Python Project - (Dataset Exploration Title)

by (Angela Imarhiagbe)

In [ ]:
```python
import pandas as pd
```

In [261…
```python
df = pd.read_csv('Prosper Loan.csv')
```

In [262…
```python
df
```

Out[262]:

	ListingKey	ListingNumber	ListingCreationDate	CreditGrade	Term	LoanStatus	ClosedDate	BorrowerAPR	BorrowerRate	LenderYield	...	LP_ServiceF
0	1021339766868145413AB3B	193129	09:29.3	C	36	Completed	14/08/2009 00:00	0.16516	0.1580	0.1380	...	-13
1	10273602499503308B223C1	1209647	28:07.9	NaN	36	Current	NaN	0.12016	0.0920	0.0820	...	(
2	0EE9337825851032864889A	81716	00:47.1	HR	36	Completed	17/12/2009 00:00	0.28269	0.2750	0.2400	...	-2
3	0EF5356002482715299901A	658116	02:35.0	NaN	36	Current	NaN	0.12528	0.0974	0.0874	...	-10
4	0F023589499656230C5E3E2	909464	38:39.1	NaN	36	Current	NaN	0.24614	0.2085	0.1985	...	-6
...	...	...	...	...	...	...	...	...	...	...	...	
113932	E6D9357655724827169606C	753087	55:02.7	NaN	36	Current	NaN	0.22354	0.1864	0.1764	...	-7
113933	E6DB353036033497292EE43	537216	42:55.3	NaN	36	FinalPaymentInProgress	NaN	0.13220	0.1110	0.1010	...	-3
113934	E6E13596170052029692BB1	1069178	49:12.7	NaN	60	Current	NaN	0.23984	0.2150	0.2050	...	-1
113935	E6EB3531504622671970D9E	539056	18:26.6	NaN	60	Completed	13/08/2013 00:00	0.28408	0.2605	0.2505	...	-23
113936	E6ED3600409833199F711B7	1140093	27:37.7	NaN	36	Current	NaN	0.13189	0.1039	0.0939	...	-

113937 rows × 81 columns

I will be visualizing the Prosper loan database, this database is for a loan company which contains all the information regarding the company

In [264…
```python
df.columns = df.columns.str.lower()
```

In [265…
```python
df.head(1)
```

Out[265]:

	listingkey	listingnumber	listingcreationdate	creditgrade	term	loanstatus	closeddate	borrowerapr	borrowerrate	lenderyield	...	lp_servicefees	lp_collectionfees	lp_
0	1021339766868145413AB3B	193129	09:29.3	C	36	Completed	14/08/2009 00:00	0.16516	0.158	0.138	...	-133.18	0.0	

1 rows × 81 columns

Lets drop some unwanted columns

In [315…
```python
df.drop('lp_nonprincipalrecoverypayments', axis=1, inplace=True)
```

In [267…
```python
df['datecreditpulled'] = pd.to_datetime(df['datecreditpulled'],errors='coerce',infer_datetime_format=True)
```

In [322…
```python
df
```

Out[322]:

	listingnumber	listingcreationdate	term	loanstatus	borrowerapr	borrowerrate	lenderyield	estimatedeffectiveyield	estimatedloss	estimatedreturn	...	monthlyloanpayn
0	193129	09:29.3	36	Completed	0.16516	0.1580	0.1380	0.00000	0.0000	0.00000	...	33
1	1209647	28:07.9	36	Current	0.12016	0.0920	0.0820	0.07960	0.0249	0.05470	...	31
2	81716	00:47.1	36	Completed	0.28269	0.2750	0.2400	0.00000	0.0000	0.00000	...	12
3	658116	02:35.0	36	Current	0.12528	0.0974	0.0874	0.08490	0.0249	0.06000	...	32
4	909464	38:39.1	36	Current	0.24614	0.2085	0.1985	0.18316	0.0925	0.09066	...	56
...	...	...	...	...	...	...	...	...	...	...	...	
113932	753087	55:02.7	36	Current	0.22354	0.1864	0.1764	0.16490	0.0699	0.09500	...	36
113933	537216	42:55.3	36	FinalPaymentInProgress	0.13220	0.1110	0.1010	0.10070	0.0200	0.08070	...	6
113934	1069178	49:12.7	60	Current	0.23984	0.2150	0.2050	0.18828	0.1025	0.08578	...	27
113935	539056	18:26.6	60	Completed	0.28408	0.2605	0.2505	0.24450	0.0850	0.15950	...	44
113936	1140093	27:37.7	36	Current	0.13189	0.1039	0.0939	0.09071	0.0299	0.06081	...	6

113937 rows × 73 columns

In [321…
```python
df.estimatedreturn = df.estimatedreturn.fillna(0)
```

In [274…
```python
df.rename(columns = {'prosperrating (numeric)':'prosperratingnumeric', 'prosperrating (numeric)':'prosperratingnumeric'}, inplace = True)
```

In [275…
```python
df.datecreditpulled = df.datecreditpulled.fillna(0)
```

In [323…
```python
df
```

Out[323]:

	listingnumber	listingcreationdate	term	loanstatus	borrowerapr	borrowerrate	lenderyield	estimatedeffectiveyield	estimatedloss	estimatedreturn	...	monthlyloanpayn
0	193129	09:29.3	36	Completed	0.16516	0.1580	0.1380	0.00000	0.0000	0.00000	...	33
1	1209647	28:07.9	36	Current	0.12016	0.0920	0.0820	0.07960	0.0249	0.05470	...	31
2	81716	00:47.1	36	Completed	0.28269	0.2750	0.2400	0.00000	0.0000	0.00000	...	12
3	658116	02:35.0	36	Current	0.12528	0.0974	0.0874	0.08490	0.0249	0.06000	...	32
4	909464	38:39.1	36	Current	0.24614	0.2085	0.1985	0.18316	0.0925	0.09066	...	56
...	...	...	...	...	...	...	...	...	...	...	...	
113932	753087	55:02.7	36	Current	0.22354	0.1864	0.1764	0.16490	0.0699	0.09500	...	36
113933	537216	42:55.3	36	FinalPaymentInProgress	0.13220	0.1110	0.1010	0.10070	0.0200	0.08070	...	6
113934	1069178	49:12.7	60	Current	0.23984	0.2150	0.2050	0.18828	0.1025	0.08578	...	27
113935	539056	18:26.6	60	Completed	0.28408	0.2605	0.2505	0.24450	0.0850	0.15950	...	44
113936	1140093	27:37.7	36	Current	0.13189	0.1039	0.0939	0.09071	0.0299	0.06081	...	6

