

Python Project - Prosper Loan Data Exploaration

by Monique Webley

Introduction

Prosper Loan is a peer-to-peer lending service based in San Francisco. The dataset contains loan information spanning several decades from 1972 to 2014. Data visualization tools, Seaborn and Matplotlib, will be used to illustrate essential insights regarding lending practices. This will involve quantitative and qualitative analysis of univariate and bivariate variables to see individual variable performance and correlated variable interactions, respectively.

Preliminary Wrangling

In [1]:

```
# import all packages

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns
```

In [2]:

```
#Load dataset and visually analyze
prosper = pd.read_csv('Prosper Loan.csv')

#Shows only first few rows
prosper.head()
```

Out[2]:

	ListingKey	ListingNumber	ListingCreationDate	CreditGrade	Term	LoanStatus	ClosedDate	BorrowerAPR	BorrowerRate	Len
0	1021339766868145413AB3B	193129	09:29.3	C	36	Completed	14/08/2009 00:00	0.16516	0.1580	
1	10273602499503308B223C1	1209647	28:07.9	NaN	36	Current	NaN	0.12016	0.0920	

	ListingKey	ListingNumber	ListingCreationDate	CreditGrade	Term	LoanStatus	ClosedDate	BorrowerAPR	BorrowerRate	Len
2	0EE9337825851032864889A	81716	00:47.1	HR	36	Completed	17/12/2009 00:00	0.28269	0.2750	
3	0EF5356002482715299901A	658116	02:35.0	NaN	36	Current	NaN	0.12528	0.0974	
4	0F023589499656230C5E3E2	909464	38:39.1	NaN	36	Current	NaN	0.24614	0.2085	

5 rows × 81 columns

```
In [3]: #programmatic analysis of dataset to see data types
prosper.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113937 entries, 0 to 113936
Data columns (total 81 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   ListingKey                           113937 non-null object
1   ListingNumber                         113937 non-null int64
2   ListingCreationDate                  113937 non-null object
3   CreditGrade                          28953 non-null  object
4   Term                                 113937 non-null int64
5   LoanStatus                           113937 non-null object
6   ClosedDate                           55089 non-null  object
7   BorrowerAPR                          113912 non-null float64
8   BorrowerRate                         113937 non-null float64
9   LenderYield                          113937 non-null float64
10  EstimatedEffectiveYield               84853 non-null  float64
11  EstimatedLoss                         84853 non-null  float64
12  EstimatedReturn                       84853 non-null  float64
13  ProsperRating (numeric)               84853 non-null  float64
14  ProsperRating (Alpha)                 84853 non-null  object
15  ProsperScore                          84853 non-null  float64
16  ListingCategory (numeric)             113937 non-null int64
17  BorrowerState                         108422 non-null object
18  Occupation                            110349 non-null object
19  EmploymentStatus                      111682 non-null object
20  EmploymentStatusDuration              106312 non-null float64
21  IsBorrowerHomeowner                  113937 non-null bool
22  CurrentlyInGroup                      113937 non-null bool
23  GroupKey                             13341 non-null  object
```

24	DateCreditPulled	113937	non-null	object
25	CreditScoreRangeLower	113346	non-null	float64
26	CreditScoreRangeUpper	113346	non-null	float64
27	FirstRecordedCreditLine	113240	non-null	object
28	CurrentCreditLines	106333	non-null	float64
29	OpenCreditLines	106333	non-null	float64
30	TotalCreditLinespast7years	113240	non-null	float64
31	OpenRevolvingAccounts	113937	non-null	int64
32	OpenRevolvingMonthlyPayment	113937	non-null	int64
33	InquiriesLast6Months	113240	non-null	float64
34	TotalInquiries	112778	non-null	float64
35	CurrentDelinquencies	113240	non-null	float64
36	AmountDelinquent	106315	non-null	float64
37	DelinquenciesLast7Years	112947	non-null	float64
38	PublicRecordsLast10Years	113240	non-null	float64
39	PublicRecordsLast12Months	106333	non-null	float64
40	RevolvingCreditBalance	106333	non-null	float64
41	BankcardUtilization	106333	non-null	float64
42	AvailableBankcardCredit	106393	non-null	float64
43	TotalTrades	106393	non-null	float64
44	TradesNeverDelinquent (percentage)	106393	non-null	float64
45	TradesOpenedLast6Months	106393	non-null	float64
46	DebtToIncomeRatio	105383	non-null	float64
47	IncomeRange	113937	non-null	object
48	IncomeVerifiable	113937	non-null	bool
49	StatedMonthlyIncome	113937	non-null	float64
50	LoanKey	113937	non-null	object
51	TotalProsperLoans	22085	non-null	float64
52	TotalProsperPaymentsBilled	22085	non-null	float64
53	OnTimeProsperPayments	22085	non-null	float64
54	ProsperPaymentsLessThanOneMonthLate	22085	non-null	float64
55	ProsperPaymentsOneMonthPlusLate	22085	non-null	float64
56	ProsperPrincipalBorrowed	22085	non-null	float64
57	ProsperPrincipalOutstanding	22085	non-null	float64
58	ScorexChangeAtTimeOfListing	18928	non-null	float64
59	LoanCurrentDaysDelinquent	113937	non-null	int64
60	LoanFirstDefaultedCycleNumber	16952	non-null	float64
61	LoanMonthsSinceOrigination	113937	non-null	int64
62	LoanNumber	113937	non-null	int64
63	LoanOriginalAmount	113937	non-null	int64
64	LoanOriginationDate	113937	non-null	object
65	LoanOriginationQuarter	113937	non-null	object
66	MemberKey	113937	non-null	object
67	MonthlyLoanPayment	113937	non-null	float64
68	LP_CustomerPayments	113937	non-null	float64

