Loan Data from Prosper

Preliminary Wrangling

> This data set contains 113,937 loans with 81 variables on each loan, including loan amount, borrower rate (or interest rate), current loan status, borrower income, and many others.

In [1]:
```
# import all packages and set plots to be embedded inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
```

In [2]:
```
# load in the dataset into a pandas dataframe, print statistics
Data = pd.read_csv('prosperLoanData.csv')
```

In [3]:
```
# high-level overview of data shape and composition
print(Data.shape)
```

```
(113937, 81)
```

In [34]:
```
features = ['LoanOriginalAmount', 'BorrowerAPR', 'BorrowerRate','StatedMonthlyInc
        'EmploymentStatus','ListingCreationDate','Occupation','AvailableBankcard(
selected_data = Data[features]
```

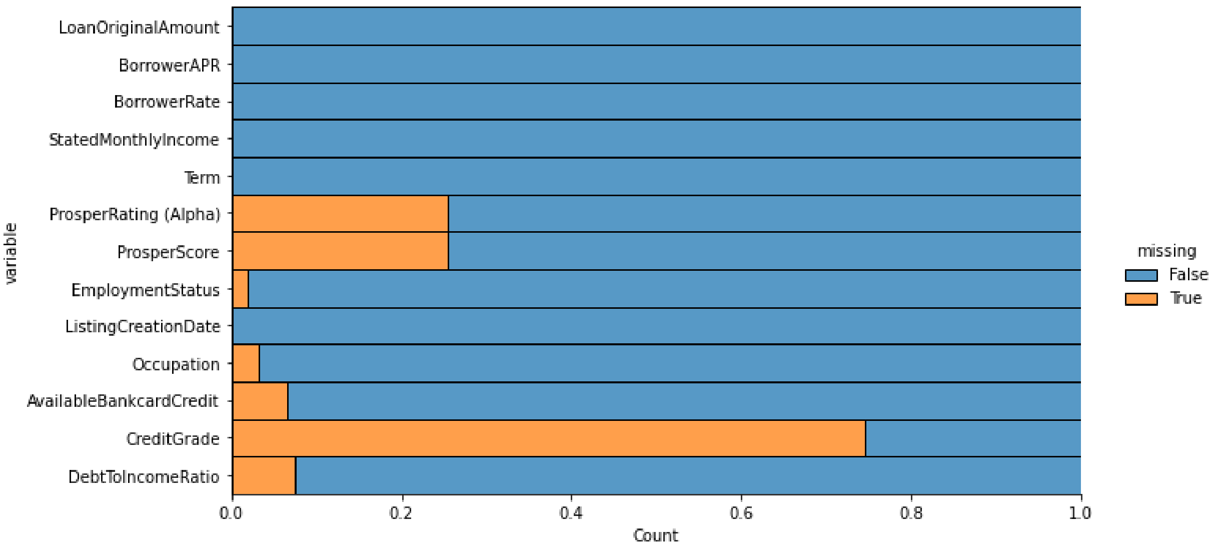
In [35]:
```
selected_data.head()
```

Out[35]:

	LoanOriginalAmount	BorrowerAPR	BorrowerRate	StatedMonthlyIncome	Term	ProsperRating (Alpha)	P
0	9425	0.16516	0.1580	3083.333333	36	NaN	
1	10000	0.12016	0.0920	6125.000000	36	A	
2	3001	0.28269	0.2750	2083.333333	36	NaN	
3	10000	0.12528	0.0974	2875.000000	36	A	
4	15000	0.24614	0.2085	9583.333333	36	D	

In [36]:
```
plt.figure(figsize=(20,20));
sns.displot(
    data=selected_data.isna().melt(value_name="missing"),
    y="variable",
    hue="missing",
    multiple="fill",
    aspect=2
);
```

```
<Figure size 1440x1440 with 0 Axes>
```



we have a greatly missed data in **Credit Grade** so maybe result of analysis of this feature isn't accurate, **ProsperRating (Alpha)**, **ProsperScore** **EmploymentStatus**, **DebtToIncomeRatio**, **Occupation**, and **AvailableBankcardCredit** all of these columns contain nulls value

In [37]:

```
# descriptive statistics for numeric variables
selected_data.describe()
```

Out[37]:

	LoanOriginalAmount	BorrowerAPR	BorrowerRate	StatedMonthlyIncome	Term	Pr
count	113937.00000	113912.000000	113937.000000	1.139370e+05	113937.000000	84
mean	8337.01385	0.218828	0.192764	5.608026e+03	40.830248	
std	6245.80058	0.080364	0.074818	7.478497e+03	10.436212	
min	1000.00000	0.006530	0.000000	0.000000e+00	12.000000	
25%	4000.00000	0.156290	0.134000	3.200333e+03	36.000000	
50%	6500.00000	0.209760	0.184000	4.666667e+03	36.000000	
75%	12000.00000	0.283810	0.250000	6.825000e+03	36.000000	
max	35000.00000	0.512290	0.497500	1.750003e+06	60.000000	

What is the structure of your dataset?

> The dataset has 113,937 loans with 81 variables on each loan. I will be interested in a subset of those variables including loan amount, borrower rate, Borrower APR, current loan status, borrower income, and many others. Variables are loan information and borrower information.

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

> I'm most interested in figuring out what features are best for predicting borrower's Annual Percentage Rate (Borrower APR) for the loan and affecting the loan status.

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

> I guess that the total loan amount will have a great effect on the APR of the loan which the most larger the total loan amount, the most the lower the APR. I also guess that the borrowers stated monthly income, loan term, Prosper rating, employment status will also have effects on the APR.

> I also guess I need to invest and find in ListingCreationDate and some borrower information such as Occupation, AvailableBankcardCredit, CreditGrade, StatedMonthlyIncome, and DebtToIncomeRatio which will also have effects on the borrower's Annual Percentage Rate and loan status.

Univariate Exploration

I'll start by looking at the distribution of the main variable of interest: **employment status**.

> **Employment Status** : The employment status of the borrower at the time they posted the listing.

In [196]:

```
# Convert ProsperRating to an ordered type
rate_order = ['HR','E','D','C','B','A','AA']
ordered_var = pd.api.types.CategoricalDtype(ordered = True, categories = rate_orc
selected_data.loc[:,'ProsperRating (Alpha)'] = selected_data.loc[:,'ProsperRating
```

In [195]:

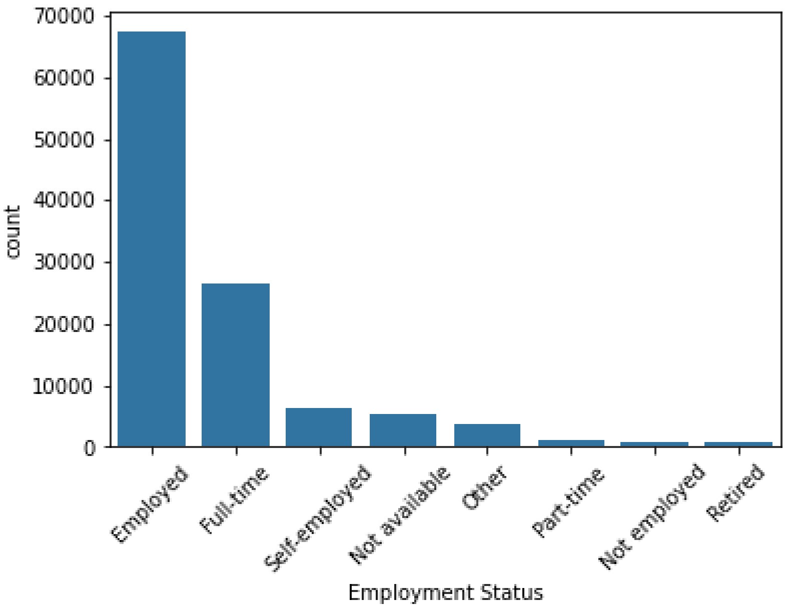
```

# Convert  Employment status to an ordered type
emp_order = ['Employed','Self-employed','Full-time','Part-time','Retired','Other'
ordered_var = pd.api.types.CategoricalDtype(ordered = True, categories = emp_orde
selected_data.loc[:,'EmploymentStatus'] = selected_data.loc[:,'EmploymentStatus']
```

In [40]:

```
# The `color_palette()` returns the the current / default palette as a list of RG
# Each tuple consists of three digits specifying the red, green, and blue channel
# Choose the first tuple of RGB colors
base_color = sns.color_palette()[0]
```

In [41]:

```
# Plot the Employment Status
type_order = selected_data['EmploymentStatus'].value_counts().index
sns.countplot(data=selected_data, x='EmploymentStatus', color=base_color, order=t
plt.xticks(rotation = 45);
plt.xlabel('Employment Status');
```