113937 rows × 73 columns

In [324…
```python
df['statedmonthlyincome']=df['statedmonthlyincome'].astype(int)
```

In [325…
```python
df.statedmonthlyincome
```

Out[325]:
```
0         3083
1         6125
2         2083
3         2875
4         9583
          ...
113932    4333
113933    8041
113934    2875
113935    3875
113936    4583
Name: statedmonthlyincome, Length: 113937, dtype: int32
```

I perform some cleaning in m dataset and also i dropped some columns which i would not be needing as as to have a clean and clear database for easier visualization.

Univariate Exploration

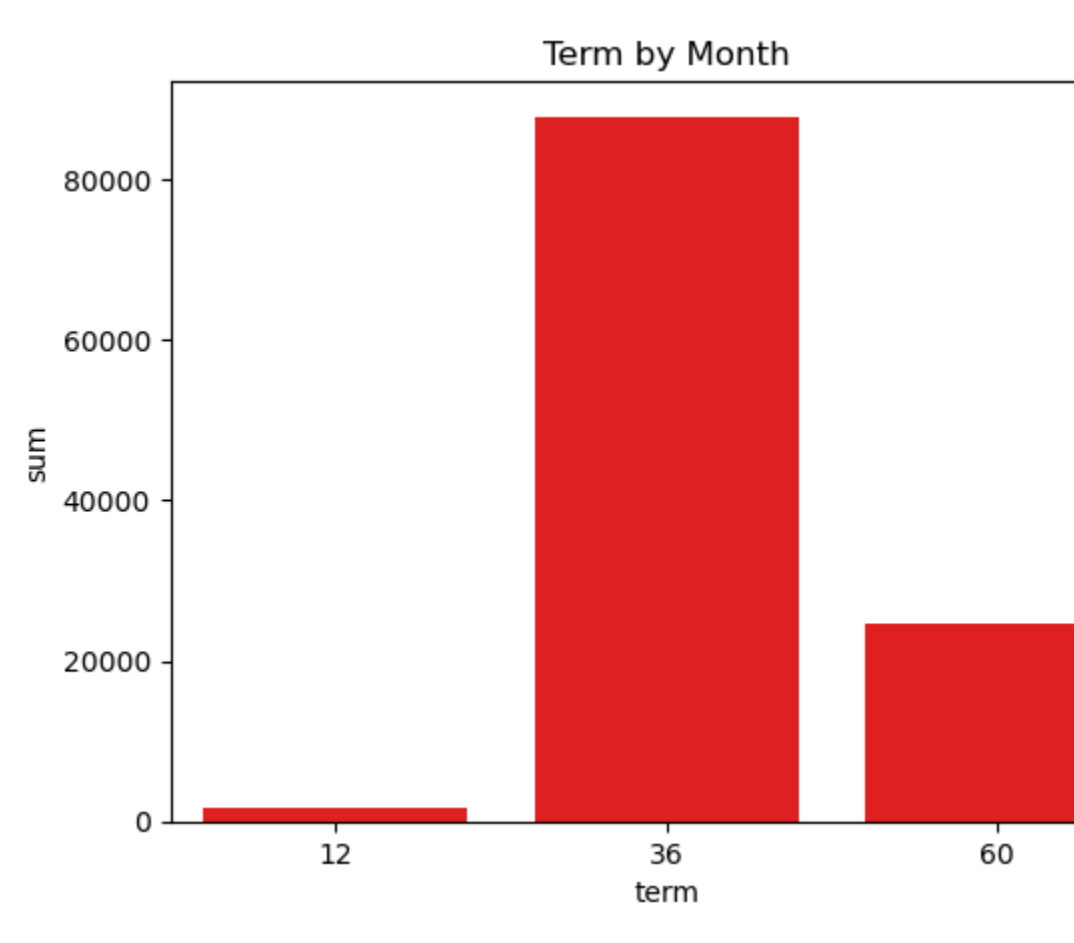
In [280…
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [326…
```python
df.term.value_counts()
```

Out[326]:
```
36    87778
60    24545
12     1614
Name: term, dtype: int64
```

In [350…
```python
sns.countplot(data=df,x='term', color='red');
plt.title('Term by Month')
plt.xlabel('term')
plt.ylabel('sum')
```

Out[350]:
```
Text(0, 0.5, 'sum')
```



This visualization shows that over 80,000 customer who borrow money from Prosper are under the "36 months term" which is also equivalent to 3 years.

