Lending Club Case Study

by

Basanth Rachakonda,

Raneeth Kumar Parsi

Problem statement

- The lending club company wants to understand the driving factors (or driver variables) behind loan default, i.e. the variables which are strong indicators of default. The company can utilise this knowledge for its portfolio and risk assessment.

Introduction

- The given data information about past loan applicants and whether they ‘defaulted’ or not. The aim is to identify patterns which indicate if a person is likely to default, which may be used for taking actions such as denying the loan, reducing the amount of loan, lending (to risky applicants) at a higher interest rate, etc.

Data Cleaning

- step-1

- we need to identify the columns with Null values, and need to remove them.
- we have 54 columns having complete Null values, those are not needed for analysis, so we are dropped them.

- step-2

- after dropping the complete Null value columns, we observed that there are some columns with more number of Null values (greater than 10000 Null values), we dropped those columns as well.
- columns with more number of Null values -["desc", "mths_since_last_delinq", "mths_since_last_record", "next_pymnt_d"]

Data Cleaning(contd..)

- step-3

- There are some customer behaviour columns those are the variables not available at the time of loan application, and thus they cannot be used as predictors for credit approval. we do not need to consider them.
- csutomer behaviour columns = ["delinq_2yrs", "earliest_cr_line", "open_acc", "pub_rec", "revol_bal", "revol_util", "total_acc", "out_prncp", "out_prncp_inv", "total_pymnt", "total_pymnt_inv", "total_rec_prncp", "total_rec_int", "total_rec_late_fee", "recoveries", "collection_recovery_fee", "last_pymnt_d", "last_pymnt_amnt", "last_credit_pull_d", "application_type"]

- step-4

- There are some mpre columns are not needed for analysis, need to drop them also. Those are ["id", "member_id", "emp_title", "url", "title", "zip_code", "addr_state"]

we dropped all unwanted columns those are not helpful for analysis..

categorical variables, continuous variables

by considering unique value count in columns we devided categorical columns and continuous columns.

- cnt_columns= ["loan_amnt", "funded_amnt", "funded_amnt_inv", "int_rate", "installment", "annual_inc", "dti"]
- cat_columns= ["term", "grade", "sub_grade", "emp_length", "home_ownership", "verification_status", "loan_status", "pymnt_plan", "purpose", "pub_rec_bankruptcies"]

cnt_columns = continuous variables, cat_columns = categorical variables

Data Standardisation & Data Manipulation

The goal of the analysis is to see who is likely to default and this can only be said in case of either fully paid or charged off loans.

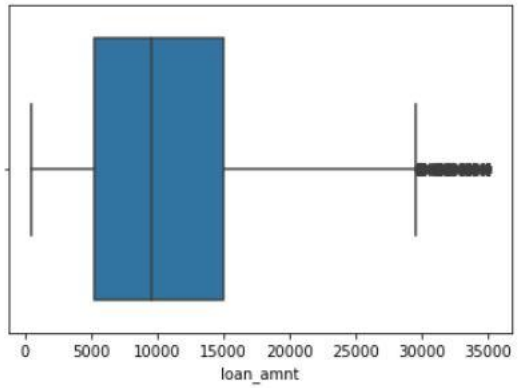
- We cannot make anything up for the current loans.
- To exclude that data , removing the records with current loan status

we obeserved that int_column is string, we converted that as float and removed ‘%’ symbol as well.

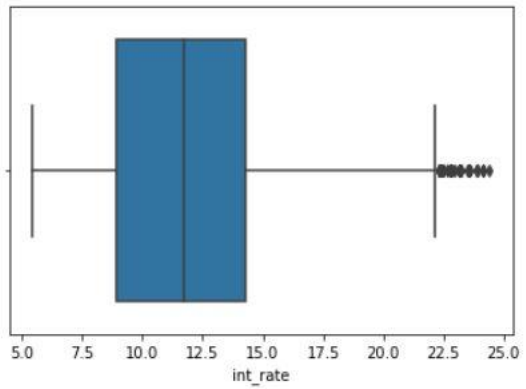
Removel of outliers

we observed ouliers for diffrent columns by ploting boxer plots.

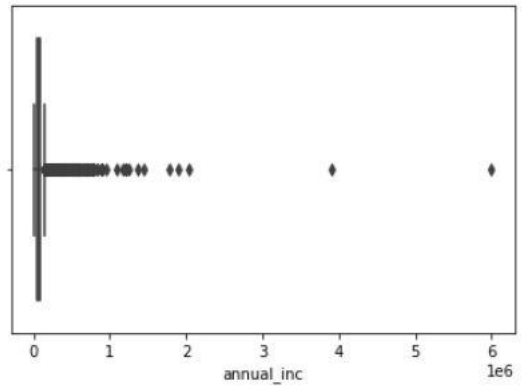
```
In [30]: sns.boxplot(data['loan_amnt'])
Out[30]: <AxesSubplot:xlabel='loan_amnt'>
```




```
In [37]: sns.boxplot(data['int_rate'])
Out[37]: <AxesSubplot:xlabel='int_rate'>
```



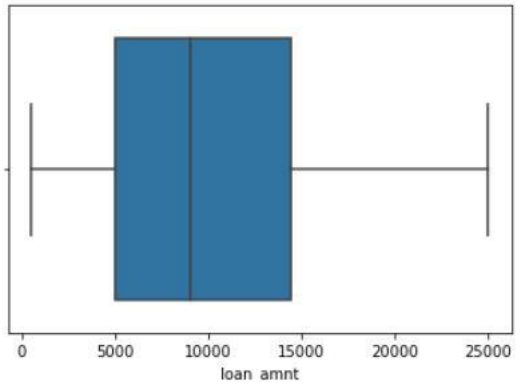

```
In [43]: sns.boxplot(data['annual_inc'])
Out[43]: <AxesSubplot:xlabel='annual_inc'>
```