69	LP_CustomerPrincipalPayments	113937	non-null	float64
70	LP_InterestandFees	113937	non-null	float64
71	LP_ServiceFees	113937	non-null	float64
72	LP_CollectionFees	113937	non-null	float64
73	LP_GrossPrincipalLoss	113937	non-null	float64
74	LP_NetPrincipalLoss	113937	non-null	float64
75	LP_NonPrincipalRecoverypayments	113937	non-null	float64
76	PercentFunded	113937	non-null	float64
77	Recommendations	113937	non-null	int64
78	InvestmentFromFriendsCount	113937	non-null	int64
79	InvestmentFromFriendsAmount	113937	non-null	float64
80	Investors	113937	non-null	int64

dtypes: bool(3), float64(49), int64(12), object(17)
memory usage: 68.1+ MB

What is the structure of the dataset?

The dataset consists of booleans, integers, floats, and objects. Each value must be non-null.

What are the main features of interest in the dataset?

Main areas of interest are those related to the success of loans based on borrower's financial status and funding provided for the loans.

Loan Status: This feature indicates the current status of the loan, which includes categories like "completed", "charged off", "current", etc.

Percent Funded: This feature represents what percentage of the loan has been funded by investors.

Income Range: This feature categorizes the income range of the borrowers in predefined intervals.

Homeownership: This feature indicates whether the borrower owns a home or not. Categorized as "True" if borrower is a homeowner and "False" if they are not.

Estimated Return: This feature estimates the return on investment for the loan.

Estimated Effective Yield: This feature estimates the effective yield on investment for the loan, which may include factors like fees and defaults.

Employment Status: potentially relevant feature for assessing borrower's financial stability.

How will the features in the dataset help support investigation into features of interest?

With these features, the relationship between borrower's financial status (income range, homeownership) and loan success metrics (estimated return, estimated effective yield) will be analyzed. Additionally, it will be explored how funding levels (percent funded) and loan status affect the success of loans. Other variables such as employment status will be used to give further insight into the dataset.