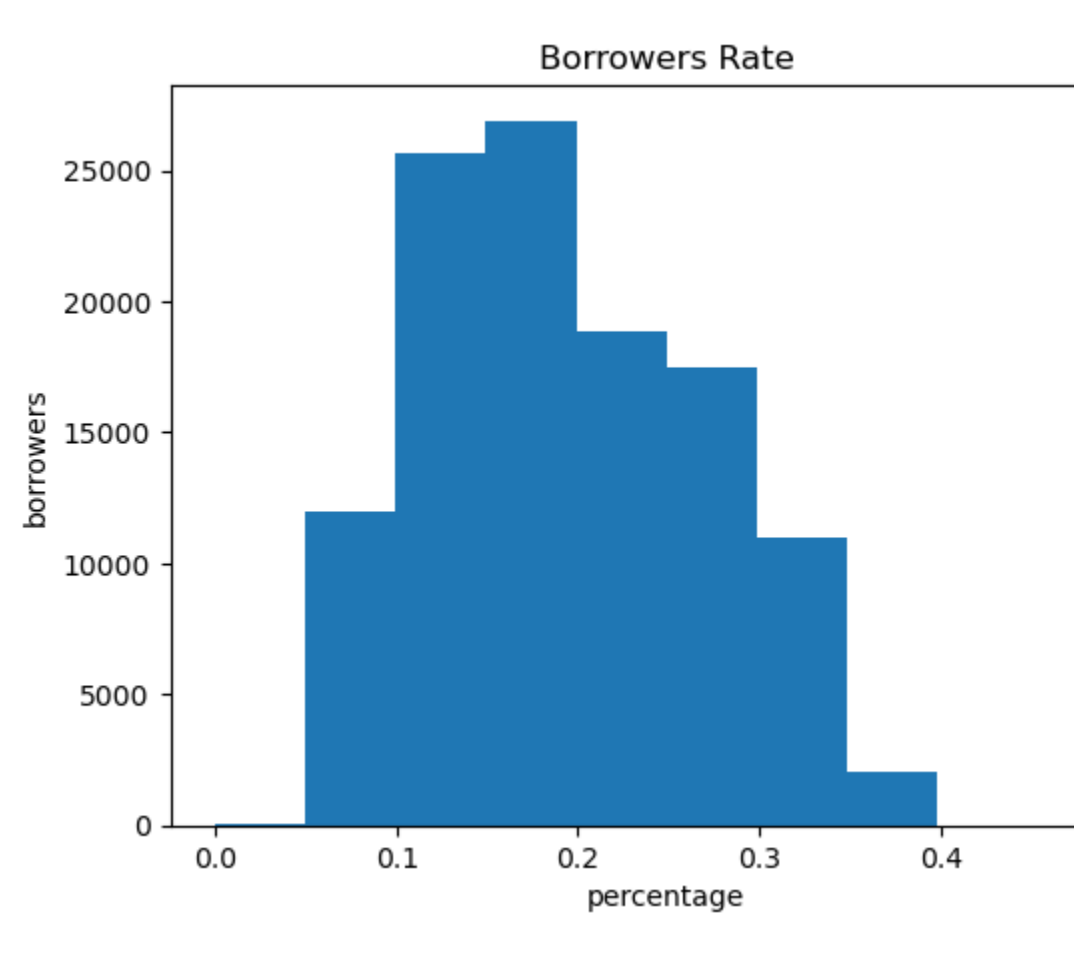
and lesss than 1,000 of their customer falls under the "12 months term" which is also equivalent to 1 year.

In [328…
```python
df.borrowerrate.value_counts()
```

Out[328]:
```
0.3177    3672
0.3500    1905
0.3199    1651
0.2900    1508
0.2699    1319
          ...
0.2201       1
0.0752       1
0.1416       1
0.2812       1
0.0739       1
Name: borrowerrate, Length: 2294, dtype: int64
```

In [329…
```python
plt.hist(data=df, x='borrowerrate');
plt.title('Borrowers Rate')
plt.xlabel('percentage')
plt.ylabel('borrowers')
```

Out[329]:
```
Text(0, 0.5, 'borrowers')
```



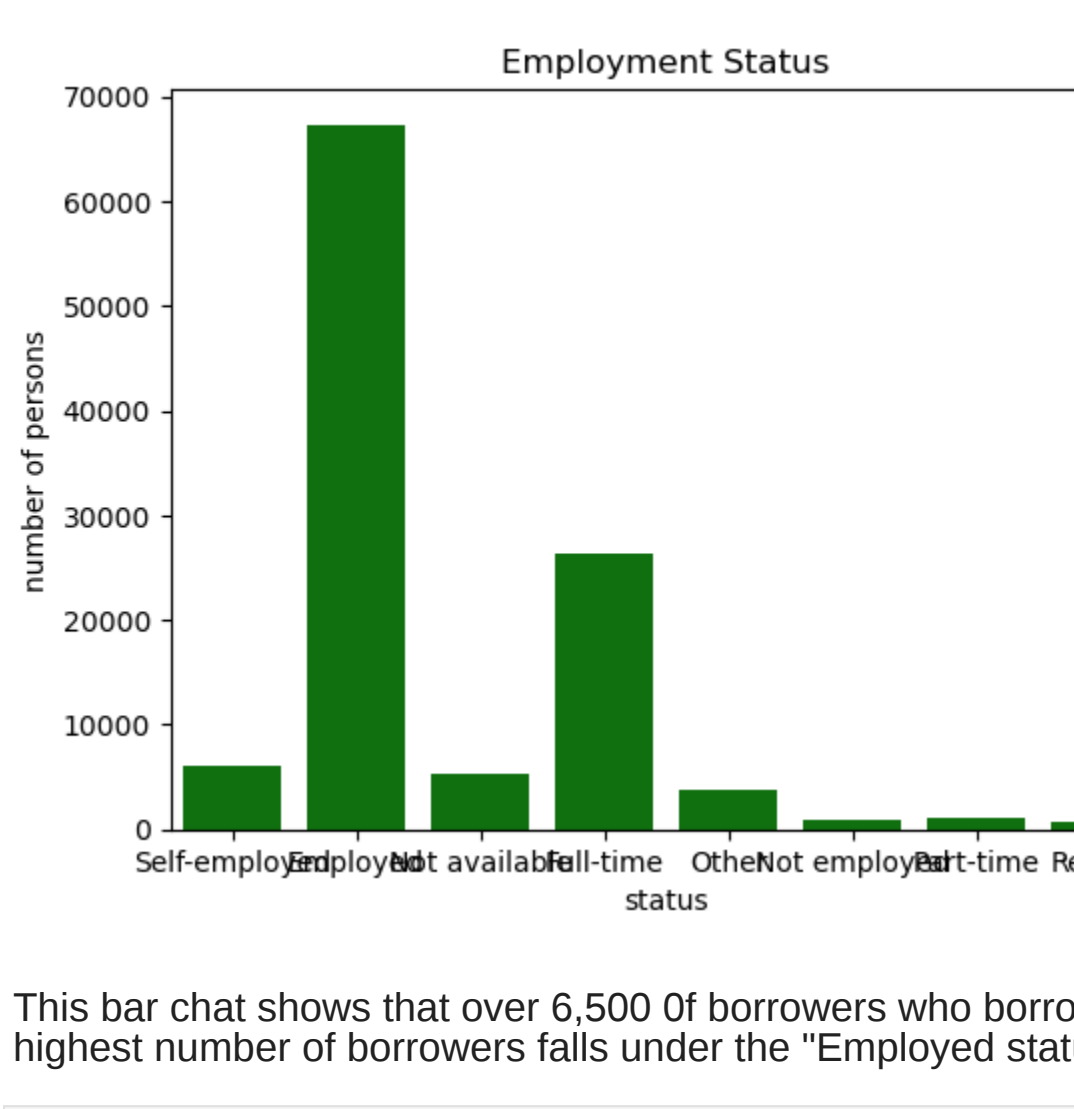
This Histogram shows that more than 25,000 of the company'scustomer are between the borrowers rate of "0.2 and 0.3" percent.

