Lending Club Case Study - EDA

- Ramkumar Raman

Data Sourcing and Data Cleaning

Code for Data Sourcing and Data cleaning

```
#importing the required libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
#To ignore warnings
warnings.filterwarnings('ignore')
#Data Sourcing
#reading the dataset in low memory set to false because data contains mixed data types
df= pd.read_csv("D:\\loan.csv", low_memory=False)
#Data Cleaning
#Including the columns which I planned to use which is having NA values
df['emp_length'].fillna("0 years", inplace=True)
#Dropping the columns having NA values
cleaned_df= df.dropna(axis=1)
#Standardizing the Annual Income data to 2 decimal values
cleaned_df['annual_inc']=cleaned_df['annual_inc'].round(2)
#Removing Outliers
Q1 = cleaned_df['annual_inc'].quantile(0.10)
Q3 = cleaned_df['annual_inc'].quantile(0.95)
IQR = Q3 - Q1
threshold = 1.5
cleaned_df_no_outliers = cleaned_df[~((cleaned_df['annual_inc'] < (Q1 - threshold * IQR)) | (cleaned_df['annual_inc'] > (Q3 +
      threshold * IQR)))]
cleaned_df_no_outliers['int_rate'] = cleaned_df_no_outliers['int_rate'].str.rstrip('%').astype(float)
```

UNIVARTIATE ANALYSIS

Before starting the analysis, Here I have listed the variables took into account for analysis. Columns for EDA (not as in the file)

1. Term
2. Interest Rate
3. Grade
4. SubGrade
5. Employment Length
6. Home Ownership
7. Annual Income
8. Verification Status
9. Address State
10. Purpose
11. Open Account
12. Total Account
13. Loan Status
14. Funded Amount
15. Funded Amount Investors
16. Loan Amount

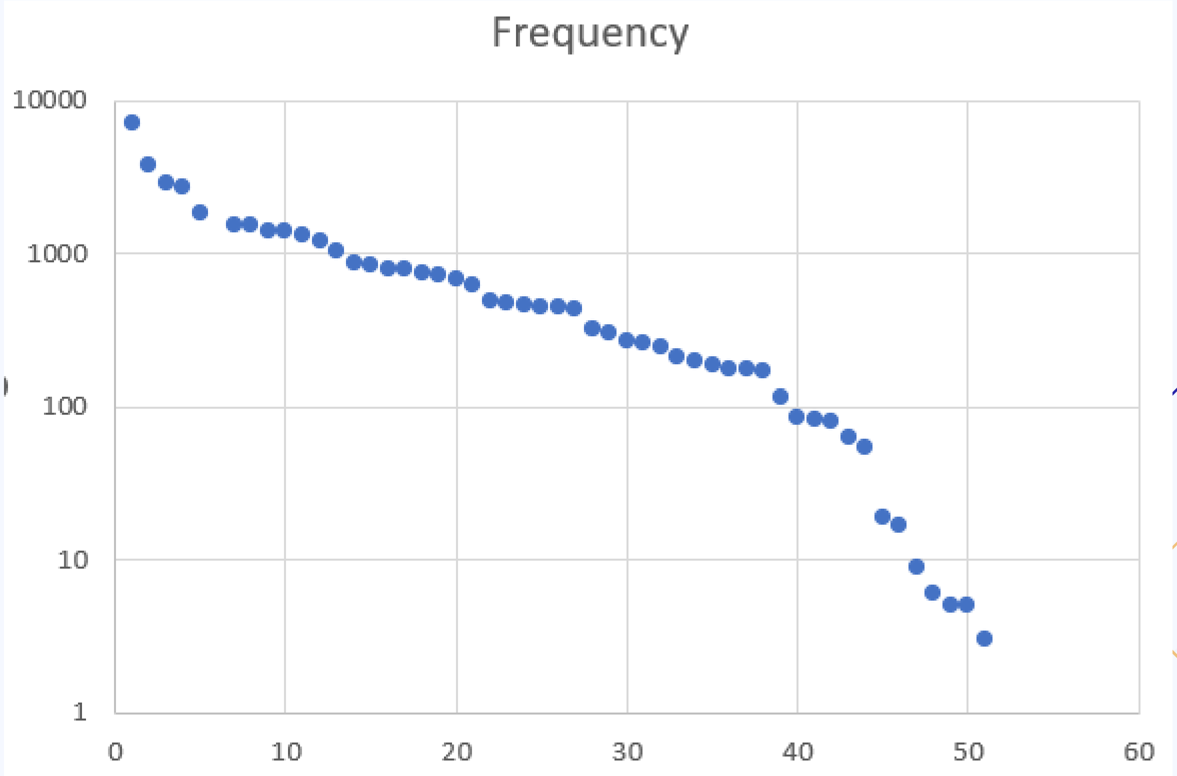
Analysis on Purpose - An Unordered Categorical Variable

As we can see below, the main reason for borrowing the money is for debt_consolidation and the least for making their energy consumption as renewable_energy

Row Labels	Count of purpose
car	1549
credit_card	5130
debt_consolidation	18641
educational	325
home_improvement	2976
house	381
major_purchase	2187
medical	693
moving	583
other	3993
renewable_energy	103
small_business	1828
vacation	381
wedding	947
Grand Total	**39717**

Analysis on Address State - An Unordered Categorical Variable

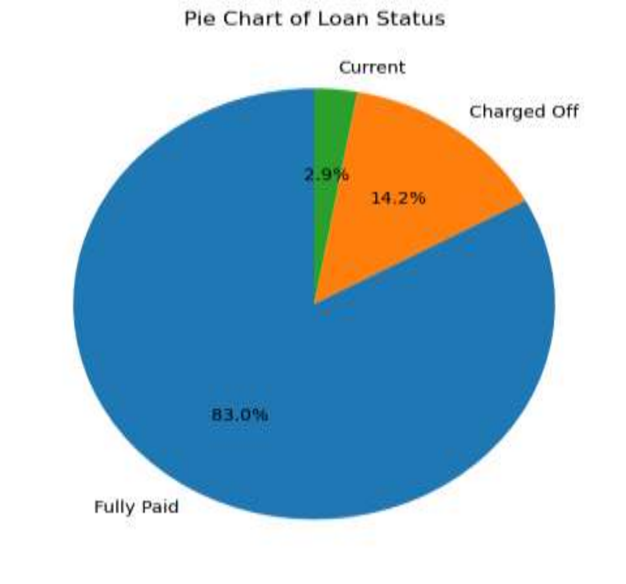
Here I have plotted a Scatter Plot for the Address State variable in terms of Rank (x-axis) and Frequency (y-axis). As per the Power Law Distribution, The Plot obeys it and there is no anomaly detected in the pattern



Code and Results for Univariate analysis

```
#Univariate Analysis
#Analysis on Loan Status

plt.figure(figsize=(6, 6))
loan_status_counts =
df['loan_status'].value_counts()
plt.pie(loan_status_counts,
labels=loan_status_counts.index,
autopct='%1.1f%%', startangle=90)
plt.title('Pie Chart of Loan Status')
plt.show()
```



As we can see, Most of the borrowers paid the loan and some of them are currently paying the loan.

Code and Results for Univariate analysis

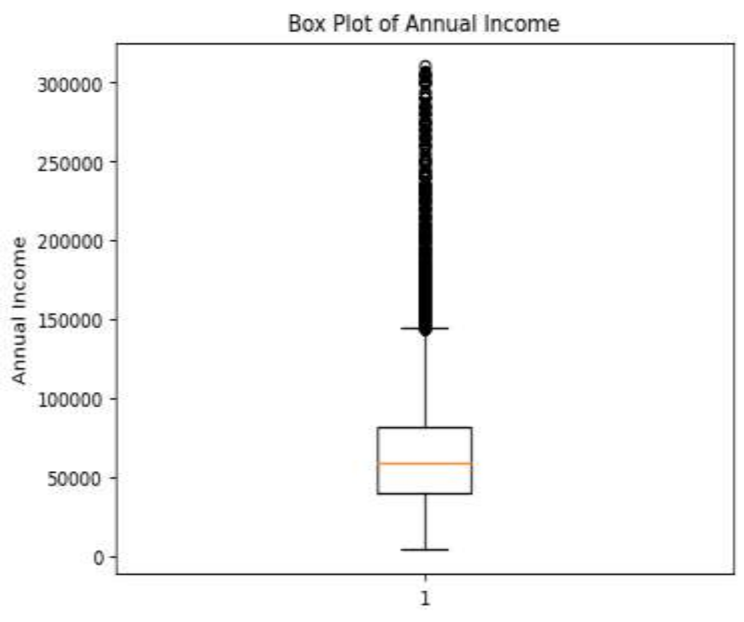
```python
#Univariate Analysis
#Analysis on Annual Income - A Quantitative Variable (After removing the outliers)
mean_income = cleaned_df_no_outliers['annual_inc'].mean().round(2)
print("Mean Income - ",mean_income)
median_income = cleaned_df_no_outliers['annual_inc'].median()
print("Median Income - ",median_income)
```

```
Mean Income -  66728.26
Median Income -  58947.0
```

From the above extracted data, we can see that the people having around 55 k to 70 k annual income likely to get a loan. I have removed the outliers using Interquartile difference already.

Code and Results for Univariate analysis

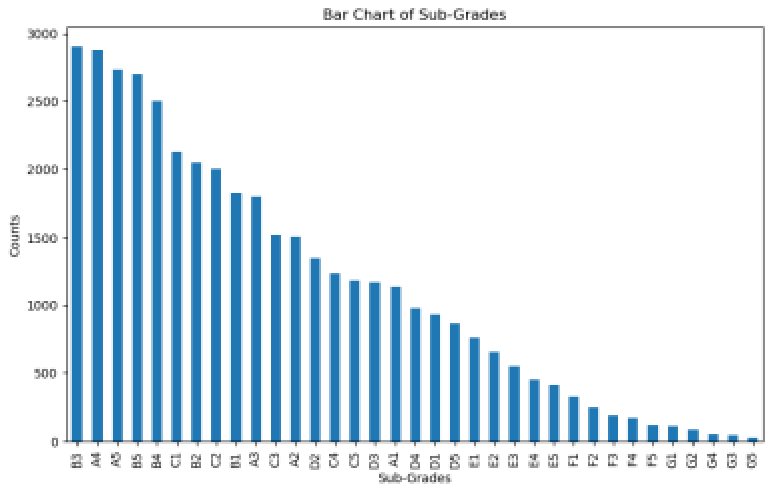
```
#Univariate Analysis
#Summary Metrics
#Box Plot on Annual Income
plt.boxplot(cleaned_df_no_outliers['
annual_inc'])
plt.title('Box Plot of Annual
Income')
plt.ylabel('Annual Income')
plt.show()
```



The above BoxPlot shows the spread of Annual Income and we can see most of the borrowers are around 40k to 70k

Code and Results for Univariate analysis

```
#Univariate Analysis
#Bar Chart on Sub Grades
SubGrade_Counts=
cleaned_df_no_outliers['sub_grade'].
value_counts()
plt.figure(figsize=(10, 6))
SubGrade_Counts.plot(kind='bar')
plt.xlabel('Sub-Grades')
plt.ylabel('Counts')
plt.title('Bar Chart of Sub-Grades')
plt.show()
```



As we can see, Most borrowers falls under sub grade of 'B3' and 'A4' and G5 sub grade has the least borrowers.

Segmented Univariate Analysis

Analysis on Annual Income across House Ownership

From below results we can see that the average annual income of people who got mortgage property is more than that of who lives for rent.

Row Labels	Average of annual_inc
MORTGAGE	83116.96395
NONE	80733.33333
OTHER	71309.71429
OWN	58863.32245
RENT	57370.32597
Grand Total	**68968.92638**

Analysis on Annual Income across Employee Experience

In below image, We can see as the experience increases , the annual income increases.

Row Labels	Average of annual_inc
< 1 year	60860.23403
1 year	62644.61963
10+ years	81706.53435
2 years	63274.65855
3 years	66787.18145
4 years	66583.75697
5 years	68225.20415
6 years	68184.6186
7 years	69153.09694
8 years	74590.46852
9 years	74474.42921
Grand Total	**69608.27721**

Analysis on Funded amount across Verification

The image shows that the Verified users are more likely to get more loan amount than the others.

Row Labels	Average of funded_amnt
Not Verified	8293.769576
Source Verified	9880.016521
Verified	15286.10547
Grand Total	**10947.7132**

Analysis on Loan Amount across Terms

As we can see here, The more the term duration, more the loan amount

Row Labels	Average of loan_amnt
36 months	9592.936314
60 months	15675.22597
Grand Total	**11219.44381**

Analysis on Loan Amount, Funded Amount and Funded Amount Investors across Grades

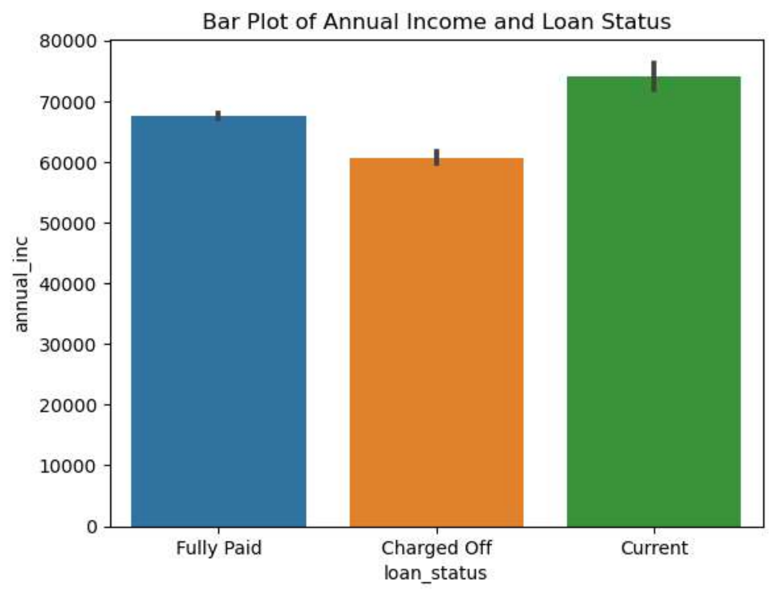
As we can see, The amounts are in the increasing order according to the grades from A to G but I can see a slight anomaly on Grade B and C.

Row Labels	Average of funded_amnt_inv	Average of loan_amnt	Average of funded_amnt
A	8155.229508	8624.928111	8402.421914
B	10327.1622	11119.0807	10861.33527
C	10051.27565	11004.67091	10779.2634
D	11381.81586	12278.19861	12069.7899
E	14461.886	15847.25545	15254.32794
F	16891.62749	18363.29838	17688.41754
G	18857.51014	20226.81962	19828.63924
Grand Total	**10397.44887**	**11219.44381**	**10947.7132**

Bivariate Analysis

Code and Results for Bivariate analysis

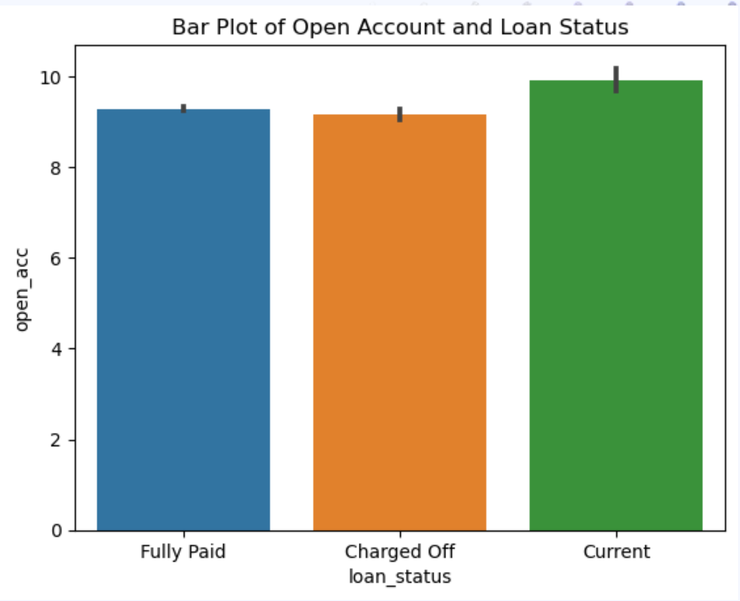
```
#Bivariate Analysis on Loan
Status and Annual Income
sns.barplot(x='loan_status',y='an
nual_inc',data=cleaned_df_no_outl
iers)
plt.title('Bar Plot of Annual
Income and Loan Status')
plt.show()
```



By comparing the annual incomes,the borrowers who has more annual income likely to repay the loan fully.

Code and Results for Bivariate analysis

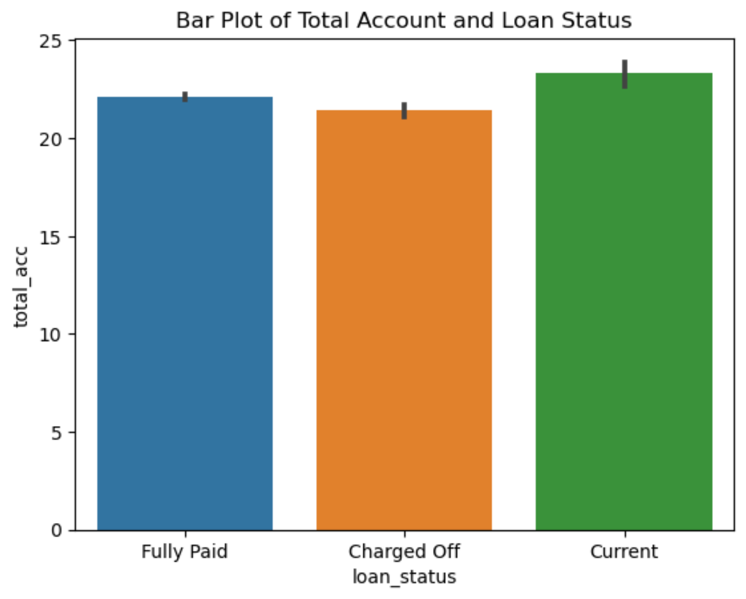
```
#Bivariate Analysis on Loan
Status and Open Account
sns.barplot(x='loan_status',y='op
en_acc',data=cleaned_df_no_outlie
rs)
plt.title('Bar Plot of Open
Account and Loan Status')
plt.show()
```



By comparing, The borrowers who fully paid the loan and defaulted has most likely the same number of open credit lines.

Code and Results for Bivariate analysis

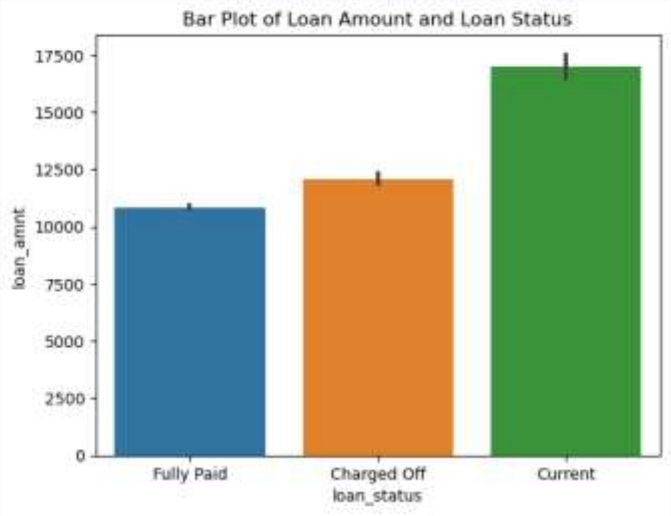
```
#Bivariate Analysis on Loan Status and Total Account
sns.barplot(x='loan_status',y='total_acc',data=cleaned_df_no_outliers)
plt.title('Bar Plot of Total Account and Loan Status')
plt.show()
```



By comparing, The borrowers who fully paid the loan has slightly having more total credit lines than Defaulters.

Code and Results for Bivariate analysis

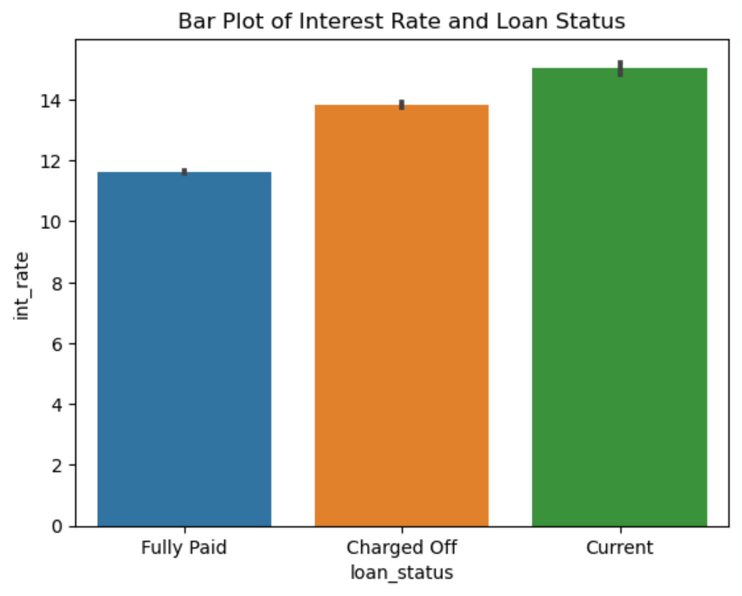
```
#Bivariate Analysis on Loan
Status and Loan Amount
sns.barplot(x='loan_status',y='lo
an_amnt',data=cleaned_df_no_outli
ers)
plt.title('Bar Plot of Loan
Amount and Loan Status')
plt.show()
```



By comparing, The fully paid borrowers likely to ask for less loan amount than the Defaulters.

Code and Results for Bivariate analysis

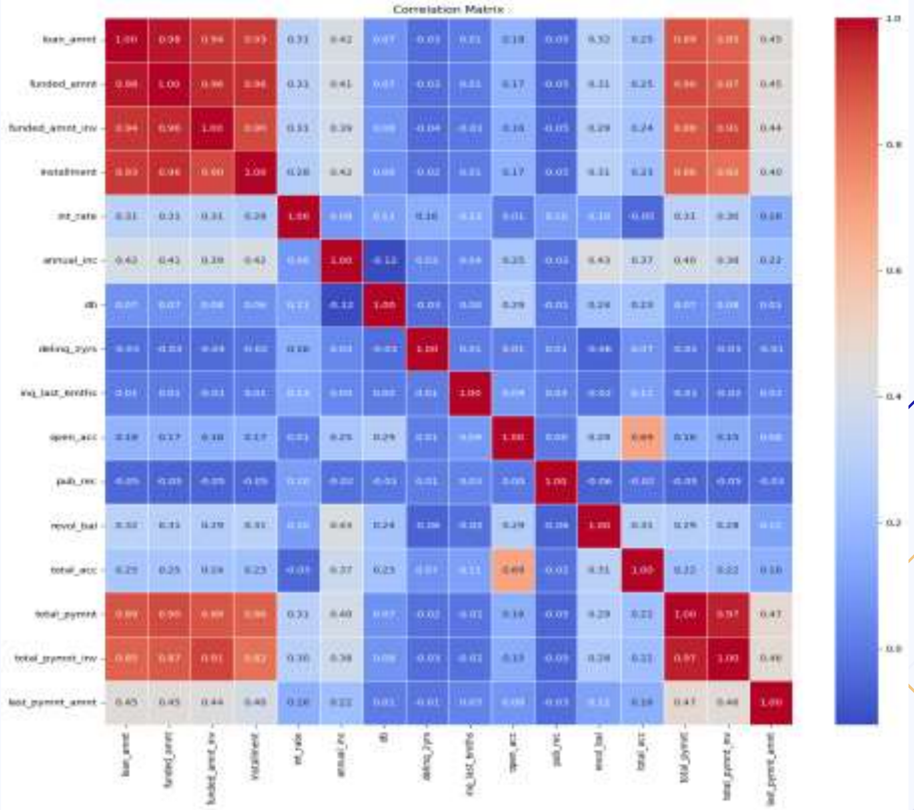
```
#Bivariate Analysis on Loan
Status and Interest Rate
sns.barplot(x='loan_status',y='in
t_rate',data=cleaned_df_no_outlie
rs)
plt.title('Bar Plot of Interest
Rate and Loan Status')
plt.show()
```



By comparing, With Less Interest rates, the loan is more likely to be paid.

Code and Results for Bivariate analysis

```
#Bivariate Analysis
#Finding Correlations
loan_correlation =
cleaned_df_no_outliers[['loan_amnt','funded_amnt','funded_am
nt_inv','installment','int_rate','annual_inc','dti','delinq_
2yrs','inq_last_6mths','open_acc','pub_rec','revol_bal','tot
al_acc',  'total_pymnt','total_pymnt_inv','last_pymnt_amnt']
]
correlation = loan_correlation.corr()
plt.figure(figsize=(15,15))
sns.heatmap(correlation, annot = True, cmap='coolwarm',
fmt='.2f', linewidths=0.5)
plt.title("Correlation Matrix")
plt.show()
```



Here the loan amount is most correlated with the funded amount i.e., as the loan amount increases/decreases, the funded amount will increase/decrease. The DTI and the annual income are less correlated i.e., as the annual income increases, the DTI decreases and vice versa.

Derived Metrics

Code and Results for Derived Metrics

```
#Adding a new Business driven column
bins = [5, 10, 20, 25]
labels = ['Low', 'Medium', 'High']
cleaned_df_no_outliers['int_rate_category'] = pd.cut(cleaned_df_no_outliers['int_rate'], bins=bins, labels=labels, right=False)
print(cleaned_df_no_outliers[['int_rate', 'int_rate_category']].to_string())
```

```
    int_rate int_rate_category
0      10.65            Medium
1      15.27            Medium
2      15.96            Medium
3      13.49            Medium
4      12.69            Medium
5       7.90               Low
6      15.96            Medium
7      18.64            Medium
8      21.28              High
9      12.69            Medium
10     14.65            Medium
11     12.69            Medium
12     13.49            Medium
13      9.91               Low
```