Univariate Exploration

In [4]:

```
#Number of Loans by Prosper Loan and their status sorted by most to least using a horizontal bar chart.

#Counts the number of Loans in the LoanStatus column
loan_status_counts = prosper['LoanStatus'].value_counts()

#sorts the counted columns in descending order
loan_status_counts_sorted = loan_status_counts.sort_values(ascending=False)

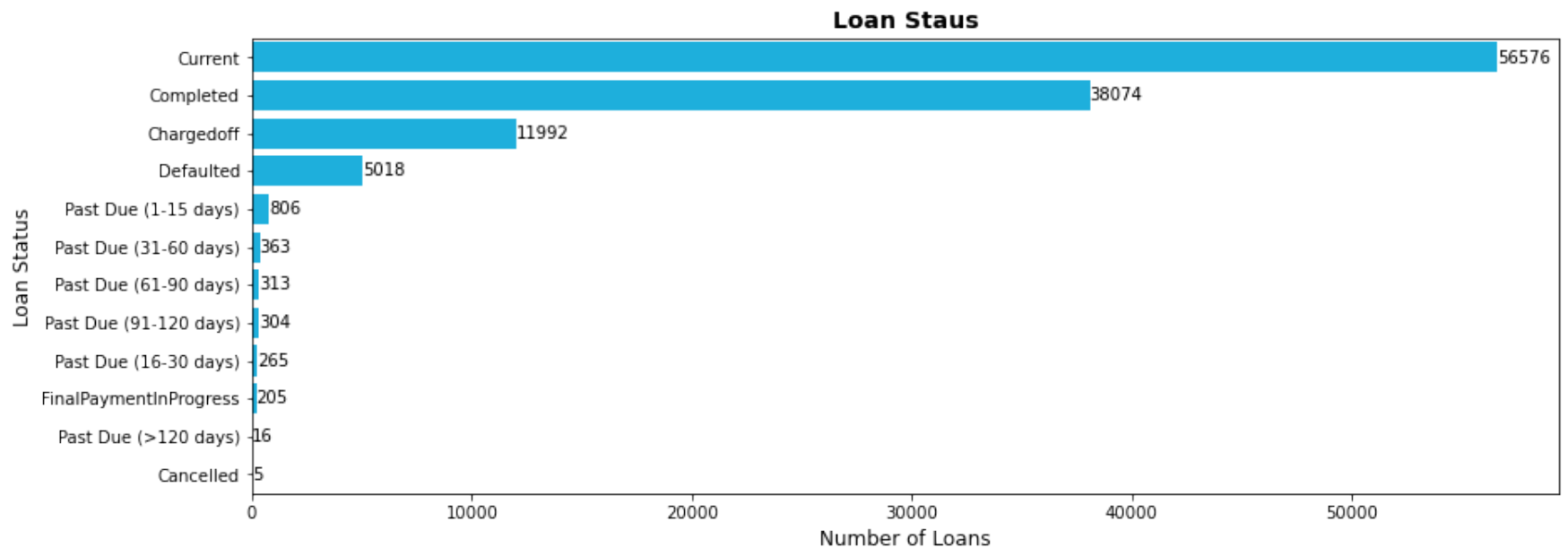
#Changes size of chart
plt.figure(figsize=(14,5))

#Changes colors of data bars
colors = '#00bfff'

#creates bar chart with loan status displaying on y axis
Loan_Status = sns.countplot(y = 'LoanStatus', data = prosper, order=loan_status_counts_sorted.index, color = colors)

#creates bar labels and specifies no decimal points
Loan_Status.bar_label(Loan_Status.containers[0],fmt = '%.f')

#creates title,bolds title, creates axis labels, sets font size, and font weight
plt.title('Loan Staus', fontsize = 14, fontweight = 'bold')
plt.xlabel('Number of Loans',fontsize = 12)
plt.ylabel('Loan Status', fontsize = 12);
```



In [5]:

```
#Percentage of borrowers who are homeowners vs not.

#Change size of pie chart
plt.figure(figsize=(14,5))

#Counts the number of homeowners compared to non-homeowners
home_status_counts = prosper['IsBorrowerHomeowner'].value_counts()

#Changes values from False to Not a homeowner and from True to Homeowner
home_status_counts.index = ['Not a Homeowner' if not x else 'Homeowner' for x in home_status_counts.index]

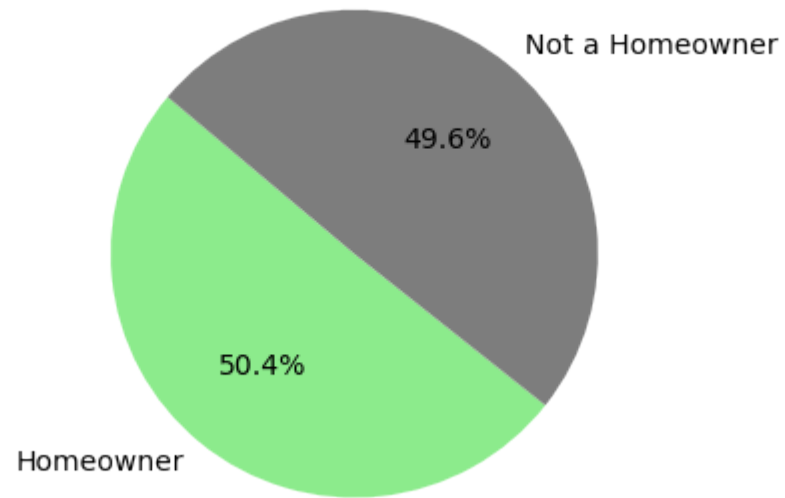
#Change colors of sections
colors = ['lightgreen', 'grey']

#Creates pie chart and sets decimal to one point
plt.pie(home_status_counts, labels=home_status_counts.index, autopct='%1.1f%%', startangle=140, colors=colors, textprops=

#adjusts font size, font weight, and title
plt.title('Homeownership Among Borrowers', fontsize=14, fontweight='bold')

#Makes the pie chart centered and have equal axes
plt.axis('equal');
```

Homeownership Among Borrowers



In [6]:

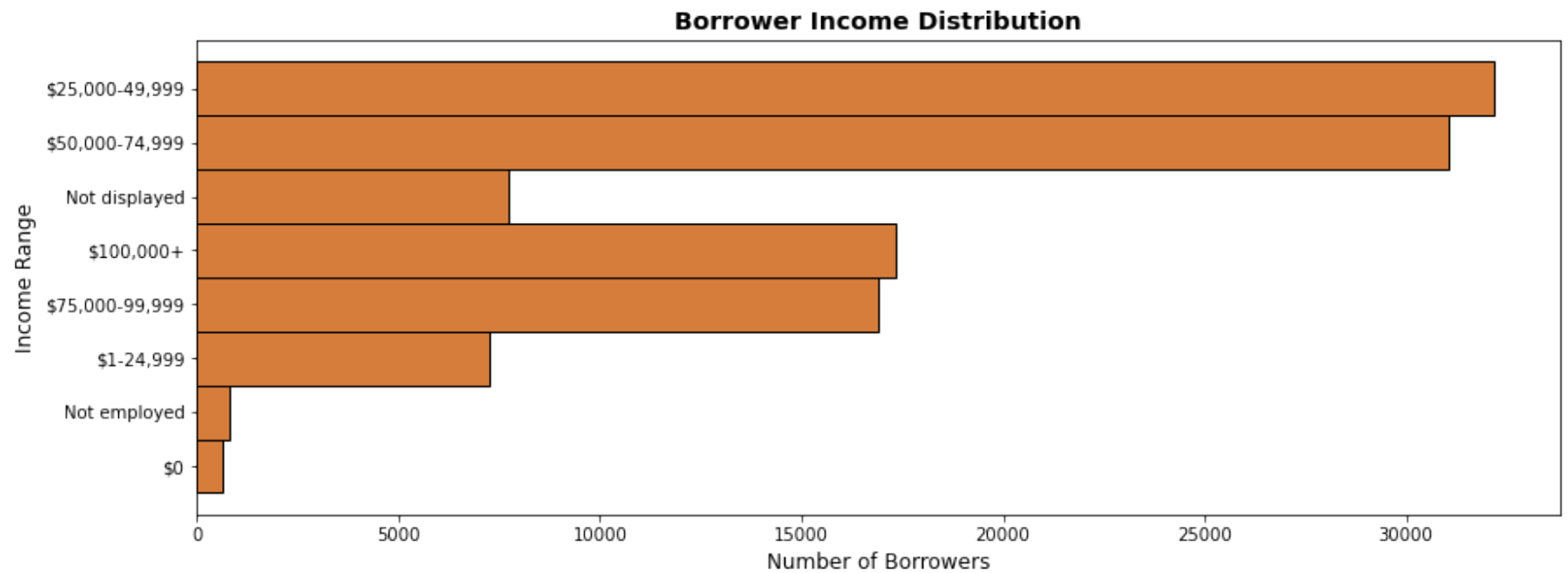
```
# Histogram displaying income range distribution among borrowers.

#Change size of histogram
plt.figure(figsize=(14,5))

#Changes colors of data bars
color = '#cd5700'

#creates histogram plotting income ranges in y axis
sns.histplot(y=prosper.IncomeRange, color=color)

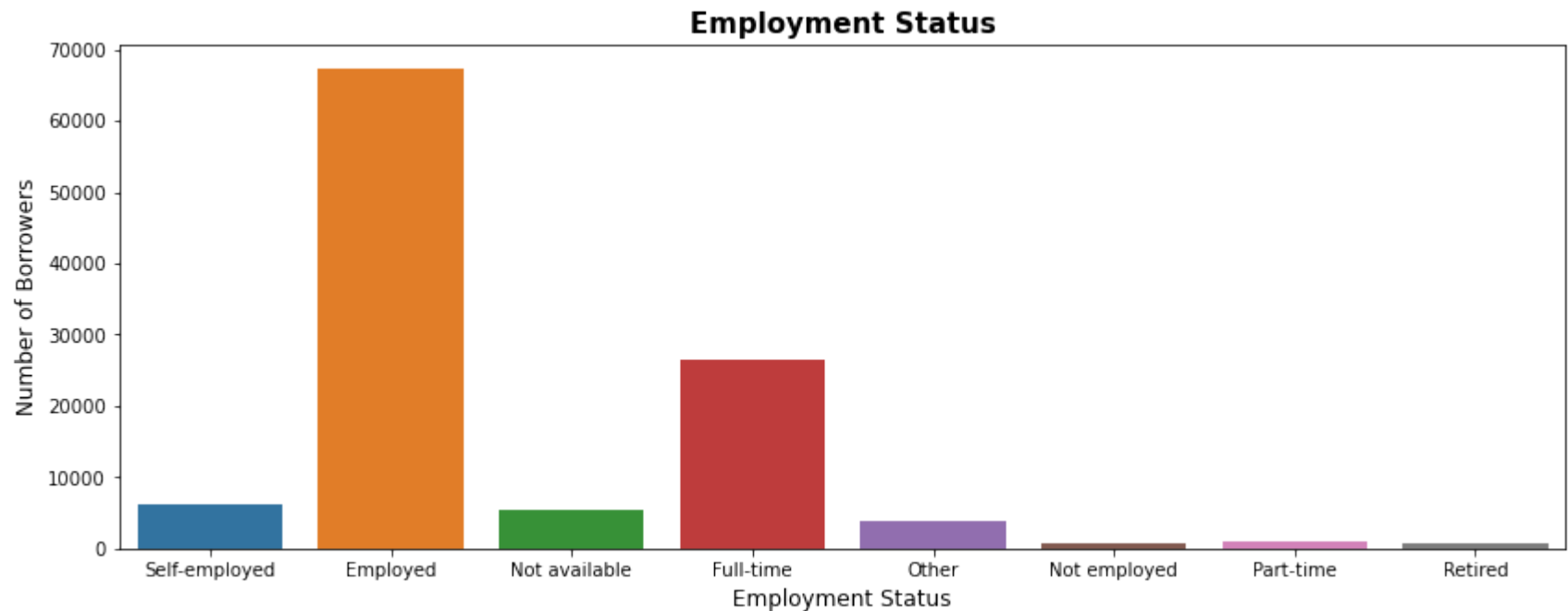
#Changes title, rename axes, font weight, and font sizes.
plt.title('Borrower Income Distribution', fontsize = 14, fontweight = 'bold')
plt.ylabel('Income Range', fontsize = 12)
plt.xlabel('Number of Borrowers', fontsize = 12);
```



```
In [7]: #Employment status for borrowers.
plt.figure(figsize=(14,5))

#Creates bar chart
sns.countplot(data = prosper, x='EmploymentStatus')

#Titles chart, sets font weight, font sizes and title axes
plt.title('Employment Status',fontsize = 15, fontweight='bold')
plt.xlabel('Employment Status',fontsize = 12)
plt.ylabel('Number of Borrowers',fontsize = 12);
```