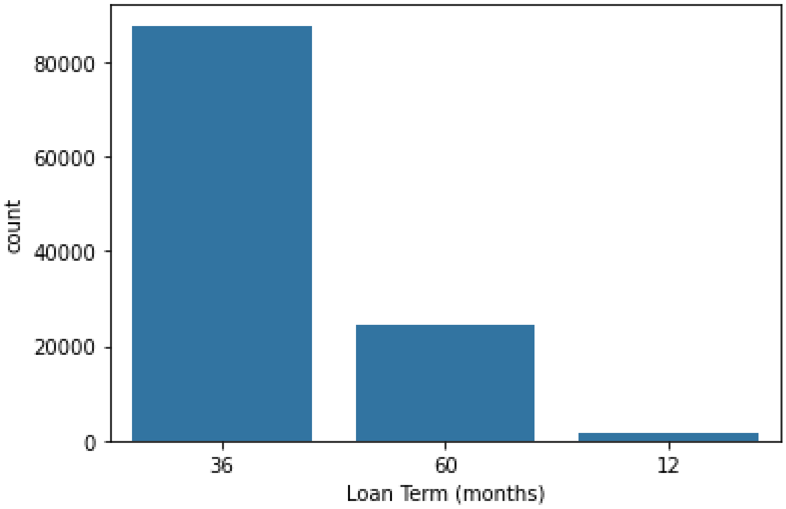
> Comman borrowers indicate **“Employed”** and **“Full-Time”** as employment status.

Next up, the first predictor variable of interest: Term.

> **Term**: The length of the loan expressed in months.

In [42]:

```
# Plot the Term
type_order = selected_data['Term'].value_counts().index
sns.countplot(data=selected_data, x='Term', color=base_color, order=type_order);
plt.xlabel('Loan Term (months)');
```

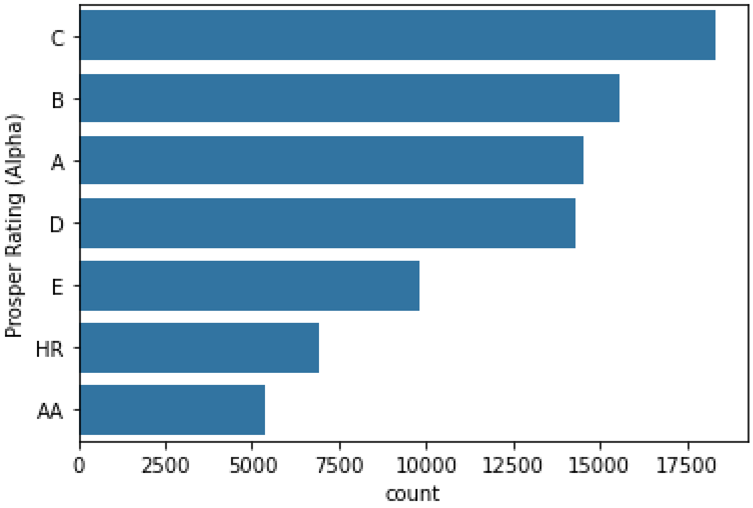


> The length of most Common of the loans are 36 months Then it was followed by a long distance of 60 months which there were significantly fewer loans for 60-month terms and almost none for 12-month terms. Actually, this affects directly in Borrower Rate

Next up, the first predictor variable of interest: Prosper Rating.

> **ProsperRating (Alpha)**: The Prosper Rating assigned at the time the listing was created between AA - HR. Applicable for loans originated after July 2009.

In [43]:
```
# Plot the Prosper Rating
type_order = selected_data['ProsperRating (Alpha)'].value_counts().index
sns.countplot(data=selected_data, y='ProsperRating (Alpha)', color=base_color, or
plt.ylabel('Prosper Rating (Alpha)');
```



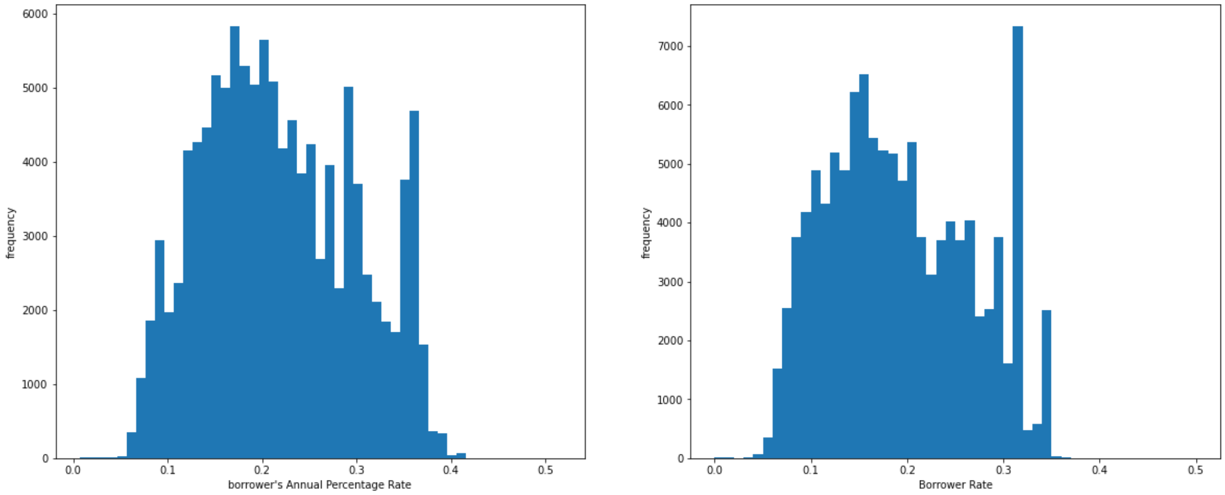
> The ratings of most common of the borrowers are among C to D

Next up, the first predictor variable of interest: BorrowerAPR and BorrowerRate

> **BorrowerAPR** : The Borrower's Annual Percentage Rate (APR) for the loan.
> **BorrowerRate** : The Borrower's interest rate for this loan.

In [44]:

```
plt.subplots(figsize = [20,8])
plt.subplot(1, 2, 1)
apr_bins = np.arange(selected_data.BorrowerAPR.min(), selected_data.BorrowerAPR.
plt.hist(data=selected_data, x='BorrowerAPR', bins=apr_bins);
plt.xlabel('borrower\'s Annual Percentage Rate');

plt.ylabel('frequency');
# Plot the distribution of BorrowerRate
plt.subplot(1, 2, 2)
rate_bins = np.arange(selected_data['BorrowerRate'].min(), selected_data.Borrower
plt.hist(data=selected_data, x='BorrowerRate', bins=rate_bins)
plt.xlabel('Borrower Rate');
plt.ylabel('frequency');
```



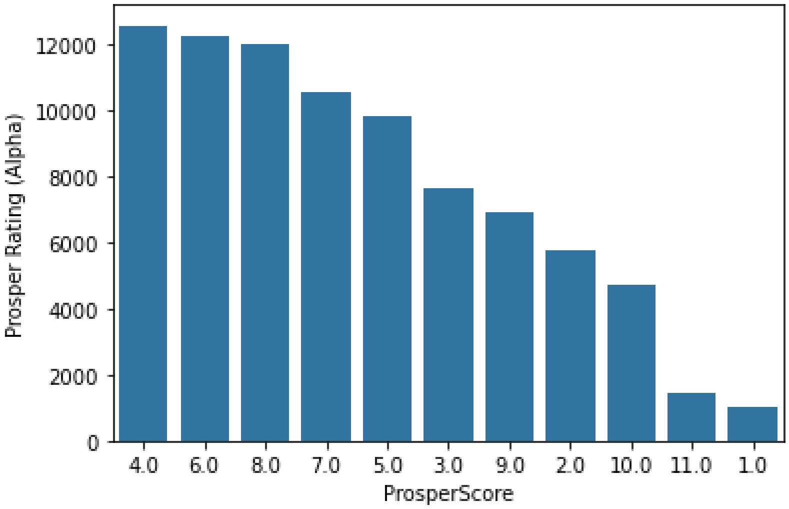
Both distributions are multimodal which the distributions are a normal distribution than skewed right. Borrower Rate and borrower's Annual Percentage Rate are similar but we considered that borrower's Annual Percentage Rate contain some fees (such as discount points, most closing costs, mortgage insurance, and loan origination fees) of course them affect on the total cost of loan.

So that borrower's Annual Percentage Rate greater than Borrower Rate

Next up, the first predictor variable of interest: ProsperScore

ProsperScore: A custom risk score built using historical Prosper data. The score ranges from 1-10, with 10 being the best, or lowest risk score. Applicable for loans originated after July 2009.

