Part1_prosper_loan_exploration

November 15, 2022

1 Part I - (Prosper Loan data exploration and analysis.)

1.1 by FIDELIS WAWERU

1.2 Introduction

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. Please find the data dictionary in the link below data dictionary

1.3 Preliminary Wrangling

[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 sb

%matplotlib inline
```

Loading data and assessing it

[2]:
```
df = pd.read_csv("prosperLoanData.csv")
```

[3]:
```
df.shape
```

[3]:
```
(113937, 81)
```

[4]:
```
df.sample(10)
```

[4]:
```
                     ListingKey  ListingNumber            ListingCreationDate  \
47018   A33F35317161657494B40A9         540850  2011-11-23 19:49:42.877000000
94865   29313580728564951A237CB         806389  2013-06-12 09:04:42.497000000
25971   C8C73594291269283D3916B        1017155  2013-11-14 12:34:16.507000000
607     00F43563947805596EC80E9         670164  2012-11-14 08:35:42.453000000
29261   F833353808572872449 47AF        555872  2012-02-01 10:44:02.337000000
91209   28D734798116285616 3460C        450600  2010-03-16 16:41:03.203000000
21940   51E73603842819749 5A6718       1177741  2014-02-24 06:32:45.687000000
102058  97CC356287396628384A1CF         674014  2012-11-19 12:56:49.987000000
```

24011	EF163535899723526B826D6	550847	2012-01-11 18:16:33.047000000
48269	536535854492047621EBAD2	847678	2013-07-22 13:34:56.820000000

	CreditGrade	Term	LoanStatus	ClosedDate	BorrowerAPR	\
47018	NaN	36	Current	NaN	0.29486	
94865	NaN	60	Current	NaN	0.11695	
25971	NaN	36	Current	NaN	0.23898	
607	NaN	60	Chargedoff	2013-12-20 00:00:00	0.35097	
29261	NaN	36	Current	NaN	0.24246	
91209	NaN	36	Completed	2013-03-12 00:00:00	0.07439	
21940	NaN	36	Current	NaN	0.16678	
102058	NaN	36	Current	NaN	0.09736	
24011	NaN	36	Current	NaN	0.11766	
48269	NaN	60	Current	NaN	0.10367	

	BorrowerRate	LenderYield	…	LP_ServiceFees	LP_CollectionFees	\
47018	0.2561	0.2461	…	-47.92	0.0	
94865	0.0949	0.0849	…	-96.00	0.0	
25971	0.2015	0.1915	…	-9.88	0.0	
607	0.3232	0.3132	…	-25.84	0.0	
29261	0.2049	0.1949	…	-44.96	0.0	
91209	0.0710	0.0610	…	-39.81	0.0	
21940	0.1305	0.1205	…	0.00	0.0	
102058	0.0839	0.0739	…	-25.73	0.0	
24011	0.0899	0.0799	…	-117.83	0.0	
48269	0.0819	0.0719	…	-73.44	0.0	

	LP_GrossPrincipalLoss	LP_NetPrincipalLoss	\
47018	0.00	0.00	
94865	0.00	0.00	
25971	0.00	0.00	
607	3753.61	3753.61	
29261	0.00	0.00	
91209	0.00	0.00	
21940	0.00	0.00	
102058	0.00	0.00	
24011	0.00	0.00	
48269	0.00	0.00	

	LP_NonPrincipalRecoverypayments	PercentFunded	Recommendations	\
47018	0.0	1.0	0	
94865	0.0	1.0	0	
25971	0.0	1.0	0	
607	0.0	1.0	0	
29261	0.0	1.0	0	
91209	0.0	1.0	0	
21940	0.0	1.0	0	

```
102058                            0.0          1.0                  0
24011                             0.0          1.0                  0
48269                             0.0          1.0                  0

       InvestmentFromFriendsCount  InvestmentFromFriendsAmount  Investors
47018                           0                          0.0         13
94865                           0                          0.0        236
25971                           0                          0.0          1
607                             0                          0.0         81
29261                           0                          0.0          1
91209                           1                         50.0        119
21940                           0                          0.0          1
102058                          0                          0.0         55
24011                           0                          0.0        125
48269                           0                          0.0        128

[10 rows x 81 columns]
```

[5]:
```
df.describe()
```

[5]:
```
       ListingNumber           Term    BorrowerAPR   BorrowerRate  \
count   1.139370e+05  113937.000000  113912.000000  113937.000000
mean    6.278857e+05      40.830248       0.218828       0.192764
std     3.280762e+05      10.436212       0.080364       0.074818
min     4.000000e+00      12.000000       0.006530       0.000000
25%     4.009190e+05      36.000000       0.156290       0.134000
50%     6.005540e+05      36.000000       0.209760       0.184000
75%     8.926340e+05      36.000000       0.283810       0.250000
max     1.255725e+06      60.000000       0.512290       0.497500

         LenderYield  EstimatedEffectiveYield  EstimatedLoss  EstimatedReturn  \
count  113937.000000             84853.000000   84853.000000     84853.000000
mean        0.182701                 0.168661       0.080306         0.096068
std         0.074516                 0.068467       0.046764         0.030403
min        -0.010000                -0.182700       0.004900        -0.182700
25%         0.124200                 0.115670       0.042400         0.074080
50%         0.173000                 0.161500       0.072400         0.091700
75%         0.240000                 0.224300       0.112000         0.116600
max         0.492500                 0.319900       0.366000         0.283700

       ProsperRating (numeric)  ProsperScore  …  LP_ServiceFees  \
count             84853.000000  84853.000000  …   113937.000000
mean                  4.072243      5.950067  …      -54.725641
std                   1.673227      2.376501  …       60.675425
min                   1.000000      1.000000  …     -664.870000
25%                   3.000000      4.000000  …      -73.180000
50%                   4.000000      6.000000  …      -34.440000
```

```
75%                  5.000000      8.000000  …      -13.920000
max                  7.000000     11.000000  …       32.060000

       LP_CollectionFees  LP_GrossPrincipalLoss  LP_NetPrincipalLoss  \
count      113937.000000          113937.000000        113937.000000
mean          -14.242698             700.446342           681.420499
std           109.232758            2388.513831          2357.167068
min         -9274.750000             -94.200000          -954.550000
25%             0.000000               0.000000             0.000000
50%             0.000000               0.000000             0.000000
75%             0.000000               0.000000             0.000000
max             0.000000           25000.000000         25000.000000

       LP_NonPrincipalRecoverypayments  PercentFunded  Recommendations  \
count                    113937.000000  113937.000000    113937.000000
mean                         25.142686       0.998584         0.048027
std                         275.657937       0.017919         0.332353
min                           0.000000       0.700000         0.000000
25%                           0.000000       1.000000         0.000000
50%                           0.000000       1.000000         0.000000
75%                           0.000000       1.000000         0.000000
max                       21117.900000       1.012500        39.000000

       InvestmentFromFriendsCount  InvestmentFromFriendsAmount      Investors
count               113937.000000                113937.000000  113937.000000
mean                     0.023460                    16.550751      80.475228
std                      0.232412                   294.545422     103.239020
min                      0.000000                     0.000000       1.000000
25%                      0.000000                     0.000000       2.000000
50%                      0.000000                     0.000000      44.000000
75%                      0.000000                     0.000000     115.000000
max                     33.000000                 25000.000000    1189.000000

[8 rows x 61 columns]
```

[6]:
```
df.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
```

1.3.1 Data Cleaning

In this section I will doing some data cleaning, ie. checking the data types, the missing data, and removing columns that are not neccessary for our analysis etc.

Remove unwanted columns Remove the columns that I will not use for my visualization and remain with the 17 coulmns that are described beow:

- `ListingCreationDate`: The date the listing was created.
- `Term`: The length of the loan expressed in months.
- `LoanStatus`: The current status of the loan: Cancelled, Chargedoff, Completed, Current, Defaulted, FinalPaymentInProgress, PastDue. The PastDue status will be accompanied by a delinquency bucket
- `BorrowerAPR`: The Borrower's Annual Percentage Rate (APR) for the loan.
- `BorrowerRate`: The Borrower's interest rate for this loan.
- `LenderYield`: The Lender yield on the loan. Lender yield is equal to the interest rate on the loan less the servicing fee.
- `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.

- `EmploymentStatus`: The employment status of the borrower at the time they posted 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 homowner if they have a mortgage on their credit profile or provide documentation confirming they are a homeowner.
- `IncomeRange`: The income range of the borrower at the time the listing was created.
- `IncomeVerifiable`: The borrower indicated they have the required documentation to support their income.
- `StatedMonthlyIncome`: The monthly income the borrower stated at the time the listing was created.
- `LoanOriginalAmount`: The origination amount of the loan.
- `LoanOriginationDate`: The date the loan was originated.
- `LoanOriginationQuarter`: The quarter in which the loan was originated.
- `MonthlyLoanPayment`: The scheduled monthly loan payment.

[7]:
```
# df1 = df.iloc[:,[2,4,5,7,8,9,16,19,20,21,47,48,49,50,51,52,63,64,65,67,80]]
df1 = df.iloc[:,[2,4,5,7,8,9,16,19,20,21,47,48,49,63,64,65,67]]
df1.head(10)
```

[7]:
```
              ListingCreationDate  Term LoanStatus  BorrowerAPR  BorrowerRate  \
0  2007-08-26 19:09:29.263000000    36  Completed      0.16516        0.1580
1  2014-02-27 08:28:07.900000000    36    Current      0.12016        0.0920
2  2007-01-05 15:00:47.090000000    36  Completed      0.28269        0.2750
3  2012-10-22 11:02:35.010000000    36    Current      0.12528        0.0974
4  2013-09-14 18:38:39.097000000    36    Current      0.24614        0.2085
5  2013-12-14 08:26:37.093000000    60    Current      0.15425        0.1314
6  2013-04-12 09:52:56.147000000    36    Current      0.31032        0.2712
7  2013-05-05 06:49:27.493000000    36    Current      0.23939        0.2019
8  2013-12-02 10:43:39.117000000    36    Current      0.07620        0.0629
9  2013-12-02 10:43:39.117000000    36    Current      0.07620        0.0629