In [330…
```python
df.employmentstatus.value_counts()
```

Out[330]:
```
Employed         67322
Full-time        26355
Self-employed     6134
Not available     5347
Other             3806
Part-time         1088
Not employed       835
Retired            795
Name: employmentstatus, dtype: int64
```

In [331…
```python
sns.countplot(data=df,x='employmentstatus', color='green');
plt.title('Employment Status')
plt.xlabel('status')
plt.ylabel('number of persons')
```

Out[331]:
```
Text(0, 0.5, 'number of persons')
```



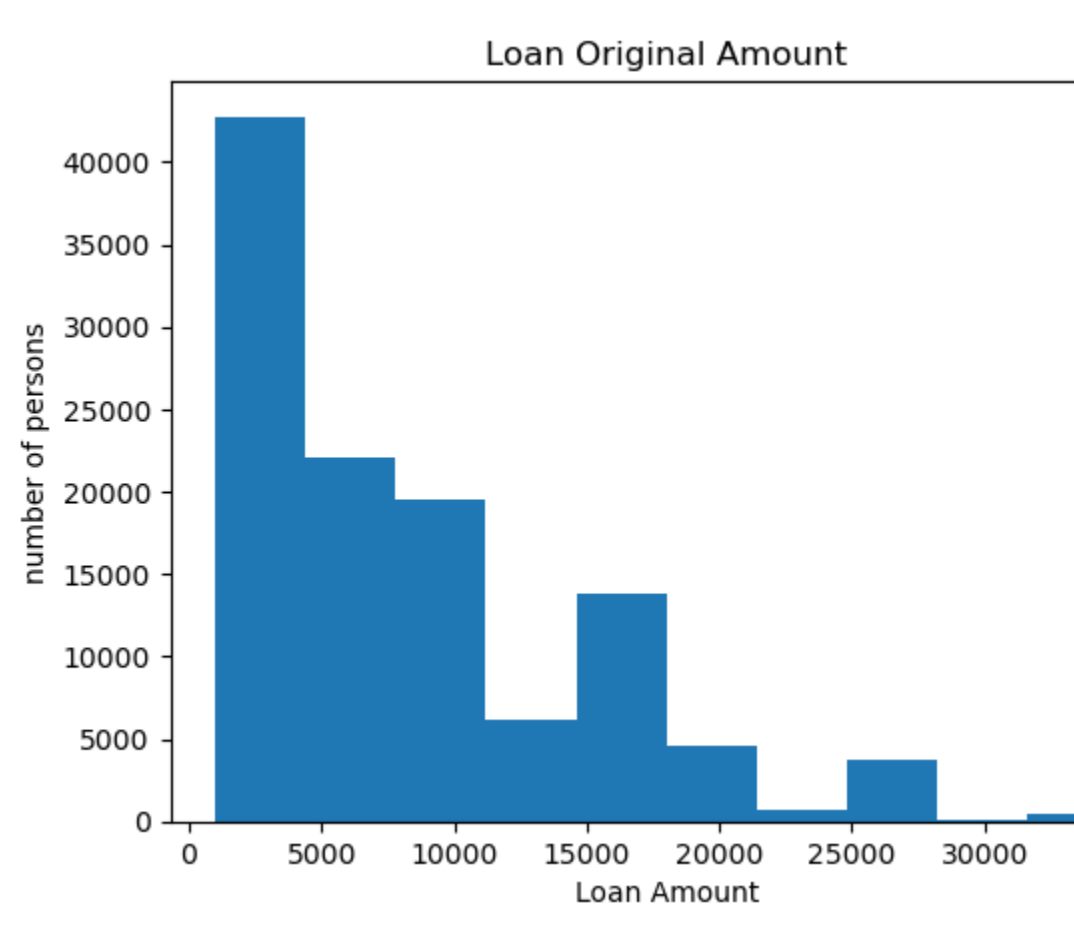
This bar chat shows that over 6,500 0f borrowers who borrows money from prosper loan are employed, which means the highest number of borrowers falls under the "Employed status"

In [332…
```python
df.loanoriginalamount.value_counts()
```

Out[332]:
```
4000     14333
15000    12407
10000    11106
5000      6990
2000      6067
         ...
5284         1
7936         1
1201         1
10593        1
4292         1
Name: loanoriginalamount, Length: 2468, dtype: int64
```

In [333…
```python
plt.hist(data=df, x='loanoriginalamount');
plt.title('Loan Original Amount')
plt.xlabel('Loan Amount')
plt.ylabel('number of persons')
```