In [45]:
```
type_order = selected_data['ProsperScore'].value_counts().index
sns.countplot(data=selected_data, x='ProsperScore', color=base_color, order=type_
plt.ylabel('Prosper Rating (Alpha)');
```



It's observed that degree of risk between **0.4 : 0.8** It's not good but also not bad which the most common rate is **0.4** this so interesting.

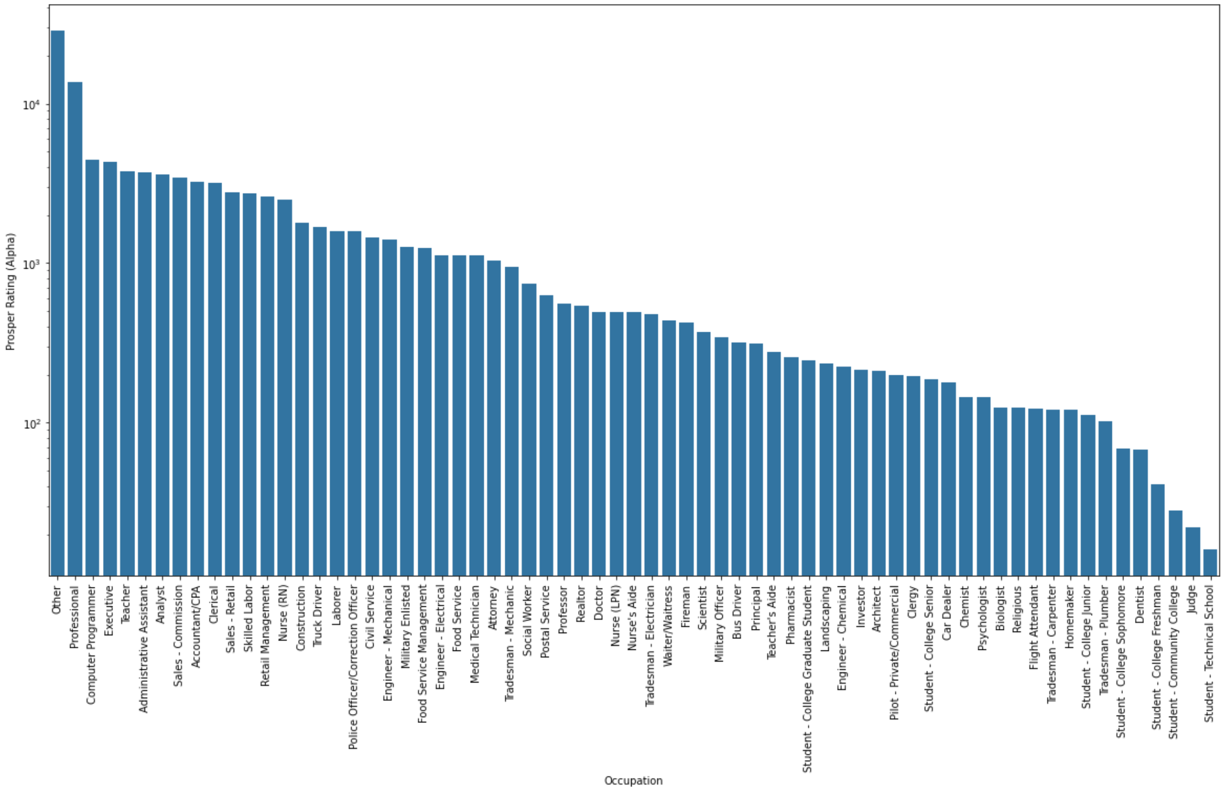
It's observed that we find rate value equal 11.0 it's out of range. This

Next up, the first predictor variable of interest: Occupation

Occupation: The Occupation selected by the Borrower at the time they created the listing.

In [46]:

```
plt.subplots(figsize = [20,10])
type_order = selected_data['Occupation'].value_counts().index
g = sns.countplot(data=selected_data, x='Occupation', color=base_color, order=typ
plt.ylabel('Prosper Rating (Alpha)');
plt.xticks(rotation=90)
g.set(yscale="log");
```



> Because I noted huge variance between highest and lowest counts. I used **Log Scale** in my chart
>
> I found The most common Occupation is **‘Others’** then **‘Professional’**, and not give us meaningful information so we will skip them and move to next ones we found that **‘computer programmer’**,**’Excutives’**, **‘Teacher’**, **'Administrative Assistant'**, and **'Analyst'**.

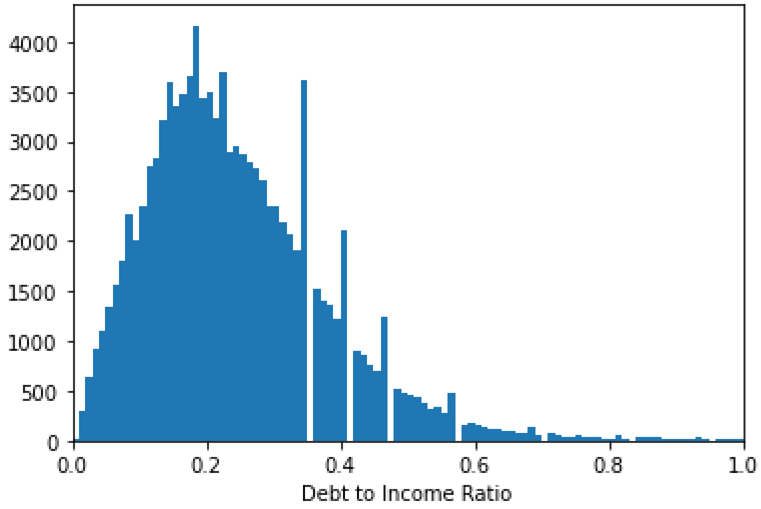
Occupation	frequency
Computer Programmer	4478

Occupation	frequency
Executive	4311
Teacher	3759
Administrative Assistant	3688
Analyst	3602
Sales - Commission	3446
Accountant/CPA	3233
Clerical	3164

Next up, the first predictor variable of interest: **DebtToIncomeRatio**

DebtToIncomeRatio: The debt to income ratio of the borrower at the time the credit profile was pulled. This value is Null if the debt to income ratio is not available. This value is capped at 10.01 (any debt to income ratio larger than 1000% will be returned as 1001%).

In [47]:
```
in_bins = np.arange(selected_data.DebtToIncomeRatio.min(), selected_data.DebtToIr
plt.hist(data=selected_data, x='DebtToIncomeRatio', bins=in_bins);
plt.xlim(0,1);
plt.xlabel('Debt to Income Ratio');
```



The distribution of the DebtToIncomeRatio was highly skewed by the presence of those with very high incomes to their debt. This isn't unexpected in a real-world scenario and no changes to the data were performed to account for this. It will be interesting to see how this affects the interest rates of the loans.

Discuss the distribution(s) of your variable(s) of interest. Were there any unusual points? Did you need to perform any transformations?

> The distribution of Borrower Rate appears as normally distributed with a slight left skew. A small peak centered at 15%, a large peak centered at 30%, and a median found between them. There is also a small peak centered 30%. Additionally, and it's observed a few loans have a Borrower Rate greater than 35%.
>
> There isn't need to implement any transformations.

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?

> The distribution of the **DebtToIncomeRatio** was highly skewed by the presence of those with very high incomes to their debt. This isn't unexpected in a real-world scenario and does not need transformations implemented on data. just I limit x-axis to focus on distribution

> For all of the bar charts that I used I reorded rank of them descending. So it's easy to detect the most common.