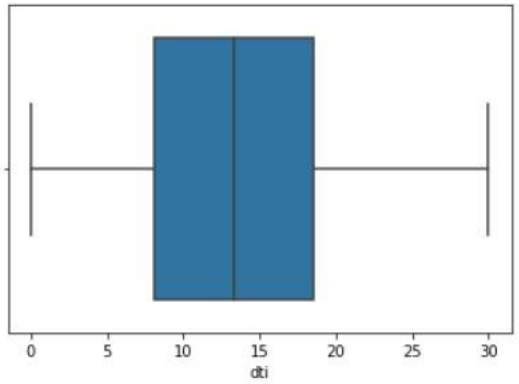
Removel of outliers(contd..)

- we removed outliers

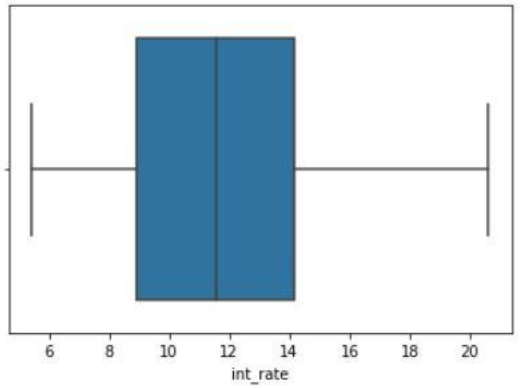
```
In [34]: sns.boxplot(data['loan_amnt'])
Out[34]: <AxesSubplot:xlabel='loan_amnt'>
```



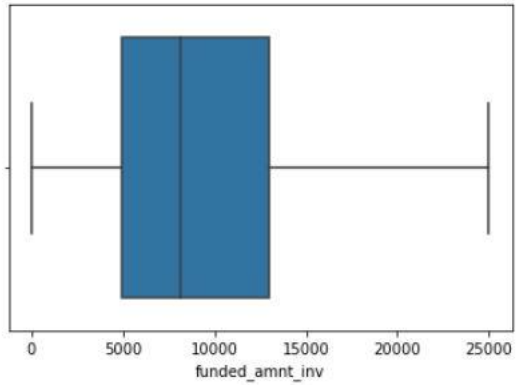

```
In [42]: sns.boxplot(data['dti'])
Out[42]: <AxesSubplot:xlabel='dti'>
```



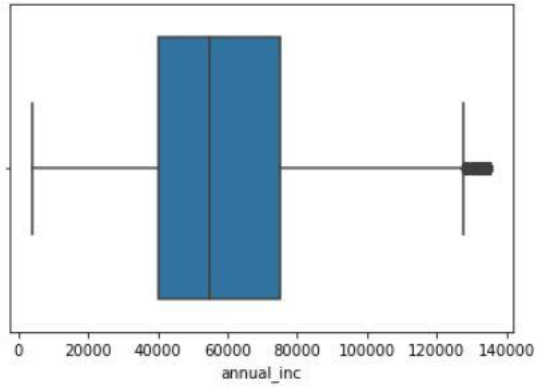

```
In [41]: sns.boxplot(data['int_rate'])
Out[41]: <AxesSubplot:xlabel='int_rate'>
```



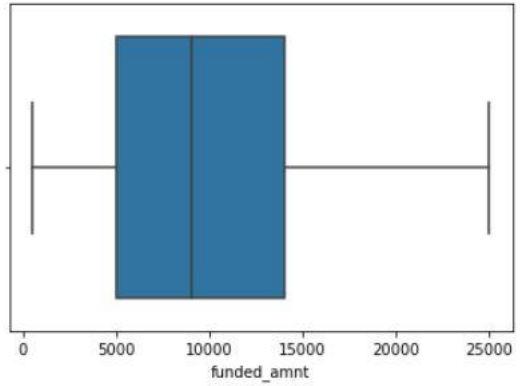

```
In [36]: sns.boxplot(data['funded_amnt_inv'])
Out[36]: <AxesSubplot:xlabel='funded_amnt_inv'>
```




```
In [46]: sns.boxplot(data['annual_inc'])
Out[46]: <AxesSubplot:xlabel='annual_inc'>
```



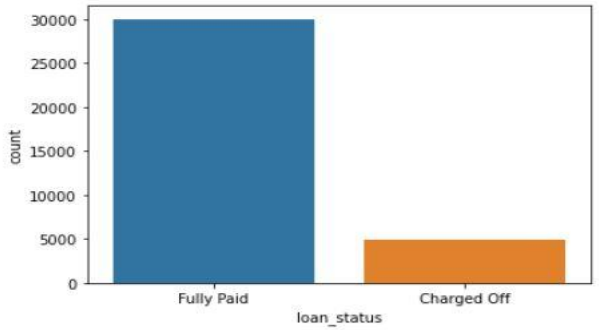

```
In [35]: sns.boxplot(data['funded_amnt'])
Out[35]: <AxesSubplot:xlabel='funded_amnt'>
```



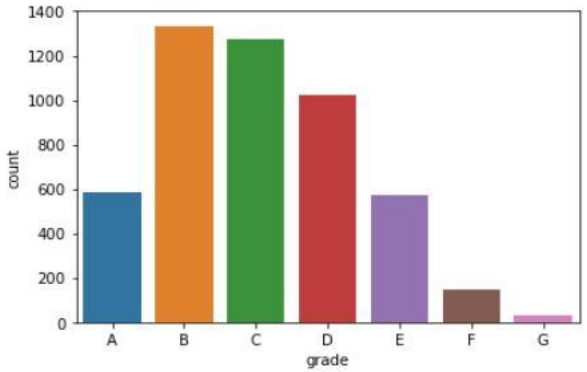
Univariate analysis on categorical columns

```
In [49]: sns.countplot(x = 'loan_status', data = data)
```

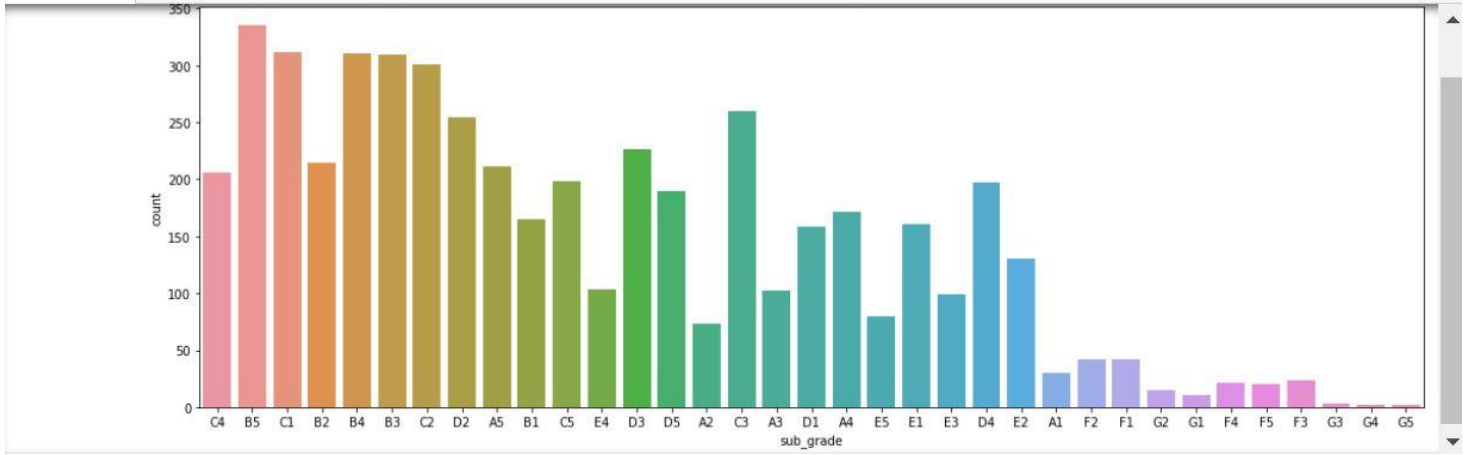
```
Out[49]: <AxesSubplot:xlabel='loan_status', ylabel='count'>
```



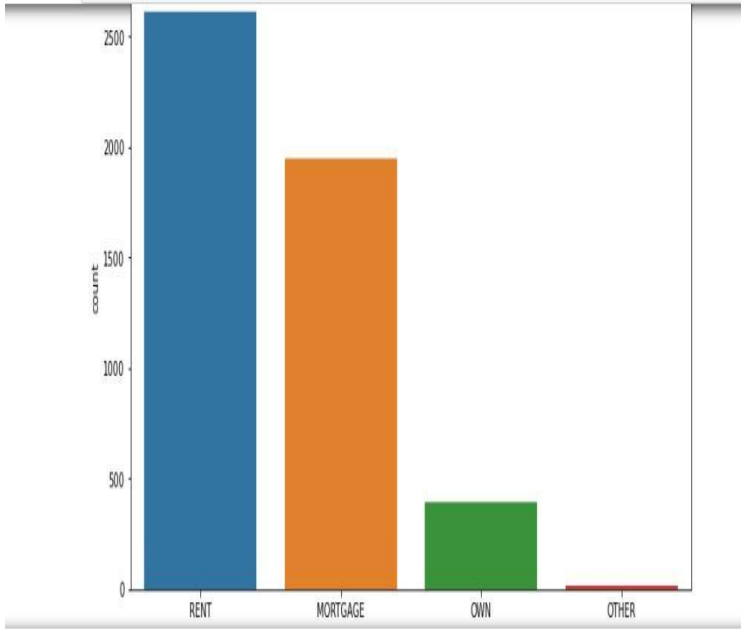

```
Out[51]: <AxesSubplot:xlabel='grade', ylabel='count'>
```




```
In [52]: fig, ax = plt.subplots(figsize = (18,6))
         sns.countplot(x = 'sub_grade', data = data[data.loan_status == 'Charged Off'])
```

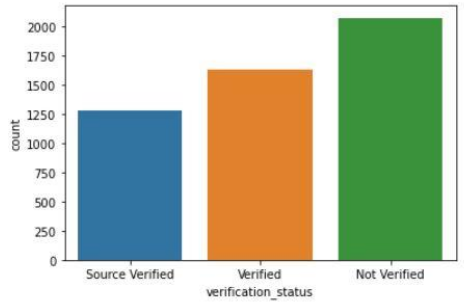



```
In [53]: fig, ax = plt.subplots(figsize = (12,6))
         sns.countplot(x = 'home_ownership', data = data[data.loan_status == 'Charged Off'])
```

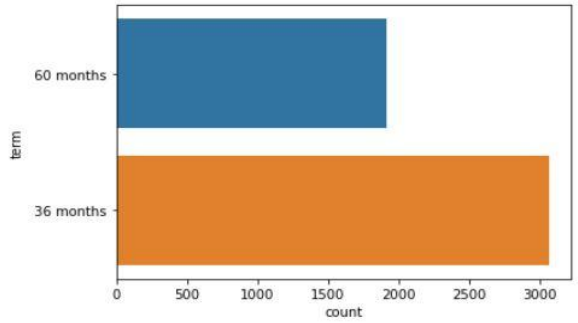


Univariate analysis on categorical columns (contd..)

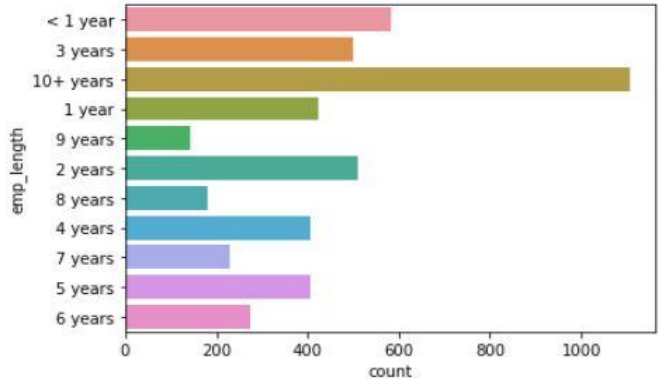
```
In [59]: sns.countplot(x ='verification_status', data=data[data.loan_status == 'Charged Off'])
Out[59]: <AxesSubplot:xlabel='verification_status', ylabel='count'>
```



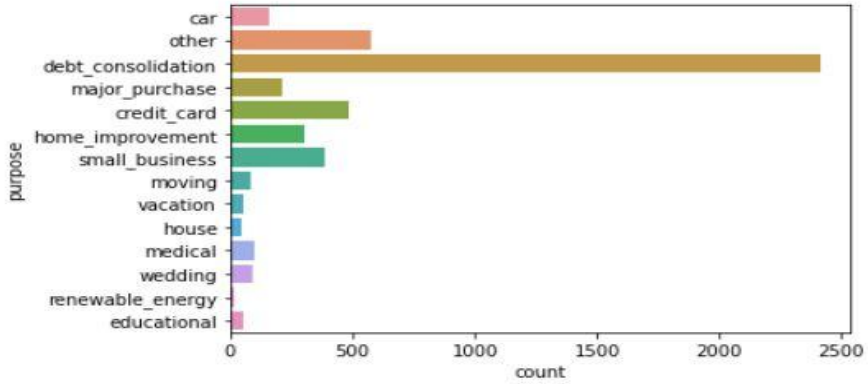

```
In [58]: sns.countplot(y ='term', data=data[data.loan_status == 'Charged Off'])
Out[58]: <AxesSubplot:xlabel='count', ylabel='term'>
```




```
In [55]: sns.countplot(y ='emp_length', data=data[data.loan_status == 'Charged Off'
Out[55]: <AxesSubplot:xlabel='count', ylabel='emp_length'>
```

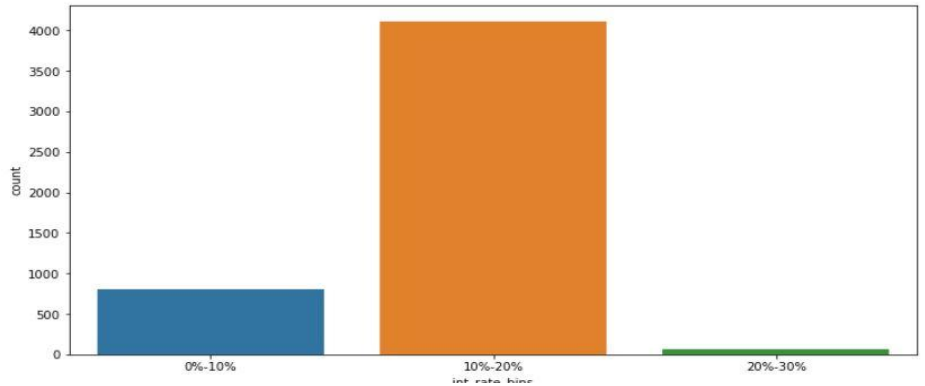



```
In [57]: sns.countplot(y ='purpose', data=data[data.loan_status == 'Charged Off'])
Out[57]: <AxesSubplot:xlabel='count', ylabel='purpose'>
```

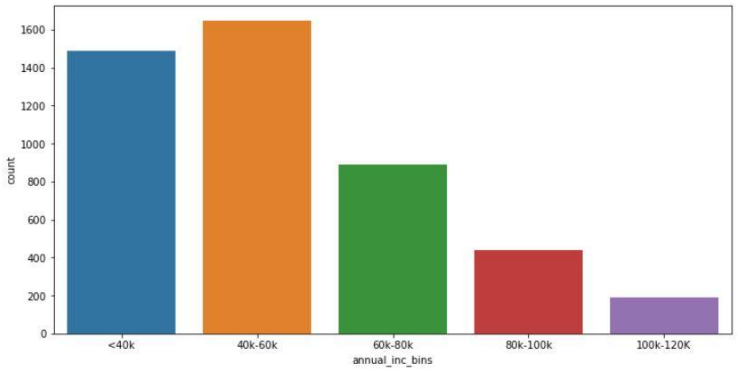


Univariate analysis on continuous columns

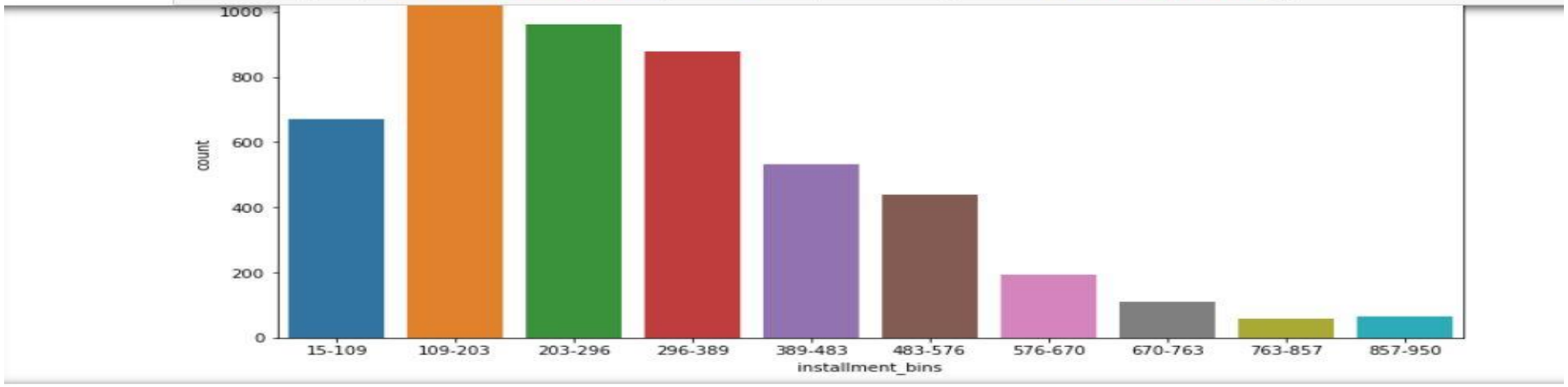
```
In [61]: bins = [0, 10, 20, 30]
         data['int_rate_bins'] = pd.cut(data['int_rate'], bins, labels= ['0%-10%', '10%-20%','20%-30%'])

In [62]: fig, ax = plt.subplots(figsize = (12,6))
         sns.countplot(x ='int_rate_bins', data=data[data.loan_status == 'Charged Off'])

Out[62]: <AxesSubplot:xlabel='int_rate_bins', ylabel='count'>
```




```
In [65]: bins=[20000,40000,60000,80000,100000,120000]
         data['annual_inc_bins'] = pd.cut(data['annual_inc'], bins, labels= ['<40k', '40k-60k', '60k-80k', '80k-100k', '100k-120K'])

In [66]: fig, ax = plt.subplots(figsize = (12,6))
         sns.countplot(x ='annual_inc_bins', data=data[data.loan_status == 'Charged Off'])

Out[66]: <AxesSubplot:xlabel='annual_inc_bins', ylabel='count'>
```

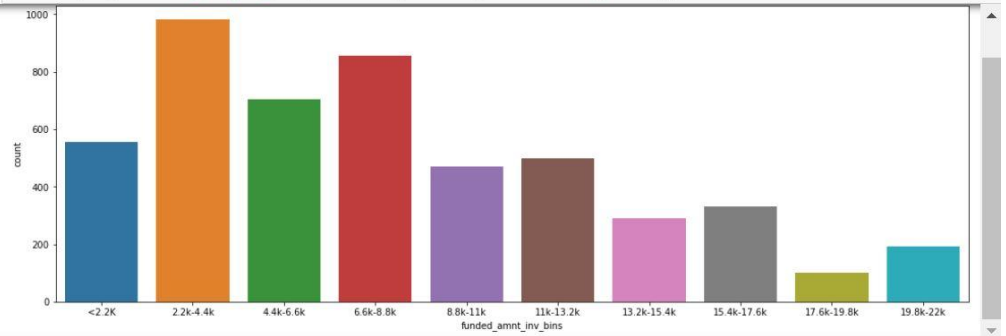



```
In [68]: fig, ax = plt.subplots(figsize = (12,6))
         sns.countplot(x ='installment_bins', data=data[data.loan_status == 'Charged Off'])
```

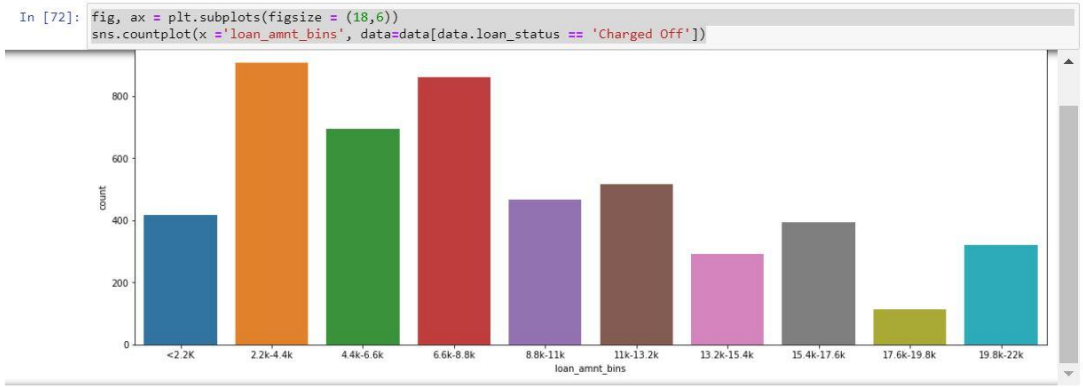


Univariate analysis on continuous columns (contd..)

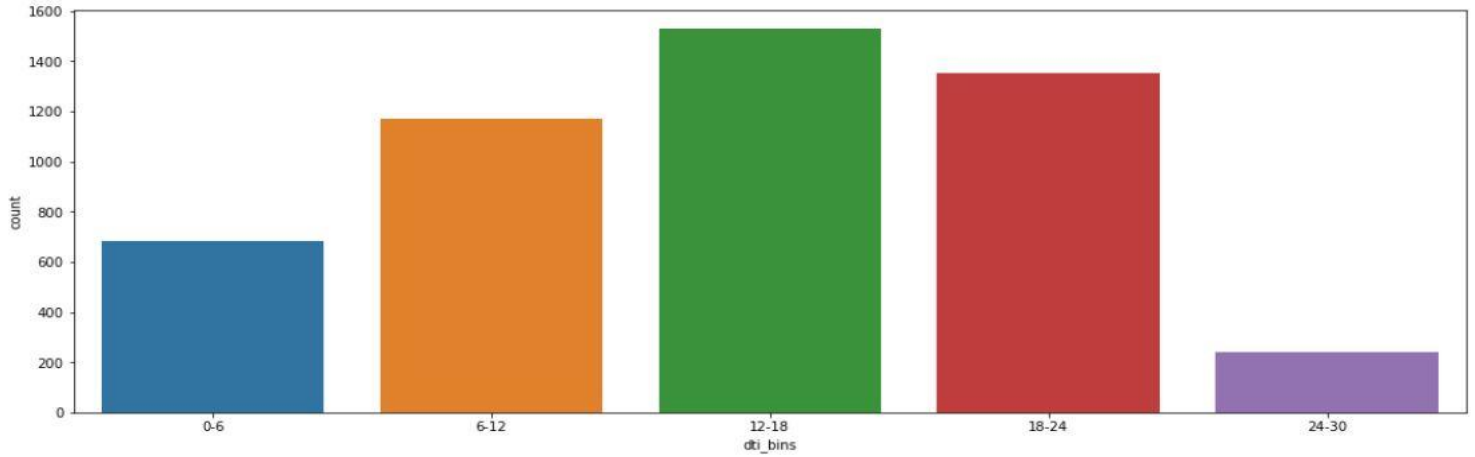
```
In [70]: fig, ax = plt.subplots(figsize = (18,6))
         sns.countplot(x ='funded_amnt_inv_bins', data=data[data.loan_status == 'Charged Off'])
```




```
In [72]: fig, ax = plt.subplots(figsize = (18,6))
         sns.countplot(x ='loan_amnt_bins', data=data[data.loan_status == 'Charged Off'])
```



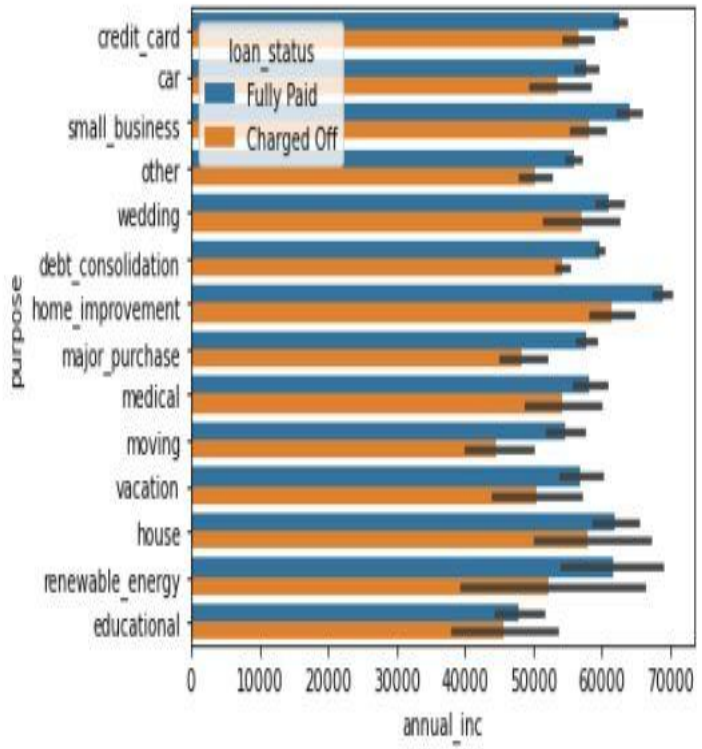

```
In [75]: fig, ax = plt.subplots(figsize = (18,6))
         sns.countplot(x ='dti_bins', data=data[data.loan_status == 'Charged Off'])
Out[75]: <AxesSubplot:xlabel='dti_bins', ylabel='count'>
```



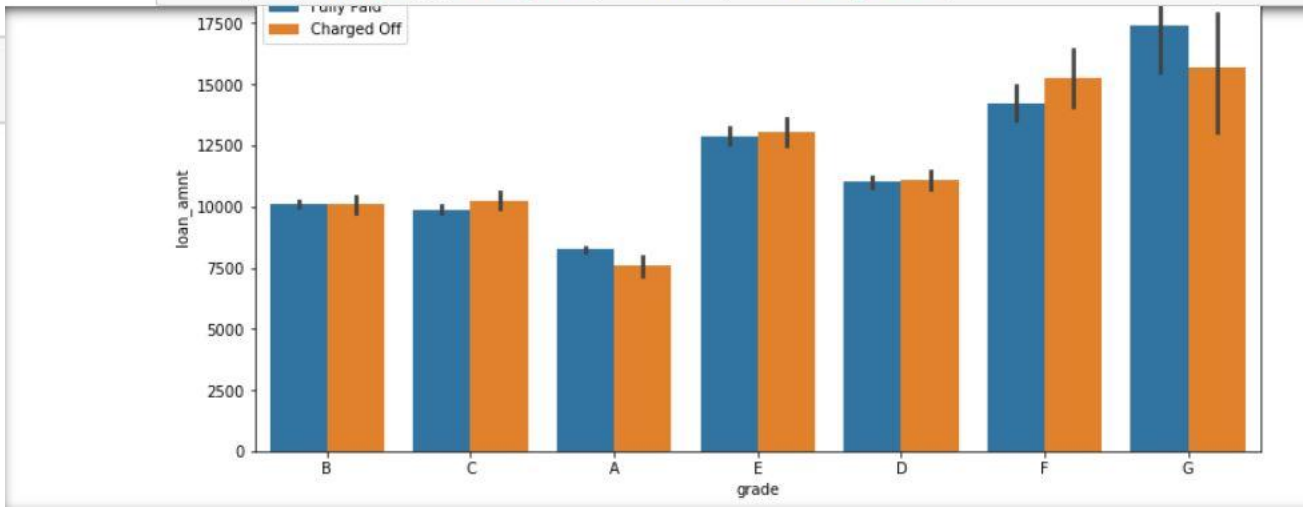
bivariate analysis

```
In [85]: sns.barplot(x ='annual_inc', y='purpose', data=data, hue ='loan_status')
```