Out[333]:
```
Text(0, 0.5, 'number of persons')
```



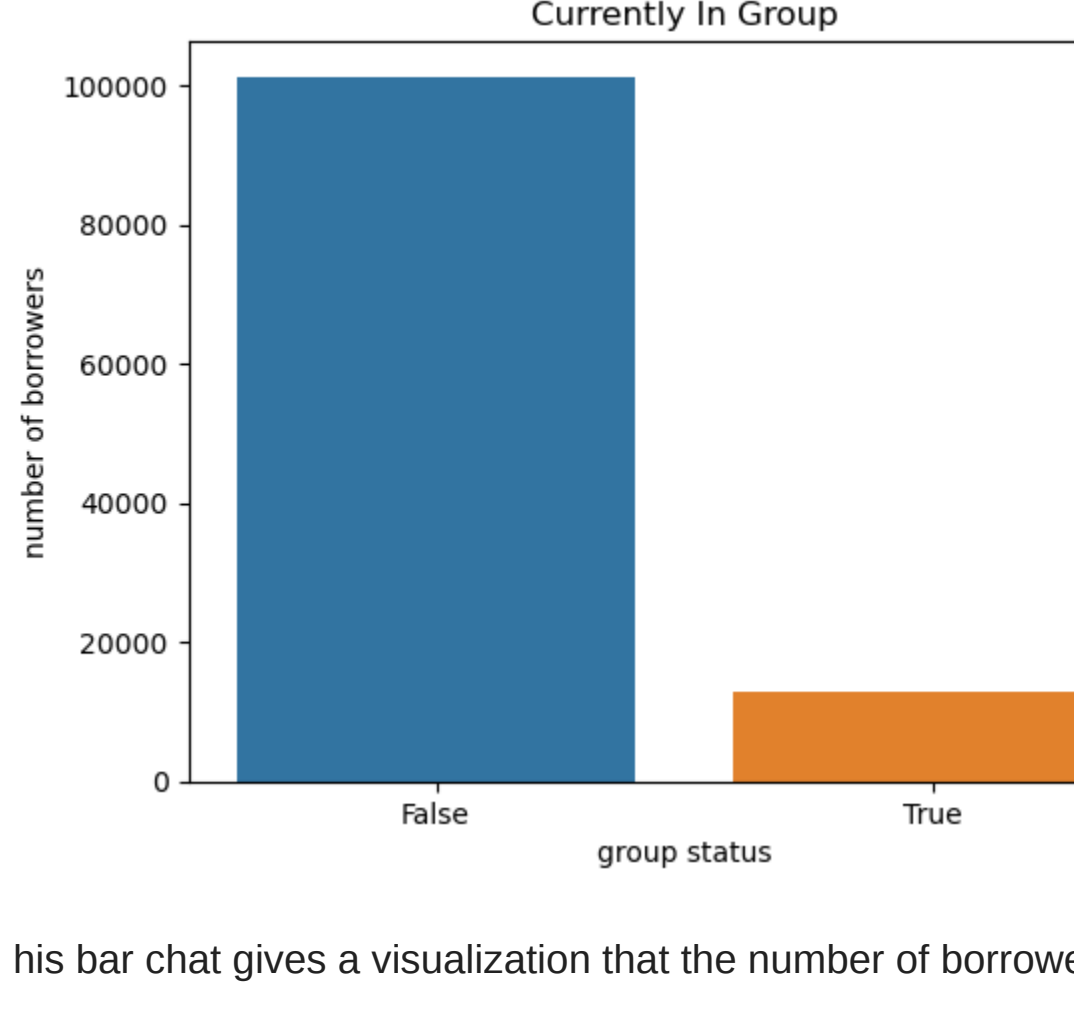
This histogram shows that most of the borrowers original loan is between 5,000 - 10,000.

In [288…
```python
df.currentlyingroup.value_counts()
```

Out[288]:
```
False    101218
True      12719
Name: currentlyingroup, dtype: int64
```

In [334…
```python
sns.countplot(data=df,x='currentlyingroup');
plt.title('Currently In Group')
plt.xlabel('group status')
plt.ylabel('number of borrowers')
```