> In **occpution** chart, I noted huge variance between highest and lowest counts. I used Log Scale in my chart

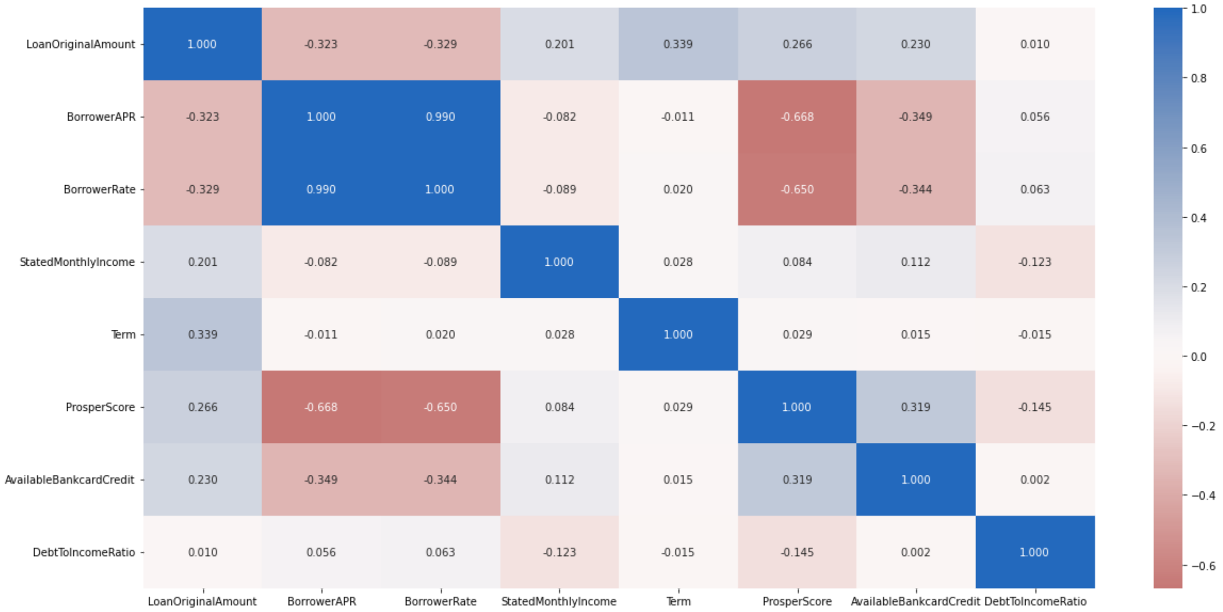
Bivariate Exploration

To start off with, I want to look at the pairwise correlations present between features in the data.

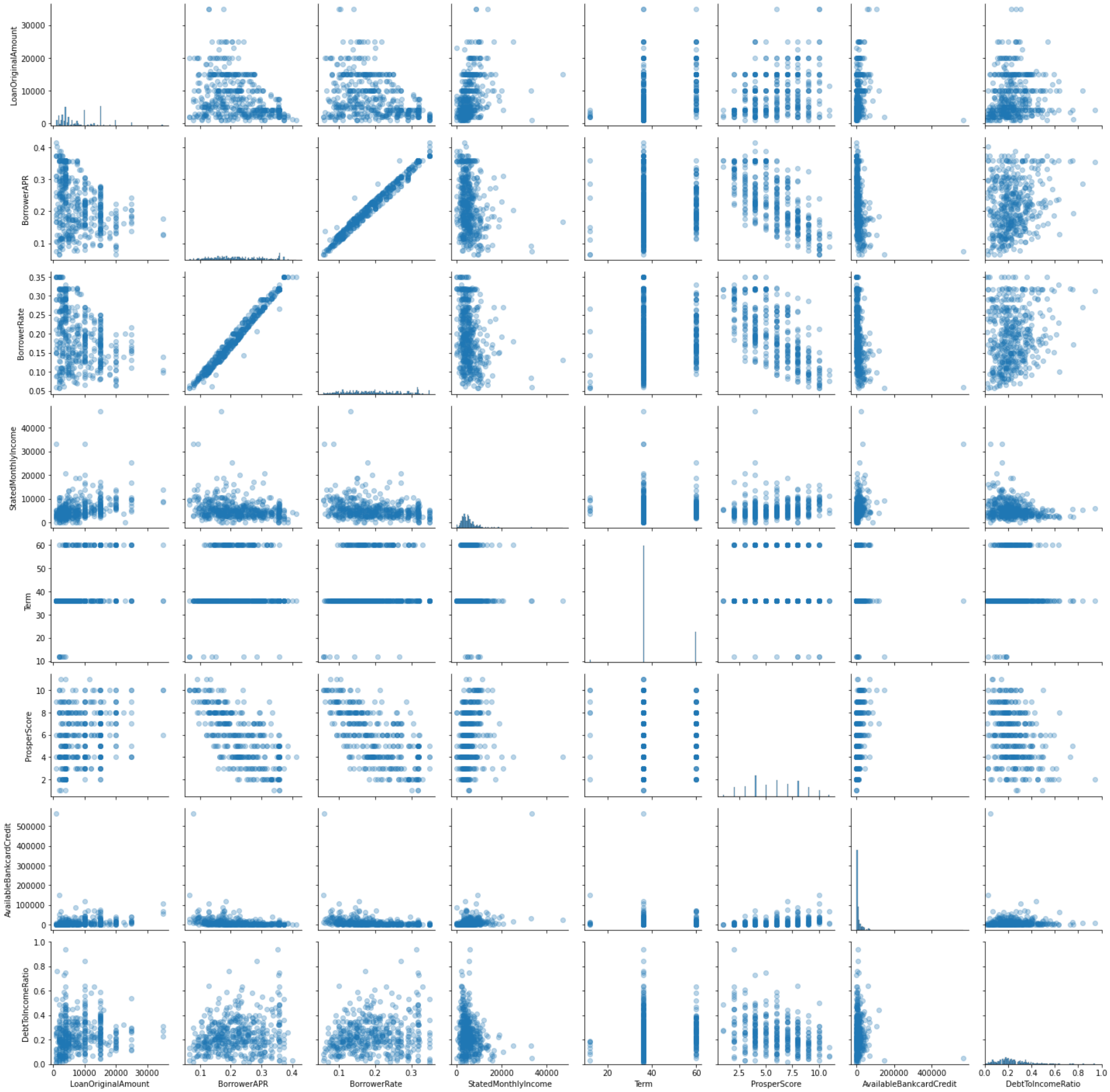
```
In [48]: numeric_vars = selected_data.select_dtypes(include='number').columns
         categoric_vars = ['Term', 'EmploymentStatus', 'ProsperRating (Alpha)']
```

```
Out[48]: Index(['LoanOriginalAmount', 'BorrowerAPR', 'BorrowerRate',
                'StatedMonthlyIncome', 'Term', 'ProsperScore',
                'AvailableBankcardCredit', 'DebtToIncomeRatio'],
               dtype='object')
```

In [49]:
```
# correlation plot
plt.figure(figsize = [20, 10])
sns.heatmap(selected_data[numeric_vars].corr(), annot = True, fmt = '.3f',
           cmap = 'vlag_r', center = 0)
plt.show()
```



In [83]:
```
# plot matrix: sample 500 diamonds so that plots are clearer and
# they render faster
samples = np.random.choice(selected_data.shape[0], 500, replace = False)
loans_samp = selected_data.loc[samples,:]

g = sns.PairGrid(data = loans_samp, vars = numeric_vars)
g = g.map_diag(sns.histplot, bins=100);
g.map_offdiag(plt.scatter, alpha=0.3);
g.axes[7,7].set_xlim(0,1);
g.axes[7,7].set_ylim(0,1);
```



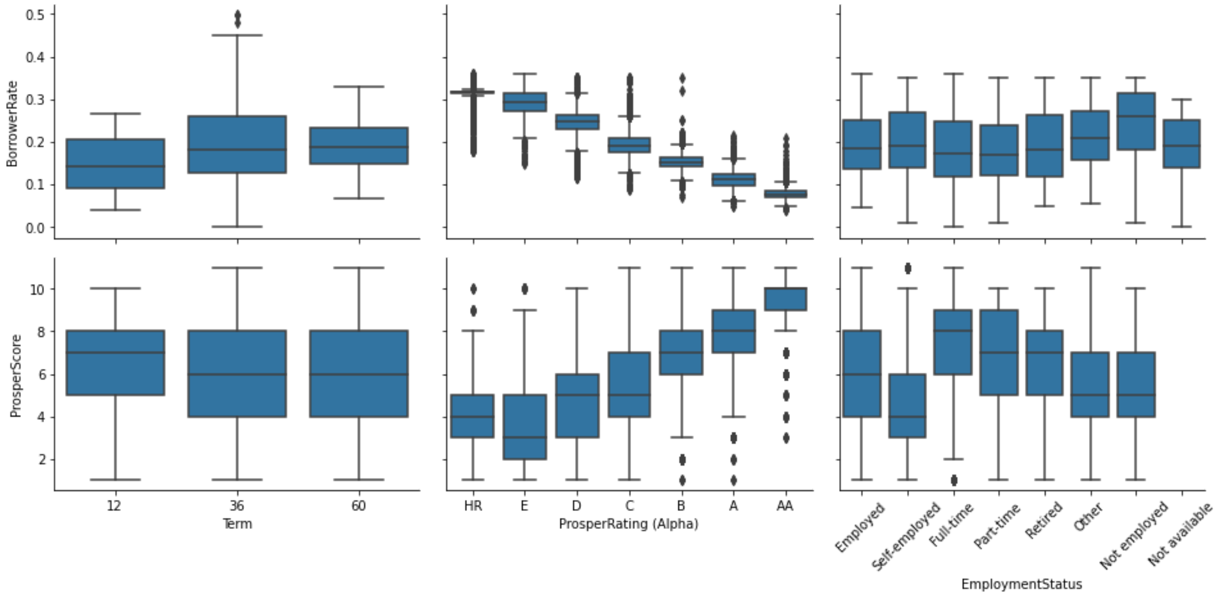
> BorrowerRate and BorrowerAPR have a great correlation between them, which makes sense since those values for a loan are similar to each other with a few diffrance, and we observe also that ProsperScore and ProsperRating have a great correlation to loan interest
> The total loan amount has a high positive correlation with the stated monthly income, it makes sense which rich people have a higher probability to borrow loans with more money rather than poor people who has small monthly income.