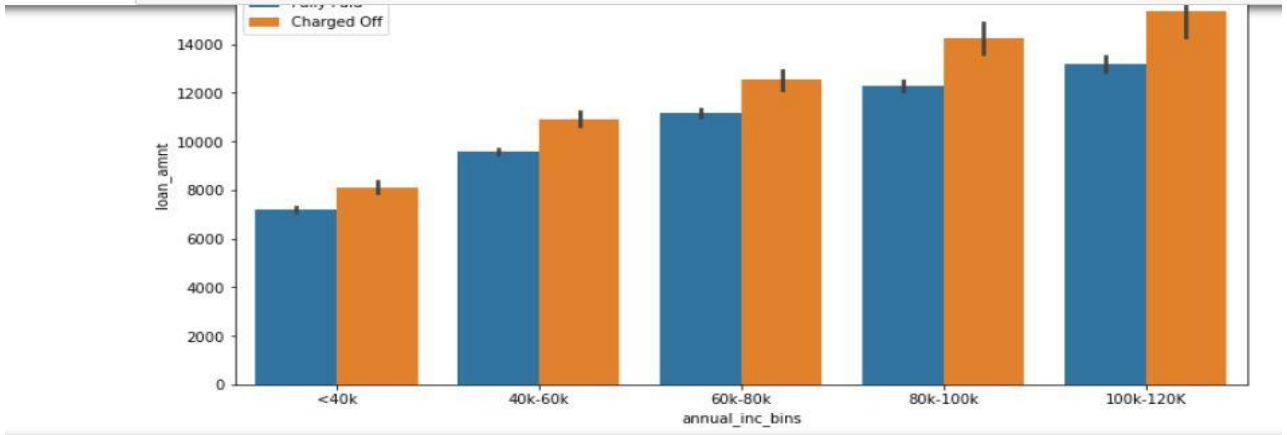
```
Out[85]: <AxesSubplot:xlabel='annual_inc', ylabel='purpose'>
```




```
In [80]: plt.figure(figsize=(12,6))
         sns.barplot(x ='grade', y='loan_amnt', data=data, hue ='loan_status')
```

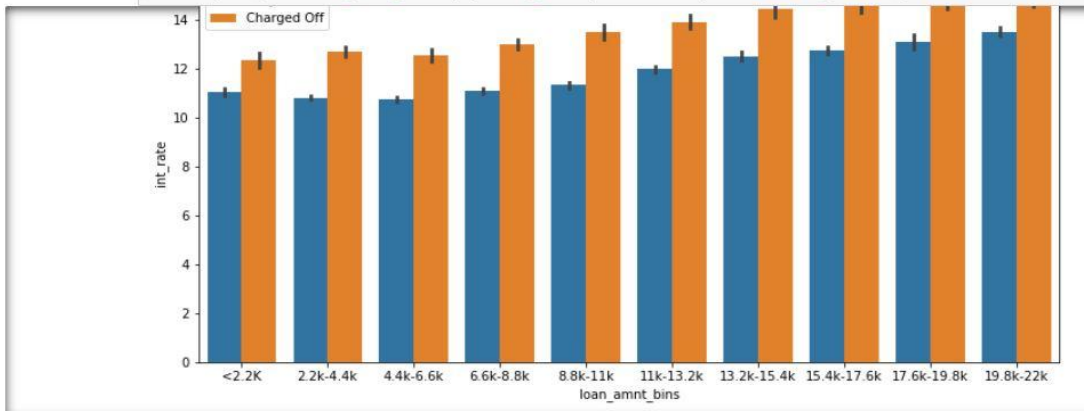



```
In [79]: plt.figure(figsize=(12,6))
         sns.barplot(x ='annual_inc_bins', y='loan_amnt', data=data, hue ='loan_status')
```

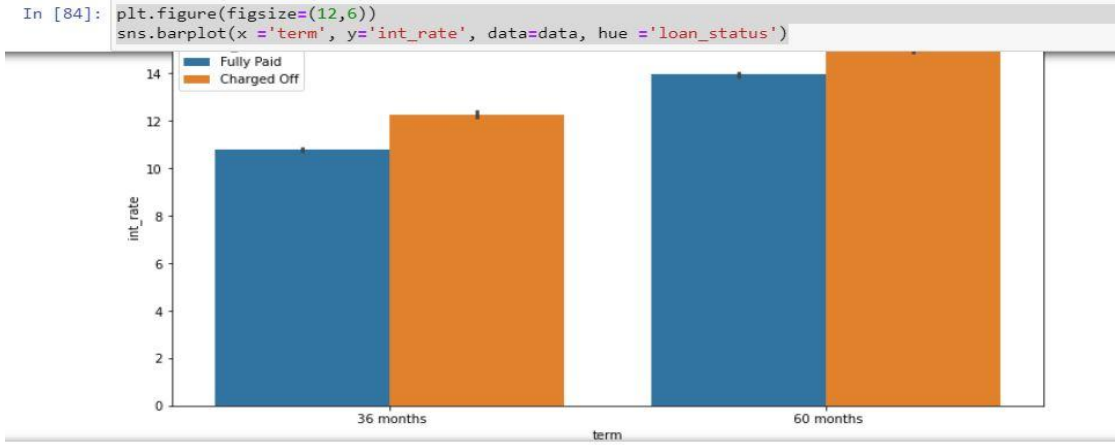


bivariate analysis(contd..)

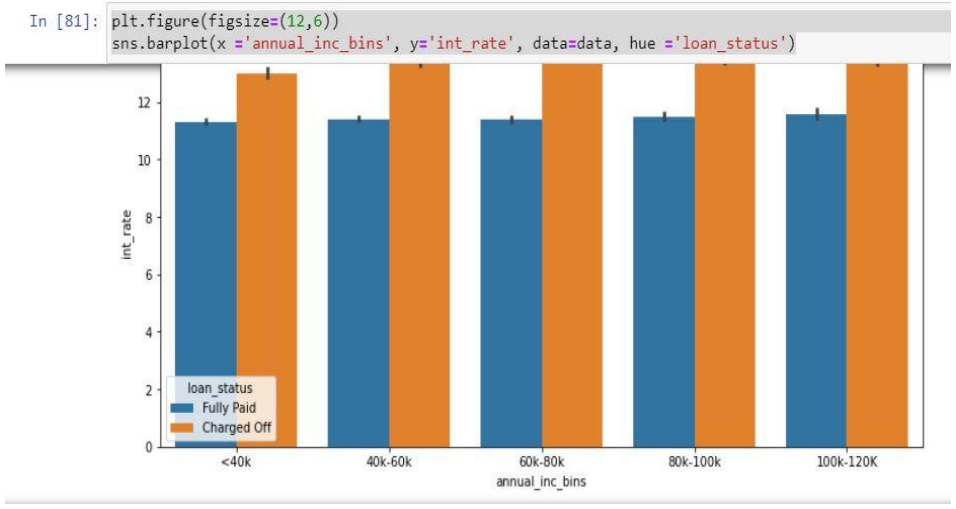
```
In [83]: plt.figure(figsize=(12,6))
         sns.barplot(x ='loan_amnt_bins', y='int_rate', data=data, hue ='loan_status')
```



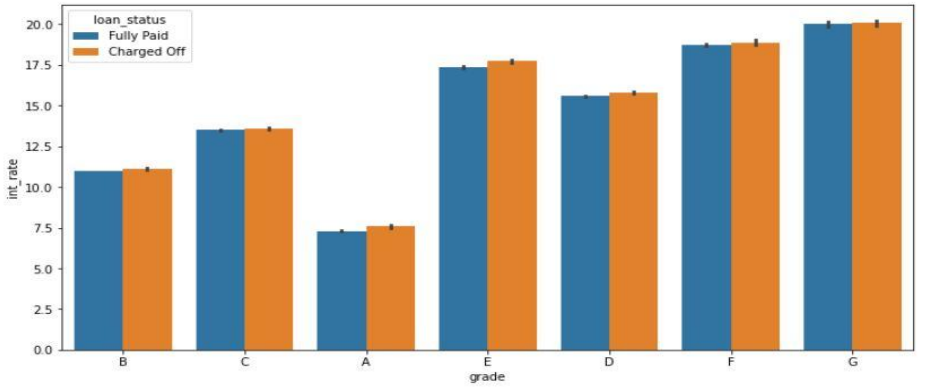

```
In [84]: plt.figure(figsize=(12,6))
         sns.barplot(x ='term', y='int_rate', data=data, hue ='loan_status')
```




```
In [81]: plt.figure(figsize=(12,6))
         sns.barplot(x ='annual_inc_bins', y='int_rate', data=data, hue ='loan_status')
```




```
In [82]: plt.figure(figsize=(12,6))
         sns.barplot(x ='grade', y='int_rate', data=data, hue ='loan_status')
Out[82]: <AxesSubplot:xlabel='grade', ylabel='int_rate'>
```



Obeservations

- 1. B and C grade are having more chances for Default/Charged off
- 2. - in grade B, sub_grade B5 has more chances for Default/Charged off,
- - in grade c, sub_grade C1 has more chances for Default/Charged off
- 3. Applicants with Rent type home_ownership has more chances for Default/Charged off
- 4. emp_length with 10+ years has more chances for Default/Charged off
- 5. Loan applicants with debt_consolidation purpose has more chances for Default/Charged off
- 6. Loan applicants with 36 months term has more chances for Default/Charged off
- 7. Loan applicants with 10%-20% intrest rate has more chances for Default/Charged off
- 8. - Loan applicants with <80K annual income has more chances for Default/Charged off
- - Loan applicants with >80K annual income has less chances for Default/Charged off
- 9. Loan applicants with <60K annual income and who are looking for home_improvement & small_business has more chances for Default/Charged off
- 10. for grade G and intrest rate > 17.5% has more chances of Default/charged off
- 11. Loan applicants with 19.8K to 22k loan amount and interest rate more than 14% has more chances for Default/Charged off
- 12. Loan applicants with 60 months term and >14% interest rate has more chances for Default/Charged off

Thank you