What were the distribution(s) of variable(s) of interest. Were there any unusual points? Were any transformations performed?

The majority of borrowers, both current and past, have either fully repaid their loans or are currently in good standing. A smaller proportion of borrowers experienced difficulties in loan repayment, with some defaulting on their loans or falling behind in payments at some point.

Regarding the "Borrower Income Distribution" chart, the data suggests a skewed distribution towards specific income ranges. The most prominent clusters of borrowers are observed within the income brackets of 25,000 - 49,999 and 50,000 - 74,999. Additionally, there is a noticeable increase in borrower counts in higher income brackets, particularly those earning 100,000 or more, and a secondary peak in the 75,000 - 99,999 range. This skewed distribution indicates that the majority of borrowers fall within average income ranges, followed by those with above-average earnings.

Of the features investigated, were there any unusual distributions? Were any operations performed on the data to tidy, adjust, or change the form of the data? If so, why?

In the "Employment Status" chart, certain data bars exhibited overlapping information. Specifically, the bars representing Employed, Full-time, and Part-time contained data that could arguably be categorized as the same value, Employed.

However, I opted against merging these categories into a single data bar. Although they could theoretically represent individuals in traditional employment roles, the Employed category might also encompass non-traditional forms of employment such as contract or seasonal work.

Furthermore, I refined the data labels in the "Homeownership Among Borrowers" chart to provide more detailed descriptors rather than using generic terms like "True" and "False". Additionally, the observed discrepancy in homeownership rates was less pronounced than expected. This unexpected finding prompted a deeper investigation into the data in the proceeding section.

Bivariate Exploration

In [8]:

```
#Relationship between homeownership and standing of loan

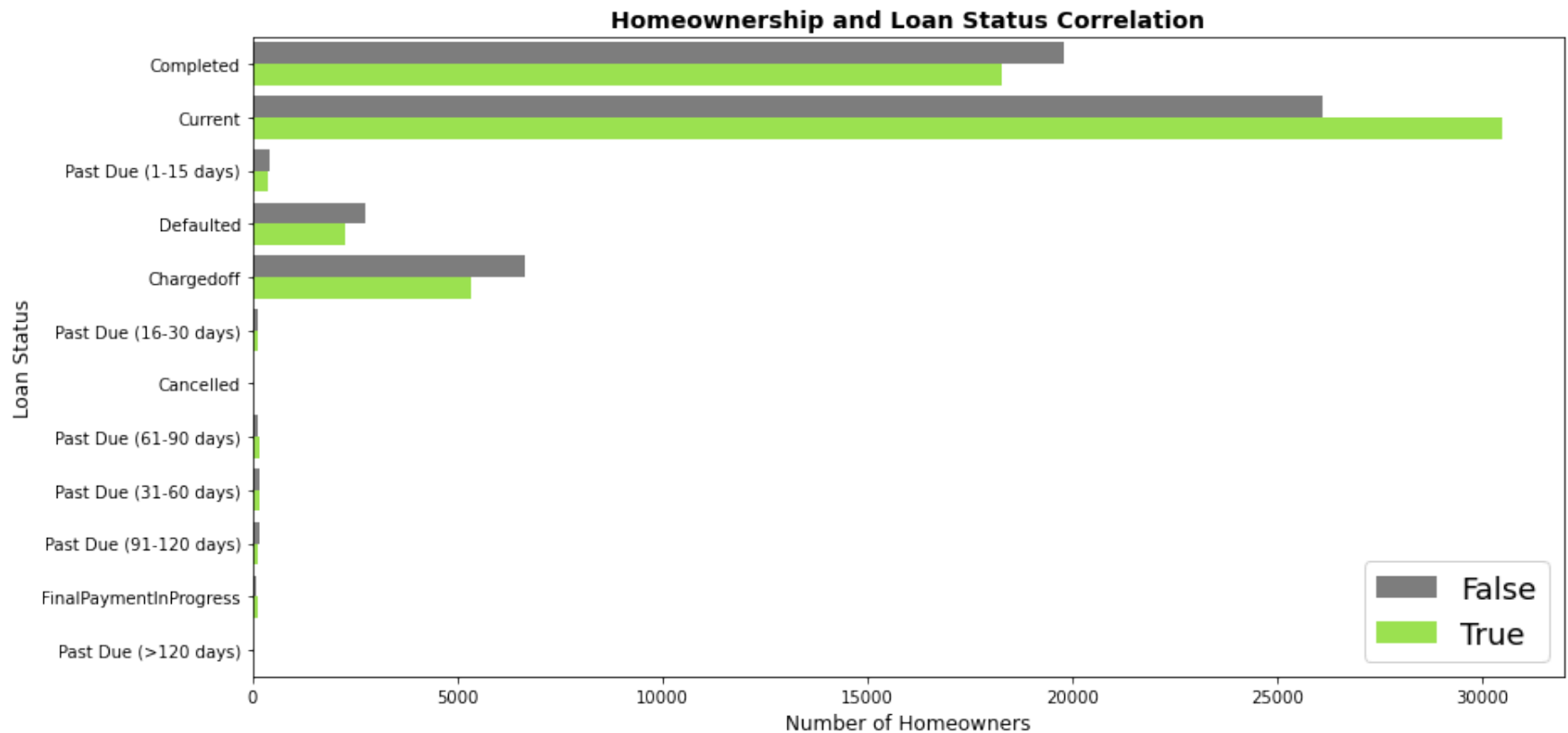
#Changes the size of the charts
plt.figure(figsize=(14,7))

#Changes the color of the chart
custom_palette = ["#808080", "#9efd38"]

#Creates horizontal bar chart displaying correlation between homeownership and loan status
sns.countplot(data = prosper, y='LoanStatus', hue = 'IsBorrowerHomeowner',palette=custom_palette)

#Changes title of chart, axes, font weight, and font sizes
plt.title('Homeownership and Loan Status Correlation', fontsize = 14, fontweight = 'bold')
plt.ylabel('Loan Status',fontsize = 12)
plt.xlabel('Number of Homeowners',fontsize = 12)

#Changes the font size of the legend and positioning on chart
plt.legend(loc='lower right', fontsize=18);
```