Let's move on to looking at how price and carat weight correlate with the categorical variables.

In [164]:

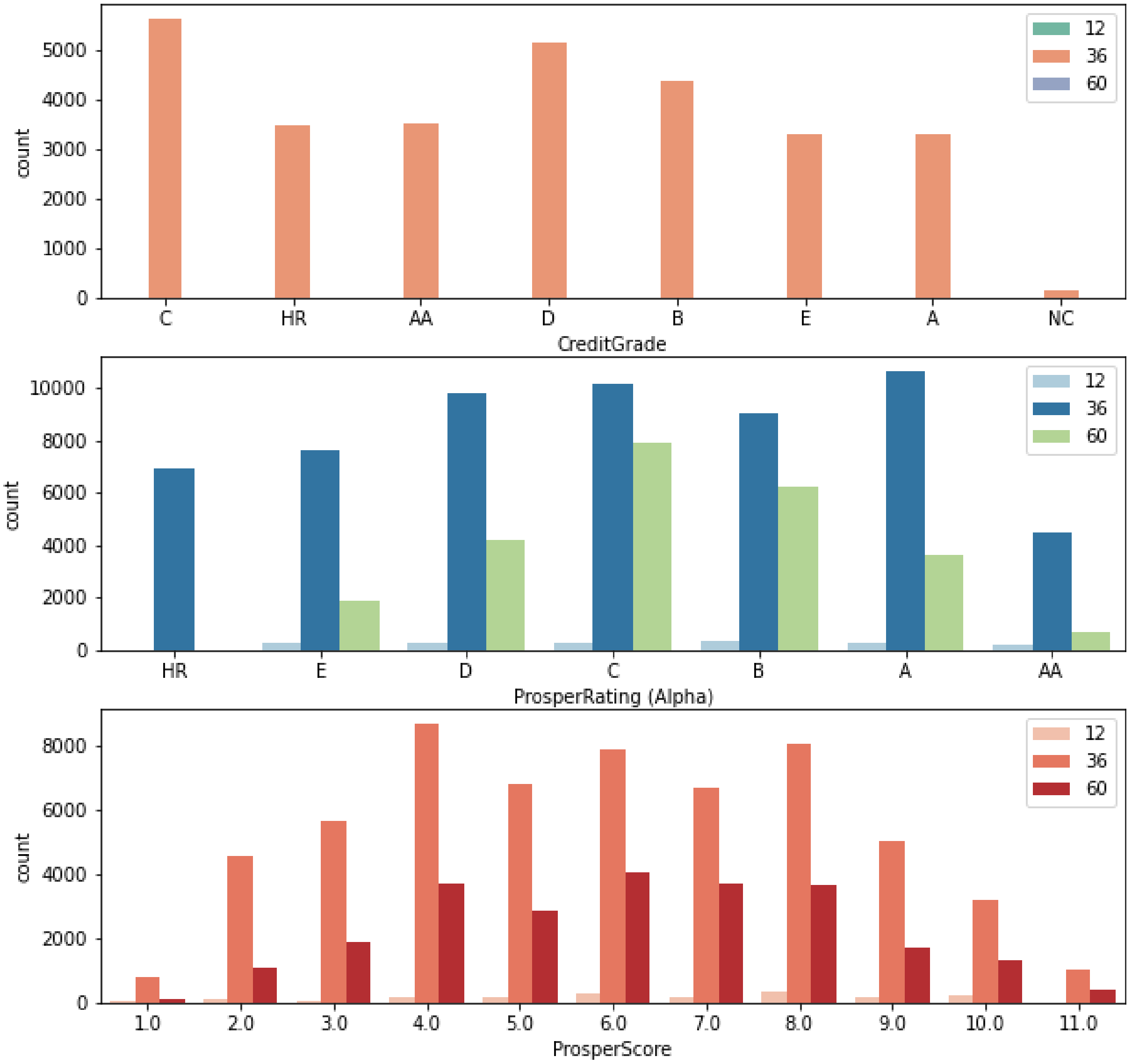
```
# plot matrix of numeric features against categorical features.
import warnings
warnings.simplefilter(action="ignore", category=FutureWarning)
def boxgrid(x, y, **kwargs):
    """ Quick hack for creating box plots with seaborn's PairGrid. """
    default_color = sns.color_palette()[0]
    sns.boxplot(x, y, color = default_color)

plt.figure(figsize = [10, 10])
g = sns.PairGrid(selected_data,y_vars = ['BorrowerRate','ProsperScore',],
                                x_vars = ['Term', 'ProsperRating (Alpha)', 'Emplo
g.map(boxgrid);
plt.xticks(rotation=45);
```

```
<Figure size 720x720 with 0 Axes>
```



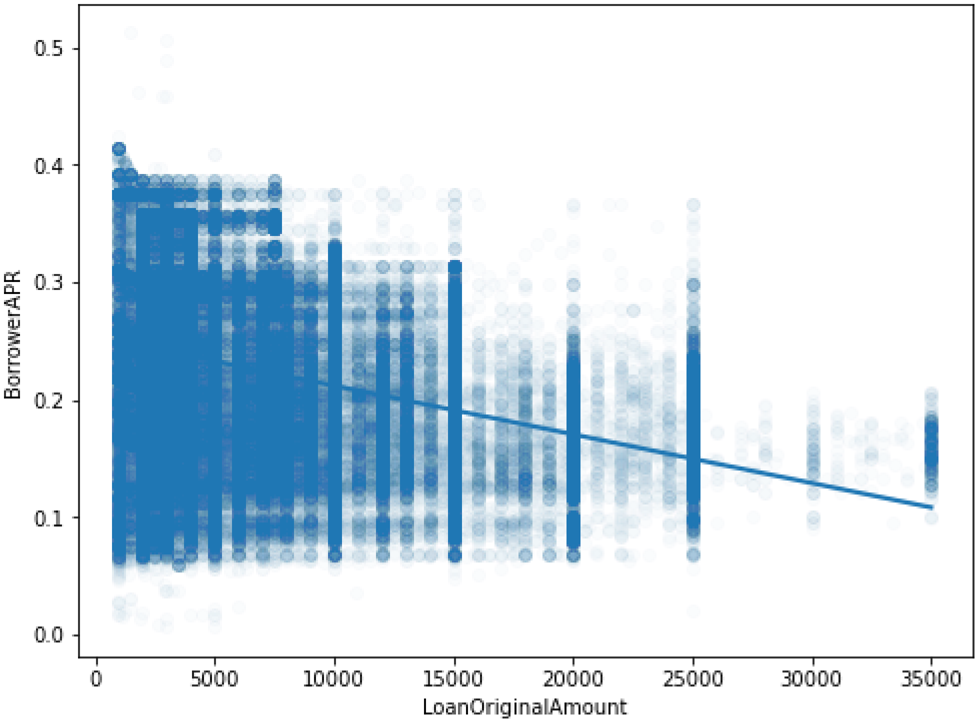
Looking at the box plots for the distribution of the different credit risks annd the BorrowerRate is that there is a negative correlation between having a higher (better) score and a lower interest rate. This is once again most pronouced in the results for the ProsperRating (Alpha) variable where the decline is steeper. With the ProsperScore there is still a decilne, but the distribution of the BorrowerRate is more distributed and the IQR is genrally larger for each rating.

```
In [115]: # Look at relationship between Term and the CreditGrade/ProsperRating(s) and Pros
          plt.figure(figsize=[10,])

          plt.subplot(3,1,1)
          sns.countplot(data=selected_data, x='CreditGrade', hue='Term', palette='Set2');
          plt.legend(loc=1);

          plt.subplot(3,1,2)
          sns.countplot(data=selected_data, x='ProsperRating (Alpha)', hue='Term', palette=
          plt.legend(loc=1);

          plt.subplot(3,1,3)
          sns.countplot(data=selected_data, x='ProsperScore', hue='Term', palette='Reds');
          plt.legend(loc=1);
```



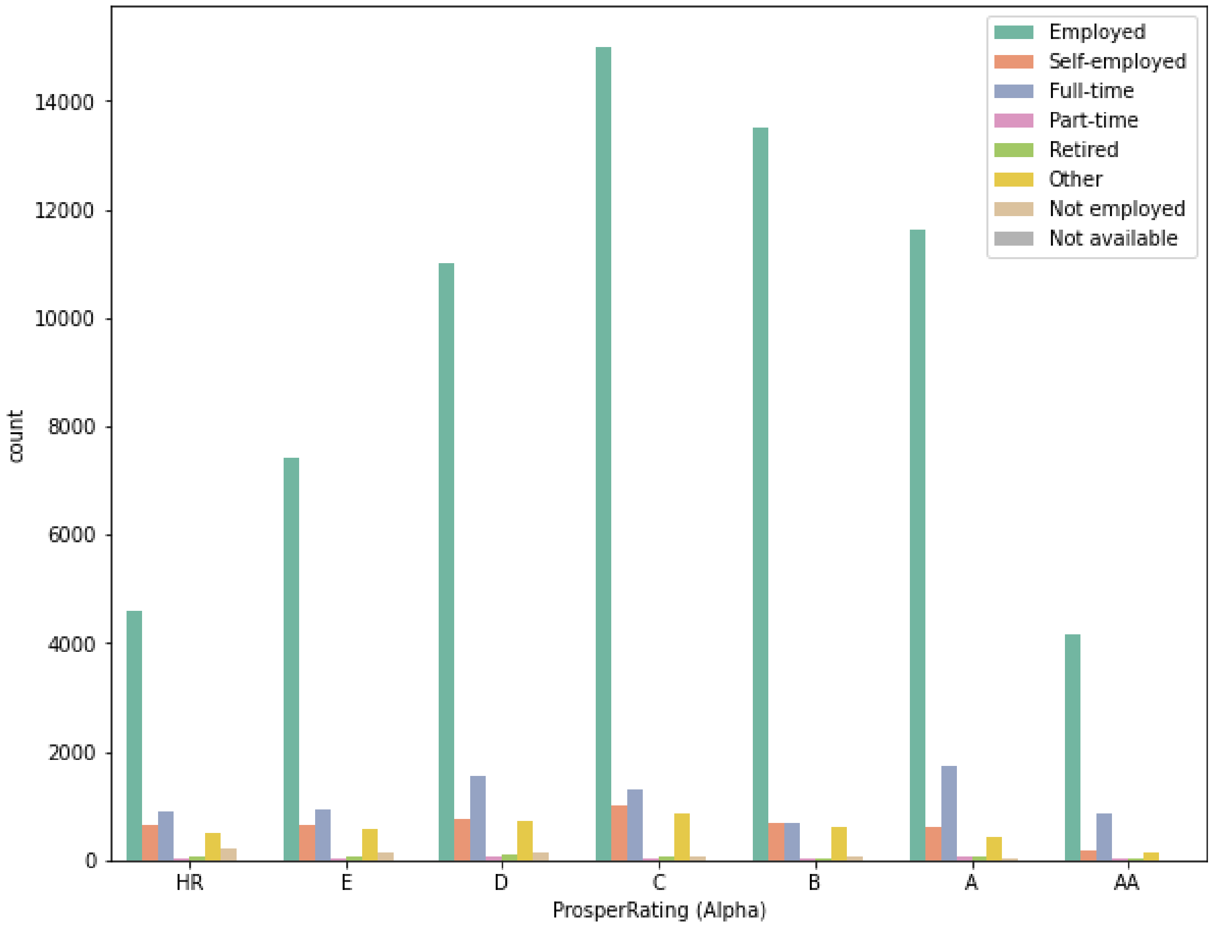
Interestingly, it appears that there are some positive relationships between the categorical variables and the two numeric variables of interest. The loans with 60-month term loans were quite popular with those with a ProsperRating of A and C. As seen before 36-month term loans are the most popular across all credit risk groups.

```
In [148]: plt.figure(figsize = [8, 6])
          sns.regplot(data = selected_data, x = 'LoanOriginalAmount', y = 'BorrowerAPR', sc
```



This graph observed that at the different sizes of the total loan amount, the borrower's APR has a large distribution, but the range of APR decreases with the increase of loan amount. So the borrower's APR is negatively correlated with the total loan amount.

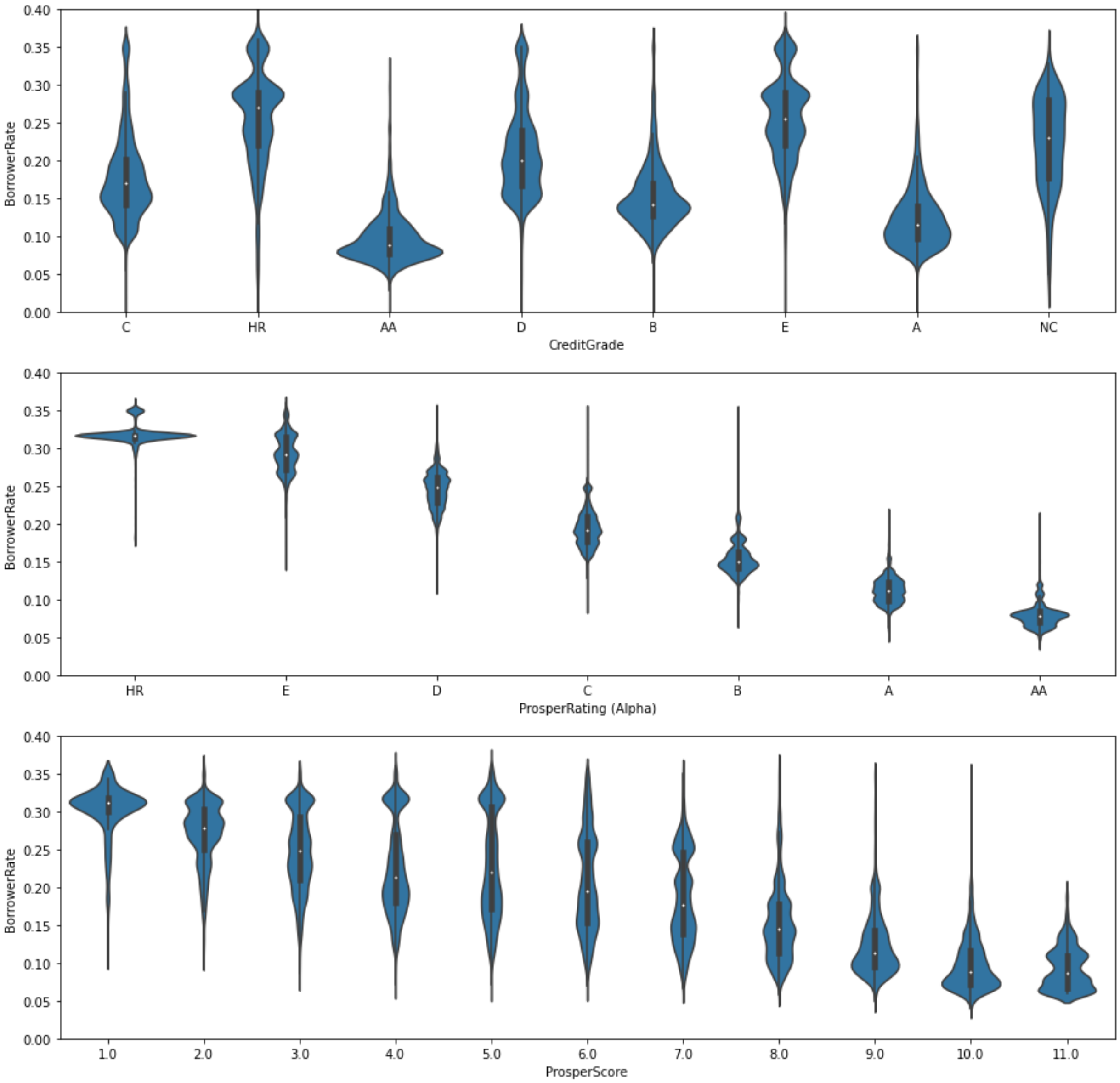
```
In [193]: plt.figure(figsize = [10, 8])
          sns.countplot(data=selected_data, x='ProsperRating (Alpha)', hue='EmploymentStatu
          plt.legend(loc=1);
```



Lower ratings were noted to have greater proportions of individuals with employment status Not Employed, Self-employed, Retired, and Part-Time.

In [168]:

```
plt.figure(figsize=[15,15])
plt.subplot(3,1,1)
sns.violinplot(data=selected_data, x='CreditGrade', y='BorrowerRate', color=base_
plt.ylim((0,0.4));

plt.subplot(3,1,2)
sns.violinplot(data=selected_data, x='ProsperRating (Alpha)', y='BorrowerRate', c
plt.ylim((0,0.4));

plt.subplot(3,1,3)
sns.violinplot(data=selected_data, x='ProsperScore', y='BorrowerRate', color=base
plt.ylim((0,0.4));
```



> Violin plots show the distribution of the different credit risks and the BorrowerRate is that there is a negative correlation between having a higher score and a lower interest rate. This is once again most pronounced in the results for the ProsperRating (Alpha) variable where the decline is steeper. With ProsperScore there is still a decline, but the distribution of the BorrowerRate is more distributed and the IQR is generally larger for each rating.

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?

> distribution of the different credit risks annd the BorrowerRate is that there is a negative correlation between having a higher (better) score and a lower interest rate. This is once again most pronouced in the results for the ProsperRating (Alpha) variable where the decline is steeper. With the ProsperScore there is still a decilne, but the distribution of the BorrowerRate is more distributed and the IQR is genrally larger for each rating.
> the different sizes of the total loan amount, the borrower's APR has a large distribution, but the range of APR decreases with the increase of loan amount. So the borrower's APR is negatively correlated with the total loan amount.

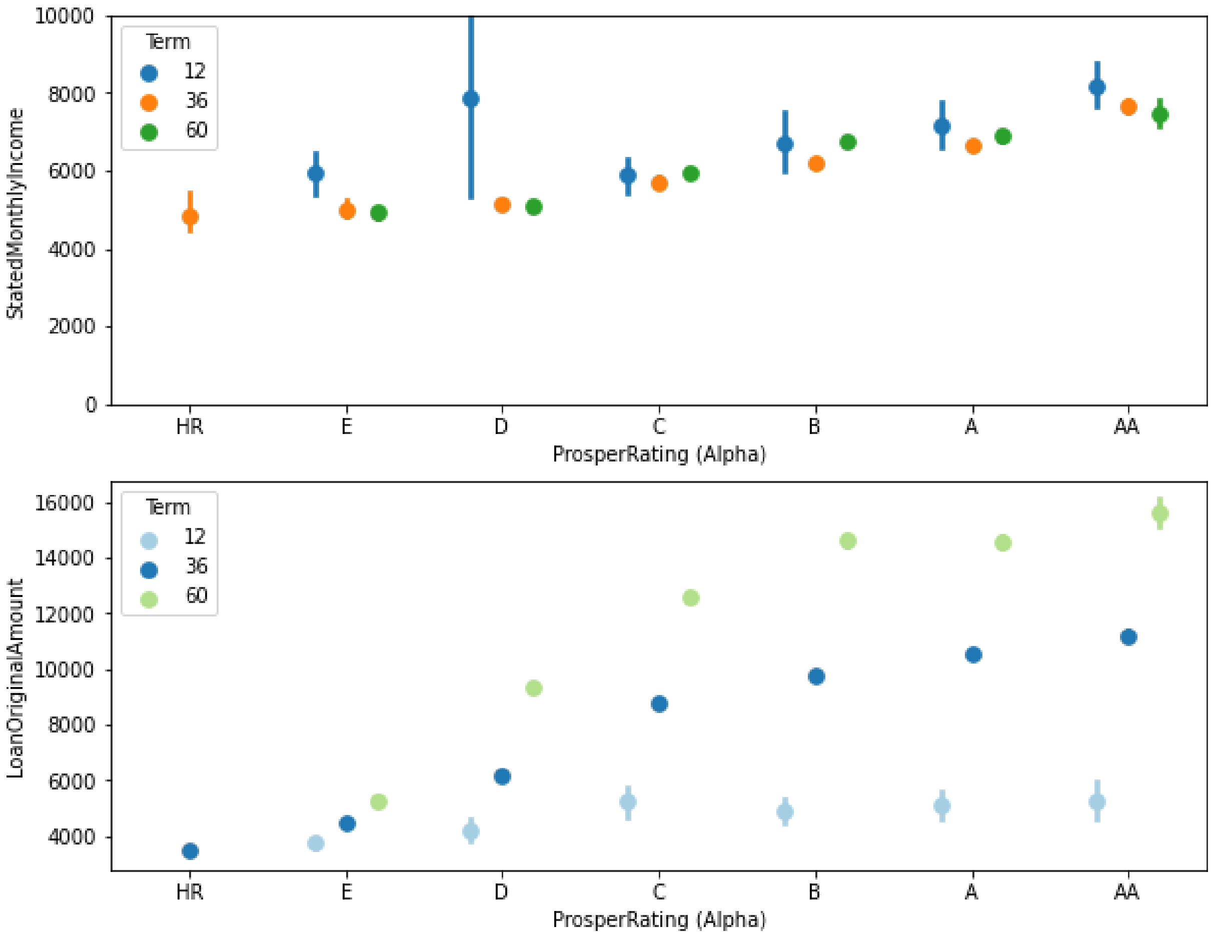
Did you observe any interesting relationships between the other features (not the main feature(s) of interest)?

> Some positive relationships between the categorical variables and the two numeric variables of interest. The loans with 60-month term loans were quite popular with those with a ProsperRating of A and C. As seen before 36-month term loans are the most popular across all credit risk groups.
>
> I was confused about how the DebtToIncomeRatio had no meaningful relation to the interest rate columns. The other features were easy to the expectation.

Multivariate Exploration

The main thing I want to explore in the rating and term effects on stated monthly income and loan original amount variables.

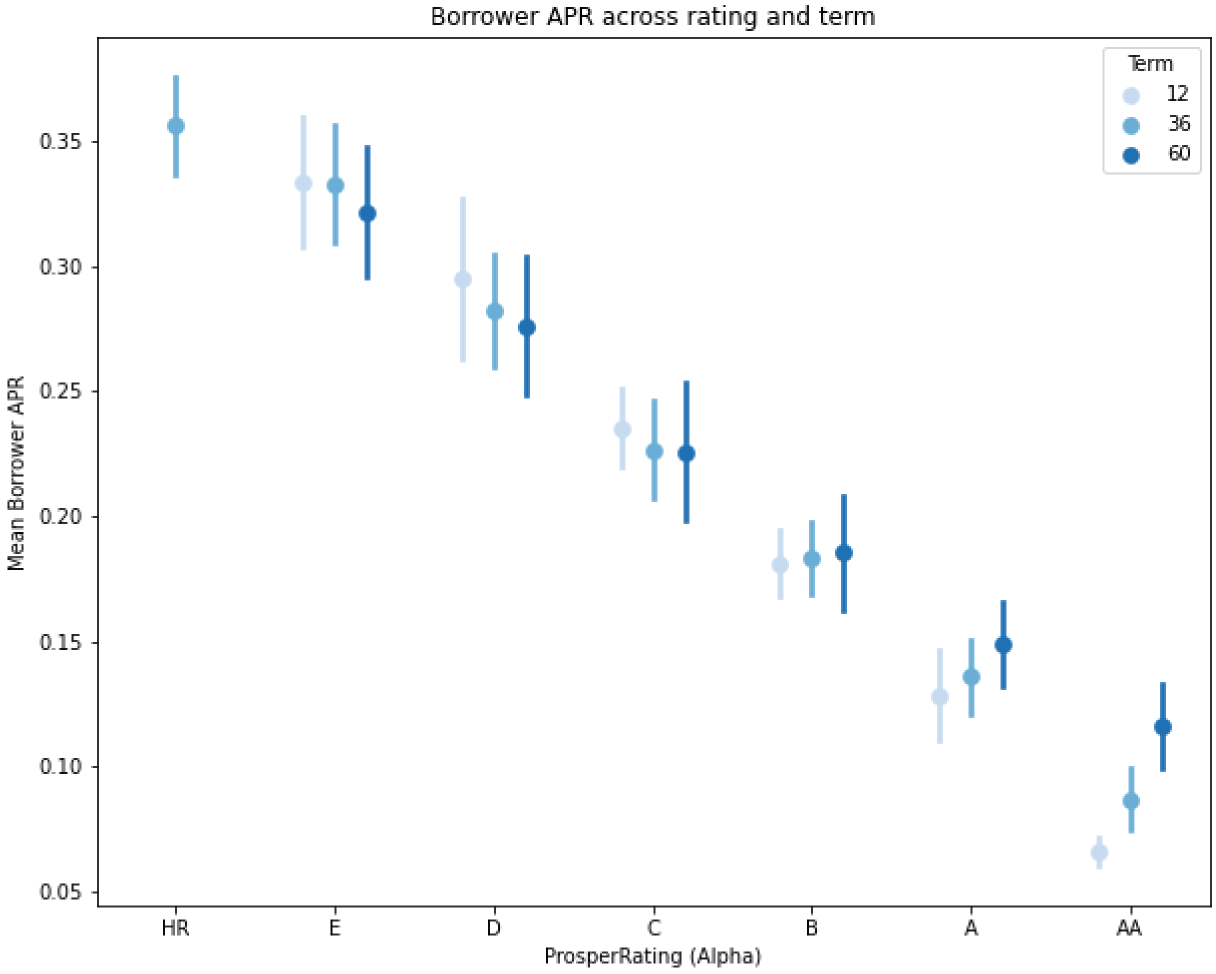
```
In [176]: plt.figure(figsize=[10,8])

          plt.subplot(2,1,1)
          sns.pointplot(data = selected_data, x = 'ProsperRating (Alpha)', y = 'StatedMonth
                     palette = 'tab10', linestyles = '', dodge = 0.4)
          plt.ylim(0,10000);
          plt.subplot(2,1,2)
          sns.pointplot(data = selected_data, x = 'ProsperRating (Alpha)', y = 'LoanOrigina
                     palette = 'Paired', linestyles = '', dodge = 0.4);
```



> About stated monthly income, it doesn't seem like there is an interaction effect between term and rating, the pattern of the term is similar among different ratings. But for the loan amount, there is an interaction between term and rating. We can see that with a better Prosper rating, the loan amount of all three terms increases, the increased amplitude of loan amount between terms also becomes larger.

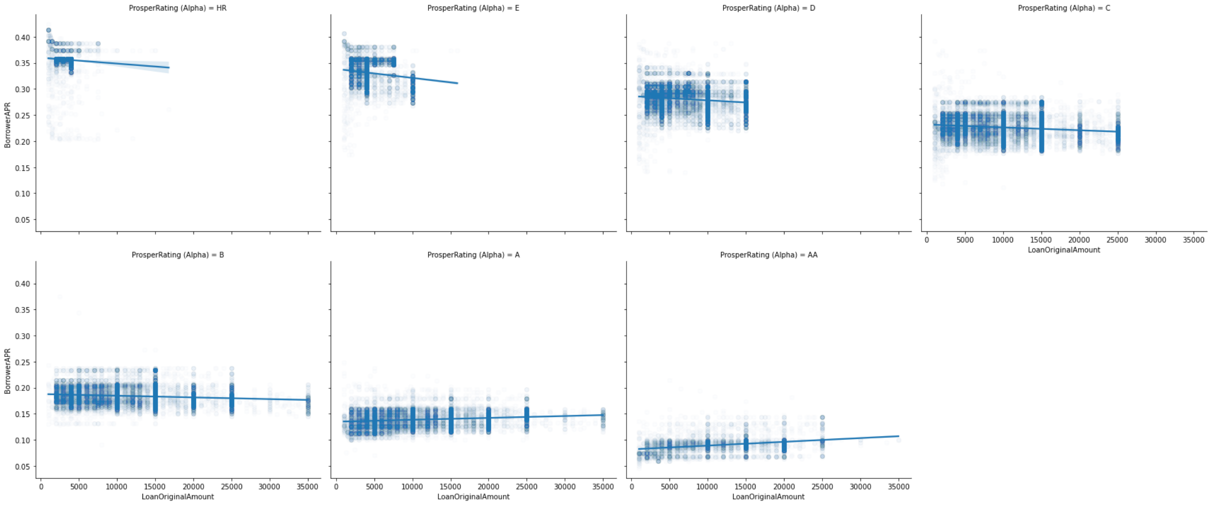
In [179]:

```
plt.figure(figsize = [10,8])
sns.pointplot(data = selected_data, x = 'ProsperRating (Alpha)', y = 'BorrowerAPR
           palette = 'Blues', linestyles = '', dodge = 0.4, ci='sd')
plt.title('Borrower APR across rating and term')
plt.ylabel('Mean Borrower APR');
```



Interestingly, the borrower APR decreases with the increase of borrow term for people with HR & C ratings. But for people with B & AA ratings, the APR increase with the increase of borrow term.

```
In [184]: # Prosper rating effect on relationship of APR and loan amount
          g=sns.FacetGrid(data=selected_data, aspect=1.2, height=5, col='ProsperRating (Alp
          g.map(sns.regplot, 'LoanOriginalAmount', 'BorrowerAPR', x_jitter=0.04, scatter_kw
          g.add_legend();
```

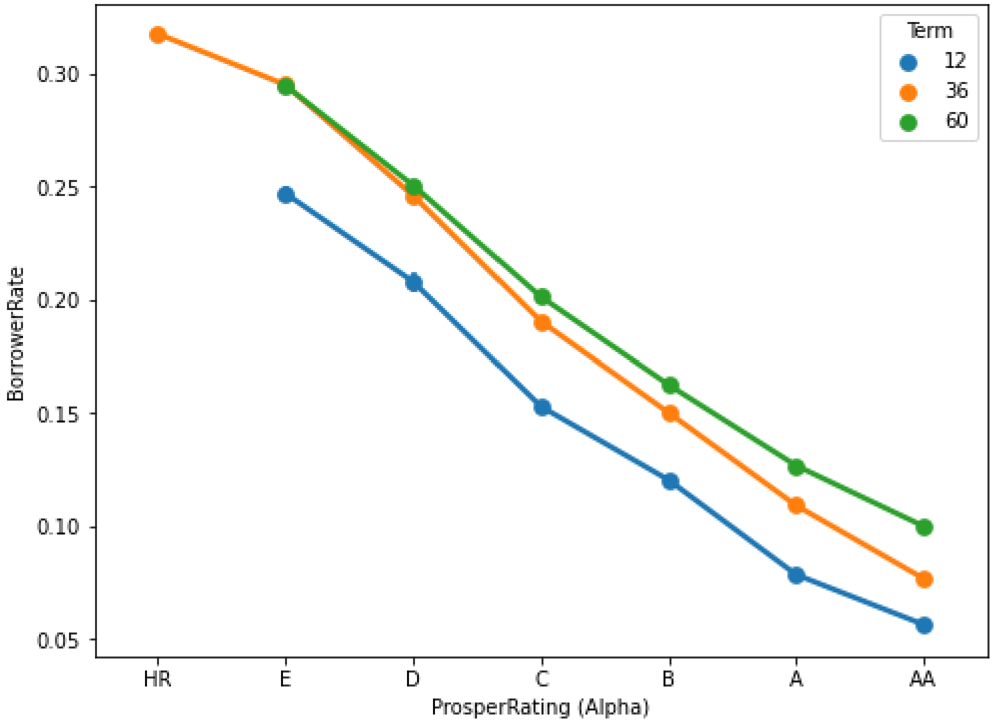


The loan amount increases with the best rating. The borrower's APR decreases with the best rating. Interestingly, the relationship between borrower APR and loan amount turns from negative to slightly positive when the Prosper ratings are increased from HR to A or higher. This may be because people with A or AA ratings tend to borrow more money, increasing the borrower's APR could prevent them borrow even more and maximizing the profit. But people with lower ratings tend to borrow less money, decreasing the borrower's APR could encourage them to borrow more.

Finally, let's look at relationships between the three categorical features.

In [190]:

```
# Create a pointplot to show how the BorrowerRate changes for different Loan Term
# when split up by ProsperRating
plt.figure(figsize=[8,6])

sns.pointplot(data=selected_data, x='ProsperRating (Alpha)', y='BorrowerRate', hu
                   palette='tab10');
plt.legend(loc=1, title='Term');
```



This seems to not make sense but for every single level of the ProsperRating, the BorrowerRate increases as for longer-term loans. I would think it to be the reverse as shorter-term loans usually carry a higher interest rate. This was not at all evident in the bivariate analysis and comes a bit as a surprise.

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?

> During investing the ProsperRating (Alpha) and the ProsperScore as it relates to the BorrowerRate I could know an explanation for why the ProsperScore wasn't as highly correlated to the BorrowerRate. It is opposite the other credit risk features must use different criteria in coming up with its value.

Were there any interesting or surprising interactions between features?

> A surprising interaction is that the borrower's APR and loan amount is negative correction when the Prosper ratings are from HR to B, but the correlation is turned to be positive when the ratings are A and AA. Another interesting thing is that the borrower APR decreases with the increase of borrow term for people with HR-C ratings. But for people with B-AA ratings, the borrower's APR increase with the borrowing term.
>
> BorrowerRate increases for longer Term loans when split up by ProsperRating (Alpha). The opposite relationship would be expected as longer-term loans generally carry a lower risk profile and have a longer time to accrue interest.