Out[334]:
```
Text(0, 0.5, 'number of borrowers')
```

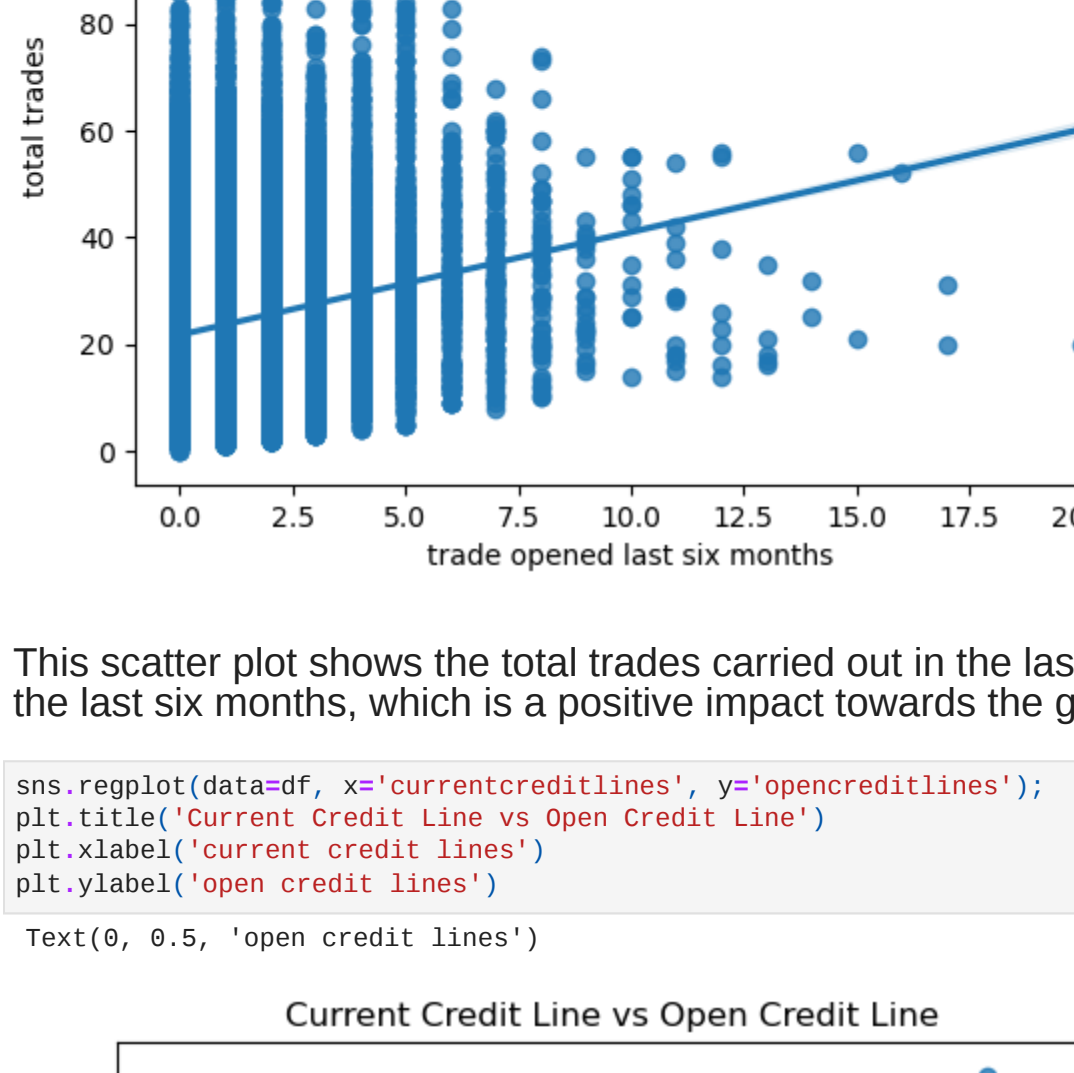


his bar chat gives a visualization that the number of borrowers who are not in a group are more than those in a group.

Bivariate Exploration

In [335…
```python
sns.regplot(data=df, x='tradesopenedlast6months', y='totaltrades');
plt.title('Total Trade In The Last Six Months')
plt.xlabel('trade opened last six months')
plt.ylabel('total trades')
```

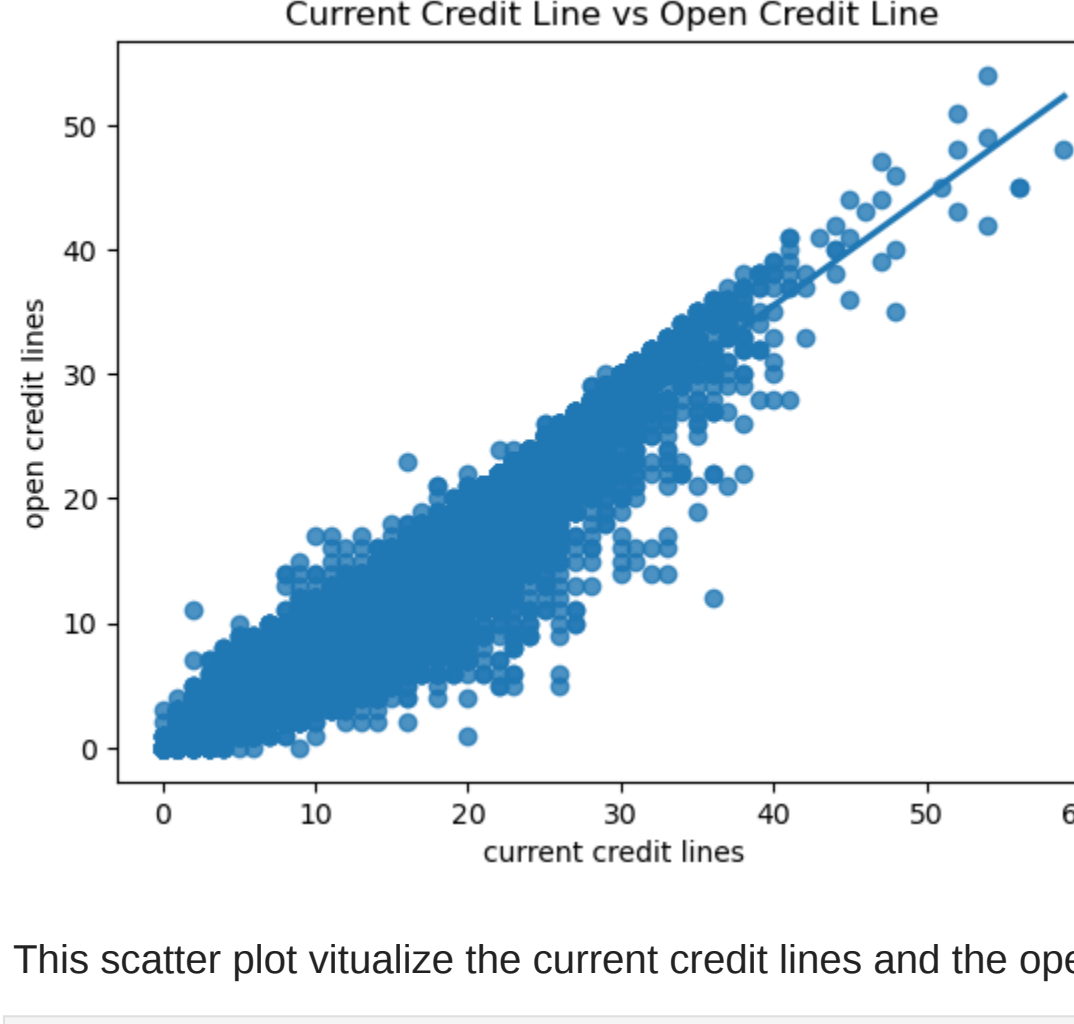
Out[335]:
```
Text(0, 0.5, 'total trades')
```



This scatter plot shows the total trades carried out in the last six months. This shows that trade has increased drastically within the last six months, which is a positive impact towards the growth of the company.