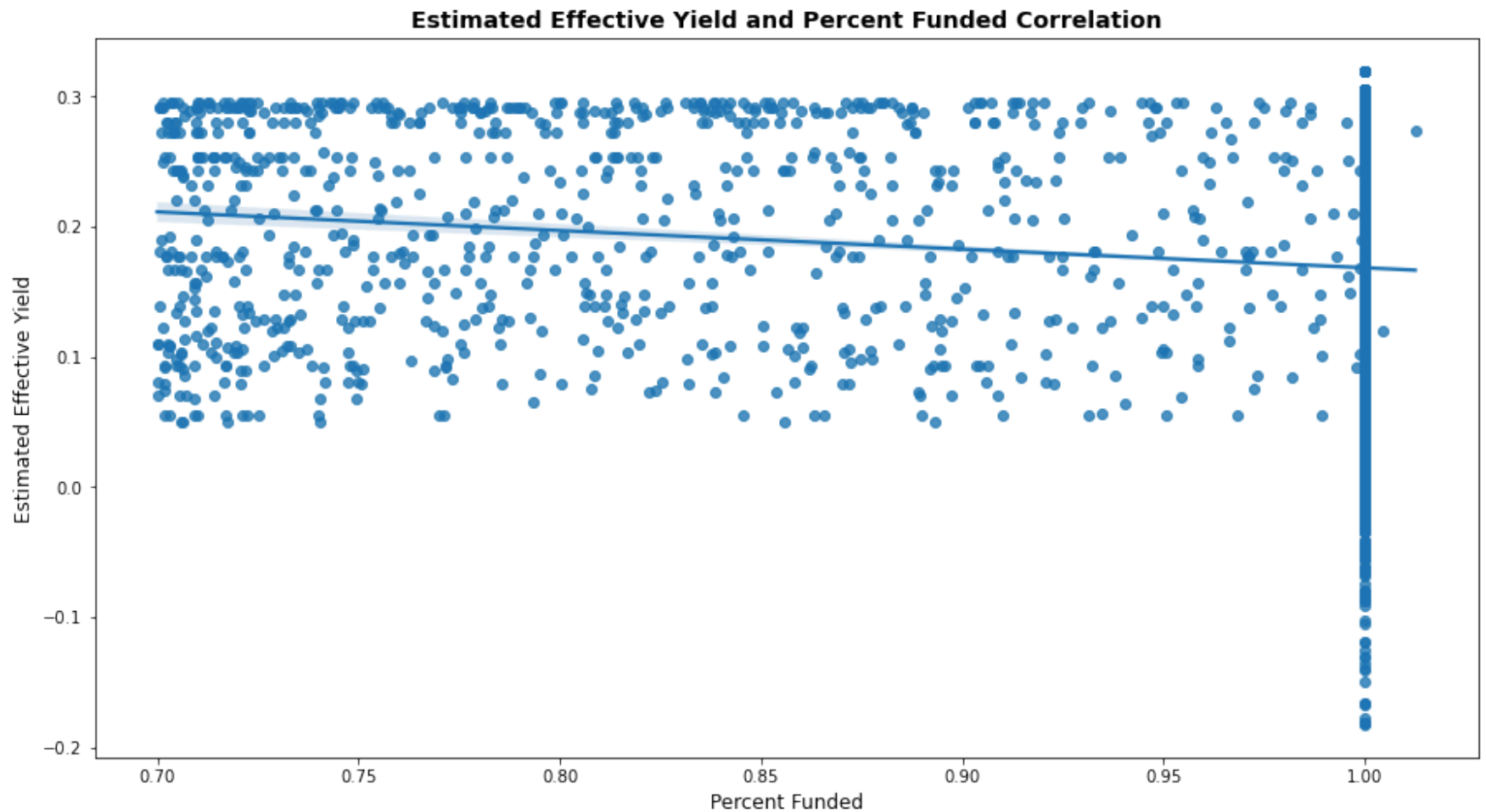
In [9]:

```
# Linear regression scatterplot showing correlation between EstimatedEffectiveYield and PercentFunded columns.

#Sizing of scatterplot
plt.figure(figsize = [15,8])

#Creation of scatter plot
sns.regplot(data = prosper, x='PercentFunded', y='EstimatedEffectiveYield')

#Changes title of chart, axes, font weight, and font sizes
plt.title('Estimated Effective Yield and Percent Funded Correlation', fontsize = 14, fontweight = 'bold')
plt.ylabel('Estimated Effective Yield',fontsize = 12)
plt.xlabel('Percent Funded',fontsize = 12);
```



```
In [10]: #Linear regression plot displaying estimated return and Loan amount correlation.

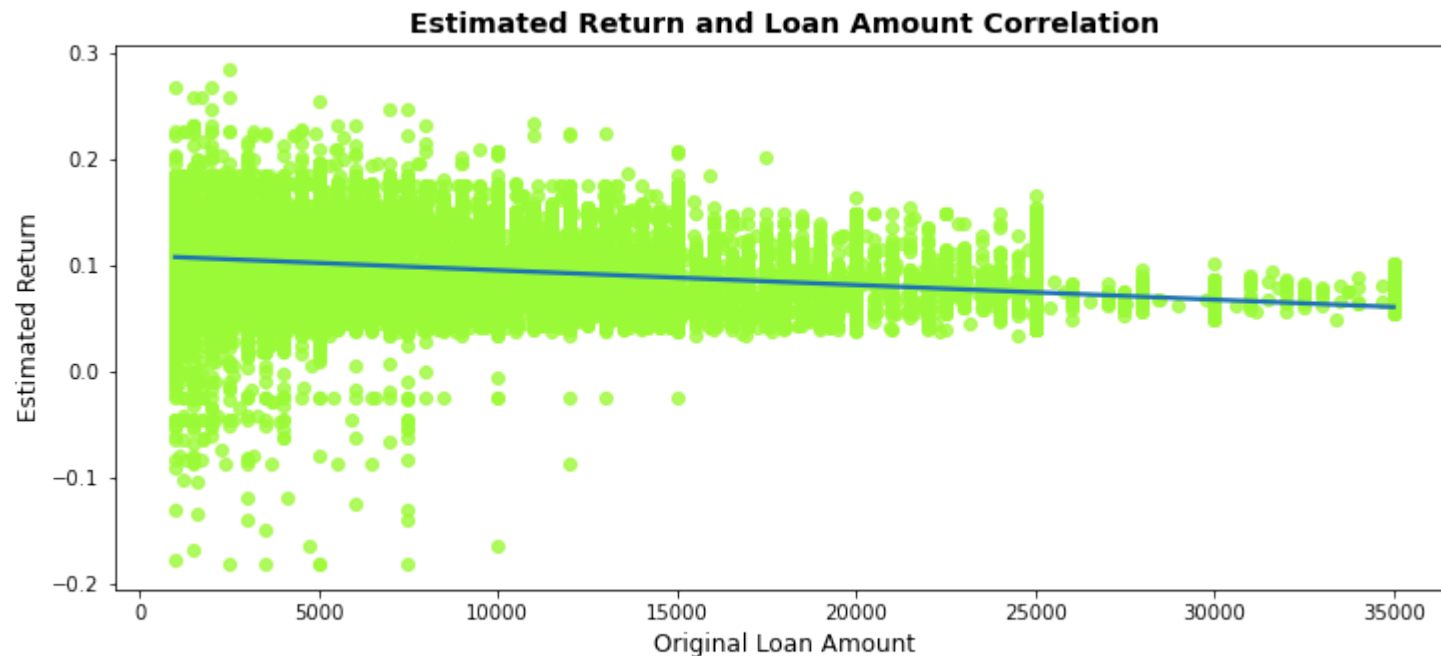
#Sizing of scatterplot
plt.figure(figsize = [12,5])

#changing color
color = ["#9efd38"]

#Creation of scatter plot
sns.regplot(data = prosper, y='EstimatedReturn', x='LoanOriginalAmount', scatter_kws={'color': color})

#Changes title of chart, axes, font weight, and font sizes
plt.title('Estimated Return and Loan Amount Correlation', fontsize = 14, fontweight = 'bold')
```

```
plt.xlabel('Original Loan Amount',fontsize = 12)
plt.ylabel('Estimated Return',fontsize = 12);
```



What were relationships observed in this part of the investigation. How did the feature(s) of interest vary with other features in the dataset?

Upon further examination of homeownership status, it became evident that a significant portion of completed loans were attributed to non-homeowners. This observation led to speculation that non-homeowners might possess greater financial resources available for loan repayment. Supporting this, among current loans, homeowners constituted a larger proportion of borrowers. Conversely, among defaulted or charged-off loans, non-homeowners represented a greater share than homeowners. This discrepancy could potentially be attributed to homeowners' presumed familiarity with managing larger debts, particularly if they have existing mortgage obligations.

Were there any observed interesting relationships between the other features (not the main feature(s) of interest)?

Unexpectedly, a negative correlation emerged between the estimated effective yield and percent funded. It was initially expected that as Prosper's backing percentage for a loan increased, the annual yield would also increase. Furthermore, there was observed a negative correlation between estimated returns and original loan amount. As the original loan amount

increased, the estimated return decreased. One possible explanation for this phenomenon might be that borrowers with higher credit scores are granted loans with lower interest rates, whereas the reverse tends to be true for borrowers with lower credit scores. This would also explain why estimated effective yield and percent funded are inversely related. It is conceivable that Prosper may have offered full funding for larger loans to borrowers with higher credit scores (lower-interest rates), thereby reducing their estimated effective yield.

Conclusions

Prosper functions as a peer-to-peer lending platform, offering a diverse range of loans with varying interest rates. This study delved into the correlation between borrowers' financial status and the outcome of their loans. The visual representations indicate that individuals with an average annual income falling within the ranges of 25,000 - 49,000 or 50,000 - 74,999, who are likely homeowners and employed, constitute a significant portion of borrowers.

Among current loans, the majority of borrowers are homeowners, whereas completed loans primarily involve non-homeowners. However, notably, non-homeowners constitute a larger proportion of borrowers who have defaulted or had their loans charged off. Furthermore, the data reveals an inverse correlation between estimated effective yield and the percentage of loan funded, as well as between estimated return and original loan amount. This trend is probably attributed to borrowers with higher credit scores securing lower interest rates for larger loan amounts. Further investigation is required to validate the relationship between credit scores and loan success.