EDA CASE STUDY

Lending Club Case Study

We work for a consumer finance company which specialises in lending various types of loans to urban customers. 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.

- When the company receives a loan application, the company has to make a decision for loan approval based on the applicant's profile. Two types of risks are associated with the bank's decision:
  - If the applicant is likely to repay the loan, then not approving the loan results in a loss of business to the company
  - If the applicant is not likely to repay the loan, i.e. he/she is likely to default, then approving the loan may lead to a financial loss for the company
- When a person applies for a loan, there are two types of decisions that could be taken by the company:
  - Loan accepted: If the company approves the loan, there are 3 possible scenarios described below:
    - Fully paid: Applicant has fully paid the loan (the principal and the interest rate)
    - Current: Applicant is in the process of paying the instalments, i.e. the tenure of the loan is not yet completed. These candidates are not labelled as 'defaulted'.
    - Charged-off: Applicant has not paid the instalments in due time for a long period of time, i.e. he/she has defaulted on the loan
  - Loan rejected: The company had rejected the loan (because the candidate does not meet their requirements etc.). Since the loan was rejected, there is no transactional history of those applicants with the company and so this data is not available with the company (and thus in this dataset)

Business Objectives

This company is the largest online loan marketplace, facilitating personal loans, business loans, and financing of medical procedures. Borrowers can easily access lower interest rate loans through a fast online interface.

Like most other lending companies, lending loans to 'risky' applicants is the largest source of financial loss (called credit loss). Credit loss is the amount of money lost by the lender when the borrower refuses to pay or runs away with the money owed. In other words, borrowers who default cause the largest amount of loss to the lenders. In this case, the customers labelled as 'charged-off' are the 'defaulters'.

If one is able to identify these risky loan applicants, then such loans can be reduced thereby cutting down the amount of credit loss. Identification of such applicants using EDA is the aim of this case study.

In other words, the 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.

Feature Description

- ACC_NOW_DELINQ : The number of accounts on which the borrower is now delinquent.
- ACC_OPEN_PAST_24MTHS : Number of trades opened in past 24 months.
- ADDR_STATE : The state provided by the borrower in the loan application
- ALL_UTIL : Balance to credit limit on all trades
- ANNUAL_INC : The self-reported annual income provided by the borrower during registration.
- ANNUAL_INC_JOINT : The combined self-reported annual income provided by the co-borrowers during registration
- APPLICATION_TYPE : Indicates whether the loan is an individual application or a joint application with two co-borrowers
- AVG_CUR_BAL : Average current balance of all accounts
- BC_OPEN_TO_BUY : Total open to buy on revolving bankcards.
- BC_UTIL : Ratio of total current balance to high credit/credit limit for all bankcard accounts.
- CHARGEOFF_WITHIN_12_MTHS : Number of charge-offs within 12 months
- COLLECTION_RECOVERY_FEE : post charge off collection fee
- COLLECTIONS_12_MTHS_EX_MED : Number of collections in 12 months excluding medical collections
- DELINQ_2YRS : The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
- DELINQ_AMNT : The past-due amount owed for the accounts on which the borrower is now delinquent.
- DESC : Loan description provided by the borrower
- DTI : A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower’s self-reported monthly income.
- DTI_JOINT : A ratio calculated using the co-borrowers' total monthly payments on the total debt obligations, excluding mortgages and the requested LC loan, divided by the co-borrowers' combined self-reported monthly income
- EARLIEST_CR_LINE : The month the borrower's earliest reported credit line was opened
- EMP_LENGTH : Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
- EMP_TITLE : The job title supplied by the Borrower when applying for the loan.*
- FICO_RANGE_HIGH : The upper boundary range the borrower’s FICO at loan origination belongs to.
- FICO_RANGE_LOW : The lower boundary range the borrower’s FICO at loan origination belongs to.
- FUNDED_AMNT : The total amount committed to that loan at that point in time.
- FUNDED_AMNT_INV : The total amount committed by investors for that loan at that point in time.
- GRADE : LC assigned loan grade
- HOME_OWNERSHIP : The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
- ID : A unique LC assigned ID for the loan listing.
- IL_UTIL : Ratio of total current balance to high credit/credit limit on all install acct
- INITIAL_LIST_STATUS : The initial listing status of the loan. Possible values are – W, F
- INQ_FI : Number of personal finance inquiries
- INQ_LAST_12M : Number of credit inquiries in past 12 months
- INQ_LAST_6MTHS : The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
- INSTALLMENT : The monthly payment owed by the borrower if the loan originates.
- INT_RATE : Interest Rate on the loan
- ISSUE_D : The month which the loan was funded
- LAST_CREDIT_PULL_D : The most recent month LC pulled credit for this loan