In [194…
```python
sns.regplot(data=df, x='currentcreditlines', y='opencreditlines');
plt.title('Current Credit Line vs Open Credit Line')
plt.xlabel('current credit lines')
plt.ylabel('open credit lines')
```

Out[194]:
```
Text(0, 0.5, 'open credit lines')
```



This scatter plot vitualize the current credit lines and the open credit lines

In [340…
```python
df.prosperratingnumeric.mean()
```

Out[340]:
```
4.07224258423391
```

In [343…
```python
df['size'] = np.where(df. prosperratingnumeric > np.mean(df. prosperratingnumeric), 'large', 'small')
```

In [344…
```python
df
```

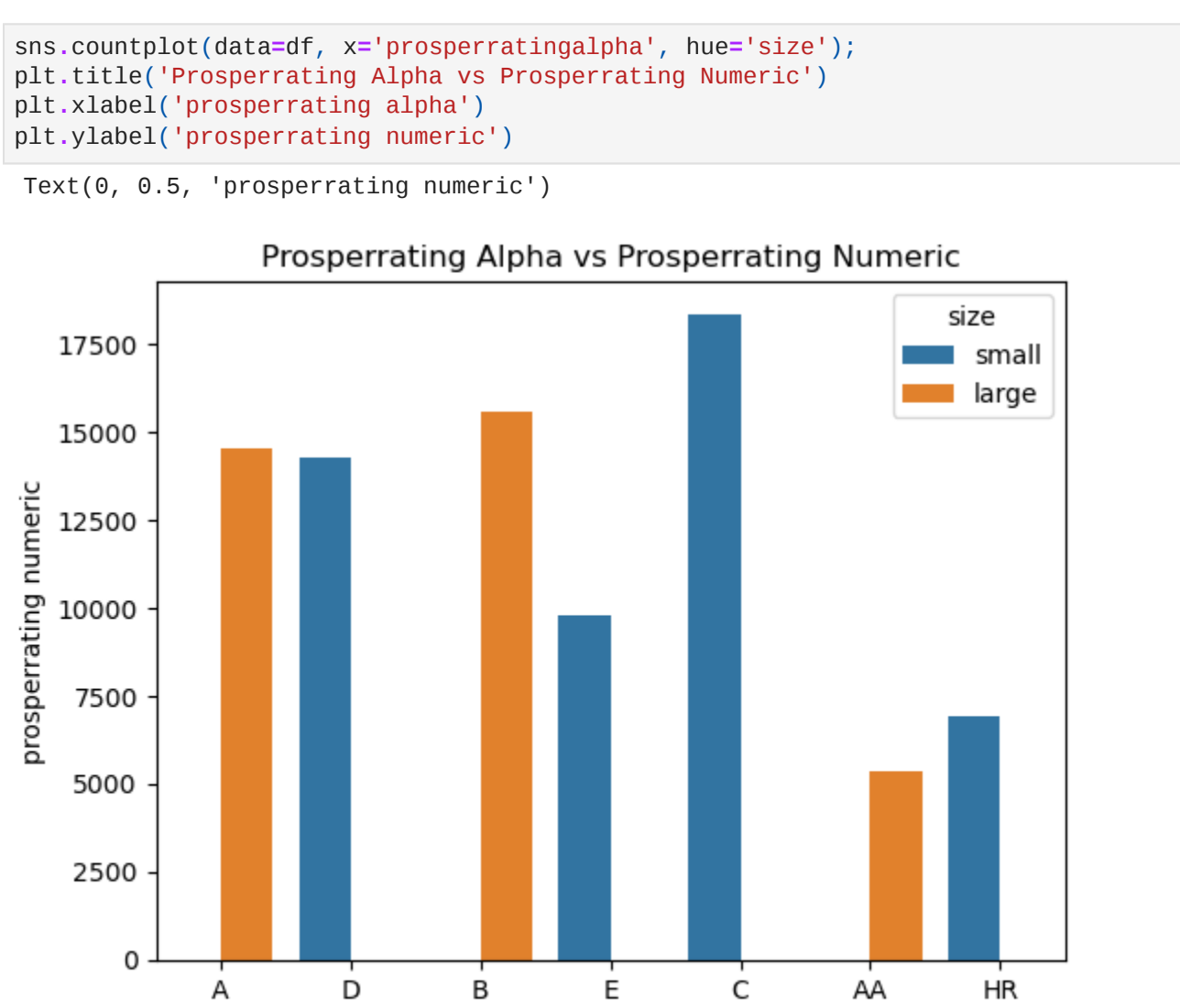
Out[344]:

	listingnumber	listingcreationdate	term	loanstatus	borrowerapr	borrowerrate	lenderyield	estimatedeffectiveyield	estimatedloss	estimatedreturn	...	lp_customerpayn
0	193129	09:29.3	36	Completed	0.16516	0.1580	0.1380	0.00000	0.0000	0.00000	...	11396
1	1209647	28:07.9	36	Current	0.12016	0.0920	0.0820	0.07960	0.0249	0.05470	...	0
2	81716	00:47.1	36	Completed	0.28269	0.2750	0.2400	0.00000	0.0000	0.00000	...	4186
3	658116	02:35.0	36	Current	0.12528	0.0974	0.0874	0.08490	0.0249	0.06000	...	5143
4	909464	38:39.1	36	Current	0.24614	0.2085	0.1985	0.18316	0.0925	0.09066	...	2819
...	...	...	...	...	...	...	...	...	...	...	...	
113932	753087	55:02.7	36	Current	0.22354	0.1864	0.1764	0.16490	0.0699	0.09500	...	3647
113933	537216	42:55.3	36	FinalPaymentInProgress	0.13220	0.1110	0.1010	0.10070	0.0200	0.08070	...	2330
113934	1069178	49:12.7	60	Current	0.23984	0.2150	0.2050	0.18828	0.1025	0.08578	...	546
113935	539056	18:26.6	60	Completed	0.28408	0.2605	0.2505	0.24450	0.0850	0.15950	...	21122
113936	1140093	27:37.7	36	Current	0.13189	0.1039	0.0939	0.09071	0.0299	0.06081	...	64