   LenderYield  ListingCategory (numeric) EmploymentStatus  \
0       0.1380                          0    Self-employed
1       0.0820                          2         Employed
2       0.2400                          0    Not available
3       0.0874                         16         Employed
4       0.1985                          2         Employed
5       0.1214                          1         Employed
6       0.2612                          1         Employed
7       0.1919                          2         Employed
8       0.0529                          7         Employed
```

```
9       0.0529                               7          Employed

   EmploymentStatusDuration  IsBorrowerHomeowner     IncomeRange  \
0                       2.0                 True  $25,000-49,999
1                      44.0                False  $50,000-74,999
2                       NaN                False   Not displayed
3                     113.0                 True  $25,000-49,999
4                      44.0                 True        $100,000+
5                      82.0                 True        $100,000+
6                     172.0                False  $25,000-49,999
7                     103.0                False  $25,000-49,999
8                     269.0                 True  $25,000-49,999
9                     269.0                 True  $25,000-49,999

   IncomeVerifiable  StatedMonthlyIncome  LoanOriginalAmount  \
0              True          3083.333333                9425
1              True          6125.000000               10000
2              True          2083.333333                3001
3              True          2875.000000               10000
4              True          9583.333333               15000
5              True          8333.333333               15000
6              True          2083.333333                3000
7              True          3355.750000               10000
8              True          3333.333333               10000
9              True          3333.333333               10000

   LoanOriginationDate LoanOriginationQuarter  MonthlyLoanPayment
0  2007-09-12 00:00:00                Q3 2007              330.43
1  2014-03-03 00:00:00                Q1 2014              318.93
2  2007-01-17 00:00:00                Q1 2007              123.32
3  2012-11-01 00:00:00                Q4 2012              321.45
4  2013-09-20 00:00:00                Q3 2013              563.97
5  2013-12-24 00:00:00                Q4 2013              342.37
6  2013-04-18 00:00:00                Q2 2013              122.67
7  2013-05-13 00:00:00                Q2 2013              372.60
8  2013-12-12 00:00:00                Q4 2013              305.54
9  2013-12-12 00:00:00                Q4 2013              305.54
```

Missing data Drop the rows with missing data

[8]:
```
df1 = df1.dropna()
```

[9]:
```
df1.sample(10)
```

[9]:
```
                       ListingCreationDate  Term  LoanStatus  BorrowerAPR  \
35505    2014-01-06 13:12:30.113000000    60     Current      0.29567
4267     2012-09-07 06:15:02.943000000    36     Current      0.13697
```

```
82933   2007-03-14 08:18:06.790000000    36   Completed      0.15713
86528   2007-08-30 14:09:34.407000000    36   Defaulted      0.13152
7052    2013-04-15 08:32:45.827000000    60   Completed      0.29341
11096   2012-06-26 07:10:14.207000000    60     Current      0.27462
38134   2012-07-29 15:17:38.707000000    36     Current      0.35797
64743   2013-06-15 06:23:15.617000000    36     Current      0.19645
108216  2008-07-15 11:54:41.810000000    36   Completed      0.28625
20627   2007-10-11 10:06:11.460000000    36  Chargedoff      0.23983

        BorrowerRate  LenderYield  ListingCategory (numeric) EmploymentStatus  \
35505         0.2694       0.2594                         15         Employed
4267          0.1089       0.0989                          1         Employed
82933         0.1500       0.1350                          0     Not employed
86528         0.1245       0.1145                          0        Full-time
7052          0.2672       0.2572                          1         Employed
11096         0.2489       0.2389                         19         Employed
38134         0.3177       0.3077                          1    Self-employed
64743         0.1599       0.1499                          1         Employed
108216        0.2708       0.2608                          3        Full-time
20627         0.2248       0.2148                          0        Full-time

        EmploymentStatusDuration  IsBorrowerHomeowner     IncomeRange  \
35505                      264.0                False  $50,000-74,999
4267                        49.0                 True  $25,000-49,999
82933                        0.0                 True    Not employed
86528                       24.0                 True       $100,000+
7052                       225.0                 True  $50,000-74,999
11096                      273.0                False  $75,000-99,999
38134                      188.0                 True       $100,000+
64743                      136.0                 True  $50,000-74,999
108216                     171.0                 True       $100,000+
20627                       46.0                False       $1-24,999

        IncomeVerifiable  StatedMonthlyIncome  LoanOriginalAmount  \
35505               True          5583.333333                4500
4267                True          3166.666667               11500
82933               True          1500.000000                8000
86528               True          8333.333333               10000
7052                True          5000.000000               12000
11096               True          6250.000000               15000
38134              False          9166.666667                4000
64743               True          4166.666667               12000
108216              True          9000.000000               13000
20627               True          1666.666667                2000

        LoanOriginationDate LoanOriginationQuarter  MonthlyLoanPayment
35505   2014-01-14 00:00:00               Q1 2014              137.25
```

```
4267    2012-09-12 00:00:00                Q3 2012              375.90
82933   2007-03-27 00:00:00                Q1 2007              277.32
86528   2007-09-12 00:00:00                Q3 2007              334.30
7052    2013-04-29 00:00:00                Q2 2013              364.42
11096   2012-07-10 00:00:00                Q3 2012              439.30
38134   2012-09-12 00:00:00                Q3 2012              173.71
64743   2013-06-26 00:00:00                Q2 2013              421.83
108216  2008-07-28 00:00:00                Q3 2008              517.89
20627   2007-10-19 00:00:00                Q4 2007               76.88
```

Round off float values to 2 decimal points.

[10]:
```
df1 = np.round(df1, 2)
```

Convert The `ListingCreationDate` and `LoanOriginationDate` datatypes to **datetime** datatype and `EmploymentStatusDuration` to **int** datatype. The `LoanStatus`,`ListingCategory`,`EmploymentStatus`,`IncomeRange`,`LoanOriginationQuarter` columns will be converted to **category**.

[11]:
```
df1["ListingCreationDate"] = pd.to_datetime(df1["ListingCreationDate"])
df1["LoanOriginationDate"] = pd.to_datetime(df1["LoanOriginationDate"])
```

[12]:
```
df1["EmploymentStatusDuration"] = df1['EmploymentStatusDuration'].astype(int)
df1["EmploymentStatusDuration"].dtype
```

[12]:
```
dtype('int32')
```

[13]:
```
df1 = df1.rename(columns={"ListingCategory (numeric)": "ListingCategory"})
```

[14]:
```
df1['LoanStatus'] = df1['LoanStatus'].astype('category')
df1['ListingCategory'] = df1['ListingCategory'].astype('category')
df1['EmploymentStatus'] = df1['EmploymentStatus'].astype('category')
df1['IncomeRange'] = df1['IncomeRange'].astype('category')
df1['LoanOriginationQuarter'] = df1['LoanOriginationQuarter'].astype('category')
```

Rename the `ListingCategory (numeric)` to just `ListingCategory` as I find the word **numeric** name not really relevant since I am going to change the numeric values of the column to their associated Category values

[15]:
```
df1["ListingCategory"] = df1["ListingCategory"].map({ 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"})
```

[16]:
```
df1.head(5)
```

[16]:
```
        ListingCreationDate  Term LoanStatus  BorrowerAPR  BorrowerRate  \
0 2007-08-26 19:09:29.263    36  Completed         0.17          0.16
1 2014-02-27 08:28:07.900    36    Current         0.12          0.09
3 2012-10-22 11:02:35.010    36    Current         0.13          0.10
4 2013-09-14 18:38:39.097    36    Current         0.25          0.21
5 2013-12-14 08:26:37.093    60    Current         0.15          0.13

   LenderYield     ListingCategory EmploymentStatus  EmploymentStatusDuration  \
0         0.14       Not Available    Self-employed                         2
1         0.08    Home Improvement         Employed                        44
3         0.09          Motorcycle         Employed                       113
4         0.20    Home Improvement         Employed                        44
5         0.12  Debt Consolidation         Employed                        82

   IsBorrowerHomeowner     IncomeRange  IncomeVerifiable  StatedMonthlyIncome  \
0                 True  $25,000-49,999              True              3083.33
1                False  $50,000-74,999              True              6125.00
3                 True  $25,000-49,999              True              2875.00
4                 True        $100,000+             True              9583.33
5                 True        $100,000+             True              8333.33

   LoanOriginalAmount LoanOriginationDate LoanOriginationQuarter  \
0                9425          2007-09-12                Q3 2007
1               10000          2014-03-03                Q1 2014
3               10000          2012-11-01                Q4 2012
4               15000          2013-09-20                Q3 2013
5               15000          2013-12-24                Q4 2013

   MonthlyLoanPayment
0              330.43
1              318.93
3              321.45
4              563.97
5              342.37
```

[17]:
```
df1.shape
```

[17]:
```
(106312, 17)
```

[18]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 106312 entries, 0 to 113936
```

```
Data columns (total 17 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   ListingCreationDate       106312 non-null  datetime64[ns]
 1   Term                      106312 non-null  int64
 2   LoanStatus                106312 non-null  category
 3   BorrowerAPR               106312 non-null  float64
 4   BorrowerRate              106312 non-null  float64
 5   LenderYield               106312 non-null  float64
 6   ListingCategory           106312 non-null  category
 7   EmploymentStatus          106312 non-null  category
 8   EmploymentStatusDuration  106312 non-null  int32
 9   IsBorrowerHomeowner       106312 non-null  bool
 10  IncomeRange               106312 non-null  category
 11  IncomeVerifiable          106312 non-null  bool
 12  StatedMonthlyIncome       106312 non-null  float64
 13  LoanOriginalAmount        106312 non-null  int64
 14  LoanOriginationDate       106312 non-null  datetime64[ns]
 15  LoanOriginationQuarter    106312 non-null  category
 16  MonthlyLoanPayment        106312 non-null  float64
dtypes: bool(2), category(5), datetime64[ns](2), float64(5), int32(1), int64(2)
memory usage: 9.2 MB
```

Set the order of the IncomeRange categorical variable

[19]:
```
from pandas.api.types import CategoricalDtype
income_range_cat = ['Not displayed', 'Not employed','$0', '$1-24,999',␣
↪'$25,000-49,999', '$50,000-74,999', '$75,000-99,999','$100,000+']
```

[20]:
```
df1.to_csv('prosperLoan_new.csv', index=False)
```

1.3.2 What is the structure of your dataset?

The Prosper data set contains 113,937 loan data with 81 variables on each loan which are described in a dictionary attached in the Link data dictionary. Some of the variables in the dataset are not releavant for my analysis, so I dropped some and remained with 106312 data items and 17 variables for my exploration. The new dataset has **category**,**int64**,**float64**,**bool** and **datetime64** data types.

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

The main features of interest from the prosper loan dataset are - The factors affecting the `BorrowerAPR` or the `BorrowerRate`. - How does the amount of income affect the amount loan to be issued and the monthly loan payment.

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

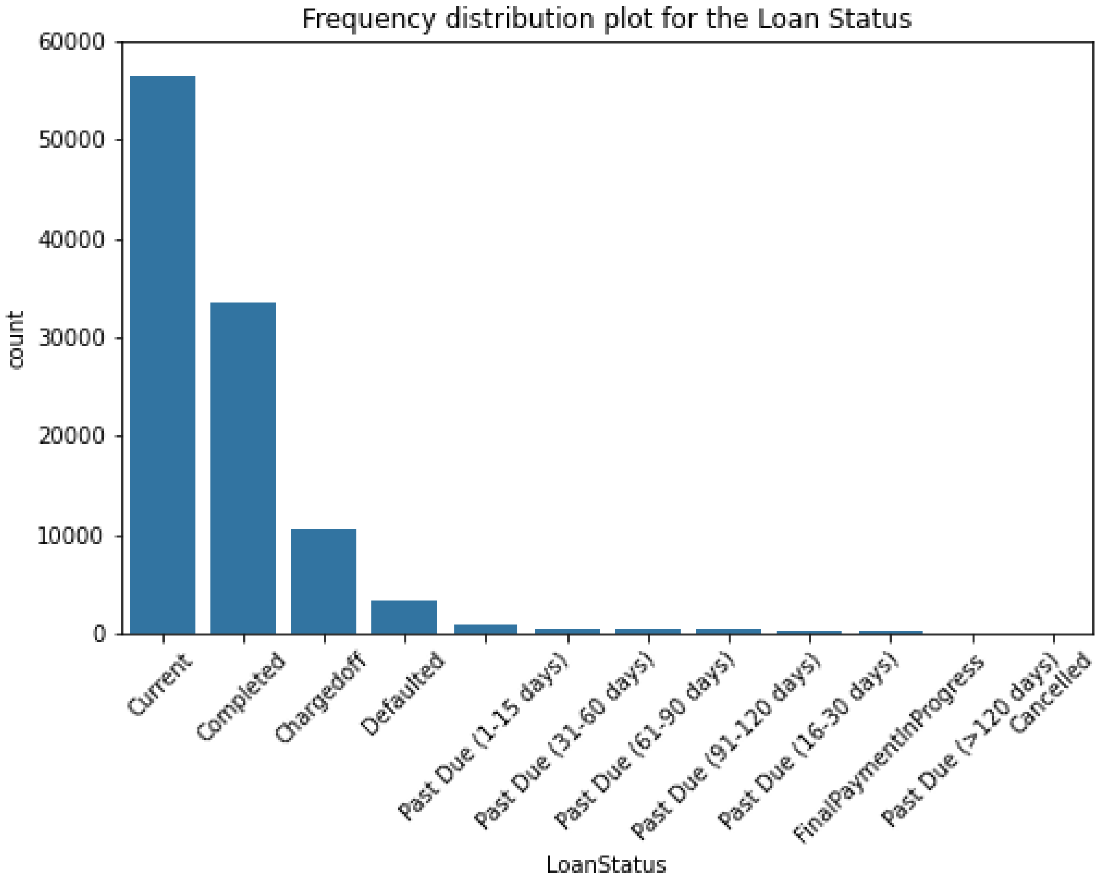
The dataset contains variables that depends on each other such as `LoanStatus`, `Term`, `BorrowerAPR`,`BorrowerRate`, `IncomeRange`, `LoanOriginalAmount`,`MonthlyLoanPayment` which are the main features in the dataset. My assumption is that the IncomeRange which highly influence the loan original amount to be issued, the Monthly loan payment also influence the BorrowerAPR and BorrowerRate. The employment status, Stated monthly income affect the loan status and term since an employed person will have income flow and will be able to pay the loan within a short time as compared to uneployed person.

1.4 Univariate Exploration

1.4.1 Qusteion 1

What is the LoanStatus frequency of the borrowers?

[133]:
```
#the LoanStatus frequency of the borrowers
plt.figure(figsize =[8, 5])
color = sb.color_palette()[0];
frequency = df1['LoanStatus'].value_counts().index;
sb.countplot(data = df1, x="LoanStatus", color=color, order = frequency);
plt.ylim(0, 60000);
plt.title("Frequency distribution plot for the Loan Status")
plt.xticks(rotation = 45);
```



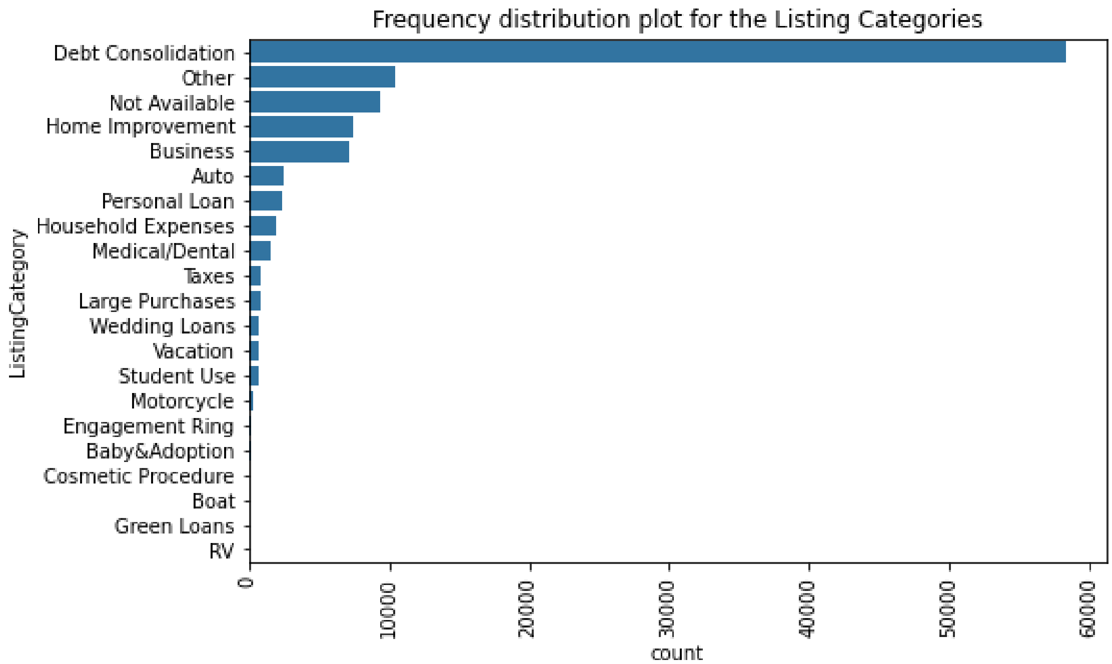
1.4.2 Observation

The plot above suggest that most of the clients have their loan status as current with a count of apploximately 55,000 thousand and it also show oly a few who have their loans past due.

1.4.3 Question 2

What is the ListingCategory of the borrowers?

[134]:
```
#ListingCategory
plt.figure(figsize =[8, 5]);
color = sb.color_palette()[0];
frequency = df1['ListingCategory'].value_counts().index;
sb.countplot(data = df1, y="ListingCategory", color=color, order = frequency);
plt.title("Frequency distribution plot for the Listing Categories")
plt.xticks(rotation = 90);
```



1.4.4 Observations

ListingCategory variable show the category of the listing selected by the borrower when posting their listing. From the dstribution above,it shows that approximately 59,000 borrowers listed debt consolidation and least listed category was student use.

1.4.5 Question 3

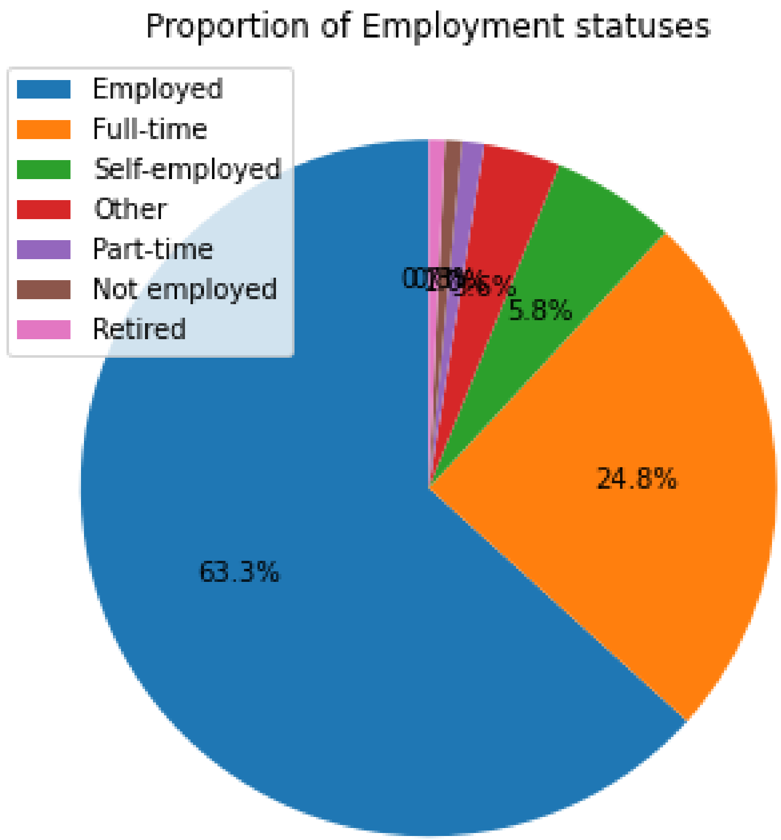
What is the EmploymentStatus of the borrowers?

[23]:
```
EmploymentStatus_counts = df1['EmploymentStatus'].value_counts()
EmploymentStatus_counts
```

[23]:
```
Employed         67321
Full-time        26342
Self-employed     6132
Other             3800
Part-time         1088
Not employed       834
Retired            795
Name: EmploymentStatus, dtype: int64
```

[136]:
```
#Employment Status
EmploymentStatus_counts = df1['EmploymentStatus'].value_counts();
# labels = ["Employed","Full-time","Self-employed","Other","Part-time","Not␣
↪employed","Retired"]
plt.figure(figsize =(8, 6));
```

```
plt.pie(EmploymentStatus_counts, autopct="%1.1f%%", startangle=90);
plt.legend(labels = EmploymentStatus_counts.index);
plt.title("Proportion of Employment statuses");
plt.show();
```



1.4.6 Observations

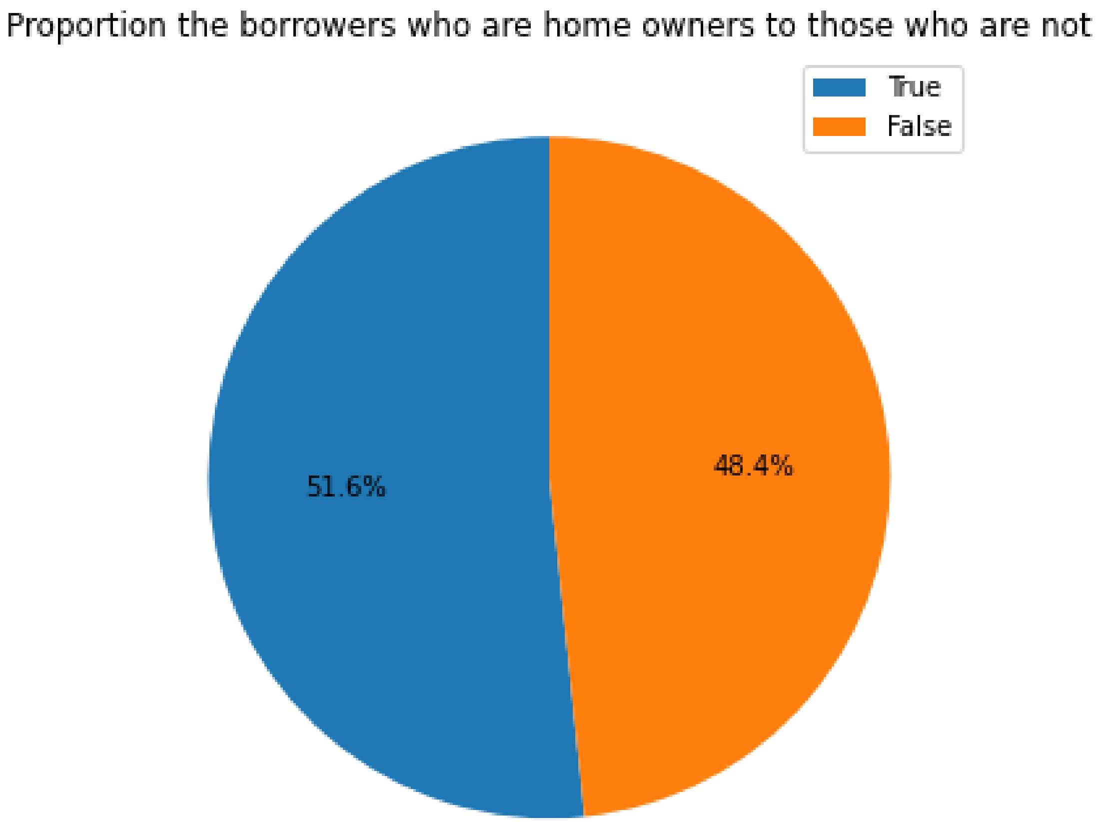
To get a clear frequency in the EmploymentStatus variable, I had to find the count of each value using the `value_counts()` function and later plotted a pie chart to show the distribution. The pie chart indicates that a 63.3% of the borrowers are employed and just a small percentage of them are retired.

1.4.7 Question 4

What is the EmploymentStatus of the borrowers?

[25]:
```
#Is the borrower a home owner or not
Homeowner_counts = df1['IsBorrowerHomeowner'].value_counts()
plt.figure(figsize =(8, 6))
```

```
plt.pie(Homeowner_counts,autopct="%1.1f%%",  startangle=90)
plt.legend(labels = Homeowner_counts.index)
plt.title("Proportion the borrowers who are home owners to those who are not")
plt.show()
```



1.4.8 Observations

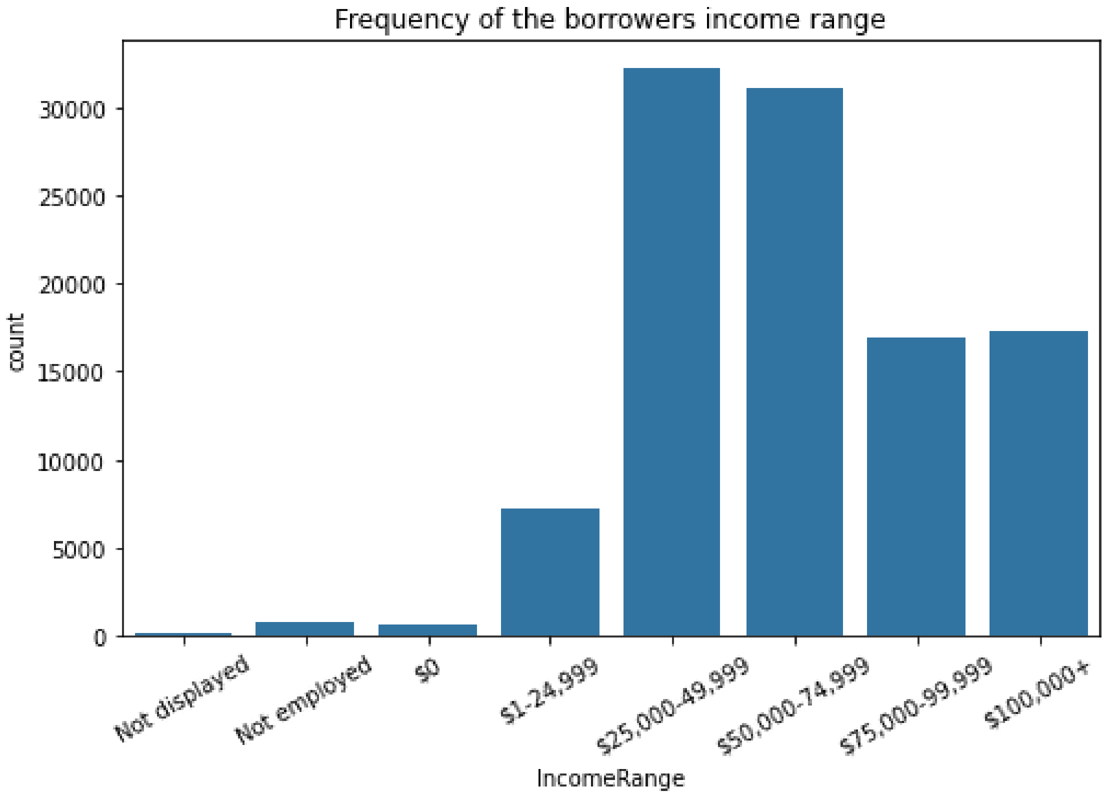
The pie chart above shows the percentage of the borrowers who are home owners and those that are'nt. I can conclude that 51.6% of the borrowers are home owners and only 48.4% are not home owners.

1.4.9 Question 5

What is the IncomeRange of the Loan borrowers?

[141]:
```
#income range
# incomerangeorder = df1["IncomeRange"].value_counts().index
plt.figure(figsize =[8, 5]);
sb.countplot(data = df1, x='IncomeRange', color=color, order=income_range_cat);
```

```
plt.title("Frequency of the borrowers income range");
plt.xticks(rotation = 30);
```



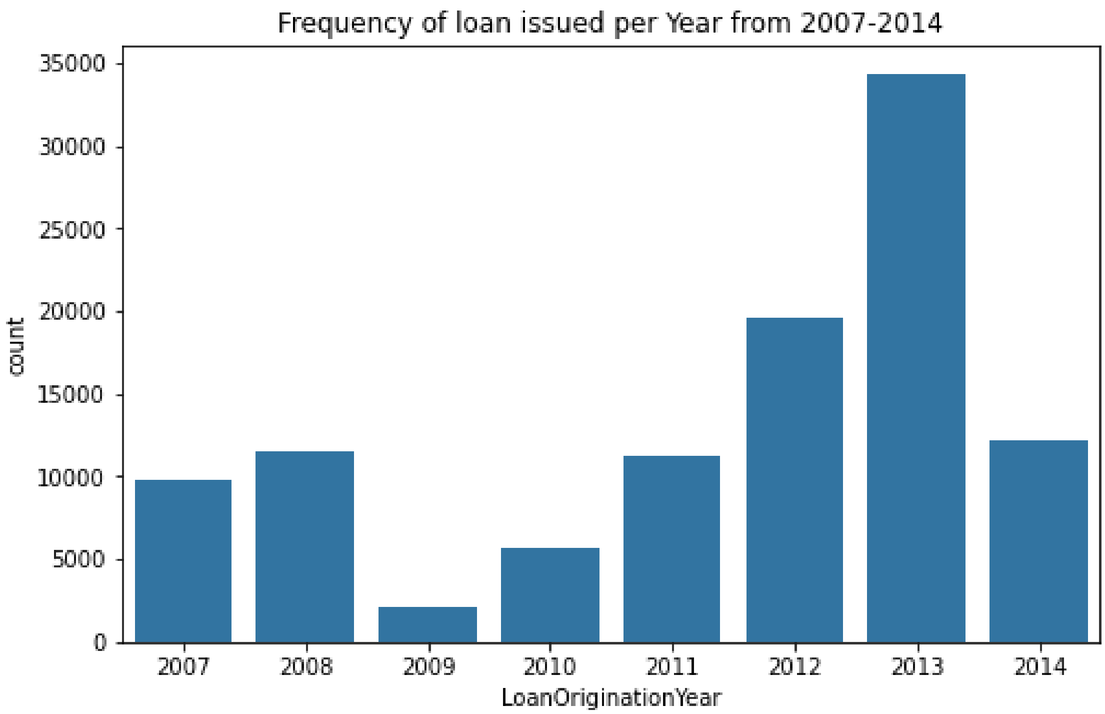
1.4.10 Observations

The distrbution above shows the frequency of the income range of the borrowers and $25,000 - $49,000 category had the highest frequency meaning that a large number of borrowers had an income range of $25,000 - $49,000 and just a few who chose not to display their income range.

1.4.11 Question 6

What is the LoanOriginationYear of the Loan borrowed?

[188]:
```
#Loan origination year
# I am going creating another variable and store it at `LoanOriginationYear`␣
 ↪variable
plt.figure(figsize =[8, 5]);
df1["LoanOriginationYear"] = df1["LoanOriginationDate"].dt.year;
plt.title("Frequency of loan issued per Year from 2007-2014");
sb.countplot(data=df1, x="LoanOriginationYear", color=color);
```



1.4.12 Observations

The prosper loan dataset have a loan original date which i extracted the loan original year and stored in a `LoanOriginalYear` variable and used it to plot a frrequency distribution. The dstribution shows that more than 30,000 borrowers took their loans in the year 2013, and less than 5,000 borrowers took their loan in 2009 which has the least count.

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

On the Loan Status, there is a large number of borrowers with their loans as Current, followed by a completed loan status. we can also say that none of the borrowers had their loans cauncelled. The IncomeRange distribution shows that many borrowers had an income range of $25,000 - 49,000. There are some who had no income who still applied for the loans, a few chose not to disply and a few were unemplyed.

1.4.14 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?

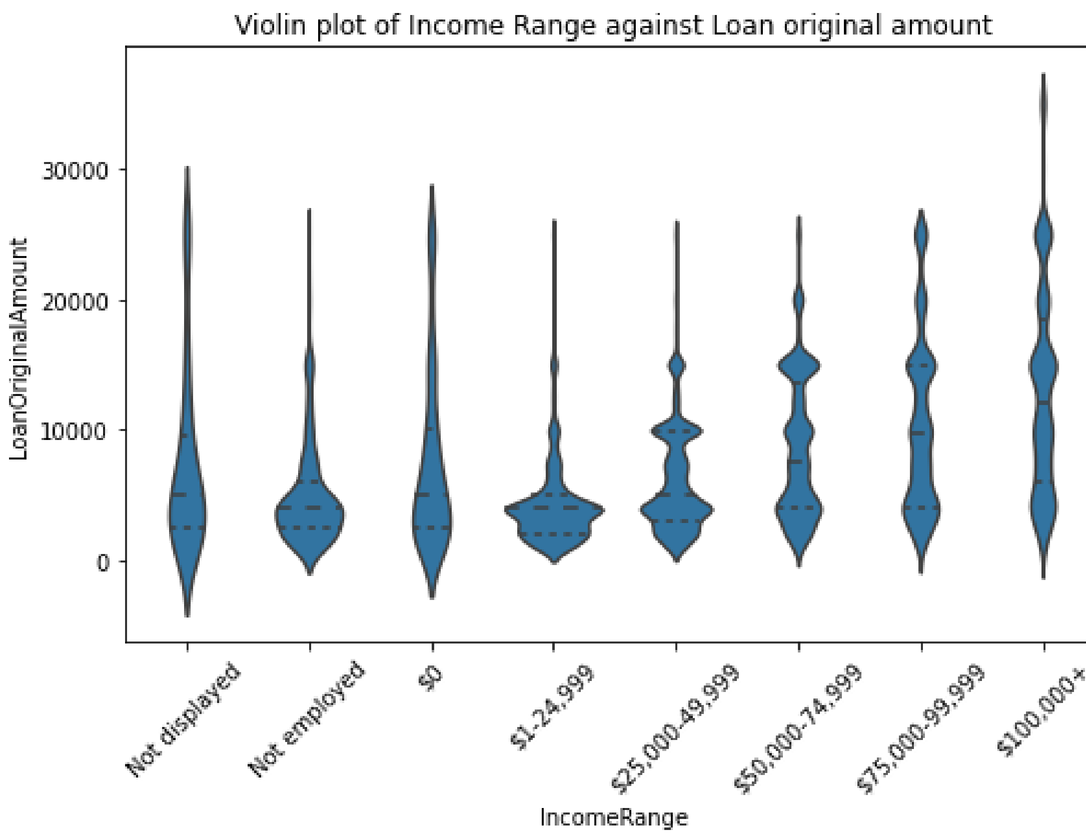
It was observed that very little proportion of loans (1.5%) were given out for a 12 month Term with `Debt consolidation` topping the charts as the major reason that borrowers obtained loans. It was also observed that employed persons obtained more loans within the period compared to other categories of borrowers. Also, an unusual distribution was observed in the year 2009 in comparison with other years as a very small proportion of loans were administered in that year. More investigation will be needed to find out why. The year 2013 however, had high proportion of loans administered when compared with other years. Certain adjustment were made on the loan data to obtained clarity of the loan origination period - one new variable was created for LoanOriginationYear.

1.5 Bivariate Exploration

1.5.1 Question 7

How does the borrowers income Range affect the loan original amount borrowed?

[144]:
```
# IR_count = [""]
plt.figure(figsize =[8, 5]);
sb.violinplot(data=df1, x="IncomeRange", y="LoanOriginalAmount",color=color,␣
  ↪inner='quartile', order=income_range_cat);
plt.title("Violin plot of Income Range against Loan original amount");
plt.xticks(rotation=45);
```



1.5.2 Observations

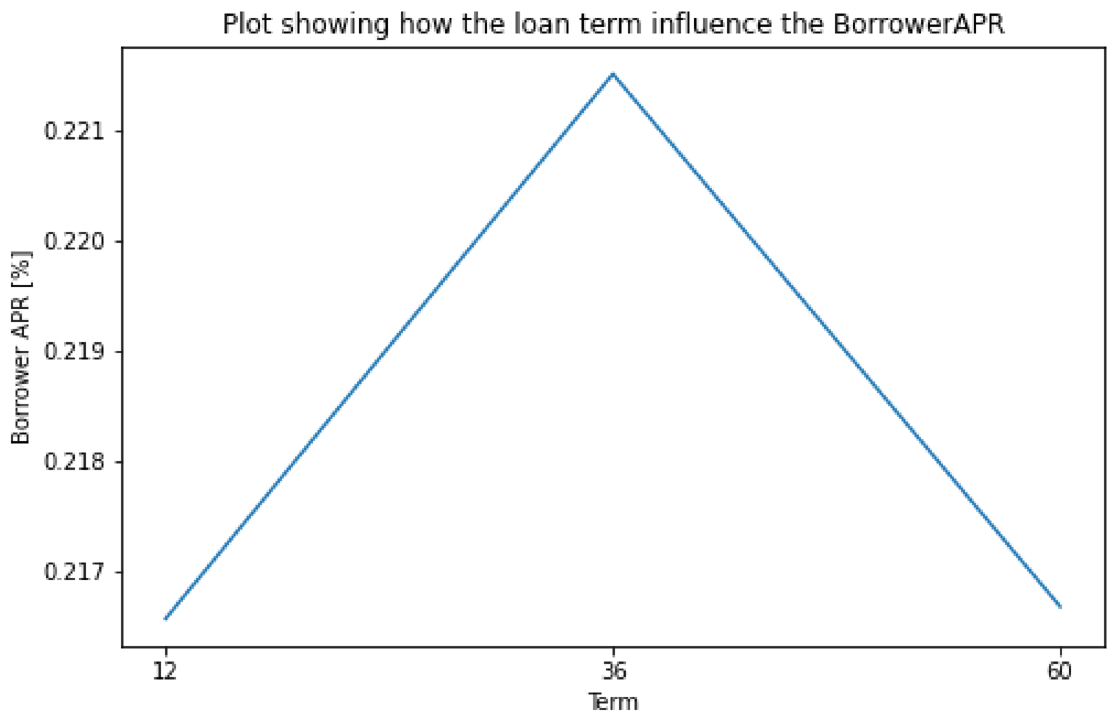
I looked at the relationship between IncomeRange and the LoanOriginalAmount using a violin plot. Each violin show the median,lower and upper quartile of the LoanOriginalAmount in every

IncomeRange category. The 100,000+ have a greater interquartile range.

1.5.3 Question 8

Should the borrowers consider the loan term when taking the loans?

[148]:
```
plt.figure(figsize=[8,5])
plt.plot(df1.groupby('Term')['BorrowerAPR'].mean());
plt.xlabel('Term');
plt.ylabel('Borrower APR [%]');
plt.title("Plot showing how the loan term influence the BorrowerAPR");
plt.xticks([12,36,60]);
plt.show();
```



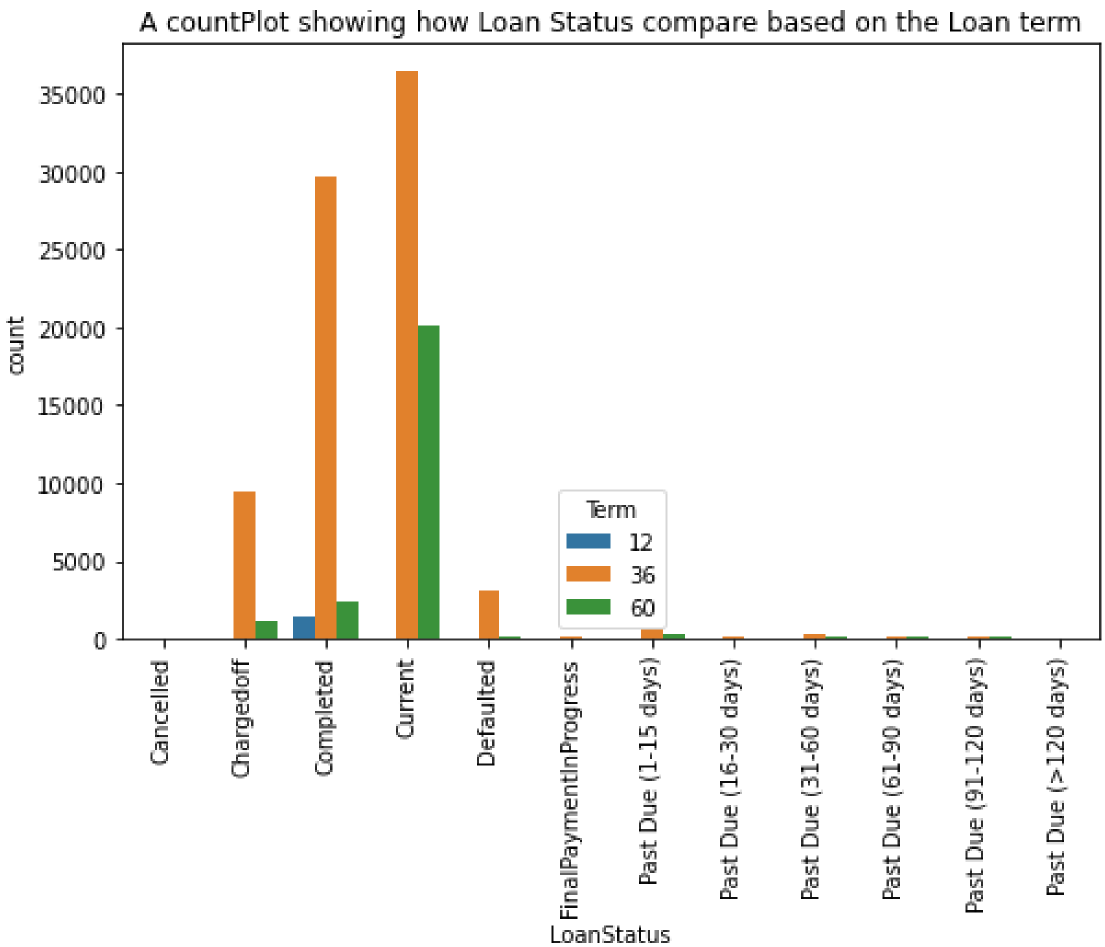
1.5.4 Observations

Term and BorrowerAPR variables are among our features of focus and have been visialized as above. The plot shows that those who paid thier loan in a 36 month term bases were charged higher as shown by the high peak above 0.221% pa.

1.5.5 Question 9

How does the Loan Status compare based on the Loan term?

[149]:
```
plt.figure(figsize=[8,5]);
sb.countplot(data=df1, x='LoanStatus', hue='Term');
plt.title("A countPlot showing how Loan Status compare based on the Loan term");
plt.xticks(rotation = 90);
```



1.5.6 Observations

We mentioned earlier that our focus from the loan status will be on completed, defaulted,finalpaymentInProgress and past due. From the above distribution, loan status in completed category have higher counts of its completed loan status in the 36 month Term and lowest in 12 month Term. Defaulted borrowers fall majorly in the 36 month Term while the total accumulation of borrowers with a Past Due status are within the 36 month Term. FinalPaymentInProgress has no noticeable Term period as seen from the plot above.

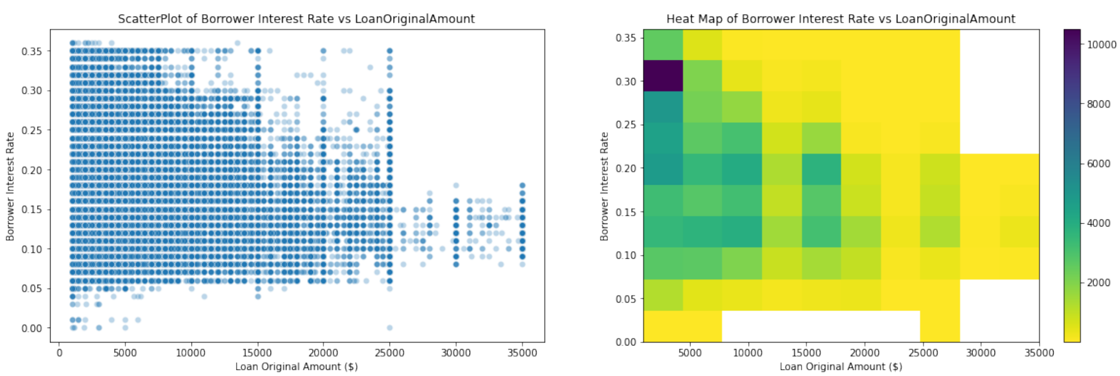
1.5.7 Question 10

How does the Loan original amoount and the Borrower rate affect each other?

```
[109]: plt.figure(figsize=[20,13])

plt.subplot(2,2,1)
sb.scatterplot(data=df1, y='BorrowerRate', x='LoanOriginalAmount', alpha=.3)
plt.ylabel('Borrower Interest Rate')
plt.xlabel('Loan Original Amount ($)')
plt.title('ScatterPlot of Borrower Interest Rate vs LoanOriginalAmount');

plt.subplot(2,2,2)
plt.hist2d(data=df1,  y='BorrowerRate', x='LoanOriginalAmount',␣
 ↪cmap='viridis_r', cmin=0.8)
plt.ylabel('Borrower Interest Rate')
plt.xlabel('Loan Original Amount ($)')
plt.title('Heat Map of Borrower Interest Rate vs LoanOriginalAmount');
plt.colorbar();
```



1.5.8 Observations

From the correlation map and the relationship shown on both the heatmap and scatter plot, a negative correlation clearly exists between the BorrowerAPR and the LoanOriginalAmount and also the Borrower Interest Rate vs Loan Original Amount.Loan original amounts greater than `$20,000` are much more prone to have lower Borrower APR and Borrower Interest Rate compared to lesser amount of `$10,000` and below which are more likely to have higher Borrower APR and Borrower Interest Rate. Thus, there is clearly a negative correlation albeit a weak one.

1.5.9 Question 11

How does the loan term affect the BorrowerAPR and the BorrowerRate?

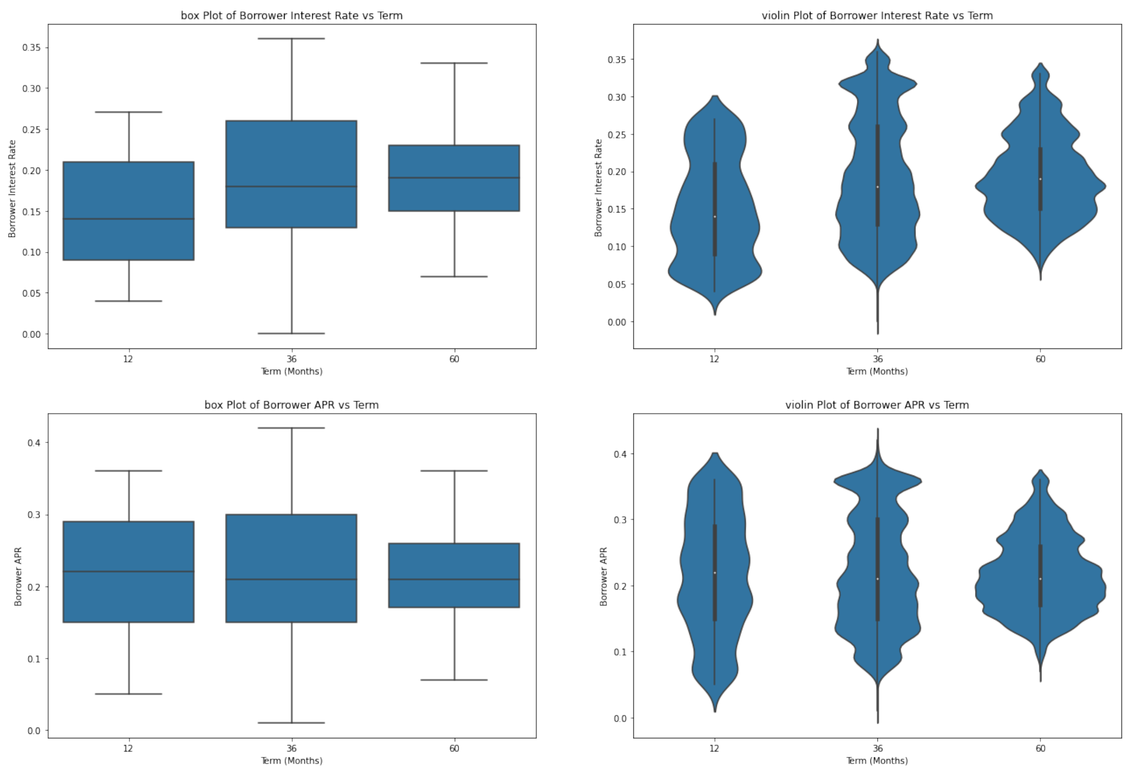
```
[154]: plt.figure(figsize=[22,15])
plt.subplot(2,2,1)
sb.boxplot(data=df1, y='BorrowerRate', x='Term',color=sb.color_palette()[0])
plt.ylabel('Borrower Interest Rate ')
plt.xlabel('Term (Months)')
```

```
plt.title('box Plot of Borrower Interest Rate vs Term');

plt.subplot(2,2,2)
sb.violinplot(data=df1,  y='BorrowerRate', x='Term',color=sb.color_palette()[0])
plt.ylabel('Borrower Interest Rate')
plt.xlabel('Term (Months)')
plt.title('violin Plot of Borrower Interest Rate vs Term');

plt.subplot(2,2,3)
sb.boxplot(data=df1, y='BorrowerAPR', x='Term',color=sb.color_palette()[0])
plt.ylabel('Borrower APR')
plt.xlabel('Term (Months)')
plt.title('box Plot of Borrower APR vs Term');

plt.subplot(2,2,4)
sb.violinplot(data=df1,  y='BorrowerAPR', x='Term',color=sb.color_palette()[0])
plt.ylabel('Borrower APR')
plt.xlabel('Term (Months)')
plt.title('violin Plot of Borrower APR vs Term');
```



1.5.10 Observations

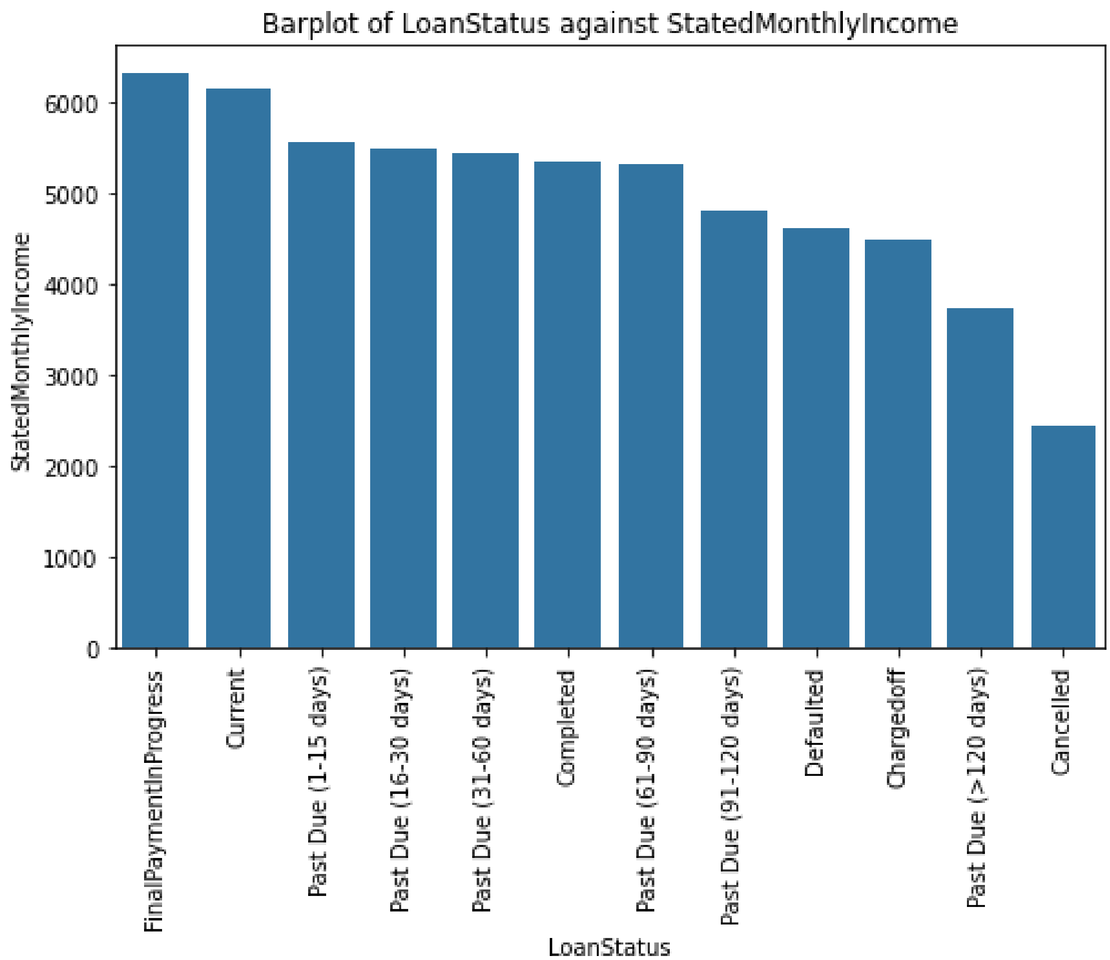
Term has a positive correlation with Borrower Rate and a negative correlation with BorrowerAPR. The violin and box plot when viewed shows Term having strong positive effect on Borrower Interest. A closer assessment using a line plot for the average Borrower APR for all loans shows although there is no considerable effect of Term on Borrower APR, loans with a 36 month term on average still have a slightly higher Borrower APR rates than a 12 and 60 month Term. With the Borrower Rate, a 36 and 60 month Term would have a higher BorrowerRate than a loan of a 12 month Term.

1.5.11 Question 12

How does the Stated monthly income affect the borrower's LoanStatus?

[155]:
```
plt.figure(figsize=[8,5])
color = sb.color_palette()[0];
plot_order = df1.groupby('LoanStatus')['StatedMonthlyIncome'].mean().
↪sort_values(ascending=False).index.values

sb.barplot(data=df1, x='LoanStatus',  y='StatedMonthlyIncome', ␣
↪ci=None,color=color, order=plot_order)
plt.title("Barplot of LoanStatus against StatedMonthlyIncome")
plt.xticks(rotation = 90);
```



1.5.12 Observations

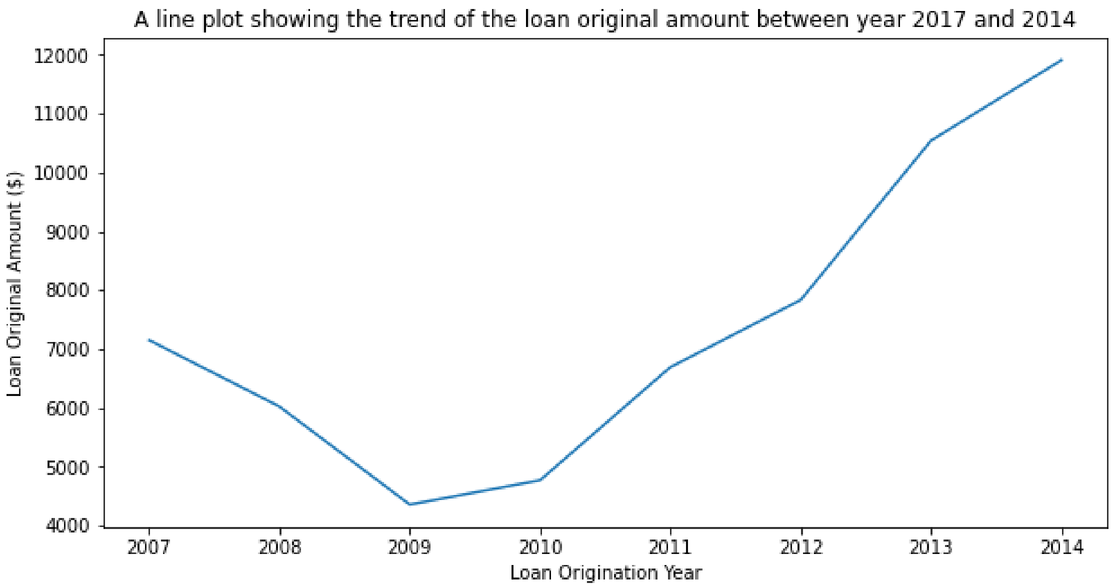
The plot above illustrates the relationship of `LoanStatus` to `StatedMonthlyIncome` and we can say that borrowers with the higher monthly income have their loan at the FinalPayment in progress. Those who have the least monthly income have their loans as cancelled due to various factors such as high interest rates which made it hard for them to pay the loans hence cancelling.

1.5.13 Question 13

What is the trend of the loan original amount between year 2017 and 2014?

[156]:
```
#Line plot for the average loan original amount vs Loan Origination Year
plt.figure(figsize=[10,5])
plt.plot(df1.groupby('LoanOriginationYear')['LoanOriginalAmount'].mean())
plt.xlabel('Loan Origination Year')
plt.ylabel('Loan Original Amount ($)')
```

```python
plt.title("A line plot showing the trend of the loan original amount between␣
 ↪year 2017 and 2014")
plt.show();
```



1.5.14 Observations

We wanted to see which year had the highest amount of loan borrowed by plotting a line gragh which shows that 2009 had the least loan amount issued of below $5,000 and since then, the rate at which the loan was issued increased over the years. This could be high cost of living which compelled people to borrow more.

1.5.15 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?

For the `BorrowerAPR`, a 36 months term is likely to have the highest interest rates paid as compared to other terms.

For the `BorrowerRate`, Loan amount of less than 10,000 tend to have a higher interest rate and that of 25,000 and above have a relatively lower rates of bwteen 0.05 and 0.2.

For the `LoanStatus`, a 36 months term was observed as the majority term in the Current, COmpleted, ChargedOff, Defaulted and all the past due categories as well.

It is also observed that borrowers of 0 income range had access to higher sizes of loans than borrowers in the income range of `Not Employed` and 1-25,000 and also had access to same sizes of loans with those within the IncomeRange of 25,000-50,000. Borrower with the income range of 25,000-100,000+ had access to the highest sizes of loans.

For the `LoanOriginalAmount` we see that 2009 recorded the least amount of loan acquired which later increased gradually over the following years rising form 4,000 to 12,000 in span of five years.

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

It is observed that borrowers with monthly income of 5,300 and above had their loans either completed, current or the the final payment in progress.

2009 have the least amount of loan issued.

We also see that, borrowers who took a loan of 36 months term paid higher interest than of the 12 months and 60 months terms.

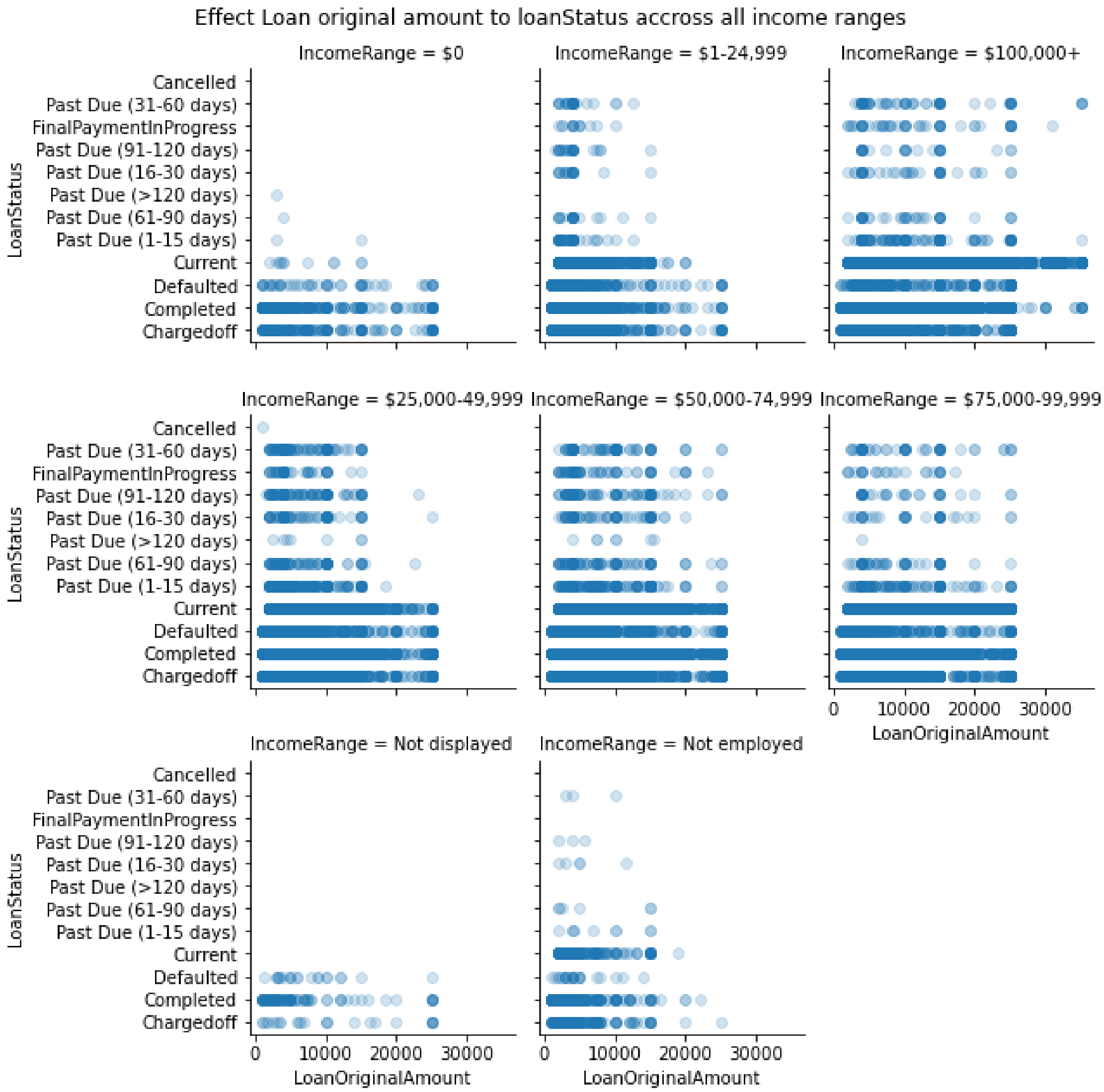
1.6 Multivariate Exploration

1.6.1 Question 14

How does the Loan original amount affect the loanStatus accross all income ranges?

[187]:
```
def mult_var():
    pd_ver = pd.__version__.split(".")
    if (int(pd_ver[0]) > 0) or (int(pd_ver[1]) >= 21): # v0.21 or later
        vclasses = pd.api.types.CategoricalDtype(ordered = True, categories =␣
↪income_range_cat)
        df1['IncomeRange'] = df1['IncomeRange'].astype(vclasses)
    else: # compatibility for v.20
        df1['IncomeRange'] = df1['IncomeRange'].astype('category', ordered =␣
↪True, categories = income_range_cat);
    # plotting
g = sb.FacetGrid(data = df1, col = 'IncomeRange', height = 3, col_wrap = 3);
g.fig.suptitle("Effect Loan original amount to loanStatus accross all income␣
↪ranges");
g.map(plt.scatter, 'LoanOriginalAmount', 'LoanStatus', alpha = 1/5);
# plt.title("Plot to show how Loan original amount affect the loanStatus␣
↪accross all income ranges")
```

[187]:
```
<seaborn.axisgrid.FacetGrid at 0x1b470ddfd60>
```

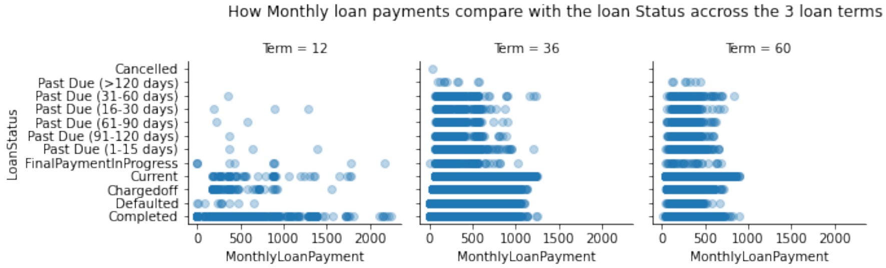


1.6.2 Question 14

How does the Monthly loan payments compare with the loan Status accross the 3 loan terms?

[186]:
```
plt.figure(figsize=[8,5])
g = sb.FacetGrid(data = df1, col = 'Term', col_wrap = 4);
g.fig.suptitle("How Monthly loan payments compare with the loan Status accross␣
↪the 3 loan terms")
g.map(plt.scatter, 'MonthlyLoanPayment','LoanStatus', alpha=0.3);
g.add_legend();
# plt.title("Scatter plot showing how Monthly loan payments compare with the␣
↪loan Status accross the 3 loan terms")
```

```
<Figure size 576x360 with 0 Axes>
```



1.6.3 Observations

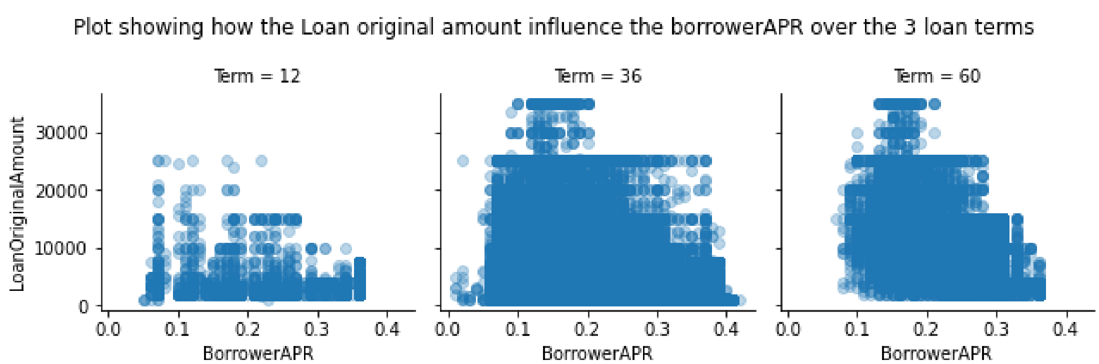
From the plot, we see that borrowers who paid their loans over a 12 months loan term have high amount o pay each month 0f between 0 and 3000 and have just a few of past due loans. However, those of 36 months and 60 months terms have a relatively lower monthly loan payment of between 0 and 1,500 and also have a relatively higher number of past due loans.

1.6.4 Question 15

How does the Loan original amount influence the borrowerAPR over the 3 loan terms?

[184]:
```
axy = sb.FacetGrid(data = df1, col="Term", col_wrap = 3);
axy.fig.suptitle("Plot showing how the Loan original amount influence the␣
↪borrowerAPR over the 3 loan terms");
axy.map(plt.scatter, "BorrowerAPR","LoanOriginalAmount", alpha=0.3);
axy.add_legend();
```

[184]: <seaborn.axisgrid.FacetGrid at 0x1b4720728b0>



1.6.5 Observations

The loan original amount of between 1000 and 25,000 have a term of 36 months and 60 months with a BorrowerAPR of between 0.1 and 0.3. However, loan original amount of below 5,000 have 12 months Loan term with a lower BorrowerAPR of less than 0.1.

1.6.6 Question 16

How does the loan original amount and the monthly loan payment related based on the employment status of the borrower?

[183]:
```
ay = sb.FacetGrid(data=df1, col= "EmploymentStatus", col_wrap=4);
ay.fig.suptitle("Plot of MonthlyLoanPayment and LoanOriginalAmount accross the EmploymentStatus categories");
ay.map(plt.scatter, 'MonthlyLoanPayment', 'LoanOriginalAmount', alpha=.3);
# ay.add_legend()
```

[183]:
```
<seaborn.axisgrid.FacetGrid at 0x1b471c3cb20>
```

1.6.7 Observations

As observed from the plot above, the higher the loan original amount acquired, the higher the monthly payment of the loan accross all the employment status categories.

1.6.8 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?

It is observed that the higher the loan original amount acquired, the higher the monthly payment of the loan accross all the employment status categories.

From the BorrowerAPR, LoanOriginalAmount and Term variables, we can deduce that Loans of less than 10,000 have a lower borrowerAPR rates and were of 12 months term. So it is advisable that borrower should consider taking loan on a short term to enjoy the benefit of the lower rates charged.

1.6.9 Were there any interesting or surprising interactions between features?

From the multivariate variable interaction of BorrowerAPR, LoanStatus, and Term was the observation that Defaulted loan status had a 12 month Term which wasn't noticed in previous exploration plots, and it had a BorrowerAPR greater than that of all the loan statuses categories and their monthly Term

1.7 Conclusions

The prosper loan dataset provided a very large amount of observations defined by 81 variables. I applied data wrangling and cleaning teachniques to make sure I am left with the data I need for my exploration. After all the wrangling and cleaning, I got 106,312 observations and 17 variables. I then added a new variable but extracting the year variable from the date making the tatol number of variables to be 18.

The main questions that guided the analysis of the dataset were as below

- What are the factors that affected the `BorrowerRate` and `Borrower interest rates`.
- How does the `monthly stated income` affect the loan status and the monthly loan payment.
- How does employment status affect the loan original amount.

From the analysis, we see the `Term` affecting the BorrowerAPR in a way that lower borrower apr rates are charged for short term loans of 12 months.

Loan original amount of less than 10,000 have a lower borrowerAPR rates and were of 12 months term. So it is advisable that borrower should consider taking loan on a short term to enjoy the benefit of the lower rates charged.

Loans of borrowers with monthly income of 2,500 and below were cancelled which could be due to high amount of monthly loan payments. However,majority of borrowers with an income of above 5,000 had their loans at the final payment, some completed,some current and also some had their loans past due.

Borrowers with higher income had higher monthly loan payments.

It was observed that employed borrowers accessed higher loans sizes than the NotEmployed/part-time borrowers.

```
[ ]:
```