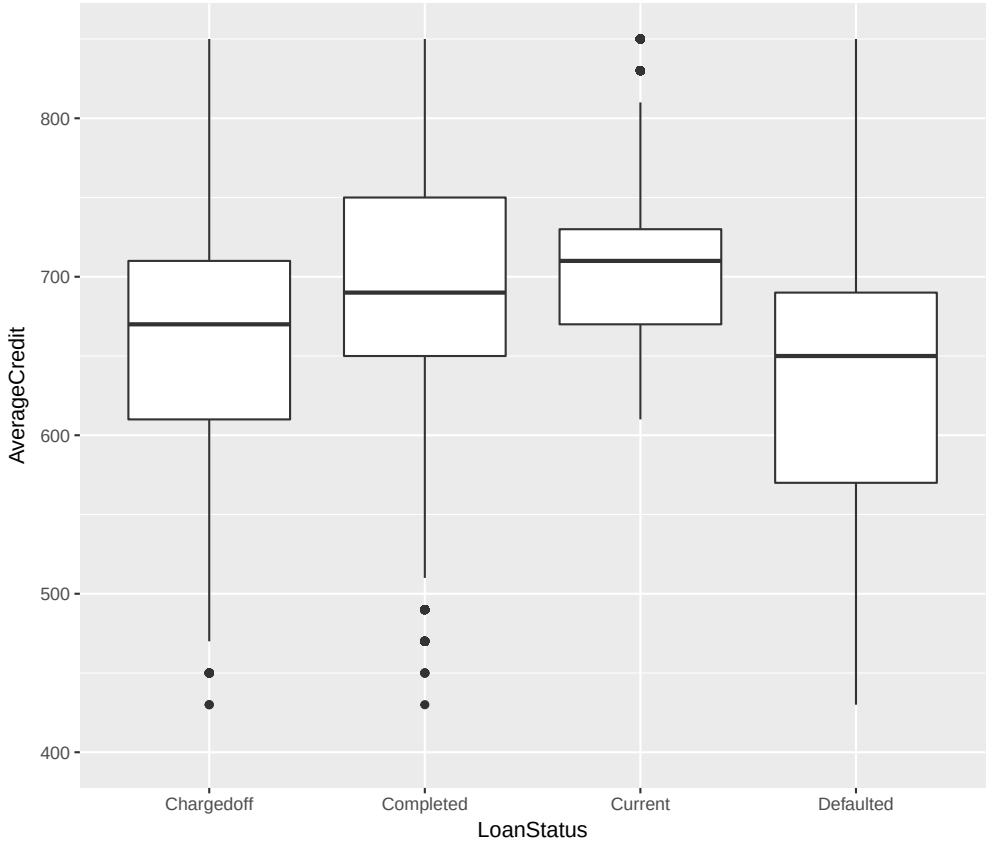
The horizontal line represents the avreage of all Average Credit Scores, and we see that the loans you would probably be too scared to invest in if you took Prosper's Ratings at face-value all have medians below the mean.

It's also interesting to note that the Average Credit Score ranges of ratings B through HR all roughly coincide, while the "safe" bets of AA and A do indeed appear to have narrower and higher ranges, as well as higher

medians. It is certainly unexpected that HR loans seem to tend towards better credit scores than E-rated loans. This suggests that credit scores are only one facet of how these ratings get assigned (or else, it's a bit of a crapshoot at the allegedly riskier end of the spectrum).

Loan Status and Average Credit

```
status_types <- levels(loans$LoanStatus)[c(2:5)]
loans_with_status <- filter(loans, LoanStatus %in% status_types)

ggplot(data=loans_with_status, aes(x=LoanStatus, y=AverageCredit)) +
  geom_boxplot() +
  ylim(400, 850)
```

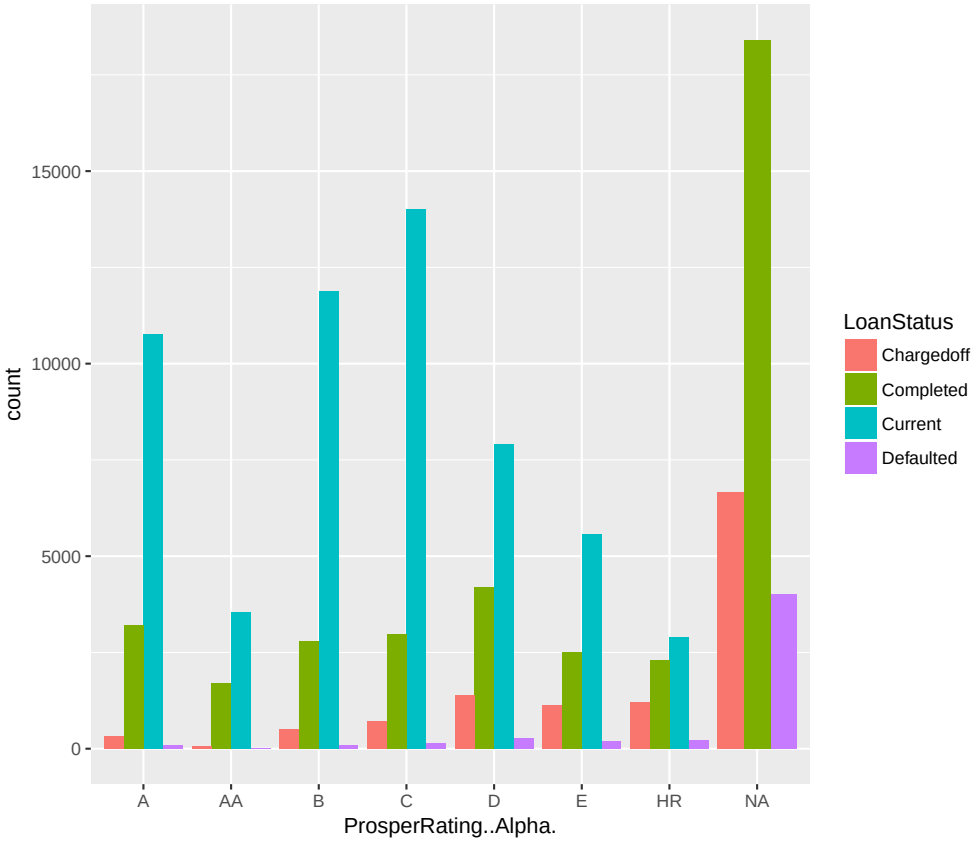


This time, we only consider loans in these four categories of **LoanStatus**. We note some interesting things– Completed loans have a wide span of credit scores, but the 50% bulk is solidly above 700. Chargedoff loans and Defaulted loans don't suggest terribly bad credit scores on average, but they're certainly more likely to be lower than those with completed payments. And, most of all, we note how strange it is that loans currently in repayment are such a delightfully high subset with extremely high ratings. Has there been some shift in Prosper's business model to explain this? Or is this an instance of extreme cherry-picking on their part?

Loan Status and Prosper Rating

```
ggplot(data=loans_with_status, aes(x=ProsperRating..Alpha., fill=LoanStatus)) +
  geom_bar(position = "dodge") +
```

```
  scale_color_brewer(type="qual")
```



Some stand-out observations: - The massive spike in Completed loans with missing Prosper Ratings. - The total lack of missing Ratings for loans currently in progress (makes sense, as investors need those scores). - Completed loans have quite similar counts across the Prosper Ratings.

Bivariate Correlations

As noted some time ago, credit score seems to have a major impact on other aspects of a given loan entry–this makes intuitive senese, since Prosper is not likely to take things like reported income, home ownership, and other self-reported factors into great account, as all of these should be accounted for in a credit history report with greater reliability than someone's written claims.

```
loans_samp <- loans[sample(1:length(loans$AverageCredit), 5000), ]
loans_samp <- loans_samp[, c(1,4:5, 8:9, 10:12, 15)]

ggcorr(loans_samp, name = "Pearson's CorrCoeff")
```

In the above correlation matrix, we see, not surprisingly, that high average credit scores correlate negatively with lender yield and also with the borrower rates attached to loans–good scores seem to indicate lower interest payments.

Other noteworthies include the fairly negative coefficient between a Credit Score Rating and the estimated %-gains for the lenders (lender yield). This indicates that those with lower credit scores are more likely to fail to make payments–at least, as per Prosper's predictions.

We see only slight correlation between the size of a loan and Average Credit Score–this weakness may perhaps have to do with the somewhat discrete nature of loan sizes (only well-defined amounts are on offer), and with the fact that people likely regard Prosper as a provider of only smallish loans (which it is).

Further, we see that there is almost no correlation between the average of the credit scores and the debt to income ratio. This is a bit of a surprise, since one would expect there to be some relationship between the amount of debt a person can take on, and that person's credit history. But perhaps this is an industry-wide phenomenon–high credit scores reflect financial prudence and thus limited indebtedness.

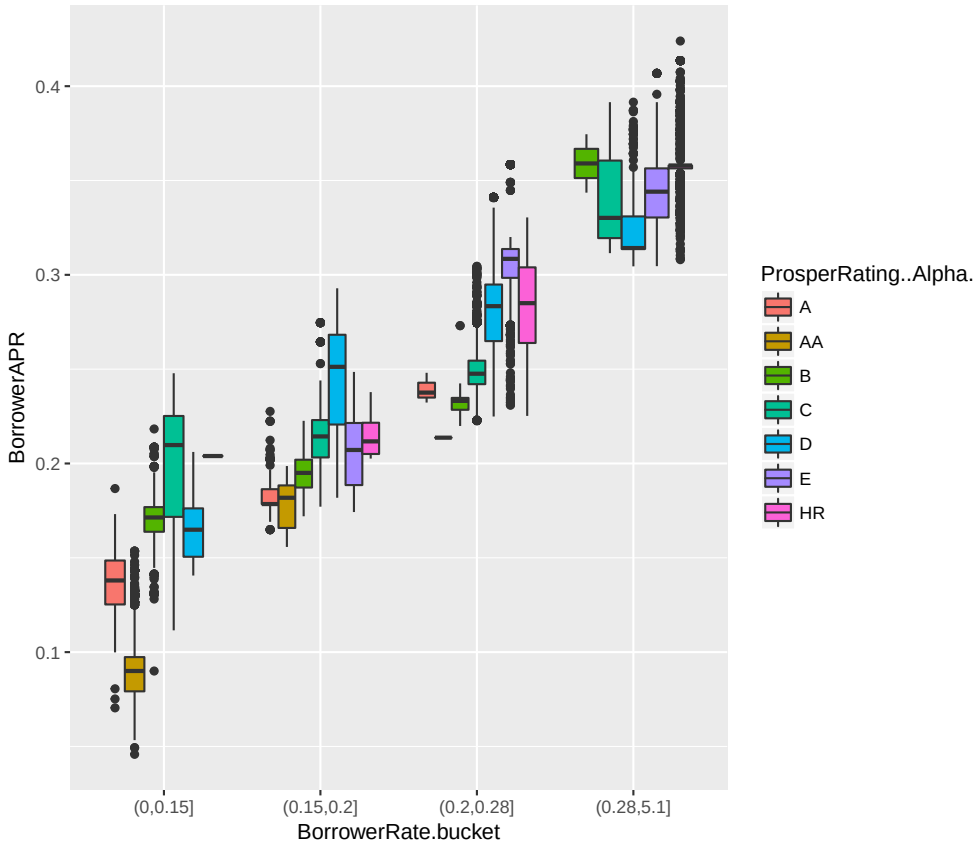
Stated monthly income has only weakly positive correlation with average credit scores, which we might interpret as further invalidating the validity of self-reported income.

Finally, we note the almost perfect correlation between a borrower's rate of interest and lender yield. The lender yield, according to the variable book which accompanied the data set, is an estimate based on the assigned interest minus the expected servicing fee. This suggests very powerfully the extent to which Prosper is dependent on credit score reports in assigning interest rates and developing it's own estimates on ROI for its microlenders.

Multivariate Plots, Grouping, and Analysis

Interest Assignments and Prosper Ratings

```
loans$BorrowerRate.bucket = cut(loans$BorrowerRate,
                                c(0. , 0.15, 0.2, 0.28, 5.1))

ggplot(aes(x = BorrowerRate.bucket,
           y = BorrowerAPR ,
           fill = ProsperRating..Alpha.),
       data = subset(loans, !is.na(ProsperRating..Alpha.)) ) +
  geom_boxplot( )
```

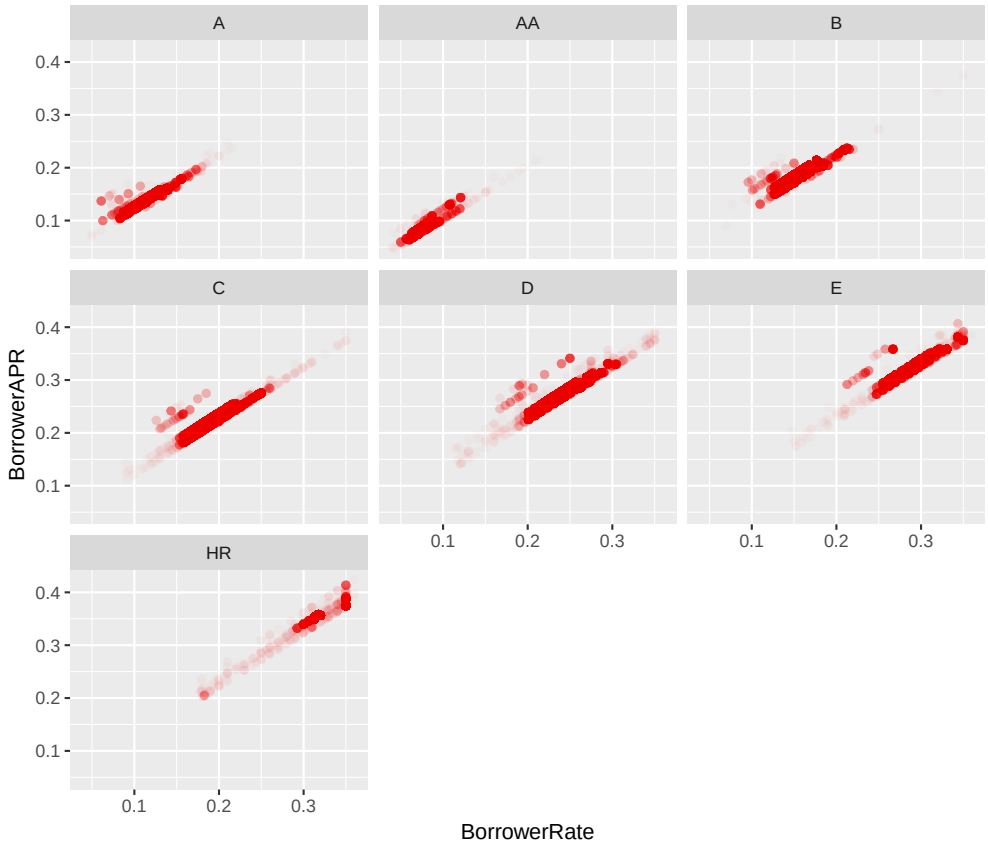


Why have I plotted a borrower's APR against the interest rate on their loan? Simply to confirm that Prosper is using a consistent method for servicing loans (unlike an interest rate, an APR includes things like service and transaction fees, a.k.a. Prosper's share of the loan pie).

We note some outlier values for many of these Rate buckets, and that the Ratings have less explanatory power the the greater the interest rate on the loan–i.e., higher APR and interest rates have less variation across the Ratings. BUt, we also note that loans rated as AA or A are absent in the highest interest rate bucket.

We can also facet scatter plots to get a clearer look at the Ratings in which the two rates diverge a bit.

```
ggplot(data=subset(loans, !is.na(ProsperRating..Alpha.)),
                   aes(x=BorrowerRate, y=BorrowerAPR)) +
  geom_point(alpha=1/50, color = "red") +
  facet_wrap(~ProsperRating..Alpha.)
```



This plot more clearly illustrates the distributions of loan interest and APR rates across Rating categories.

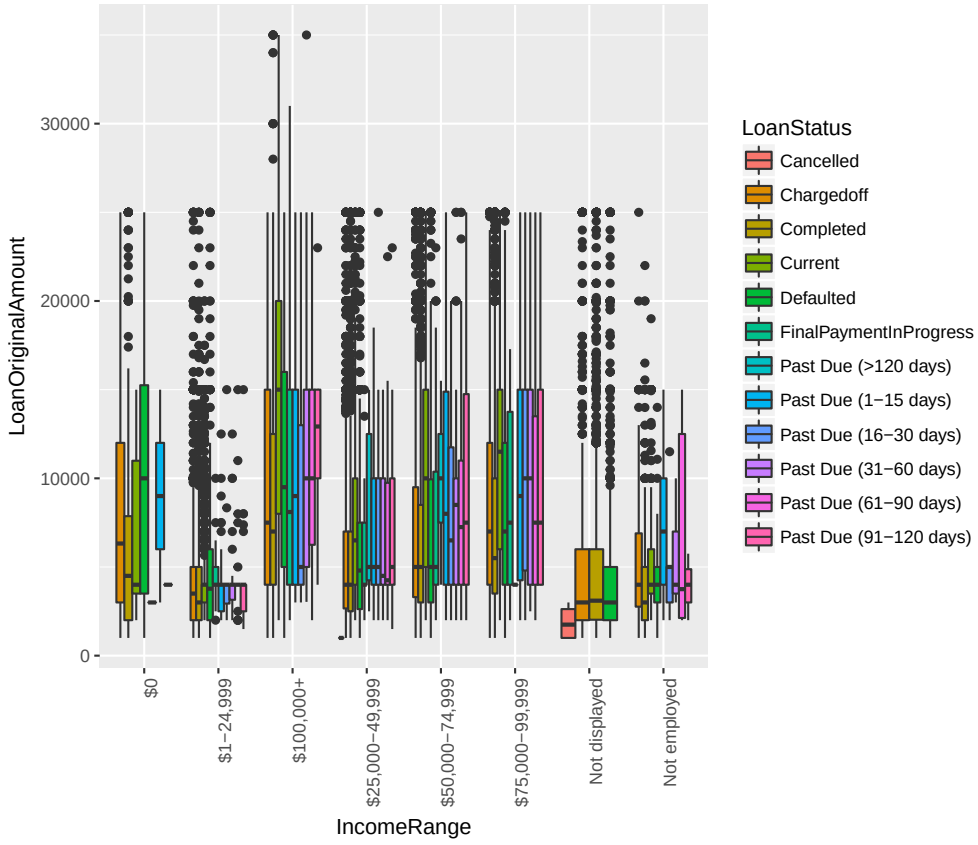
Without surprise we note that well-rated loans enjoy generally lower interest rates, though there is quite a wide range of borrowing rates across the Prosper ratings. Is there a graduated interest rate related to the size of the loan?

We shelve that question for a moment, to be pursued in a bit–for now, we note that there are different APRs assigned to the same borrower interest rates even within Rating categories. What's that about?

For the most part, these varying APRs seem to range discretely, i.e., there are simply categories of APR, which we infer by the fact that each facet graph seems to be composed of four or five distinct linear trends, representing the relationship between APR and loan interest rate. This would suggest that certain categories of loans get assigned a certain percentage on top of the loan interest rate to yield an APR–i.e. the percent-increase of the additional interest is not simply a function of a Prosper Rating.

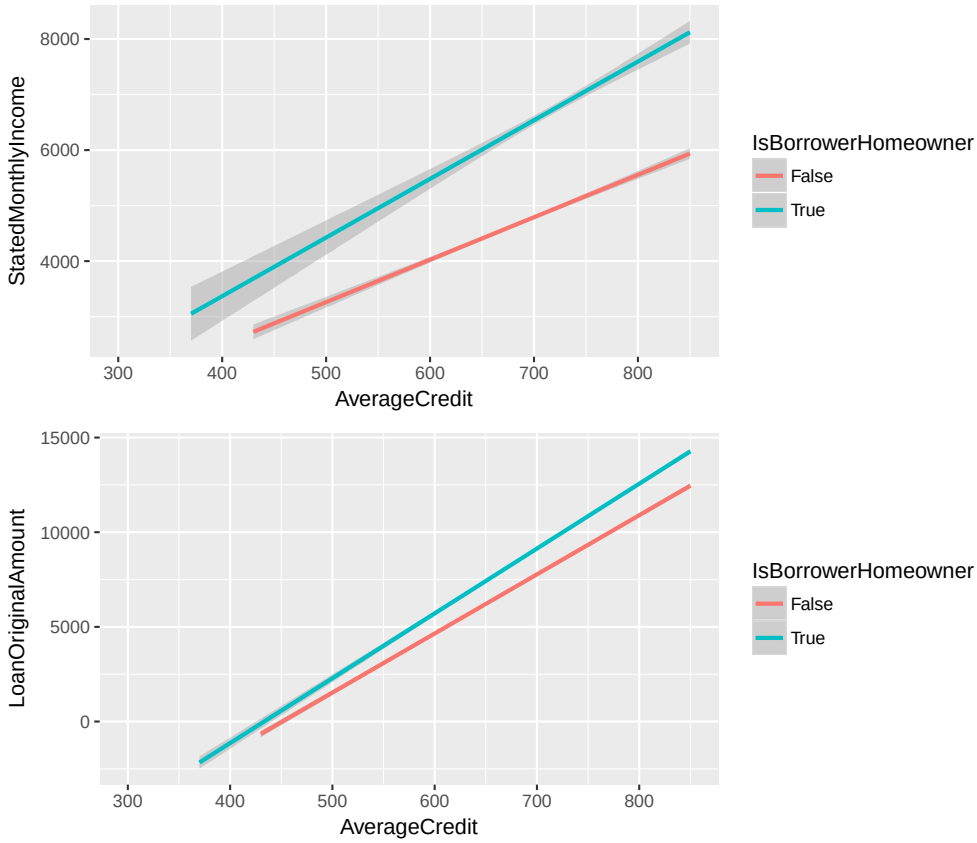
Income Group, Loan size, and Employment Status

```
ggplot(aes(x = IncomeRange,
           y = LoanOriginalAmount),
           data=subset(loans, (StatedMonthlyIncome < 11000))) +
   geom_boxplot(aes(fill = LoanStatus) )   +
      theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



AverageCredit versus Reported Income, versus Loan Amount, Faceted for Home Ownership Status

```
p1 <-  ggplot(data=subset(loans, StatedMonthlyIncome > 0),
              aes(x = AverageCredit, y=StatedMonthlyIncome)) +
  geom_smooth(method="lm", aes(color=IsBorrowerHomeowner)) + xlim(300, 850)

p2 <- ggplot(data=loans, aes(x=AverageCredit, y=LoanOriginalAmount)) +
  geom_smooth(method="lm", aes(color=IsBorrowerHomeowner)) + xlim(300, 850)

grid.arrange(p1,p2, ncol = 1)
```




```
lm(AverageCredit ~ BorrowerRate,
   data=subset(loans, IsBorrowerHomeowner == "True"))
```

```
##
## Call:
## lm(formula = AverageCredit ~ BorrowerRate, data = subset(loans,
##     IsBorrowerHomeowner == "True"))
##
## Coefficients:
##  (Intercept)  BorrowerRate
##          787          -395
```

```
lm(AverageCredit ~ BorrowerRate,
   data=subset(loans, IsBorrowerHomeowner == "False"))
```

```
##
## Call:
## lm(formula = AverageCredit ~ BorrowerRate, data = subset(loans,
##     IsBorrowerHomeowner == "False"))
##
## Coefficients:
##  (Intercept)  BorrowerRate
##        750.6        -367.9
```

The oddly truncated lines owe to the fact that I have restricted the AverageCredit values to the range of possible FICO scores.

Apart from the obvious fact that homeowners tend to report higher monthly incomes across both distribu-

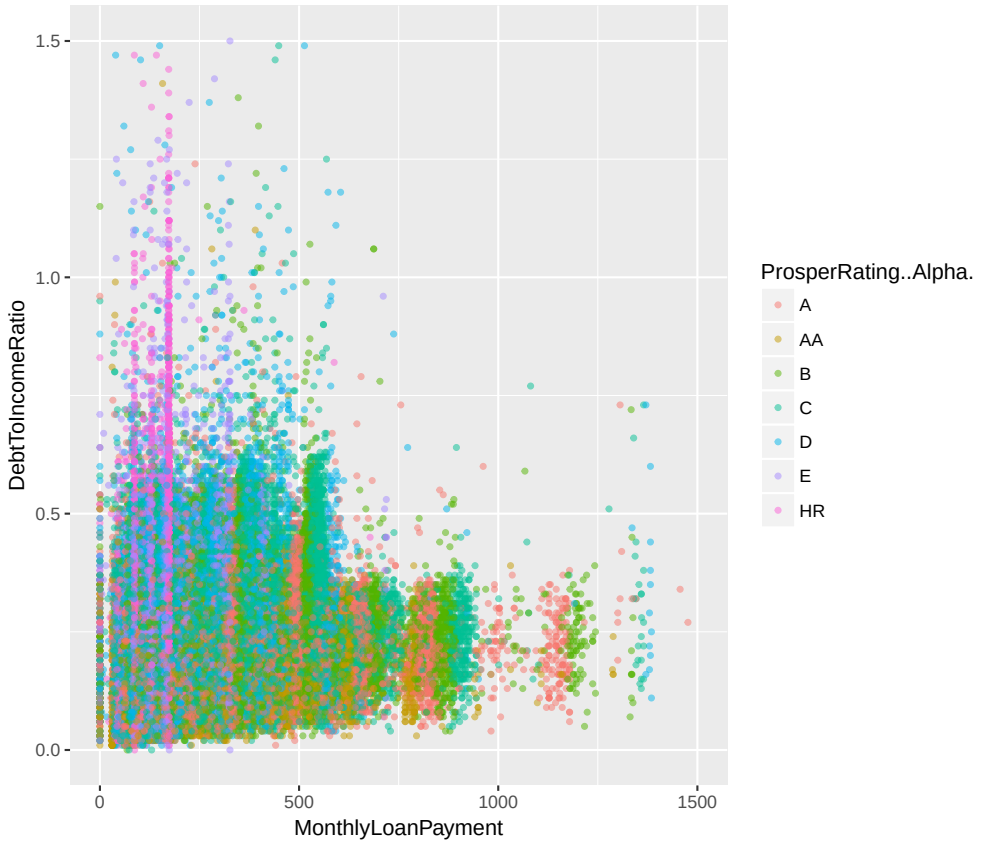
tions, we see clear linear relationships between AverageCredit scores and both StatedMonthlyIncome and LoanOriginalAmount.

The lack of variation (the grey coloring around a line) between datapoints in the Credit vs. Loans graph likely owes to the fact that these values, **LoanOriginalAmount** in particular, are essentially discrete. A look at the min and max values and their counts shows us that **LoanOriginalAmounts** are well-defined values ranging from $1000-$35000. Prosper thus seems to offer only a limited set of loan values.

Average Credit scores are similarly semi-discrete in nature, as our earlier histogram showed us. Further, homeowners seem to enjoy substantially higher minimum loans as well as an average higher income over their renter-peers.

Monthly Payments and Debt/Income Ratios by Prosper Rating

```
ggplot(data=subset(loans, !is.na(ProsperRating..Alpha.)),
       aes(x = MonthlyLoanPayment, y = DebtToIncomeRatio,
           color = ProsperRating..Alpha.)) +
  geom_point(alpha = 0.5, size = 1) +
  scale_x_continuous(limits=c(0, 1500)) +
  scale_y_continuous(limits = c(0.0, 1.5))
```



We have a great deal of overplotting here, somewhat mitigated by reducing the intensity of individual points, but there is a trade-off between this and the legibility of the legend.

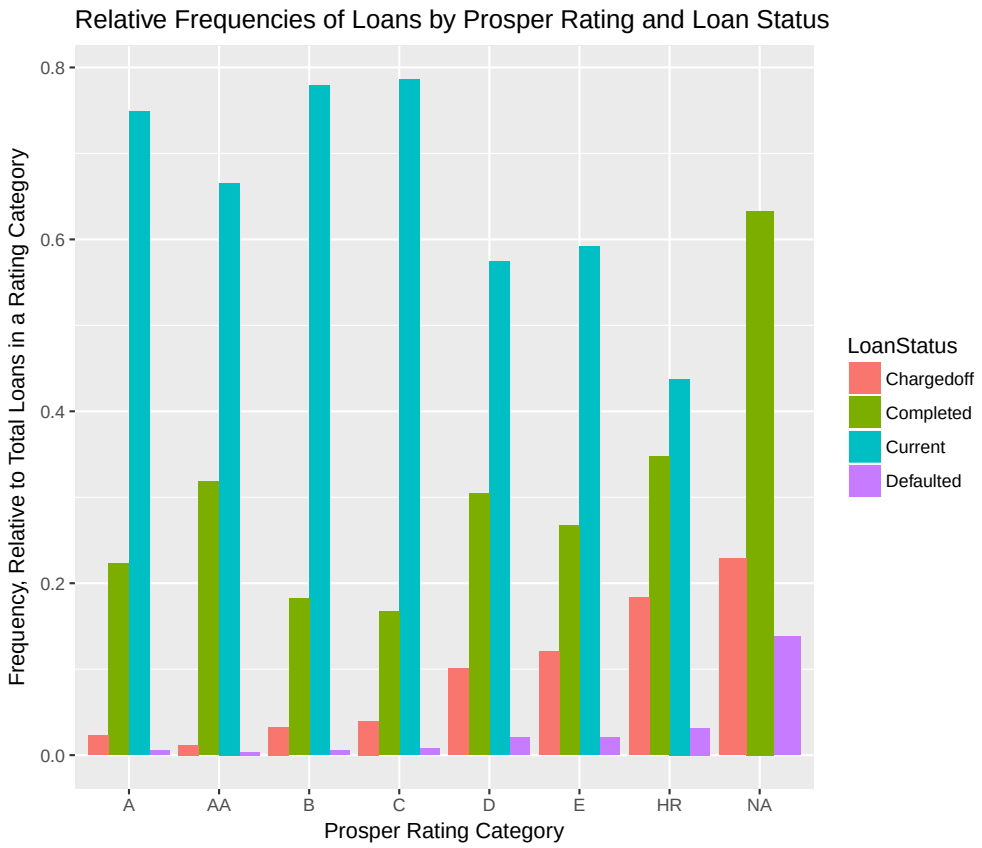
Yet, we can see that lower rated loans cluster at the low-payment side of the chart, with wide variability around the **DebtToIncomeRatio**. We also see that more highly rated loans span the range of payment sizes, but generally are in less financially precarious straits (lower ratios).

Final Plots and Summary

Plot One: ProsperRating and Loan Status Revisited

This time, we normalize the bars according to the total types of Loan Status for the different Prosper Ratings, using some **dplyr** trickery courtesy of StackOverflow.

```
grouped_status <- loans_with_status %>%
  group_by(ProsperRating..Alpha., LoanStatus) %>%
  summarize(Counts = n()) %>%
  group_by(ProsperRating..Alpha.) %>%
  mutate(Relative.Freq = Counts/sum(Counts)) %>%
  arrange(ProsperRating..Alpha.)

ggplot(data=grouped_status, aes(x=ProsperRating..Alpha.,
                                y=Relative.Freq,
                                fill=LoanStatus)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_color_brewer(type="seq") +
  ggtitle("Relative Frequencies of Loans by Prosper Rating and Loan Status") +
  xlab("Prosper Rating Category") +
  ylab("Frequency, Relative to Total Loans in a Rating Category")
```



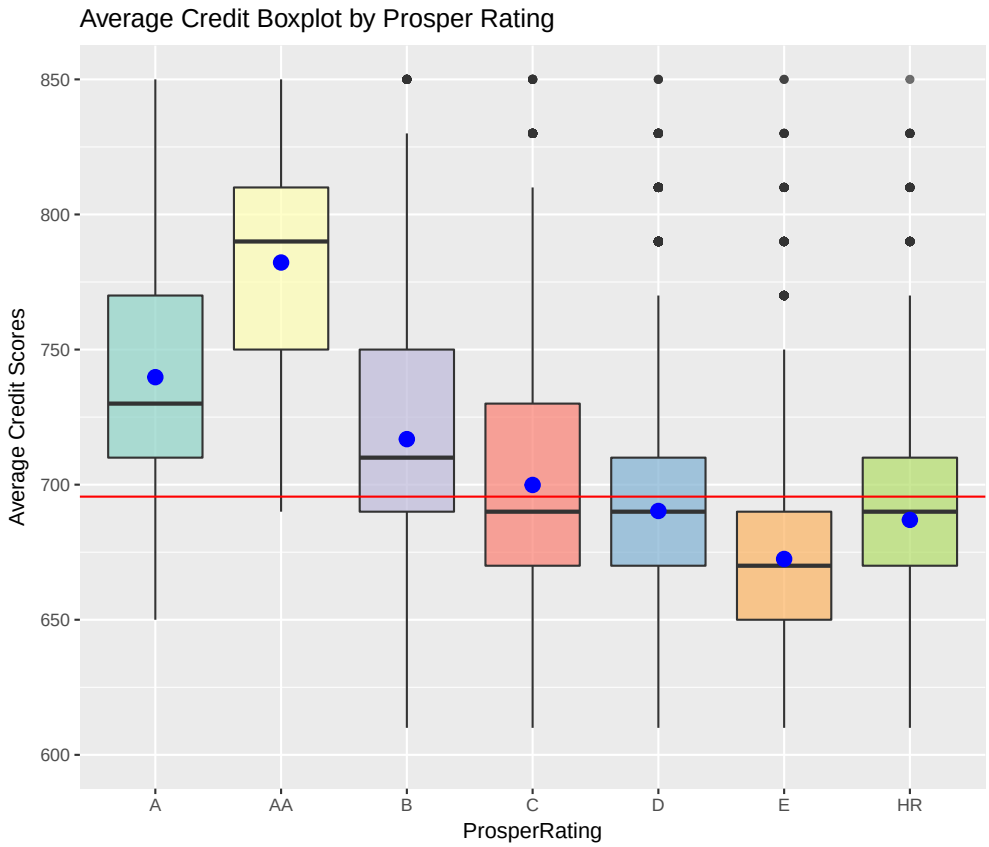

```
#dplyr help courtesy of this SO thread, which is itself courtesy of mentor Myles
#on the Udacity discussion forum:
#https://stackoverflow.com/questions/24576515/relative-frequencies-proportions-with-dplyr
```

Okay, this looks less bad than we previously thought. It remains strange that the NA category leads the way, in relative terms and absolute terms still, with loan repayment failures. One may assume that all loans that are no longer current have such information scrubbed at some regular juncture, but one could also wax paranoid and suspect Prosper of deliberately removing such essential information from bad loans as soon as possible.

That said, it is more than passing odd that AA and D ratings have roughly the same completion rates in this set, that HR loans are proportionally the most likely to be properly paid-off, and that the HR rating has the smallest proportion of current, trouble-free loans in the whole dataset.

Plot Two: AverageCredit Scores Across Prosper Ratings Boxplot

```
ggplot(subset(loans, !is.na(ProsperRating..Alpha.)),
              aes(x=ProsperRating..Alpha., y=AverageCredit,
                  fill=ProsperRating..Alpha.)) +
  ggtitle("Average Credit Boxplot by Prosper Rating") +
  geom_boxplot(alpha=0.7) +
  theme(legend.position = 'none') +
  geom_hline(yintercept = mean(loans$AverageCredit, na.rm = T),
             col="red") +
  stat_summary(fun.y=mean,
               geom="point",
               shape=20,
               size=5,
               color="blue",
               fill="red") +
  scale_fill_brewer(palette="Set3") +
  ylim(600, 850) +
  xlab("ProsperRating") +
  ylab("Average Credit Scores")
```




```
#adapted from http://www.r-graph-gallery.com/269-ggplot2-boxplot-with-average-value/
```

This far more colorful version of the boxplot has **AverageCreditScore** means for each Rating represented by the blue dots, as well as the overall mean represented once more by the horizontal line.

We note that the individual means do not diverge as much as we might expect, given the many outliers represented by the black dots beyond the whiskers. This is likely due to the large number of observations we have, and/or due to the outliers being relatively few in quantity.

Plot Three: Debt to Income Ratio and Monthly Loan Payment

```
ggplot(data=subset(loans, !is.na(ProsperRating..Alpha.)),
       aes(x = MonthlyLoanPayment, y = DebtToIncomeRatio,
           color = ProsperRating..Alpha.)) +
  ggtitle("Debt to Income Ratio Versus Stated Monthly Income") +
  geom_point(alpha = 0.8, size = 1) +
  stat_ellipse(size=2) +
  scale_x_continuous(limits=c(0, 1500)) +
  scale_y_continuous(limits = c(0.0, 1.5)) +
  xlab(label = "Monthly Loan Payments in USD") +
  ylab(label = "Ratio of Debt to Borrower Income") +
  scale_color_brewer(type='seq', palette = "PuBuGn",
                     guide=guide_legend(title='ProsperRating'))
```

This graphic does an interesting job of revealing the distribution of these variables, and we note that well-rated loans seem to range widely across payment amounts while largely avoiding high ratios of indebtedness. We see that the less well-rated loans are more likely to have higher Debt:Income ratios while also seeming to cluster at the lower end of the monthly payment range, probably because these people simply don't qualify for large loans.

And yet, this plot points to a recurring problem that has dogged this analysis–data that we might expect to be continuous are not quite so. For human-generated values like **StatedMonthlyIncome**, it is a well-known phenomenon that values provided by people cluster around multiples of ten–it is simply too tempting to round up or down to pad a value with zeroes when providing a number, perhaps for reasons related to aesthetics or cognitive-load.

In the case of fields with value ranges determined by Prosper, such as potential loan size, interest rates, and monthly payment sizes, the data seem limited to a predetermined set of discrete values–as our plot reveals with its well-defined vertical stacking of points at particlar payment amounts.

Reflection

This project has been a mammoth undertaking–having 81 variables to choose from meant from the start attempting to make guesses about what fields might prove to be of interest *after* the actual analysis.I found myself on four or five occasions having to go back and introduce new fields and remove others, as the exploratory data analysis process led me on to new questions and new exploratory paths.

I think, even at this early stage in the process of the data project cycle, I would have been greatly helped by having had greater background information into the data generation processes, the validity of some of the fields, and these would have in turn led me to formulate a more specific set of questions that would have kept this analysis from getting out of hand.

I will chalk that up to lessons learned: Never start a massive EDA project without having some well-defined questions prepared!