113937 rows × 74 columns

In [347…
```python
sns.countplot(data=df, x='prosperratingalpha', hue='size');
plt.title('Prosperrating Alpha vs Prosperrating Numeric')
plt.xlabel('prosperrating alpha')
plt.ylabel('prosperrating numeric')
```

Out[347]:
```
Text(0, 0.5, 'prosperrating numeric')
```



This chat shows that "category c" which is the middle category has the highest rating.

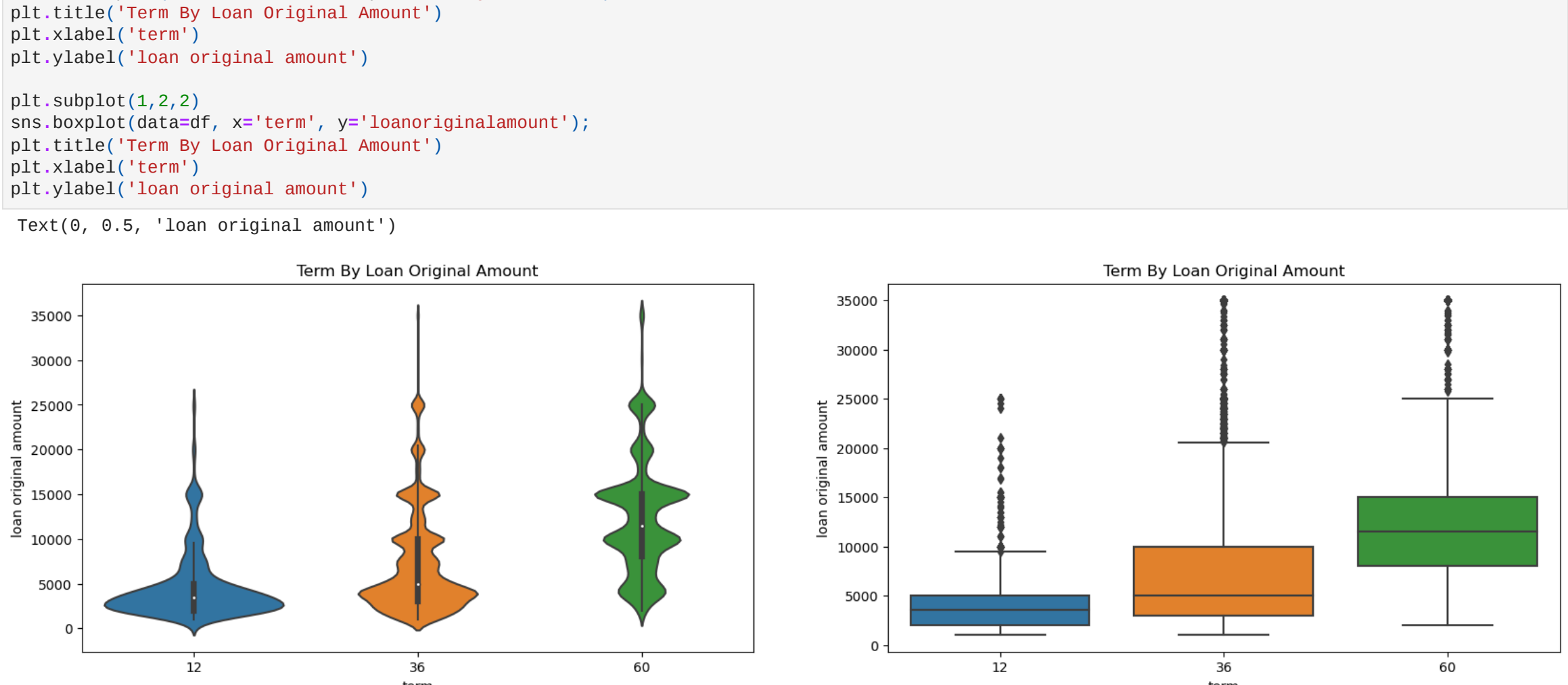
for the category and category bivariate exploration, i had to look for the mean to be able to plot my clustered bar chat. i had to look for a way that would make it work and also be able to vitualize my database.

In [342…
```python
plt.figure(figsize=[20,5])

plt.subplot(1,2,1)
sns.violinplot(data=df, x='term', y='loanoriginalamount');
plt.title('Term By Loan Original Amount')
plt.xlabel('term')
plt.ylabel('loan original amount')

plt.subplot(1,2,2)
sns.boxplot(data=df, x='term', y='loanoriginalamount');
plt.title('Term By Loan Original Amount')
plt.xlabel('term')
plt.ylabel('loan original amount')
```

Out[342]:
```
Text(0, 0.5, 'loan original amount')
```



IN CONCLUSION: This is a dataset that shows relationships between varables in a loan company. Univariate Exploration focuses on the term also refers to as the period the loan is been giving for, the borrowers rate, borrowers employment status which shows over 60% of the borrowers are employed, the loan original amount and the grouping status.

The Bivariate Exploration focuses on relationship between 2 different variables, a visualization shows trades in the last six months, this kind of visualization aid the company in decison making and to trade over time. the current credit line vs open credit line also gives the company inside on how the different magin between both and if the company is runnung as a loss.

In [ ]: