Lending club case study

Contents

Table of Contents

Study for Loan Data

❖ Data Sourcing

- **Import the useful libraries**
    - Numpy
    - Pandas
    - Seaborn
    - Matplotlib.pyplot

- **Read the Data set**

- **Apply Filter to keep only required records for the analysis**

    loanDf=loan[(loan.loan_status.str.strip() =='Fully Paid') | (loan.loan_status.str.strip() =='Charged Off')

❖ Data Cleaning

- **Dropping unnecessary columns**

    Drop columns that are not required in analysis

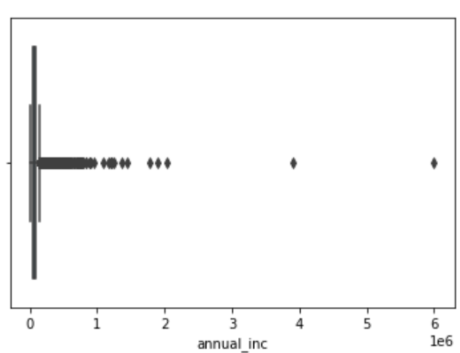
```
#We can remove id columns,not required columns as those are not helpful in analysis
loanDfwoId=loanDf_nonmissing.drop(["id","member_id","url","desc","zip_code","mths_since_last_delinq","addr_state","last_credit_pu
```

- **Impute/Remove missing values**
    1. calculate the percentage of the missing values
    2. drop the columns having more than 30% missing

```
#As we can see around 60 columns having most of the values as null . We can remove them to make data cleaner
loanDf_nonmissing=loanDf[loanDf.columns[loanDf.isnull().mean()<=0.7]]
```

    3. Imputing missing values for emp_length. We replaced missing with 0.
    4. Imputing pub_rec_bankruptcies with ‘unknown’ category.
    5. Remove outliers on annual income as we can see from boxplot:

```
In [74]: sns.boxplot(loanDf.annual_inc)

C:\Users\bhavit\anaconda\lib\site-packages\seaborn\_decorators.py:36: Fu
rg: x. From version 0.12, the only valid positional argument will be `d
yword will result in an error or misinterpretation.
  warnings.warn(

Out[74]: <AxesSubplot:xlabel='annual_inc'>
```



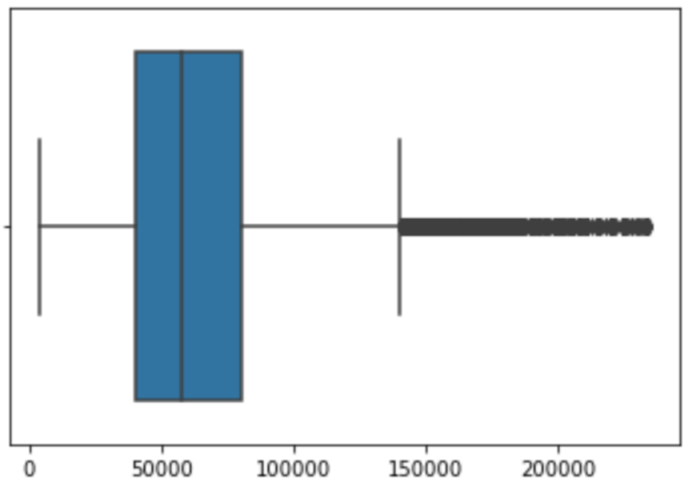

```
#Remove Outliers quantile .99
loanDfwoId = loanDfwoId[loanDfwoId["annual_inc"] < loanDfwoId["annual_inc"].quantile(0.99)]
```

Post removal of outliers:

```
sns.boxplot(loanDfwoId.annual_inc)
```

```
C:\Users\bhavit\anaconda\lib\site-packages\seaborn\_decorators.py:36: Future
rg: x. From version 0.12, the only valid positional argument will be `data`,
yword will result in an error or misinterpretation.
  warnings.warn(
```

```
<AxesSubplot:xlabel='annual_inc'>
```



6. clean interest rate ,revolt rate by removing % sign as well as clean term.

```
# clean interest rate ,revolt rate by removing % sign as well as clean term
loanDfwoId['int_rate'] = loanDfwoId['int_rate'].str.replace('%','')
loanDfwoId['revol_util'] = loanDfwoId['revol_util'].str.replace('%','')
loanDfwoId['term'] = loanDfwoId['term'].str.replace(' months','')
# Clean emp_length column to have only numbers.
replace_dict=dict(zip(["years", "year", "\+","<"," "], [""]*5))
loanDfwoId['emp_length']=loanDfwoId['emp_length'].str.strip().replace(replace_dict,regex=True)
```

7. convert amount columns into numeric data. So that we can see the correlation between these columns.

```
# convert amount columns into numeric data.

amt_cols = ['loan_amnt','funded_amnt','int_rate','funded_amnt_inv','installment','annual_inc','dti','emp_length','total_pymnt']
loanDfwoId[amt_cols] = loanDfwoId[amt_cols].apply(pd.to_numeric)
```

- **Derive new columns from existing ones**

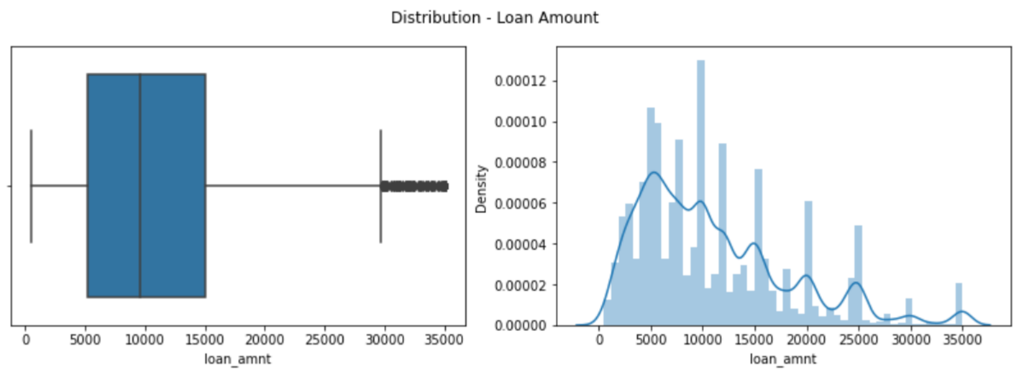
Derived new columns from the existing one like month and year from issue date. Also created bucketed column on loan amount, interest rates, debt to income ratio(dti) as well as on annual income.

```
#### derive columns from existing columns
# create month and year columns separately
loanDfwoId['issue_d'] = pd.to_datetime(loanDfwoId.issue_d, format='%b-%y')
loanDfwoId['month'] = loanDfwoId.issue_d.dt.strftime('%b')
loanDfwoId['year']=loanDfwoId['issue_d'].dt.year

# categorise loan amounts into buckets .
loanDfwoId['loan_amnt_cats'] = pd.cut(loanDfwoId['loan_amnt'], [0, 5000, 10000,15000, 20000,

# categorise annual incomes into buckets
loanDfwoId['annual_inc_cats'] = pd.cut(loan['annual_inc'], [0, 20000, 40000, 60000, 80000,100

# categorise intrest rates into buckets
loanDfwoId['int_rate_cats'] = pd.cut(loanDfwoId['int_rate'], [0, 10, 12.5, 15, 20], labels=['

# categorise dti into buckets .
loanDfwoId['dti_cats'] = pd.cut(loanDfwoId['dti'], [0, 5, 10, 15, 20, 25], labels=['0-5', '05
```

❖ Analysis

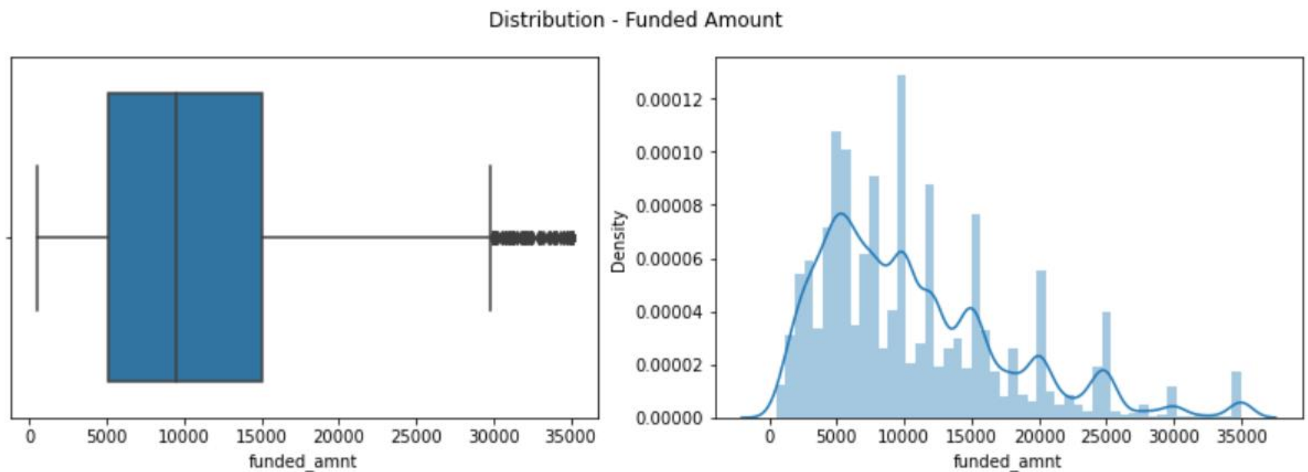
Univariate Analysis

- **Distribution of loan amount**



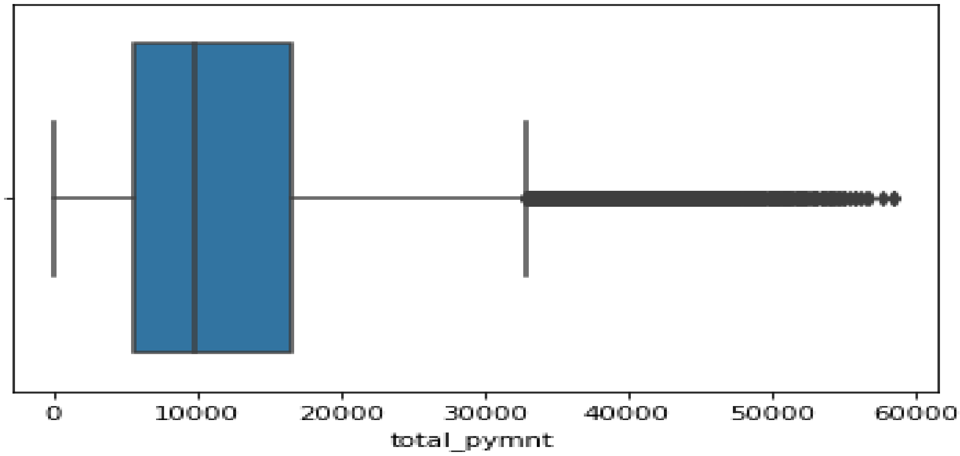
From the above plots, we can see that loan amount is varying between 5000 and 15000 for almost 50 % people and median is around 9600. 90 percent of loans are below 21,000.

- **Distribution of funded amount:**



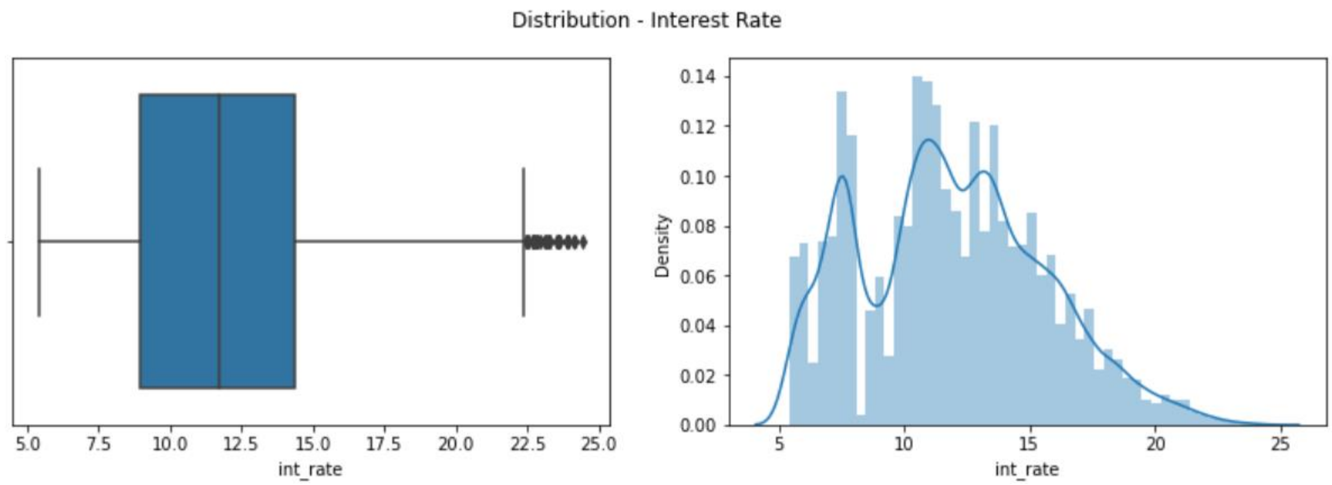
Observations:
Funded amount data has same distribution as of loan Amount, so we can say that approved loan is almost same as Applied loan amount.

- **Distribution of total payment:**



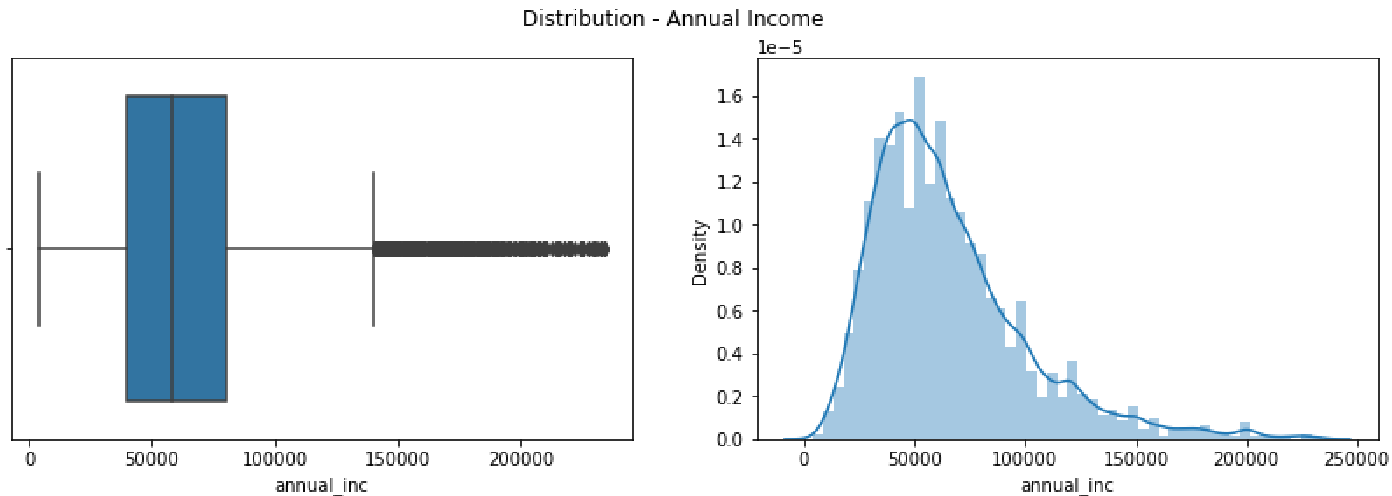
Payment amount data shows variation between 6000 and 16000 for 50% people, so we can say that loan having around 10% return on loan_amount (as its having variation between 5000 and 15000) .

- **Distribution of interest rates:**



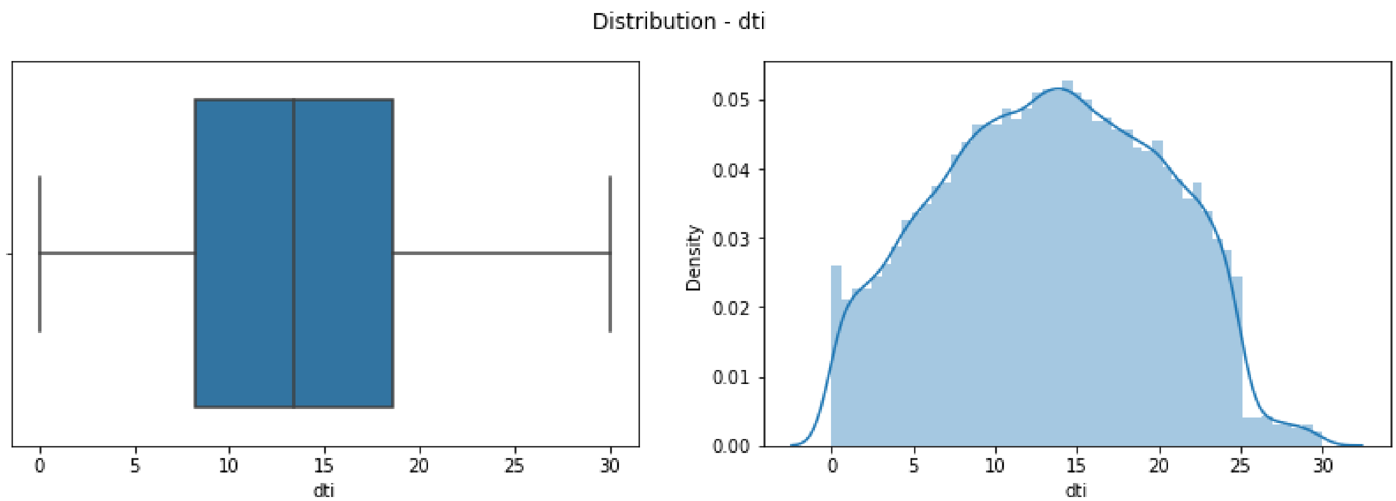
Observations:
From the above 2 plots of interest rates we can conclude that most of the interest rates lies are in the range of **9% to 14.5%**. There are some exceptions/outliars i.e., **22.5+** %

- **Distribution of Annual Income:**



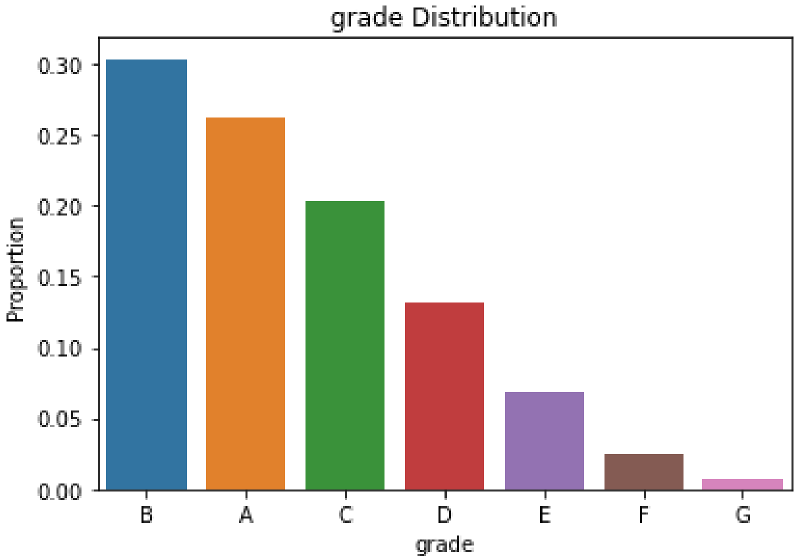
Observations:
Business observation: Most of the people having annual income between 40k and 80K who got loan approved.

- **Distribution of Debt-to-Income ratio:**



Observations:
As you can see dti is between 2 and 30. So, it’s a healthy sign that loan is disbursed to good saving people so that loan defaulters can be less.

- **Analysis of grade:**



Observations: As you can see most of the borrower having B and A grades.

Segmented Univariate Analysis

Bivariate Analysis

Overall correlation between important numeric feature columns