

prosperLoanData__part1_exploratory

October 10, 2022

1 Loan Data (from Prosper) Exploration

1.1 Table of Contents

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

Preliminary Wrangling

Univariate Exploration of Data

Question 1: What is the trend of the `LoanStatus` variable ?

Question 2: What The most used `CreditGrade` ?

Question 3: The most contracted loan `Term` ?

Question 4: `BorrowerAPR` distribution

Question 5: `ProsperRating` distribution

Question 6: `ProsperScore` distribution

Question 7: `ListingCategory` distribution

Question 8: `Investors` distribution

Question 9: `EmploymentStatus`, `Occupation`, `EmploymentStatusDuration` and `IsBorrowerHomeowner` distribution

Question 10: `IncomeRange` distribution

Univariate Exploration Summary

Bivariate Exploration of Data

Question 1: Relationship between `BorrowerAPR` and `BorrowerState`

Question 2: Relationship between `ProsperRating` (Alpha) and `ProsperScore`

Question 3: Relationship between `ProsperScore` and `ProsperRating` (numeric)

Question 4: Relationship between `Term` and `BorrowerState`

Question 5: Relationship between `BorrowerState` per `ProsperScore`

Bivariate Exploration Summary

Multivariate Exploration of Data

Question 1: Relationship between ProsperScore and BorrowerAPR per ProsperRating (Alpha)

Question 2: Relationship between ProsperScore and LenderYield per ProsperRating (Alpha)

Question 3: Relationship between LoanOriginalAmount and BorrowerAPR per ProsperRating (Alpha) per IncomeRange

Multivariate Exploration Summary

Conclusions

Feedback

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

I don't know now but I expect to find out what features can explain loans interest.

What factors affect a loan's outcome status?

What affects the borrower's APR or interest rate?

Are there differences between loans depending on how large the original loan amount was?

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.

- This [data dictionary](#) explains the variables in the data set.
- You are not expected to explore all of the variables in the dataset! Focus your exploration on about 10-15 of them.

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[unicodedecodeerror-utf-8-codec-while-reading-a-csv-file](#)

```
[2]: prosperLoanData = pd.read_csv('prosperLoanData.csv', encoding='utf-8',
    ↪engine='python')
```

We need to expand the number of displayable columns so it fits the number of columns of the DataFrame which is 81.

[how-do-i-expand-the-output-display-to-see-more-columns-of-a-pandas-dataframe](#)

```
[3]: pd.set_option('display.max_rows', 50)
pd.set_option('display.max_columns', 81)
#pd.set_option('display.width', 1000)
```

```
[4]: prosperLoanData.head(10)
```

```
[4]:
```

	ListingKey	ListingNumber	ListingCreationDate	\
0	1021339766868145413AB3B	193129	2007-08-26 19:09:29.263000000	
1	10273602499503308B223C1	1209647	2014-02-27 08:28:07.900000000	
2	0EE9337825851032864889A	81716	2007-01-05 15:00:47.090000000	
3	0EF5356002482715299901A	658116	2012-10-22 11:02:35.010000000	
4	0F023589499656230C5E3E2	909464	2013-09-14 18:38:39.097000000	
5	0F05359734824199381F61D	1074836	2013-12-14 08:26:37.093000000	
6	0F0A3576754255009D63151	750899	2013-04-12 09:52:56.147000000	
7	0F1035772717087366F9EA7	768193	2013-05-05 06:49:27.493000000	
8	0F043596202561788EA13D5	1023355	2013-12-02 10:43:39.117000000	
9	0F043596202561788EA13D5	1023355	2013-12-02 10:43:39.117000000	

	CreditGrade	Term	LoanStatus	ClosedDate	BorrowerAPR	\
0	C	36	Completed	2009-08-14 00:00:00	0.16516	
1	NaN	36	Current	NaN	0.12016	
2	HR	36	Completed	2009-12-17 00:00:00	0.28269	
3	NaN	36	Current	NaN	0.12528	
4	NaN	36	Current	NaN	0.24614	
5	NaN	60	Current	NaN	0.15425	
6	NaN	36	Current	NaN	0.31032	
7	NaN	36	Current	NaN	0.23939	
8	NaN	36	Current	NaN	0.07620	
9	NaN	36	Current	NaN	0.07620	

	BorrowerRate	LenderYield	EstimatedEffectiveYield	EstimatedLoss	\
0	0.1580	0.1380	NaN	NaN	
1	0.0920	0.0820	0.07960	0.0249	
2	0.2750	0.2400	NaN	NaN	
3	0.0974	0.0874	0.08490	0.0249	
4	0.2085	0.1985	0.18316	0.0925	
5	0.1314	0.1214	0.11567	0.0449	
6	0.2712	0.2612	0.23820	0.1275	
7	0.2019	0.1919	0.17830	0.0799	
8	0.0629	0.0529	0.05221	0.0099	
9	0.0629	0.0529	0.05221	0.0099	

	EstimatedReturn	ProsperRating (numeric)	ProsperRating (Alpha)	\
0	NaN	NaN	NaN	
1	0.05470	6.0	A	
2	NaN	NaN	NaN	
3	0.06000	6.0	A	
4	0.09066	3.0	D	
5	0.07077	5.0	B	
6	0.11070	2.0	E	
7	0.09840	4.0	C	
8	0.04231	7.0	AA	
9	0.04231	7.0	AA	

	ProsperScore	ListingCategory (numeric)	BorrowerState	Occupation \
0	NaN	0	CO	Other
1	7.0	2	CO	Professional
2	NaN	0	GA	Other
3	9.0	16	GA	Skilled Labor
4	4.0	2	MN	Executive
5	10.0	1	NM	Professional
6	2.0	1	KS	Sales - Retail
7	4.0	2	CA	Laborer
8	9.0	7	IL	Food Service
9	11.0	7	IL	Food Service

	EmploymentStatus	EmploymentStatusDuration	IsBorrowerHomeowner \
0	Self-employed	2.0	True
1	Employed	44.0	False
2	Not available	NaN	False
3	Employed	113.0	True
4	Employed	44.0	True
5	Employed	82.0	True
6	Employed	172.0	False
7	Employed	103.0	False
8	Employed	269.0	True
9	Employed	269.0	True

	CurrentlyInGroup	GroupKey	DateCreditPulled \
0	True	NaN	2007-08-26 18:41:46.780000000
1	False	NaN	2014-02-27 08:28:14
2	True	783C3371218786870A73D20	2007-01-02 14:09:10.060000000
3	False	NaN	2012-10-22 11:02:32
4	False	NaN	2013-09-14 18:38:44
5	False	NaN	2013-12-14 08:26:40
6	False	NaN	2013-04-12 09:52:53
7	False	NaN	2013-05-05 06:49:25
8	False	NaN	2013-12-02 10:43:39
9	False	NaN	2013-12-02 10:43:39

	CreditScoreRangeLower	CreditScoreRangeUpper	FirstRecordedCreditLine \
0	640.0	659.0	2001-10-11 00:00:00
1	680.0	699.0	1996-03-18 00:00:00
2	480.0	499.0	2002-07-27 00:00:00
3	800.0	819.0	1983-02-28 00:00:00
4	680.0	699.0	2004-02-20 00:00:00
5	740.0	759.0	1973-03-01 00:00:00
6	680.0	699.0	2000-09-29 00:00:00
7	700.0	719.0	1999-02-25 00:00:00
8	820.0	839.0	1993-04-01 00:00:00

9 820.0 839.0 1993-04-01 00:00:00

	CurrentCreditLines	OpenCreditLines	TotalCreditLinespast7years	\
0	5.0	4.0	12.0	
1	14.0	14.0	29.0	
2	NaN	NaN	3.0	
3	5.0	5.0	29.0	
4	19.0	19.0	49.0	
5	21.0	17.0	49.0	
6	10.0	7.0	20.0	
7	6.0	6.0	10.0	
8	17.0	16.0	32.0	
9	17.0	16.0	32.0	

	OpenRevolvingAccounts	OpenRevolvingMonthlyPayment	InquiriesLast6Months	\
0	1	24.0	3.0	
1	13	389.0	3.0	
2	0	0.0	0.0	
3	7	115.0	0.0	
4	6	220.0	1.0	
5	13	1410.0	0.0	
6	6	214.0	0.0	
7	5	101.0	3.0	
8	12	219.0	1.0	
9	12	219.0	1.0	

	TotalInquiries	CurrentDelinquencies	AmountDelinquent	\
0	3.0	2.0	472.0	
1	5.0	0.0	0.0	
2	1.0	1.0	NaN	
3	1.0	4.0	10056.0	
4	9.0	0.0	0.0	
5	2.0	0.0	0.0	
6	0.0	0.0	0.0	
7	16.0	0.0	0.0	
8	6.0	0.0	0.0	
9	6.0	0.0	0.0	

	DelinquenciesLast7Years	PublicRecordsLast10Years	\
0	4.0	0.0	
1	0.0	1.0	
2	0.0	0.0	
3	14.0	0.0	
4	0.0	0.0	
5	0.0	0.0	
6	0.0	0.0	
7	0.0	1.0	

8	0.0	0.0
9	0.0	0.0

	PublicRecordsLast12Months	RevolvingCreditBalance	BankcardUtilization \
0	0.0	0.0	0.00
1	0.0	3989.0	0.21
2	NaN	NaN	NaN
3	0.0	1444.0	0.04
4	0.0	6193.0	0.81
5	0.0	62999.0	0.39
6	0.0	5812.0	0.72
7	0.0	1260.0	0.13
8	0.0	9906.0	0.11
9	0.0	9906.0	0.11

	AvailableBankcardCredit	TotalTrades	TradesNeverDelinquent (percentage) \
0	1500.0	11.0	0.81
1	10266.0	29.0	1.00
2	NaN	NaN	NaN
3	30754.0	26.0	0.76
4	695.0	39.0	0.95
5	86509.0	47.0	1.00
6	1929.0	16.0	0.68
7	2181.0	10.0	0.80
8	77696.0	29.0	1.00
9	77696.0	29.0	1.00

	TradesOpenedLast6Months	DebtToIncomeRatio	IncomeRange \
0	0.0	0.17	\$25,000-49,999
1	2.0	0.18	\$50,000-74,999
2	NaN	0.06	Not displayed
3	0.0	0.15	\$25,000-49,999
4	2.0	0.26	\$100,000+
5	0.0	0.36	\$100,000+
6	0.0	0.27	\$25,000-49,999
7	0.0	0.24	\$25,000-49,999
8	1.0	0.25	\$25,000-49,999
9	1.0	0.25	\$25,000-49,999

	IncomeVerifiable	StatedMonthlyIncome	LoanKey \
0	True	3083.333333	E33A3400205839220442E84
1	True	6125.000000	9E3B37071505919926B1D82
2	True	2083.333333	6954337960046817851BCB2
3	True	2875.000000	A0393664465886295619C51
4	True	9583.333333	A180369302188889200689E
5	True	8333.333333	C3D63702273952547E79520
6	True	2083.333333	CE963680102927767790520

7	True	3355.750000	0C87368108902149313D53B
8	True	3333.333333	02163700809231365A56A1C
9	True	3333.333333	02163700809231365A56A1C

	TotalProsperLoans	TotalProsperPaymentsBilled	OnTimeProsperPayments \
0	NaN	NaN	NaN
1	NaN	NaN	NaN
2	NaN	NaN	NaN
3	NaN	NaN	NaN
4	1.0	11.0	11.0
5	NaN	NaN	NaN
6	NaN	NaN	NaN
7	NaN	NaN	NaN
8	NaN	NaN	NaN
9	NaN	NaN	NaN

	ProsperPaymentsLessThanOneMonthLate	ProsperPaymentsOneMonthPlusLate \
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN
4	0.0	0.0
5	NaN	NaN
6	NaN	NaN
7	NaN	NaN
8	NaN	NaN
9	NaN	NaN

	ProsperPrincipalBorrowed	ProsperPrincipalOutstanding \
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN
4	11000.0	9947.9
5	NaN	NaN
6	NaN	NaN
7	NaN	NaN
8	NaN	NaN
9	NaN	NaN

	ScorexChangeAtTimeOfListing	LoanCurrentDaysDelinquent \
0	NaN	0
1	NaN	0
2	NaN	0
3	NaN	0
4	NaN	0
5	NaN	0

6	NaN	0
7	NaN	0
8	NaN	0
9	NaN	0

	LoanFirstDefaultedCycleNumber	LoanMonthsSinceOrigination	LoanNumber \
0	NaN	78	19141
1	NaN	0	134815
2	NaN	86	6466
3	NaN	16	77296
4	NaN	6	102670
5	NaN	3	123257
6	NaN	11	88353
7	NaN	10	90051
8	NaN	3	121268
9	NaN	3	121268

	LoanOriginalAmount	LoanOriginationDate	LoanOriginationQuarter \
0	9425	2007-09-12 00:00:00	Q3 2007
1	10000	2014-03-03 00:00:00	Q1 2014
2	3001	2007-01-17 00:00:00	Q1 2007
3	10000	2012-11-01 00:00:00	Q4 2012
4	15000	2013-09-20 00:00:00	Q3 2013
5	15000	2013-12-24 00:00:00	Q4 2013
6	3000	2013-04-18 00:00:00	Q2 2013
7	10000	2013-05-13 00:00:00	Q2 2013
8	10000	2013-12-12 00:00:00	Q4 2013
9	10000	2013-12-12 00:00:00	Q4 2013

	MemberKey	MonthlyLoanPayment	LP_CustomerPayments \
0	1F3E3376408759268057EDA	330.43	11396.14
1	1D13370546739025387B2F4	318.93	0.00
2	5F7033715035555618FA612	123.32	4186.63
3	9ADE356069835475068C6D2	321.45	5143.20
4	36CE356043264555721F06C	563.97	2819.85
5	874A3701157341738DE458F	342.37	679.34
6	AA4535764146102879D5959	122.67	1226.70
7	737F347089545035681C074	372.60	3353.40
8	49A53699682291323D04D66	305.54	611.08
9	49A53699682291323D04D66	305.54	611.08

	LP_CustomerPrincipalPayments	LP_InterestandFees	LP_ServiceFees \
0	9425.00	1971.14	-133.18
1	0.00	0.00	0.00
2	3001.00	1185.63	-24.20
3	4091.09	1052.11	-108.01
4	1563.22	1256.63	-60.27

5	351.89	327.45	-25.33
6	604.25	622.45	-22.95
7	1955.89	1397.51	-69.21
8	505.58	105.50	-16.77
9	505.58	105.50	-16.77

	LP_CollectionFees	LP_GrossPrincipalLoss	LP_NetPrincipalLoss	\
0	0.0	0.0	0.0	
1	0.0	0.0	0.0	
2	0.0	0.0	0.0	
3	0.0	0.0	0.0	
4	0.0	0.0	0.0	
5	0.0	0.0	0.0	
6	0.0	0.0	0.0	
7	0.0	0.0	0.0	
8	0.0	0.0	0.0	
9	0.0	0.0	0.0	

	LP_NonPrincipalRecoverypayments	PercentFunded	Recommendations	\
0	0.0	1.0	0	
1	0.0	1.0	0	
2	0.0	1.0	0	
3	0.0	1.0	0	
4	0.0	1.0	0	
5	0.0	1.0	0	
6	0.0	1.0	0	
7	0.0	1.0	0	
8	0.0	1.0	0	
9	0.0	1.0	0	

	InvestmentFromFriendsCount	InvestmentFromFriendsAmount	Investors
0	0	0.0	258
1	0	0.0	1
2	0	0.0	41
3	0	0.0	158
4	0	0.0	20
5	0	0.0	1
6	0	0.0	1
7	0	0.0	1
8	0	0.0	1
9	0	0.0	1

```
[5]: # Getting basic informations about the dataset
prosperLoanData.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	float64
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(50), int64(11), object(17)

memory usage: 68.1+ MB

Univariate Exploration of Data

Question 1: What is the trend of the LoanStatus variable ?

We'll start looking at the trend of the LoanStatus variable.

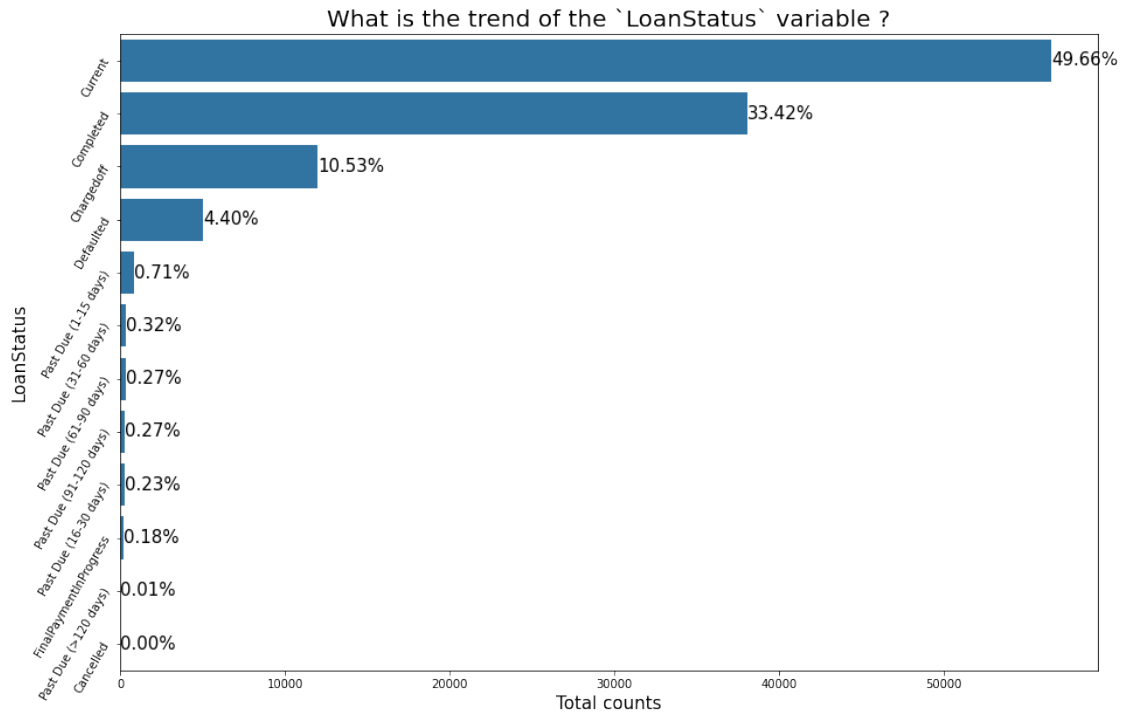
LoanStatus : The current status of the loan: Cancelled, Chargedoff, Completed, Current, Defaulted, FinalPaymentInProgress, PastDue. The PastDue status will be accompanied by a delinquency bucket.

```
[6]: def special_barplot(data, colname):
    """This fuction will help to plot bar charts with percentage of each bar
    ↪overall the data for better insights."""
    # Getting order from the greatest to the lowest
    prosperLoanData = data
    col_name = colname

    type_counts = prosperLoanData[col_name].value_counts()
    order = type_counts.index
    n_total = type_counts.sum()
    color = sns.color_palette()[0]
    plt.figure(figsize = (15, 10))
    f = sns.countplot(data=prosperLoanData, y=col_name, order=order,
    ↪color=color);
    try:
        for index in range(type_counts.shape[0]):
            count = type_counts[index]
            percent_str = f'{100 * count / n_total:0.2f}%'
            plt.text(x=count+3, y=index, s=percent_str, va='center',
    ↪fontdict={'fontsize': 15});

            xticks = list(type_counts.values)[::-1]
            #print(xticks)
            xlabels = [f'{elt:0.1f}' for elt in xticks]
            yticks = list(type_counts.index)
            #print(yticks)
            ylabels = [f'{elt}' for elt in yticks]
            #plt.xticks(ticks=xticks, labels=xlabels, rotation=30)
            plt.yticks(ticks=np.arange(len(yticks)), labels=ylabels, rotation=60)
    except Exception as e :
        #print(e)
        pass
```

```
[7]: special_barplot(data=prosperLoanData, colname='LoanStatus')
    #f.set_xticklabels(labels = list(type_counts.values), minor=True, rotation=360)
    plt.title('What is the trend of the `LoanStatus` variable ?',
    ↪fontdict={'fontsize': 20});
    plt.xlabel(xlabel='Total counts', fontdict={'fontsize': 15});
    plt.ylabel(ylabel='LoanStatus', fontdict={'fontsize': 15});
```

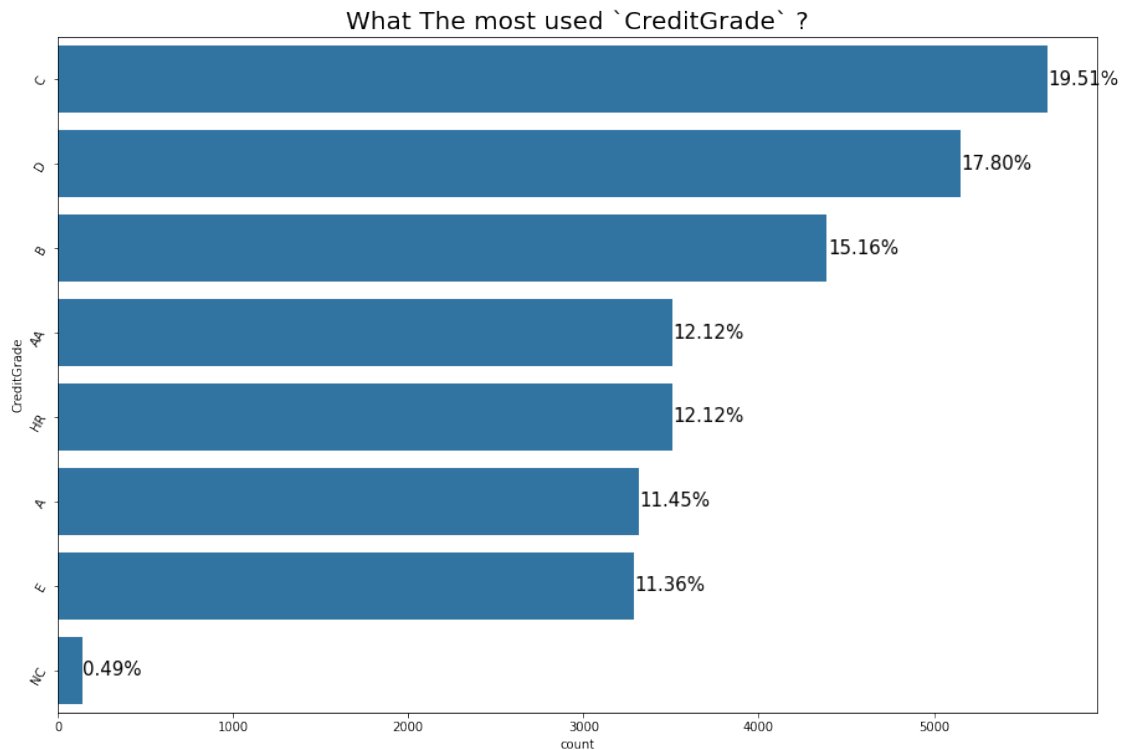


- As we can see, most of the loans are ongoing 49.66%, some are already completed 33.42% and the less are chargedoff 10.53%. All the other statuses are not so relevant because they have little percentages.

Question 2: What The most used CreditGrade ?

CreditGrade : The Credit rating that was assigned at the time the listing went live. Applicable for listings pre-2009 period and will only be populated for those listings.

```
[8]: special_barplot(data=prosperLoanData, colname='CreditGrade');
plt.title('What The most used `CreditGrade` ?', fontdict={'fontsize': 20});
```

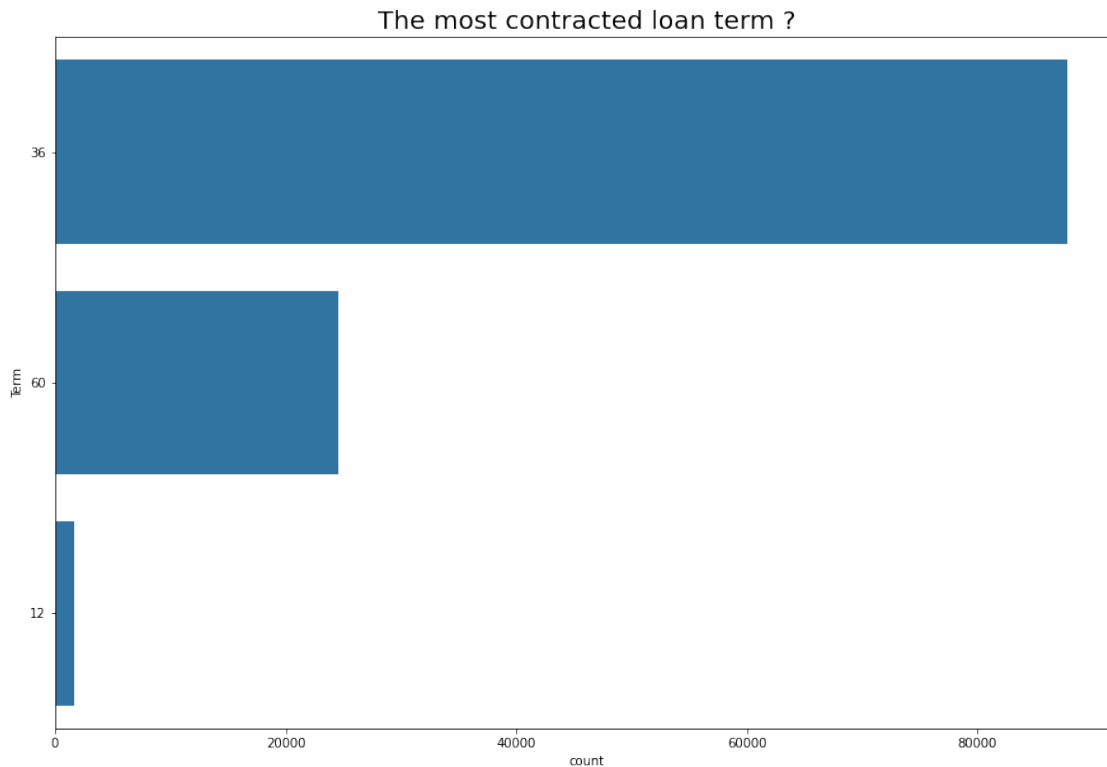


- The grade C is the most represented with 19.51%
- The grade D is the second most represented with 17.80%
- The grade B is the third most represented with 15.16%

Question 3: The most contracted loan Term ?

Term : The length of the loan expressed in months.

```
[9]: special_barplot(data=prosperLoanData, colname='Term');
plt.title('The most contracted loan term ?', fontdict={'fontsize': 20});
```



The most contracted loan term is the length of 36 months with more than 80000 contracts.

Question 4: BorrowerAPR distribution

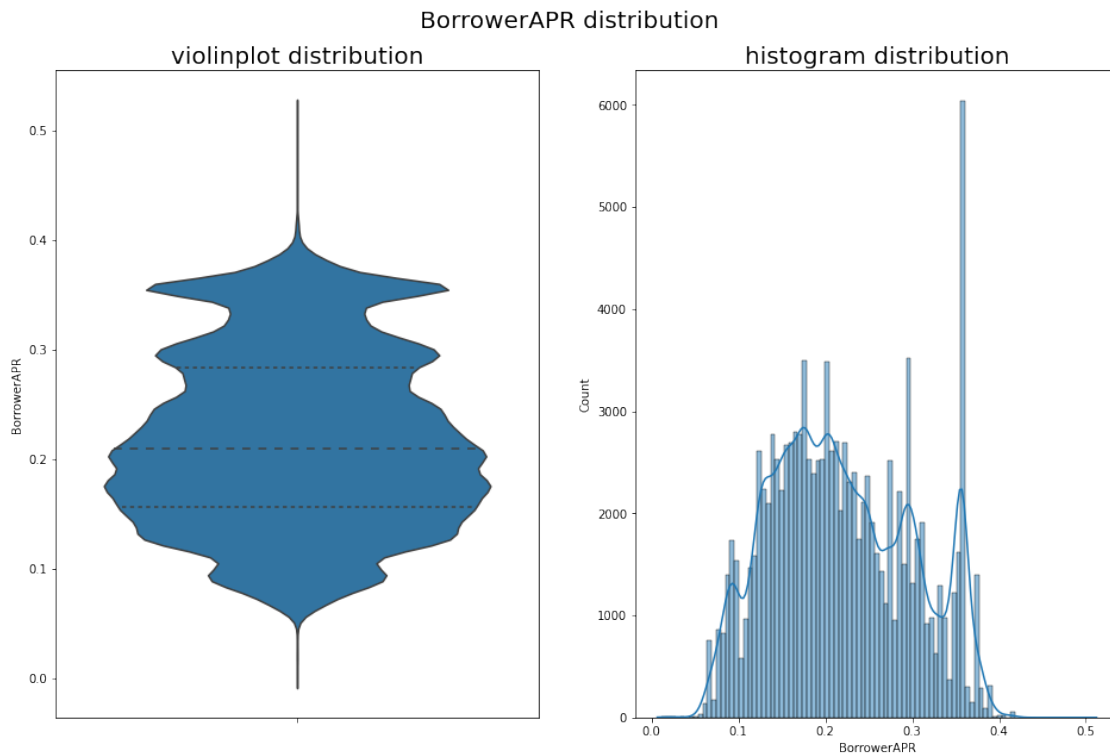
BorrowerAPR : The Borrower's Annual Percentage Rate (APR) for the loan.

```
[10]: prosperLoanData['BorrowerAPR'].describe()
```

```
[10]: count    113912.000000
      mean      0.218828
      std       0.080364
      min       0.006530
      25%       0.156290
      50%       0.209760
      75%       0.283810
      max       0.512290
      Name: BorrowerAPR, dtype: float64
```

```
[11]: plt.figure(figsize = (16, 10));
      plt.suptitle('BorrowerAPR distribution', x=0.5, y=0.95, fontproperties={'size': 20});
      plt.subplot(1, 2, 1)
      sns.violinplot(data=prosperLoanData, y='BorrowerAPR', inner='quartile');
      plt.title('violinplot distribution', fontdict={'fontsize': 20});
```

```
plt.subplot(1, 2, 2)
sns.histplot(data=prosperLoanData, x='BorrowerAPR', kde=True);
plt.title('histogram distribution', fontdict={'fontsize': 20});
```

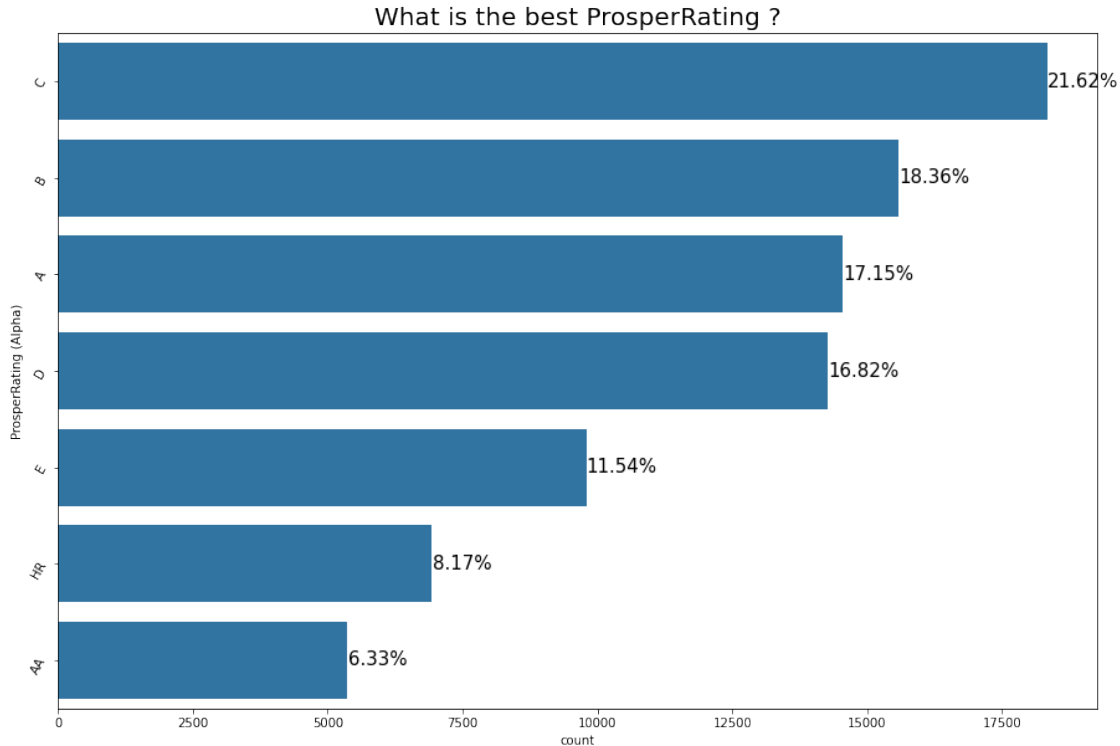


- The distribution is slightly normal
- With a standard deviation of 0.080364 the median value of the BorrowerAPR is 0.209760 and it is closest to the mean 0.218828. It means that the BorrowerAPR is really good for loans.

Question 5: ProsperRating distribution

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

```
[12]: special_barplot(data=prosperLoanData, colname='ProsperRating (Alpha)');
plt.title('What is the best ProsperRating ?', fontdict={'fontsize': 20});
```

The best ProsperRating is * The grade C is the most represented with 21.62% and followed by * The grade B is the second most represented with 18.36% and followed by * The grade A is the third most represented with 17.15%

Question 6: ProsperScore distribution

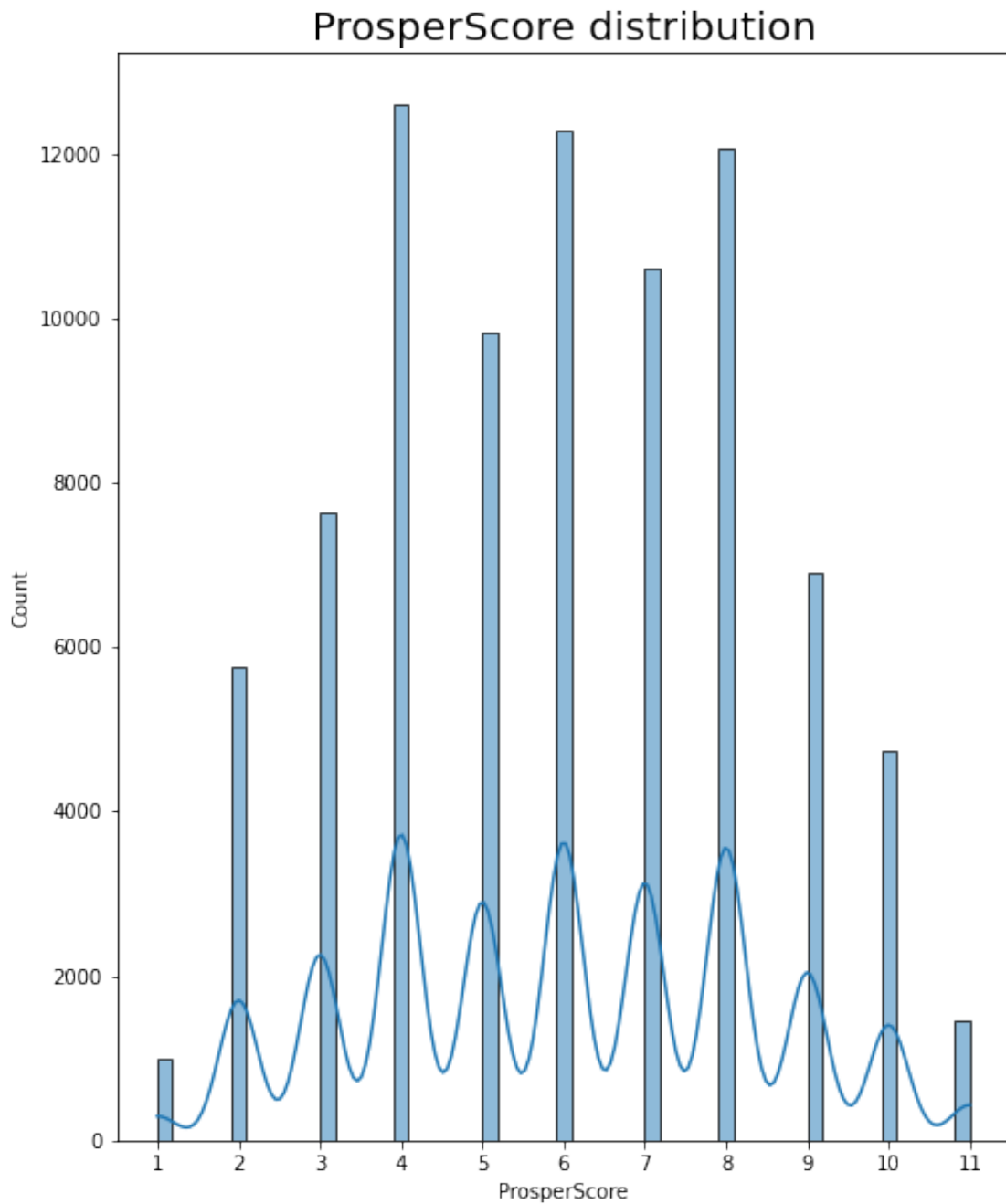
ProsperScore : A custom risk score built using historical Prosper data. The score ranges from 1-10, with 10 being the best, or lowest risk score. Applicable for loans originated after July 2009.

```
[13]: prosperLoanData['ProsperScore'].describe()
```

```
[13]: count      84853.000000
      mean         5.950067
      std          2.376501
      min          1.000000
      25%          4.000000
      50%          6.000000
      75%          8.000000
      max         11.000000
      Name: ProsperScore, dtype: float64
```

```
[14]: plt.figure(figsize = (8, 10));
      sns.histplot(data=prosperLoanData, x='ProsperScore', kde=True);
      plt.xticks(ticks=range(1, 11 + 1), labels=[f'{i}' for i in range(1, 11 + 1)])
```

```
plt.title('ProsperScore distribution', fontdict={'fontsize': 20});
```



- The most highest ProsperScore are 4 then 6 then 8. More than 12000 loans are related to houses with such ProsperScore.

Question 7: ListingCategory distribution

ListingCategory : The category of the listing that the borrower selected when posting their

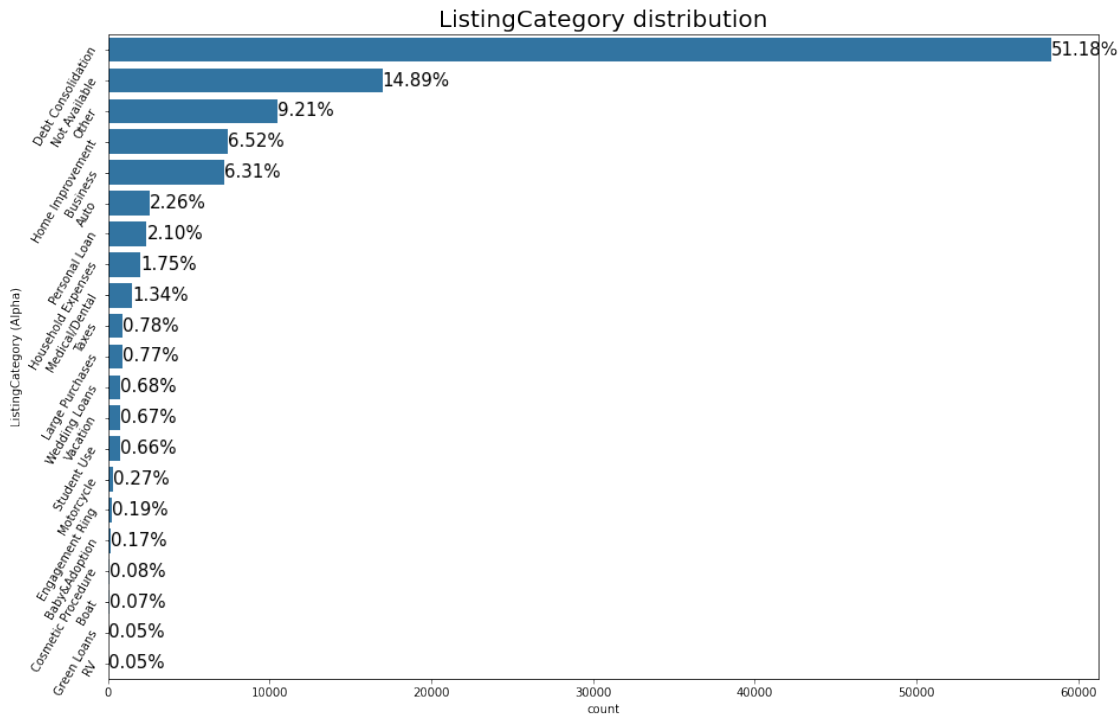
listing: 0 - Not Available, 1 - Debt Consolidation, 2 - Home Improvement, 3 - Business, 4 - Personal Loan, 5 - Student Use, 6 - Auto, 7- Other, 8 - Baby&Adoption, 9 - Boat, 10 - Cosmetic Procedure, 11 - Engagement Ring, 12 - Green Loans, 13 - Household Expenses, 14 - Large Purchases, 15 - Medical/Dental, 16 - Motorcycle, 17 - RV, 18 - Taxes, 19 - Vacation, 20 - Wedding Loans

```
[15]: prosperLoanData['ListingCategory (numeric)'].describe()
```

```
[15]: count      113937.000000
      mean        2.774209
      std         3.996797
      min         0.000000
      25%         1.000000
      50%         1.000000
      75%         3.000000
      max         20.000000
      Name: ListingCategory (numeric), dtype: float64
```

```
[16]: # Creating a dict for ListingCategory
ListingCategoryDict = {0 : 'Not Available', 1 : 'Debt Consolidation', 2 : 'Home_
↳Improvement', 3 : 'Business', 4 : 'Personal Loan', 5 : 'Student Use', 6 :_
↳'Auto', 7 : 'Other', 8 : 'Baby&Adoption', 9 : 'Boat', 10 : 'Cosmetic_
↳Procedure', 11 : 'Engagement Ring', 12 : 'Green Loans', 13 : 'Household_
↳Expenses', 14 : 'Large Purchases', 15 : 'Medical/Dental', 16 : 'Motorcycle',_
↳17 : 'RV', 18 : 'Taxes', 19 : 'Vacation', 20 : 'Wedding Loans'}
```

```
[17]: #plt.figure(figsize = (8, 10));
# Creating a variable for coresponding listing category numbers to their_
↳meaning.
prosperLoanData['ListingCategory (Alpha)'] = prosperLoanData['ListingCategory_
↳(numeric)'].apply(lambda val: ListingCategoryDict[val]);
special_barplot(data=prosperLoanData, colname='ListingCategory (Alpha)');
plt.title('ListingCategory distribution', fontdict={'fontsize': 20});
```



Most of the loans are for Debt Consolidation. Really inspiring.

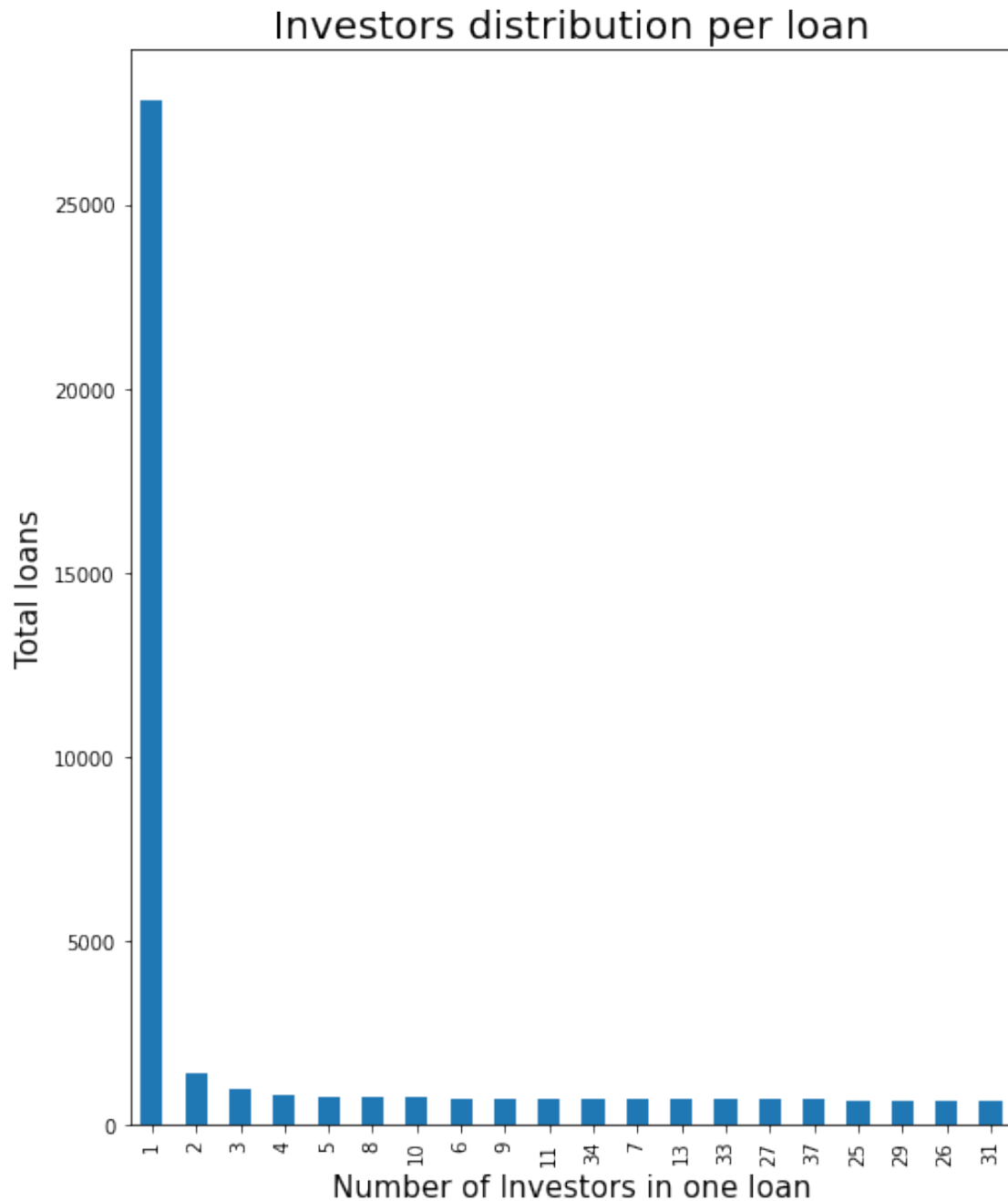
Question 8: Investors distribution

Investors : The number of investors that funded the loan.

```
[18]: prosperLoanData['Investors'].value_counts().nlargest(5)
```

```
[18]: 1    27814
      2    1386
      3     991
      4     827
      5     753
      Name: Investors, dtype: int64
```

```
[19]: plt.figure(figsize = (8, 10));
      prosperLoanData['Investors'].value_counts().nlargest(20).plot(kind='bar');
      plt.title('Investors distribution per loan', fontdict={'fontsize': 20});
      plt.xlabel(xlabel='Number of Investors in one loan', fontdict={'fontsize': 15});
      plt.ylabel(ylabel='Total loans', fontdict={'fontsize': 15});
```



Almost all loans are funded by one investor. (More than 25000 loans.)

Question 9: EmploymentStatus, Occupation, EmploymentStatusDuration and IsBorrowerHomeowner distribution

EmploymentStatus : The employment status of the borrower at the time they posted the listing.

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

EmploymentStatusDuration : The length in months of the employment status at the time the listing was created.

IsBorrowerHomeowner : A Borrower will be classified as a homeowner if they have a mortgage on their credit profile or provide documentation confirming they are a homeowner.

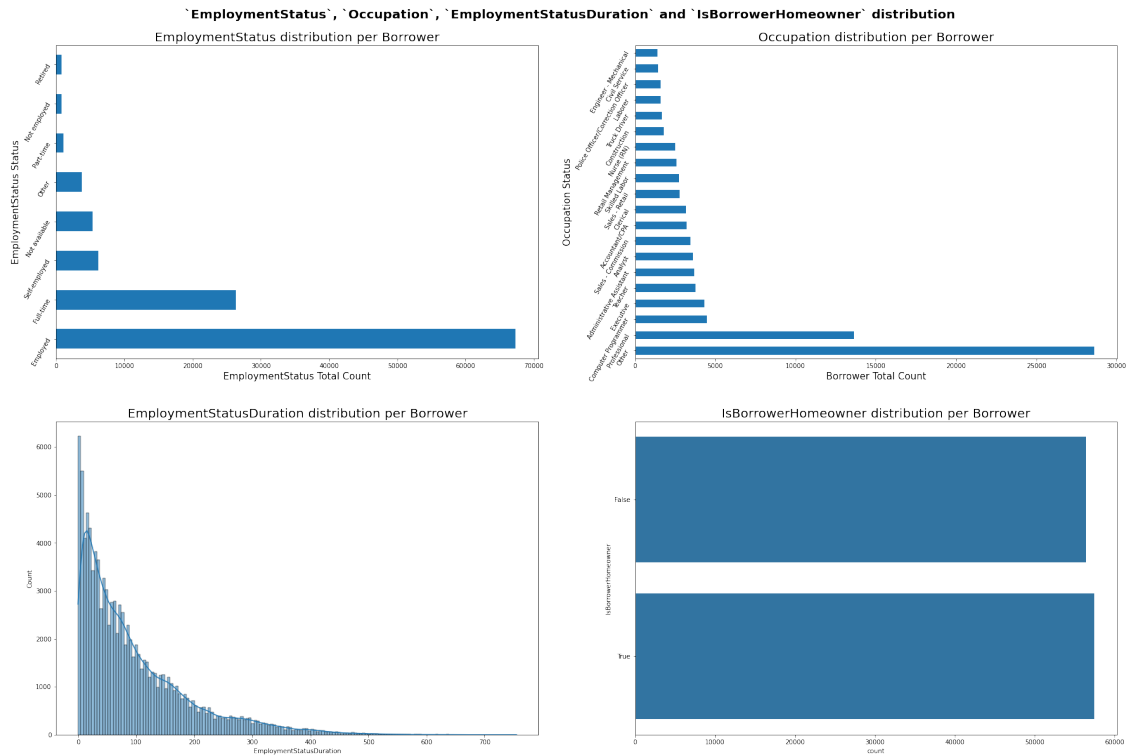
```
[20]: plt.figure(figsize = (30, 20));
plt.suptitle(`EmploymentStatus`, `Occupation`, `EmploymentStatusDuration` and
↳ `IsBorrowerHomeowner` distribution', fontsize=20, fontweight='bold', x=0.5,
↳ y=0.92)

plt.subplot(2, 2, 1)
prosperLoanData['EmploymentStatus'].value_counts().nlargest(20).
↳ plot(kind='barh');
plt.title('EmploymentStatus distribution per Borrower', fontdict={'fontsize':
↳ 20});
plt.xlabel(xlabel='EmploymentStatus Total Count', fontdict={'fontsize': 15});
plt.ylabel(ylabel='EmploymentStatus Status', fontdict={'fontsize': 15});
plt.yticks(rotation=60);

plt.subplot(2, 2, 2)
prosperLoanData['Occupation'].value_counts().nlargest(20).plot(kind='barh');
plt.title('Occupation distribution per Borrower', fontdict={'fontsize': 20});
plt.xlabel(xlabel='Borrower Total Count', fontdict={'fontsize': 15});
plt.ylabel(ylabel='Occupation Status', fontdict={'fontsize': 15});
plt.yticks(rotation=60);

plt.subplot(2, 2, 3)
sns.histplot(data=prosperLoanData, x='EmploymentStatusDuration', kde=True);
plt.title('EmploymentStatusDuration distribution per Borrower',
↳ fontdict={'fontsize': 20});

plt.subplot(2, 2, 4);
sns.countplot(data=prosperLoanData, y='IsBorrowerHomeowner', color=sns.
↳ color_palette()[0]);
plt.title('IsBorrowerHomeowner distribution per Borrower', fontdict={'fontsize':
↳ 20});
```

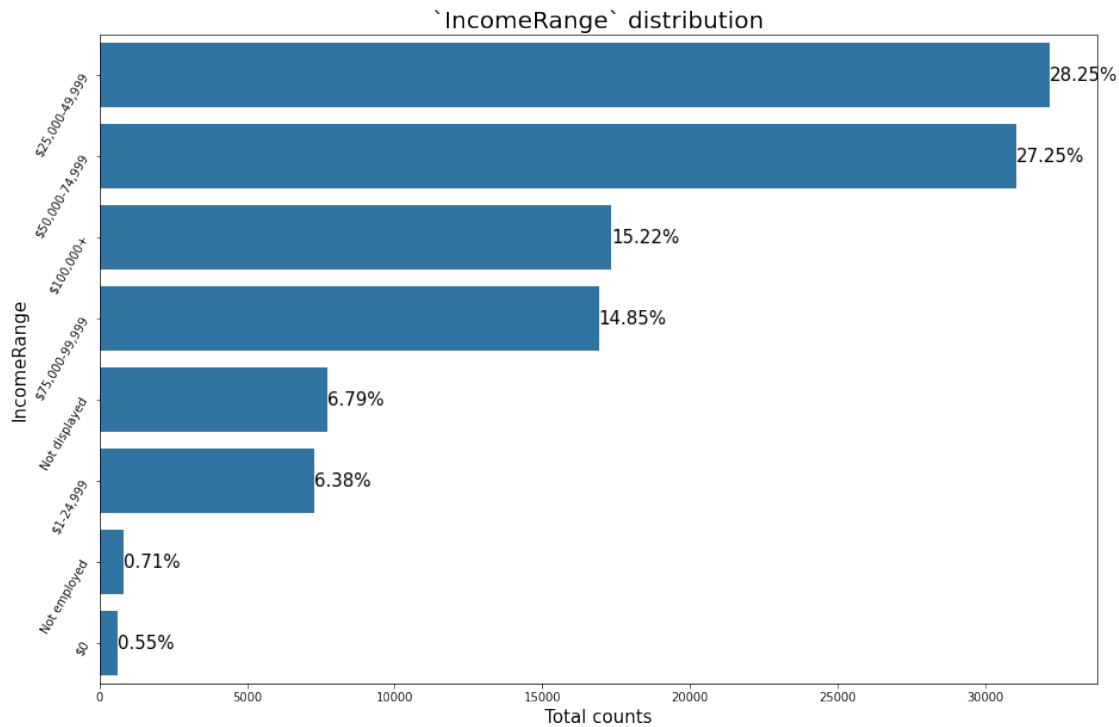


- Most of the Borrowers are employed(Employed, Full-time or Self-employed)
- Most of the Borrowers don't specify their Occupations.
 - More than 10000 Borrowers are **Professionals**
 - Around 5000 Borrowers are **Computer Programmer** and **Executive**
 - Around 4000 Borrowers are **Teacher**, **Administrative Assistant** and **Analyst**
- Around 50% of Borrowers are either Homeowners or not Homeowners.
- Looks like there needs to be an employee for some time (about 100 months) in order to the a borrower, assuming your employment come with high revenues. (to be able to be Homeowner.)

Question 10: **IncomeRange** distribution

IncomeRange : The income range of the borrower at the time the listing was created.

```
[21]: special_barplot(data=prosperLoanData, colname='IncomeRange')
      #f.set_xticklabels(labels = list(type_counts.values), minor=True, rotation=360)
      plt.title(`IncomeRange` distribution', fontdict={'fontsize': 20});
      plt.xlabel(xlabel='Total counts', fontdict={'fontsize': 15});
      plt.ylabel(ylabel='IncomeRange', fontdict={'fontsize': 15});
```



- Clearly you must have a minimum `IncomeRange` of \$25,000 to be sure to have a loan.

Univariate Exploration Summary

Question 1: What is the trend of the `LoanStatus` variable ?

- As we can see, most of the loans are ongoing 49.66%, some are already completed 33.42% and the less are chargedoff 10.53%. All the other statuses are not so relevant because they have little percentages.

Question 2: What The most used `CreditGrade` ?

- The grade C is the most represented with 19.51%
- The grade D is the second most represented with 17.80%
- The grade B is the third most represented with 15.16%

Question 3: The most contracted loan `Term` ?

- The most contracted loan term is the length of 36 months with more than 80000 contracts.

Question 4: `BorrowerAPR` distribution

- The distribution is slightly normal
- With a standard deviation of 0.080364 the median value of the `BorrowerAPR` is 0.209760 and it is closest to the mean 0.218828. It means that the `BorrowerAPR` is really good for loans.

Question 5: What is the best ProsperRating ? The best ProsperRating is * The grade C is the most represented with 21.62% and followed by * The grade B is the second most represented with 18.36% and followed by * The grade A is the third most represented with 17.15%

Question 6: ProsperScore distribution

- The most highest ProsperScore are 4 then 6 then 8. More than 12000 loans are related to houses with such ProsperScore.

Question 7: ListingCategory distribution

- Most of the loans are for Debt Consolidation. Really inspiring.

Question 8: Investors distribution

- Almost all loans are funded by one investor. (More than 25000 loans.)

Question 9: EmploymentStatus, Occupation, EmploymentStatusDuration and IsBorrowerHomeowner distribution

- Most of the Borrowers are employed (Employed, Full-time or Self-employed)
- Most of the Borrowers don't specify their Occupations.
 - More than 10000 Borrowers are Professionals
 - Around 5000 Borrowers are Computer Programmer and Executive
 - Around 4000 Borrowers are Teacher, Administrative Assistant and Analyst
- Around 50% of Borrowers are either Homeowners or not Homeowners.
- Looks like there needs to be an employee for some time (about 100 months) in order to be a borrower, assuming your employment comes with high revenues. (to be able to be Homeowner.)

Question 10: IncomeRange distribution

- Clearly you must have a minimum IncomeRange of \$25,000 to be sure to have a loan.

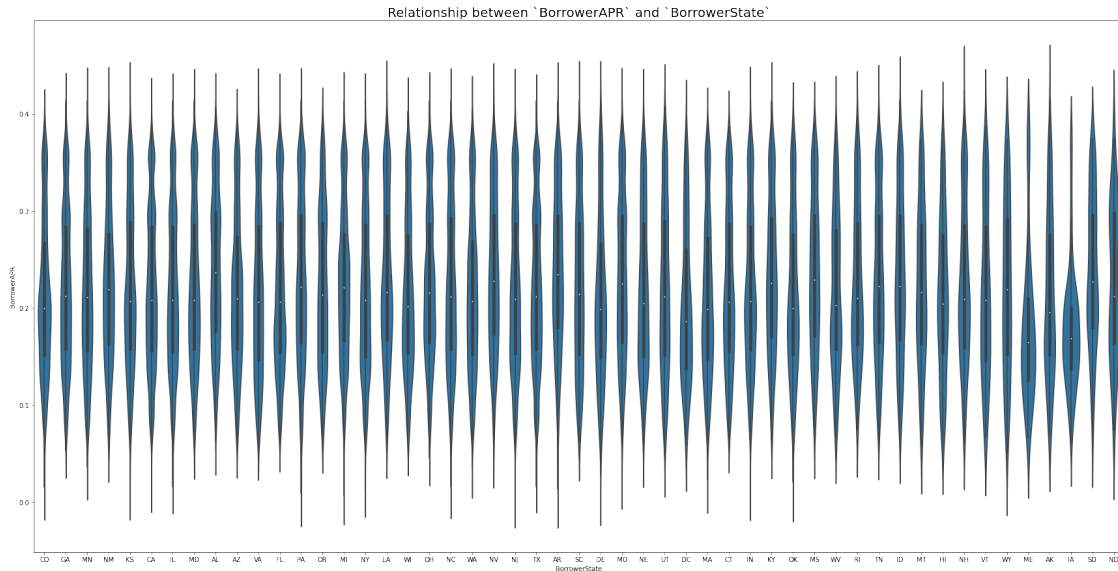
Bivariate Exploration of Data

Question 1: Relationship between BorrowerAPR and BorrowerState

BorrowerAPR : The Borrower's Annual Percentage Rate (APR) for the loan.

BorrowerState : The two letter abbreviation of the state of the address of the borrower at the time the Listing was created.

```
[22]: color = sns.color_palette()[0]
plt.figure(figsize = (30, 15));
sns.violinplot(data=prosperLoanData, x='BorrowerState', y='BorrowerAPR',
               inner='box', color=color);
plt.title('Relationship between `BorrowerAPR` and `BorrowerState`',
          fontdict={'fontsize': 20});
```



Per each `BorrowerState`, the `BorrowerAPR` are slightly the same. There is not much difference of `BorrowerAPR` according to different `BorrowerState`.

Question 2: Relationship between `ProsperRating (Alpha)` and `ProsperScore`

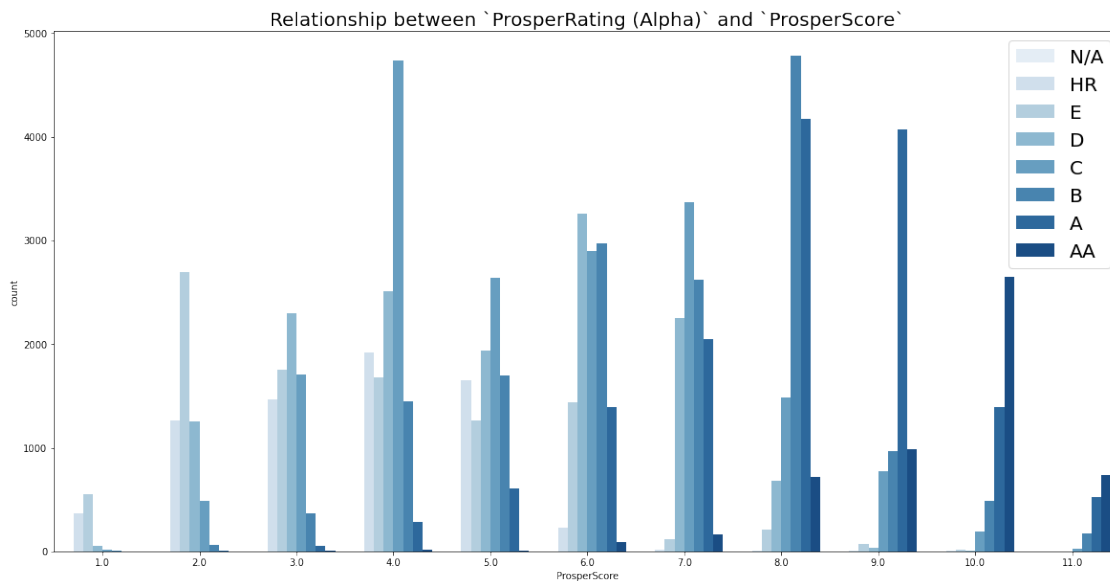
`ProsperRating (Alpha)` : The Prosper Rating assigned at the time the listing was created between AA - HR. Applicable for loans originated after July 2009.

`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.

```
[23]: # Let's convert ProsperRating (Alpha) to a CategoricalDtype
classes = ['N/A', 'HR', 'E', 'D', 'C', 'B', 'A', 'AA']
# Creating Category
cat_classes = pd.api.types.CategoricalDtype(categories=classes, ordered=True)
# Converting to CategoricalDtype
prosperLoanData['ProsperRating (Alpha)'] = prosperLoanData['ProsperRating_
↳(Alpha)'].astype(cat_classes)
# Testing
prosperLoanData['ProsperRating (Alpha)'].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 113937 entries, 0 to 113936
Series name: ProsperRating (Alpha)
Non-Null Count  Dtype
-----
84853 non-null  category
dtypes: category(1)
memory usage: 111.7 KB
```

```
[24]: color = sns.color_palette()[0]
plt.figure(figsize = (20, 10));
sns.countplot(data=prosperLoanData, x='ProsperScore', hue='ProsperRating_
↳(Alpha)', palette='Blues');
plt.legend(fontsize=20, loc='upper right');
plt.title('Relationship between `ProsperRating (Alpha)` and `ProsperScore`',
↳fontdict={'fontsize': 20});
```



- Loans with high ProsperScore (like 8, 9 or 10) are associated with high ProsperRating (like B, A or AA)
- Loans with median ProsperScore (like 5, 6 or 7) are associated with low or median ProsperRating (like E, D or C)
- Loans with low ProsperScore (like 1, 2 or 4) are associated with low ProsperRating (like HR, E or D)

Question 3: Relationship between ProsperScore and ProsperRating (numeric)

ProsperRating (numeric) : The Prosper Rating assigned at the time the listing was created: 0 - N/A, 1 - HR, 2 - E, 3 - D, 4 - C, 5 - B, 6 - A, 7 - AA. Applicable for loans originated after July 2009.

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.

```
[ ]: color = sns.color_palette()[0]
plt.figure(figsize = (20, 10));
df = prosperLoanData[['ProsperScore', 'ProsperRating (numeric)']].corr()
sns.heatmap(data=df, annot=True)
```

```
#sns.regplot(data=prosperLoanData, x='ProsperScore', y='ProsperRating_
↳(numeric)', x_jitter=0.2, y_jitter=0.2, truncate=False, scatter_kws={'alpha':
↳ 1/30});
plt.title('Relationship between `ProsperScore` and `ProsperRating (numeric)`',
↳fontdict={'fontsize': 20});
```

- The higher the ProsperScore, the higher the ProsperRating (numeric)
- There is a positive correlation between ProsperScore and ProsperRating (numeric)

Question 4: Relationship between Term and BorrowerState

Term : The length of the loan expressed in months.

BorrowerState : The two letter abbreviation of the state of the address of the borrower at the time the Listing was created.

```
[ ]: plt.figure(figsize = (30, 15));
sns.countplot(data=prosperLoanData, x='Term', hue='BorrowerState',
↳palette='viridis');
plt.title('Relationship between `Term` and `BorrowerState`',
↳fontdict={'fontsize': 20});
```

- Most of the loans are for the length of 36 months and the first five(06) BorrowerState's codes with high loans are CA CO, GA, MN, NM and KS

Question 5: Relationship between BorrowerState per ProsperScore

BorrowerState : The two letter abbreviation of the state of the address of the borrower at the time the Listing was created.

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.

```
[ ]: color = sns.color_palette()[0]
plt.figure(figsize = (100, 10));
sns.countplot(data=prosperLoanData, x='BorrowerState', hue='ProsperScore',
↳palette='viridis_r');
plt.legend(fontsize=20, loc='upper right');
plt.title('Relationship between `BorrowerState` per `ProsperScore`',
↳fontdict={'fontsize': 20});
```

- There are five BorrowerState with Highest ProsperScore: CA, FL, NY, TX and IL

Bivariate Exploration Summary

Question 1: Relationship between BorrowerAPR and BorrowerState

- Per eachBorrowerState, the BorrowerAPR are slightly the same. There is not much difference of BorrowerAPR according to different BorrowerState.

Question 2: Relationship between ProsperRating (Alpha) and ProsperScore

- Loans with high ProsperScore (like 8, 9 or 10) are associated with high ProsperRating (like B, A or AA)
- Loans with median ProsperScore (like 5, 6 or 7) are associated with low or median ProsperRating (like E, D or C)
- Loans with low ProsperScore (like 1, 2 or 4) are associated with low ProsperRating (like HR, E or D)

Question 3: Relationship between ProsperScore and ProsperRating (numeric)

- The most contracted loan term is the length of 36 months with more than 80000 contracts.

Question 4: Relationship between Term and BorrowerState

- Most of the loans are for the length of 36 months and the first five (05) BorrowerState's codes with high loans are CO, GA, MN, NM and KS

Question 5: Relationship between BorrowerState per ProsperScore

- There are five BorrowerState with Highest ProsperScore: CA, FL, NY, TX and IL

Multivariate Exploration of Data

Question 1: Relationship between ProsperScore and BorrowerAPR per ProsperRating (Alpha)

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.

BorrowerAPR : The Borrower's Annual Percentage Rate (APR) for the loan.

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

```
[ ]: color = sns.color_palette()[0]
#plt.figure(figsize = (30, 20));
#plt.suptitle('`EmploymentStatus`, `Occupation`, `EmploymentStatusDuration` and
↳ `IsBorrowerHomeowner` distribution', fontsize=20, fontweight='bold', x=0.5,
↳ y=0.92)
plt.figure(figsize = (40, 10));

plt.subplot(1, 2, 1)
sns.regplot(data=prosperLoanData, x='ProsperScore', y='BorrowerAPR',
↳ truncate=False, x_jitter=0.1, scatter_kws={'alpha': 1/20});
plt.xticks(ticks=range(1, 11 + 1), labels=[f'{i}' for i in range(1, 11 + 1)])
plt.title('Relationship between `ProsperScore` and `BorrowerAPR`',
↳ fontdict={'fontsize': 20});

plt.subplot(1, 2, 2)
sns.scatterplot(data=prosperLoanData, x='ProsperScore', y='BorrowerAPR',
↳ hue='ProsperRating (Alpha)', x_jitter=0.3, palette='magma_r', alpha=1/10);
```

```
plt.xticks(ticks=range(1, 11 + 1), labels=[f' {i}' for i in range(1, 11 + 1)])
plt.legend(fontsize=20, loc='upper right');
plt.title('Relationship between `ProsperScore` and `BorrowerAPR` per
↳ `ProsperRating (Alpha)`', fontdict={'fontsize': 20});
```

- The higher the ProsperScore the lower the BorrowerAPR. There is a negative correlation between these variables.
- The lower the ProsperRating is (HR or E or D) the higher the BorrowerAPR is, among every ProsperScore values
- The better the ProsperRating is (A or AA), the lower the BorrowerAPR is.
- The better the ProsperRating is (A or AA) and much more the higher the ProsperScore is (with 10 being the best, or lowest risk score), the most lower the BorrowerAPR is

Question 2: Relationship between ProsperScore and LenderYield per ProsperRating (Alpha)

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.

LenderYield : The Lender yield on the loan. Lender yield is equal to the interest rate on the loan less the servicing fee.

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

```
[ ]: plt.figure(figsize = (30, 15));
sns.scatterplot(data=prosperLoanData, x='ProsperScore', y='LenderYield',
↳ hue='ProsperRating (Alpha)', x_jitter=0.3, palette='magma_r', alpha=1/10);
plt.title('Relationship between `ProsperScore` and `LenderYield` per
↳ `ProsperRating (Alpha)`', fontdict={'fontsize': 20});
```

- The higher the ProsperScore, the higher the ProsperRating (Alpha) the lower the LenderYield

Question 3: Relationship between LoanOriginalAmount and BorrowerAPR per ProsperRating (Alpha) per IncomeRange

```
[ ]: prosperLoanData['IncomeRange'].unique()
```

```
[ ]: # Let's convert IncomeRange to a CategoricalDtype
classes = ['Not displayed', 'Not employed', '$0', '$1-24,999',
↳ '$25,000-49,999', '$50,000-74,999', '$75,000-99,999', '$100,000+']
# Creating Category
cat_classes = pd.api.types.CategoricalDtype(categories=classes, ordered=True)
# Converting to CategoricalDtype
prosperLoanData['IncomeRange'] = prosperLoanData['IncomeRange'].
↳ astype(cat_classes)
# Testing
```

```
prosperLoanData['IncomeRange'].info()
```

```
[ ]: plt.figure(figsize = (20, 10));  
g = sns.FacetGrid(data=prosperLoanData, col='IncomeRange', row='ProsperRating_  
↳(Alpha)', margin_titles=True);  
g.map_dataframe(sns.scatterplot, x='LoanOriginalAmount', y='BorrowerAPR',  
↳alpha=0.4);
```

- For IncomeRange = 'Not employed' it is really difficult to have loan with even a little amountLoanOriginalAmount , with even a lower ProsperRating (Alpha) score
- For 'IncomeRange =\$0' it is merely impossible to have a loan no matter the ProsperRating (Alpha) score
- The BorrowerAPR range tend to be lower when the IncomeRange tend to be higher and the ProsperRating (Alpha) score higher
- The LoanOriginalAmount range tend to be higher when the IncomeRange tend to be higher and the ProsperRating (Alpha) score higher

Multivariate Exploration Summary

Question 1: Relationship between ProsperScore and BorrowerAPR per ProsperRating (Alpha)

- The higher the ProsperScore the lower the BorrowerAPR. There is a neagitive correlation between these variables.
- The lower the ProsperRating is (HR or E or D) the higher the BorrowerAPR is, among every ProsperScore values
- The better the ProsperRating is (A or AA), the lower the BorrowerAPR is.
- The better the ProsperRating is (A or AA) and much more the higher the ProsperScore is (with 10 being the best, or lowest risk score), the most lower the BorrowerAPR is

Question 2: Relationship between ProsperScore and LenderYield per ProsperRating (Alpha)

- The higher the ProsperScore, the higher the ProsperRating (Alpha) the lower the LenderYield

Question 3: Relationship between LoanOriginalAmount and BorrowerAPR per ProsperRating (Alpha) per IncomeRange

- For IncomeRange = 'Not employed' it is really difficult to have loan with even a little amountLoanOriginalAmount , with even a lower ProsperRating (Alpha) score
- For 'IncomeRange =\$0' it is merely impossible to have a loan no matter the ProsperRating (Alpha) score
- The BorrowerAPR range tend to be lower when the IncomeRange tend to be higher and the ProsperRating (Alpha) score higher
- The LoanOriginalAmount range tend to be higher when the IncomeRange tend to be higher and the ProsperRating (Alpha) score higher

Conclusions

1.1.1 Univariate Exploration Summary

Question 1: What is the trend of the `LoanStatus` variable ?

- As we can see, most of the loans are ongoing 49.66%, some are already completed 33.42% and the less are chargedoff 10.53%. All the other statuses are not so relevant because they have little percentages.

Question 2: What The most used `CreditGrade` ?

- The grade C is the most represented with 19.51%
- The grade D is the second most represented with 17.80%
- The grade B is the third most represented with 15.16%

Question 3: The most contracted loan `Term` ?

- The most contracted loan term is the length of 36 months with more than 80000 contracts.

Question 4: `BorrowerAPR` distribution

- The distribution is slightly normal
- With a standard deviation of 0.080364 the median value of the `BorrowerAPR` is 0.209760 and it is closest to the mean 0.218828. It means that the `BorrowerAPR` is really good for loans.

Question 5: What is the best `ProsperRating` ? The best `ProsperRating` is * The grade C is the most represented with 21.62% and followed by * The grade B is the second most represented with 18.36% and followed by * The grade A is the third most represented with 17.15%

Question 6: `ProsperScore` distribution

- The most highest `ProsperScore` are 4 then 6 then 8. More than 12000 loans are related to houses with such `ProsperScore`.

Question 7: `ListingCategory` distribution

- Most of the loans are for Debt Consolidation. Really inspiring.

Question 8: `Investors` distribution

- Almost all loans are funded by one investor. (More than 25000 loans.)

Question 9: `EmploymentStatus`, `Occupation`, `EmploymentStatusDuration` and `IsBorrowerHomeowner` distribution

- Most of the Borrowers are employed (Employed, Full-time or Self-employed)
- Most of the Borrowers don't specify their Occupations.
 - More than 10000 Borrowers are Professionals
 - Around 5000 Borrowers are Computer Programmer and Executive

- Around 4000 Borrowers are Teacher, Administrative Assistant and Analyst
- Around 50% of Borrowers are either Homeowners or not Homeowners.
- Looks like there needs to be an employee for some time (about 100 months) in order to the a borrower, assuming your employment come with high revenues. (to be able to be Homeowner.)

Question 10: IncomeRange distribution

- Clearly you must have a minimum IncomeRange of \$25,000 to be sure to have a loan.

1.1.2 Bivariate Exploration Summary

Question 1: Relationship between BorrowerAPR and BorrowerState

- Per eachBorrowerState, the BorrowerAPR are slightly the same. There is not much difference of BorrowerAPR according to differents BorrowerState.

Question 2: Relationship between ProsperRating (Alpha) and ProsperScore

- Loans with high ProsperScore(like 8, 9 or 10) are associated with high ProsperRating(like B, A or AA)
- Loans with median ProsperScore(like 5, 6 or 7) are associated with low or median ProsperRating(like E, D or C)
- Loans with low ProsperScore(like 1, 2 or 4) are associated with low ProsperRating(like HR, E or D)

Question 3: Relationship between ProsperScore and ProsperRating (numeric)

- The most contracted loan term is the length of 36 months with more than 80000 contracts.

Question 4: Relationship between Term and BorrowerState

- Most of the loans are for the lenght of 36 months and the first five(05) BorrowerState's codes with high loans are CO, GA, MN, NM and KS

Question 5: Relationship between BorrowerState per ProsperScore

- There are five BorrowerState with Highest ProsperScore: CA, FL, NY, TX and IL

1.1.3 Multivariate Exploration Summary

Question 1: Relationship between ProsperScore and BorrowerAPR per ProsperRating (Alpha)

- The higher the ProsperScore the lower the BorrowerAPR. There is a neagtive correlation between these variables.
- The lower the ProsperRating is (HR or E or D) the higher the BorrowerAPR is, among every ProsperScore values
- The better the ProsperRating is (A or AA), the lower the BorrowerAPR is.

- The better the ProsperRating is (A or AA) and much more the higher the ProsperScore is (with 10 being the best, or lowest risk score), the most lower the BorrowerAPR is

Question 2: Relationship between ProsperScore and LenderYield per ProsperRating (Alpha)

- The higher the ProsperScore, the higher the ProsperRating (Alpha) the lower the LenderYield

Question 3: Relationship between LoanOriginalAmount and BorrowerAPR per ProsperRating (Alpha) per IncomeRange

- For IncomeRange = 'Not employed' it is really difficult to have loan with even a little amountLoanOriginalAmount , with even a lower ProsperRating (Alpha) score
- For 'IncomeRange = \$0' it is merely impossible to have a loan no matter the ProsperRating (Alpha) score
- The BorrowerAPR range tend to be lower when the IncomeRange tend to be higher and the ProsperRating (Alpha) score higher
- The LoanOriginalAmount range tend to be higher when the IncomeRange tend to be higher and the ProsperRating (Alpha) score higher

Feedback

Thank you for paying attention to my work. Please I need your review to improve myself.

- What do you notice about each visualization?
- What questions do you have about the data?
- What relationships do you notice?
- What do you think is the main takeaway from the report / presentation?
- Is there anything that you don't understand from the plots?

Kindly reach me:

- LinkedIn : [jozias-tema](#)
- GitHub : [jozias-tema](#)
- Gmail : [jozias-tema](#) temajozias@gmail.com

Let's export to html and pdf

```
[ ]: from subprocess import call
call(['python', '-m', 'nbconvert', 'prosperLoanData_part1_exploratory.ipynb',
      '--to', 'pdf'])
```

```
[ ]: from subprocess import call
call(['python', '-m', 'nbconvert', 'prosperLoanData_part1_exploratory.ipynb',
      '--to', 'html'])
```