- LAST_FICO_RANGE_HIGH : The upper boundary range the borrower’s last FICO pulled belongs to.
- LAST_FICO_RANGE_LOW : The lower boundary range the borrower’s last FICO pulled belongs to.
- LAST_PYMNT_AMNT : Last total payment amount received
- LAST_PYMNT_D : Last month payment was received
- LOAN_AMNT : The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
- LOAN_STATUS : Current status of the loan
- MAX_BAL_BC : Maximum current balance owed on all revolving accounts
- MEMBER_ID : A unique LC assigned Id for the borrower member.
- MO_SIN_OLD_IL_ACCT : Months since oldest bank installment account opened
- MO_SIN_OLD_REV_TL_OP : Months since oldest revolving account opened
- MO_SIN_RCNT_REV_TL_OP : Months since most recent revolving account opened
- MTHS_SINCE_LAST_DELINQ : The number of months since the borrower's last delinquency.
- MTHS_SINCE_LAST_MAJOR_DEROG : Months since most recent 90-day or worse rating
- MTHS_SINCE_LAST_RECORD : The number of months since the last public record.
- MTHS_SINCE_RCNT_IL : Months since most recent installment accounts opened
- MTHS_SINCE_RECENT_BC : Months since most recent bankcard account opened.
- MTHS_SINCE_RECENT_BC_DLQ : Months since most recent bankcard delinquency
- MTHS_SINCE_RECENT_INQ : Months since most recent inquiry.
- MTHS_SINCE_RECENT_REVOL_DELINQ : Months since most recent revolving delinquency.
- NEXT_PYMNT_D : Next scheduled payment date
- NUM_ACCTS_EVER_120_PD : Number of accounts ever 120 or more days past due
- NUM_ACTV_BC_TL : Number of currently active bankcard accounts
- NUM_ACTV_REV_TL : Number of currently active revolving trades
- NUM_BC_SATS : Number of satisfactory bankcard accounts
- NUM_BC_TL : Number of bankcard accounts
- NUM_IL_TL : Number of installment accounts
- NUM_OP_REV_TL : Number of open revolving accounts
- NUM_REV_ACCTS : Number of revolving accounts
- NUM_REV_TL_BAL_GT_0 : Number of revolving trades with balance >0
- NUM_SATS : Number of satisfactory accounts
- NUM_TL_120DPD_2M : Number of accounts currently 120 days past due (updated in past 2 months)
- NUM_TL_30DPD : Number of accounts currently 30 days past due (updated in past 2 months)
- NUM_TL_90G_DPD_24M : Number of accounts 90 or more days past due in last 24 months
- NUM_TL_OP_PAST_12M : Number of accounts opened in past 12 months
- OPEN_ACC : The number of open credit lines in the borrower's credit file.
- OPEN_ACC_6M : Number of open trades in last 6 months
- OPEN_IL_12M : Number of installment accounts opened in past 12 months
- OPEN_IL_24M : Number of installment accounts opened in past 24 months
- OPEN_IL_6M : Number of currently active installment trades
- OPEN_RV_12M : Number of revolving trades opened in past 12 months
- OPEN_RV_24M : Number of revolving trades opened in past 24 months
- OUT_PRNCP : Remaining outstanding principal for total amount funded
- OUT_PRNCP_INV : Remaining outstanding principal for portion of total amount funded by investors
- PCT_TL_NVR_DLQ : Percent of trades never delinquent
- PERCENT_BC_GT_75 : Percentage of all bankcard accounts > 75% of limit.
- POLICY_CODE : publicly available policy_code=1 new products not publicly available policy_code=2

- PUB_REC : Number of derogatory public records
- PUB_REC_BANKRUPTCIES : Number of public record bankruptcies
- PURPOSE : A category provided by the borrower for the loan request.
- PYMNT_PLAN : Indicates if a payment plan has been put in place for the loan
- RECOVERIES : post charge off gross recovery
- REVOL_BAL : Total credit revolving balance
- REVOL_UTIL : Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
- SUB_GRADE : LC assigned loan subgrade
- TAX_LIENS : Number of tax liens
- TERM : The number of payments on the loan. Values are in months and can be either 36 or 60.
- TITLE : The loan title provided by the borrower
- TOT_COLL_AMT : Total collection amounts ever owed
- TOT_CUR_BAL : Total current balance of all accounts
- TOT_HI_CRED_LIM : Total high credit/credit limit
- TOTAL_ACC : The total number of credit lines currently in the borrower's credit file
- TOTAL_BAL_EX_MORT : Total credit balance excluding mortgage
- TOTAL_BAL_IL : Total current balance of all installment accounts
- TOTAL_BC_LIMIT : Total bankcard high credit/credit limit
- TOTAL_CU_TL : Number of finance trades
- TOTAL_IL_HIGH_CREDIT_LIMIT : Total installment high credit/credit limit
- TOTAL_PYMNT : Payments received to date for total amount funded
- TOTAL_PYMNT_INV : Payments received to date for portion of total amount funded by investors
- TOTAL_REC_INT : Interest received to date
- TOTAL_REC_LATE_FEE : Late fees received to date
- TOTAL_REC_PRNCP : Principal received to date
- TOTAL_REV_HI_LIM : Total revolving high credit/credit limit
- URL : URL for the LC page with listing data.
- VERIFICATION_STATUS : Indicates if income was verified by LC, not verified, or if the income source was verified
- VERIFIED_STATUS_JOINT : Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified
- ZIP_CODE : The first 3 numbers of the zip code provided by the borrower in the loan application.

Importing the Libraries

In [292]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Reading the Dataset

In [293]:
```
df=pd.read_csv("loan.csv")
```

Analyzing/Understanding the Dataset with various commands

In [294]:
```
#shape of the dataset
df.shape
```

Out[294]: (39717, 111)

In [295]:
```
# viewing the first 5 rows of the dataset to get an idea
df.head()
```

Out[295]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installm
0	1077501	1296599	5000	5000	4975.0	36 months	10.65%	162
1	1077430	1314167	2500	2500	2500.0	60 months	15.27%	59
2	1077175	1313524	2400	2400	2400.0	36 months	15.96%	84
3	1076863	1277178	10000	10000	10000.0	36 months	13.49%	339
4	1075358	1311748	3000	3000	3000.0	60 months	12.69%	67

5 rows × 111 columns

In [296]:
```
# viewing the columns of the dataset
df.columns.tolist()
```

Out[296]:
```
['id',
 'member_id',
 'loan_amnt',
 'funded_amnt',
 'funded_amnt_inv',
 'term',
 'int_rate',
 'installment',
 'grade',
 'sub_grade',
 'emp_title',
 'emp_length',
 'home_ownership',
 'annual_inc',
 'verification_status',
 'issue_d',
 'loan_status',
 'pymnt_plan',
 'url',
 'desc',
 'purpose',
 'title',
 'zip_code',
 'addr_state',
 'dti',
 'delinq_2yrs',
 'earliest_cr_line',
 'inq_last_6mths',
 'mths_since_last_delinq',
 'mths_since_last_record',
 'open_acc',
 'pub_rec',
 'revol_bal',
 'revol_util',
 'total_acc',
 'initial_list_status',
 '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',
 'next_pymnt_d',
 'last_credit_pull_d',
 'collections_12_mths_ex_med',
 'mths_since_last_major_derog',
 'policy_code',
 'application_type',
 'annual_inc_joint',
 'dti_joint',
 'verification_status_joint',
```

```
 'acc_now_delinq',
 'tot_coll_amt',
 'tot_cur_bal',
 'open_acc_6m',
 'open_il_6m',
 'open_il_12m',
 'open_il_24m',
 'mths_since_rcnt_il',
 'total_bal_il',
 'il_util',
 'open_rv_12m',
 'open_rv_24m',
 'max_bal_bc',
 'all_util',
 'total_rev_hi_lim',
 'inq_fi',
 'total_cu_tl',
 'inq_last_12m',
 'acc_open_past_24mths',
 'avg_cur_bal',
 'bc_open_to_buy',
 'bc_util',
 'chargeoff_within_12_mths',
 'delinq_amnt',
 'mo_sin_old_il_acct',
 'mo_sin_old_rev_tl_op',
 'mo_sin_rcnt_rev_tl_op',
 'mo_sin_rcnt_tl',
 'mort_acc',
 'mths_since_recent_bc',
 'mths_since_recent_bc_dlq',
 'mths_since_recent_inq',
 'mths_since_recent_revol_delinq',
 'num_accts_ever_120_pd',
 'num_actv_bc_tl',
 'num_actv_rev_tl',
 'num_bc_sats',
 'num_bc_tl',
 'num_il_tl',
 'num_op_rev_tl',
 'num_rev_accts',
 'num_rev_tl_bal_gt_0',
 'num_sats',
 'num_tl_120dpd_2m',
 'num_tl_30dpd',
 'num_tl_90g_dpd_24m',
 'num_tl_op_past_12m',
 'pct_tl_nvr_dlq',
 'percent_bc_gt_75',
 'pub_rec_bankruptcies',
 'tax_liens',
 'tot_hi_cred_lim',
 'total_bal_ex_mort',
 'total_bc_limit',
 'total_il_high_credit_limit']
```

In [297]:
```
# getting the necessary gist of the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39717 entries, 0 to 39716
Columns: 111 entries, id to total_il_high_credit_limit
dtypes: float64(74), int64(13), object(24)
memory usage: 33.6+ MB
```

In [298]:
```
# viewing some analytical measures of the dataset
df.describe()
```

Out[298]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	installme
count	3.971700e+04	3.971700e+04	39717.000000	39717.000000	39717.000000	39717.0000
mean	6.831319e+05	8.504636e+05	11219.443815	10947.713196	10397.448868	324.5619
std	2.106941e+05	2.656783e+05	7456.670694	7187.238670	7128.450439	208.8748
min	5.473400e+04	7.069900e+04	500.000000	500.000000	0.000000	15.6900
25%	5.162210e+05	6.667800e+05	5500.000000	5400.000000	5000.000000	167.0200
50%	6.656650e+05	8.508120e+05	10000.000000	9600.000000	8975.000000	280.2200
75%	8.377550e+05	1.047339e+06	15000.000000	15000.000000	14400.000000	430.7800
max	1.077501e+06	1.314167e+06	35000.000000	35000.000000	35000.000000	1305.1900

8 rows × 87 columns

Here , we see that many of the cols have null values. this should be noted. Also there are many columns, which can br dropped since they don't constitute enough in loan defaulter analysis. We need to remove them too.

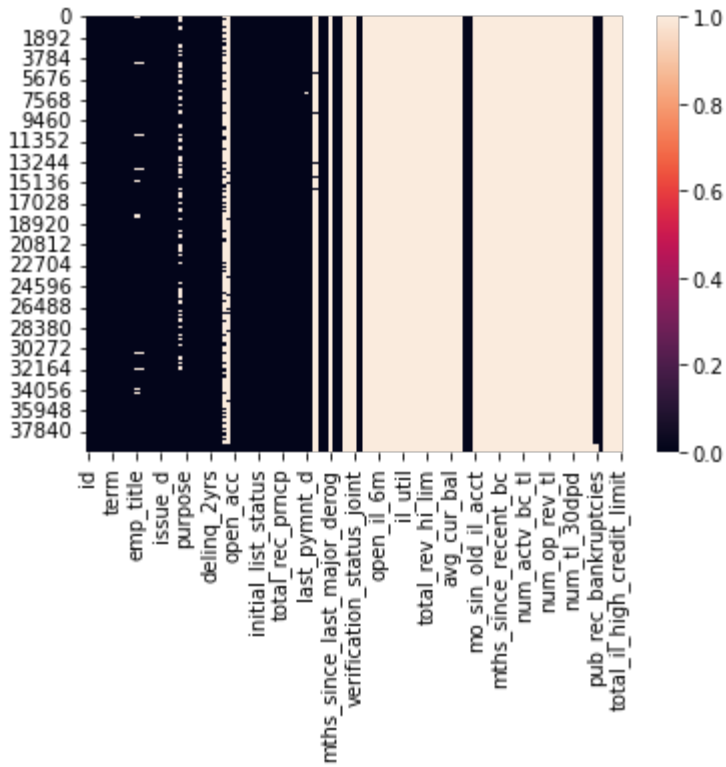
In [299]:
```
# checking the total percentage of null values the dataset
(df.isnull().sum().sum())/(df.shape[0]*df.shape[1])
```

Out[299]: 0.5133989643393677

We can see that around 51% of the values are null in this dataset. we will now look at how many columns have more null values and how many have less. we will divide them on the basis of null value percentages in every column

In [300]:
```
# Null value visualization
sns.heatmap(df.isnull())
```

Out[300]: <AxesSubplot:>



Here, we see that many columns are absolute null i.e they are totally empty. we will create a dataframe to see the how many columns have how much percentage of null values

In [301]:
```
ten_perc_cols = len(df.columns[((df.isnull().sum())/len(df))<0.1])
twe_perc_cols = len(df.columns[((df.isnull().sum())/len(df))<0.2])
thi_perc_cols = len(df.columns[((df.isnull().sum())/len(df))<0.3])
for_perc_cols = len(df.columns[((df.isnull().sum())/len(df))<0.4])
fif_perc_cols = len(df.columns[((df.isnull().sum())/len(df))<0.5])
six_perc_cols = len(df.columns[((df.isnull().sum())/len(df))<0.6])
sev_perc_cols = len(df.columns[((df.isnull().sum())/len(df))<0.7])
eig_perc_cols = len(df.columns[((df.isnull().sum())/len(df))<0.8])
nin_perc_cols = len(df.columns[((df.isnull().sum())/len(df))<0.9])
hun_perc_cols = len(df.columns[((df.isnull().sum())/len(df))<1.0])
```

In [302]:
```
temp={'perc_null_values':['less than 10','less than 20','less than 
30','less than 40','less than 50','less than 60','less than 70','le
ss than 80','less than 90','less than 100'],'No of cols':[ten_perc_
cols,twe_perc_cols,thi_perc_cols,for_perc_cols,fif_perc_cols,six_pe
rc_cols,sev_perc_cols,eig_perc_cols,nin_perc_cols,hun_perc_cols]}
```

In [303]:
```
null_perc_df = pd.DataFrame(temp)
null_perc_df
```

Out[303]:

	perc_null_values	No of cols
0	less than 10	53
1	less than 20	53
2	less than 30	53
3	less than 40	54
4	less than 50	54
5	less than 60	54
6	less than 70	55
7	less than 80	55
8	less than 90	55
9	less than 100	57

In [304]:
```
# Total number of null columns
len(df.columns[((df.isnull().sum())/len(df))==1.0])
```

Out[304]: 54

In [305]:
```
# Viewing totally null columns
df.columns[(df.isnull().sum()/len(df))==1]
```

Out[305]:
```
Index(['mths_since_last_major_derog', 'annual_inc_joint', 'dti_joint
',
       'verification_status_joint', 'tot_coll_amt', 'tot_cur_bal',
       'open_acc_6m', 'open_il_6m', 'open_il_12m', 'open_il_24m',
       'mths_since_rcnt_il', 'total_bal_il', 'il_util', 'open_rv_12m
',
       'open_rv_24m', 'max_bal_bc', 'all_util', 'total_rev_hi_lim',
'inq_fi',
       'total_cu_tl', 'inq_last_12m', 'acc_open_past_24mths', 'avg_c
ur_bal',
       'bc_open_to_buy', 'bc_util', 'mo_sin_old_il_acct',
       'mo_sin_old_rev_tl_op', 'mo_sin_rcnt_rev_tl_op', 'mo_sin_rcnt
_tl',
       'mort_acc', 'mths_since_recent_bc', 'mths_since_recent_bc_dlq
',
       'mths_since_recent_inq', 'mths_since_recent_revol_delinq',
       'num_accts_ever_120_pd', 'num_actv_bc_tl', 'num_actv_rev_tl',
       'num_bc_sats', 'num_bc_tl', 'num_il_tl', 'num_op_rev_tl',
       'num_rev_accts', 'num_rev_tl_bal_gt_0', 'num_sats', 'num_tl_1
20dpd_2m',
       'num_tl_30dpd', 'num_tl_90g_dpd_24m', 'num_tl_op_past_12m',
       'pct_tl_nvr_dlq', 'percent_bc_gt_75', 'tot_hi_cred_lim',
       'total_bal_ex_mort', 'total_bc_limit', 'total_il_high_credit_
limit'],
      dtype='object')
```

Above hypothesis shows us that 54 columns are totally null. They should be removed as they won't contribute in analysis

In [306]:
```
# Removing the totally null columns
df=df[df.columns[-((df.isnull().sum()/len(df))==1)]]
```

In [307]:
```
df.shape
```

Out[307]:
```
(39717, 57)
```

In [308]:
```
# Checking columns which have more than 60% null values after remov
ing totally null columns
len(df.columns[df.isnull().sum()/len(df)>0.6])
```

Out[308]:
```
3
```

Now there are 3 columns which have more than 60% null values. We shall remove them too.

In [309]:
```
df=df[df.columns[-((df.isnull().sum()/len(df))>0.6)]]
```

In [310]:
```
df.shape
```

Out[310]:
```
(39717, 54)
```

Now our dataset has 54 columns to analyse. If we look at the null percentage dataframe(null_perc_df), we could see that 54 columns have less than 40% null values which is same as 60%

We need to now analyze the reamaning features and look which should be kept and which should be deleted

In [311]:
```
for i in df.columns:
    print(df[i].value_counts())
    print('*'*80)
```

```
1077501    1
568534     1
568659     1
567165     1
568531     1
          ..
785667     1
785659     1
785630     1
785626     1
87023      1
Name: id, Length: 39717, dtype: int64
*********************************************************************
************
1296599    1
731393     1
731544     1
729629     1
731390     1
          ..
989001     1
988993     1
988959     1
988954     1
86999      1
Name: member_id, Length: 39717, dtype: int64
*********************************************************************
************
10000    2833
12000    2334
5000     2051
6000     1908
15000    1895
         ...
22875       1
8175        1
19475       1
21225       1
22550       1
Name: loan_amnt, Length: 885, dtype: int64
*********************************************************************
************
10000    2741
12000    2244
5000     2040
6000     1898
15000    1784
         ...
26250       1
24725       1
31750       1
22625       1
22350       1
Name: funded_amnt, Length: 1041, dtype: int64
*********************************************************************
************
```

```
5000.000000     1309
10000.000000    1275
6000.000000     1200
12000.000000    1069
8000.000000      900
                ...
4944.213109        1
18400.281660       1
14659.820000       1
6294.151315        1
11808.924370       1
Name: funded_amnt_inv, Length: 8205, dtype: int64
**********************************************************************
************
 36 months    29096
 60 months    10621
Name: term, dtype: int64
**********************************************************************
************
10.99%    956
13.49%    826
11.49%    825
7.51%     787
7.88%     725
         ...
18.36%      1
16.96%      1
16.15%      1
16.01%      1
17.44%      1
Name: int_rate, Length: 371, dtype: int64
**********************************************************************
************
311.11     68
180.96     59
311.02     54
150.80     48
368.45     46
           ..
1224.46     1
63.44       1
157.67      1
492.34      1
255.43      1
Name: installment, Length: 15383, dtype: int64
**********************************************************************
************
B    12020
A    10085
C     8098
D     5307
E     2842
F     1049
G      316
Name: grade, dtype: int64
**********************************************************************
```

```
************
B3    2917
A4    2886
A5    2742
B5    2704
B4    2512
C1    2136
B2    2057
C2    2011
B1    1830
A3    1810
C3    1529
A2    1508
D2    1348
C4    1236
C5    1186
D3    1173
A1    1139
D4     981
D1     931
D5     874
E1     763
E2     656
E3     553
E4     454
E5     416
F1     329
F2     249
F3     185
F4     168
F5     118
G1     104
G2      78
G4      56
G3      48
G5      30
Name: sub_grade, dtype: int64
********************************************************************
************
US Army                              134
Bank of America                      109
IBM                                   66
AT&T                                  59
Kaiser Permanente                     56
                                    ...
Community College of Philadelphia      1
AMEC                                   1
lee county sheriff                     1
Bacon County Board of Education        1
Evergreen Center                       1
Name: emp_title, Length: 28820, dtype: int64
********************************************************************
************
10+ years    8879
< 1 year     4583
2 years      4388
```

```
3 years      4095
4 years      3436
5 years      3282
1 year       3240
6 years      2229
7 years      1773
8 years      1479
9 years      1258
Name: emp_length, dtype: int64
**********************************************************************
************
RENT        18899
MORTGAGE    17659
OWN          3058
OTHER          98
NONE            3
Name: home_ownership, dtype: int64
**********************************************************************
************
60000.0    1505
50000.0    1057
40000.0     876
45000.0     830
30000.0     825
           ...
56820.0       1
45314.0       1
53913.0       1
62880.0       1
27376.0       1
Name: annual_inc, Length: 5318, dtype: int64
**********************************************************************
************
Not Verified       16921
Verified           12809
Source Verified     9987
Name: verification_status, dtype: int64
**********************************************************************
************
Dec-11    2260
Nov-11    2223
Oct-11    2114
Sep-11    2063
Aug-11    1928
Jul-11    1870
Jun-11    1827
May-11    1689
Apr-11    1562
Mar-11    1443
Jan-11    1380
Feb-11    1297
Dec-10    1267
Oct-10    1132
Nov-10    1121
Jul-10    1119
Sep-10    1086
```

```
Aug-10    1078
Jun-10    1029
May-10     920
Apr-10     827
Mar-10     737
Feb-10     627
Nov-09     602
Dec-09     598
Jan-10     589
Oct-09     545
Sep-09     449
Aug-09     408
Jul-09     374
Jun-09     356
May-09     319
Apr-09     290
Mar-09     276
Feb-09     260
Jan-09     239
Mar-08     236
Dec-08     223
Nov-08     184
Feb-08     174
Jan-08     171
Apr-08     155
Oct-08      96
Dec-07      85
Jul-08      83
May-08      71
Aug-08      71
Jun-08      66
Oct-07      47
Nov-07      37
Aug-07      33
Sep-08      32
Jul-07      30
Sep-07      18
Jun-07       1
Name: issue_d, dtype: int64
********************************************************************
************
Fully Paid     32950
Charged Off     5627
Current         1140
Name: loan_status, dtype: int64
********************************************************************
************
n    39717
Name: pymnt_plan, dtype: int64
********************************************************************
************
https://lendingclub.com/browse/loanDetail.action?loan_id=1077501
1
https://lendingclub.com/browse/loanDetail.action?loan_id=568534
1
https://lendingclub.com/browse/loanDetail.action?loan_id=568659
```

```
1
https://lendingclub.com/browse/loanDetail.action?loan_id=567165
1
https://lendingclub.com/browse/loanDetail.action?loan_id=568531
1

..
https://lendingclub.com/browse/loanDetail.action?loan_id=785667
1
https://lendingclub.com/browse/loanDetail.action?loan_id=785659
1
https://lendingclub.com/browse/loanDetail.action?loan_id=785630
1
https://lendingclub.com/browse/loanDetail.action?loan_id=785626
1
https://lendingclub.com/browse/loanDetail.action?loan_id=87023
1
Name: url, Length: 39717, dtype: int64
**********************************************************************
************

210
Debt Consolidation
8
Camping Membership
6
personal loan
3
credit card consolidation
3

...
  Borrower added on 05/13/11 > I have a very stable income and have
&quot;NEVER&quot; been delinquent on any accounts. I am interested i
n consolidating my credit card accounts along with a personal loan f
or the benefit of paying one payment a month versus multiple. Thank
you.<br/>
1
  Borrower added on 05/13/11 > This loan is to partially finance a c
ar. The payments will be very manageable for me.<br/>
1
  Borrower added on 05/13/11 > I am consolidating my bills to make i
t cheaper on bills. I am up for a promotion at my job and have been
here three years already. I also have a fiance that has a very relia
ble job who also helps with finances. My requirement is that all bil
ls are paid on time if not early. This is to keep good standings wit
h all businesses and keep a awesome credit score.<br/> Borrower adde
d on 05/16/11 > I am asking for this loan to pay off bills with high
er interest rates and have a lower payment every month.<br/>
1
  Borrower added on 05/13/11 > Debt Consolidation<br/> Borrower adde
d on 05/13/11 > I plan to use this money to consolidate bills with h
igh monthly payments and improve cash flow.<br/> Borrower added on 0
5/13/11 > I have good credit and I have a very stable, solid and pro
fessional job that I have held for a long time - over 20 years.  I h
ave a Bachelor's degree and I'm currently working on a masters.<br/>
```

```
Borrower added on 05/13/11 > My mortgage and utilities run approx le
ss than 2k per month including taxes.  I have sufficient income to p
ay off this loan. I simply want to take my open accounts and consoli
date them into one easy payment.<br/>      1
I plan to consolidate over $7,000 of debt: a combination of credit c
ards and student loans.
1
Name: desc, Length: 26527, dtype: int64
********************************************************************
************
debt_consolidation    18641
credit_card            5130
other                  3993
home_improvement       2976
major_purchase         2187
small_business         1828
car                    1549
wedding                 947
medical                 693
moving                  583
vacation                381
house                   381
educational             325
renewable_energy        103
Name: purpose, dtype: int64
********************************************************************
************
Debt Consolidation                 2184
Debt Consolidation Loan            1729
Personal Loan                       659
Consolidation                       517
debt consolidation                  505
                                   ...
your rate is better than my rate      1
Concession Trailer                    1
gregs                                 1
EZover                                1
JAL Loan                              1
Name: title, Length: 19615, dtype: int64
********************************************************************
************
100xx    597
945xx    545
112xx    516
606xx    503
070xx    473
        ...
381xx      1
378xx      1
739xx      1
396xx      1
469xx      1
Name: zip_code, Length: 823, dtype: int64
********************************************************************
************
CA    7099
```

```
NY    3812
FL    2866
TX    2727
NJ    1850
IL    1525
PA    1517
VA    1407
GA    1398
MA    1340
OH    1223
MD    1049
AZ     879
WA     840
CO     792
NC     788
CT     751
MI     720
MO     686
MN     615
NV     497
SC     472
WI     460
AL     452
OR     451
LA     436
KY     325
OK     299
KS     271
UT     258
AR     245
DC     214
RI     198
NM     189
WV     177
HI     174
NH     171
DE     114
MT      85
WY      83
AK      80
SD      64
VT      54
MS      19
TN      17
IN       9
ID       6
IA       5
NE       5
ME       3
Name: addr_state, dtype: int64
**********************************************************************
************
0.00     183
12.00     51
18.00     45
19.20     40
```

```
13.20     39
         ...
29.13      1
25.31      1
29.76      1
28.42      1
25.43      1
Name: dti, Length: 2868, dtype: int64
********************************************************************
************
0     35405
1      3303
2       687
3       220
4        62
5        22
6        10
7         4
8         2
9         1
11        1
Name: delinq_2yrs, dtype: int64
********************************************************************
************
Nov-98    370
Oct-99    366
Dec-98    348
Oct-00    346
Dec-97    329
         ...
Feb-66      1
Dec-61      1
Oct-54      1
Jun-72      1
Oct-74      1
Name: earliest_cr_line, Length: 526, dtype: int64
********************************************************************
************
0    19300
1    10971
2     5812
3     3048
4      326
5      146
6       64
7       35
8       15
Name: inq_last_6mths, dtype: int64
********************************************************************
************
7     4018
6     3946
8     3936
9     3718
10    3223
5     3183
```

```
11    2746
4     2343
12    2273
13    1911
3     1493
14    1487
15    1177
16     940
17     741
2      605
18     533
19     396
20     289
21     244
22     143
23      97
24      81
25      55
26      34
28      25
27      22
30      15
29      13
31       7
34       5
32       4
35       4
33       3
36       2
39       1
38       1
44       1
41       1
42       1
Name: open_acc, dtype: int64
********************************************************************
************
0    37601
1     2056
2       51
3        7
4        2
Name: pub_rec, dtype: int64
********************************************************************
************
0        994
298       14
255       14
1         12
682       11
        ...
21424      1
30747      1
23862      1
20197      1
85607      1
```

```
Name: revol_bal, Length: 21711, dtype: int64
**********************************************************************
************
0%         977
0.20%       63
63%         62
40.70%      58
66.70%      58
          ...
25.74%       1
47.36%       1
24.65%       1
10.61%       1
7.28%        1
Name: revol_util, Length: 1089, dtype: int64
**********************************************************************
************
16    1471
15    1462
17    1457
14    1445
20    1428
      ...
74       1
77       1
78       1
87       1
90       1
Name: total_acc, Length: 82, dtype: int64
**********************************************************************
************
f    39717
Name: initial_list_status, dtype: int64
**********************************************************************
************
0.00       38577
1972.60        2
827.13         2
2277.11        2
2963.24        2
           ...
782.23         1
2296.41        1
1928.85        1
1061.32        1
79.24          1
Name: out_prncp, Length: 1137, dtype: int64
**********************************************************************
************
0.00       38577
1972.60        2
1664.64        2
827.13         2
1863.21        1
           ...
782.23         1
```

```
2289.14        1
1928.85        1
1061.32        1
79.24          1
Name: out_prncp_inv, Length: 1138, dtype: int64
********************************************************************
************
11196.569430    26
0.000000        16
11784.232230    16
10956.775960    16
5478.387981     15
                ..
17768.430010     1
12794.806580     1
6193.803706      1
34797.769170     1
9195.263334      1
Name: total_pymnt, Length: 37850, dtype: int64
********************************************************************
************
0.00        165
6514.52      16
5478.39      14
13148.14     14
11196.57     12
           ...
17702.50      1
19026.06      1
7355.24       1
387.55        1
980.83        1
Name: total_pymnt_inv, Length: 37518, dtype: int64
********************************************************************
************
10000.00    2293
12000.00    1805
5000.00     1702
6000.00     1637
15000.00    1400
            ...
1097.81        1
1410.30        1
6968.65        1
3477.49        1
16077.42       1
Name: total_rec_prncp, Length: 7976, dtype: int64
********************************************************************
************
0.00       71
1196.57    26
514.52     19
956.78     17
1784.23    17
           ..
494.53      1
```

```
1119.88     1
62.31       1
2656.10     1
1695.26     1
Name: total_rec_int, Length: 35148, dtype: int64
********************************************************************
************
0.000000     37671
15.000000      255
15.000000       58
30.000000       55
15.000000       47
             ...
35.286832        1
15.000000        1
14.777500        1
14.967774        1
19.890000        1
Name: total_rec_late_fee, Length: 1356, dtype: int64
********************************************************************
************
0.00       35499
11.29          4
10.40          4
10.66          3
44.92          3
           ...
764.69         1
653.08         1
1080.96        1
878.19         1
21.29          1
Name: recoveries, Length: 4040, dtype: int64
********************************************************************
************
0.0000      35935
2.0000         12
1.2000         10
3.7100          9
1.8800          8
            ...
3.7900          1
773.4900        1
272.8250        1
1.7697          1
0.2300          1
Name: collection_recovery_fee, Length: 2616, dtype: int64
********************************************************************
************
May-16    1256
Mar-13    1026
Dec-14     945
May-13     907
Feb-13     869
          ...
Jun-08      10
```

```
Nov-08      10
Mar-08       5
Jan-08       4
Feb-08       1
Name: last_pymnt_d, Length: 101, dtype: int64
********************************************************************
************
0.00       74
276.06     21
200.00     17
50.00      16
100.00     15
           ..
1763.87     1
172.27      1
889.67      1
150.73      1
256.59      1
Name: last_pymnt_amnt, Length: 34930, dtype: int64
********************************************************************
************
May-16    10308
Apr-16     2547
Mar-16     1123
Feb-13      843
Feb-16      736
          ...
May-08        1
Jun-08        1
Jul-08        1
May-07        1
Jul-07        1
Name: last_credit_pull_d, Length: 106, dtype: int64
********************************************************************
************
0.0    39661
Name: collections_12_mths_ex_med, dtype: int64
********************************************************************
************
1    39717
Name: policy_code, dtype: int64
********************************************************************
************
INDIVIDUAL    39717
Name: application_type, dtype: int64
********************************************************************
************
0    39717
Name: acc_now_delinq, dtype: int64
********************************************************************
************
0.0    39661
Name: chargeoff_within_12_mths, dtype: int64
********************************************************************
************
0    39717
```

```
Name: delinq_amnt, dtype: int64
**********************************************************************
0.0    37339
1.0     1674
2.0        7
Name: pub_rec_bankruptcies, dtype: int64
```

Our main objective is to identify the driving factors of loan defaulting before approving loan, so we can safely remove the columns which don't contribute in understanding if the defaulter analysis.

- The columns which show properties of after approval of loan should be removed as they don't contribute in defaulter analysis. These attributes are
  - delinq_2yrs
  - revol_bal
  - out_prncp
  - total_pymnt
  - mths_since_last_record
  - collection_recovery_fee
  - total_rec_prncp
  - total_rec_int
  - mths_since_last_delinq
  - recoveries
  - total_rec_late_fee
  - last_pymnt_d
  - last_pymnt_amnt
  - next_pymnt_d
  - chargeoff_within_12_mths
- We can remove 'id', 'member_id','url','zip_code','last_credit_pull_d','addr_state', since they are not contributing to loan defaulters.
- 'funded_amnt' can be reomved as we have 'funded_amnt_inv' which shows how much the investor has funded.
- 'desc' has description of the loan from which we can't analyse for now. so removing it
- 'out_prncp_inv', 'total_pymnt_inv', 'emp_title', 'title' also don't contribute to the loan defaulting analysis. So removing them.
- there are many single valued columns 'pymnt_plan', 'initial_list_status','collections_12_mths_ex_med','policy_code','acc_now_delinq', 'application_type', 'pub_rec_bankruptcies', 'tax_liens', 'delinq_amnt', which should be deleted.

In [312]:
```
# removing above mentioned columns
df.drop(["id", "member_id", "url", "zip_code", "last_credit_pull_d", "addr_state","desc","out_prncp_inv","total_pymnt_inv","funded_amnt", "delinq_2yrs", "revol_bal", "out_prncp", "total_pymnt", "total_rec_prncp", "total_rec_int", "total_rec_late_fee", "recoveries", "collection_recovery_fee", "last_pymnt_d", "last_pymnt_amnt", "chargeoff_within_12_mths",'pymnt_plan', "initial_list_status",'collections_12_mths_ex_med','policy_code','acc_now_delinq', 'application_type', 'pub_rec_bankruptcies', 'tax_liens', 'delinq_amnt','emp_title','title'], axis = 1, inplace = True)
```

In [313]:
```
#checking the shape of dataset after removing these columns
df.shape
```

Out[313]:
```
(39717, 21)
```

In [314]:
```
df.columns
```

Out[314]:
```
Index(['loan_amnt', 'funded_amnt_inv', 'term', 'int_rate', 'installment',
       'grade', 'sub_grade', 'emp_length', 'home_ownership', 'annual_inc',
       'verification_status', 'issue_d', 'loan_status', 'purpose', 'dti',
       'earliest_cr_line', 'inq_last_6mths', 'open_acc', 'pub_rec',
       'revol_util', 'total_acc'],
      dtype='object')
```

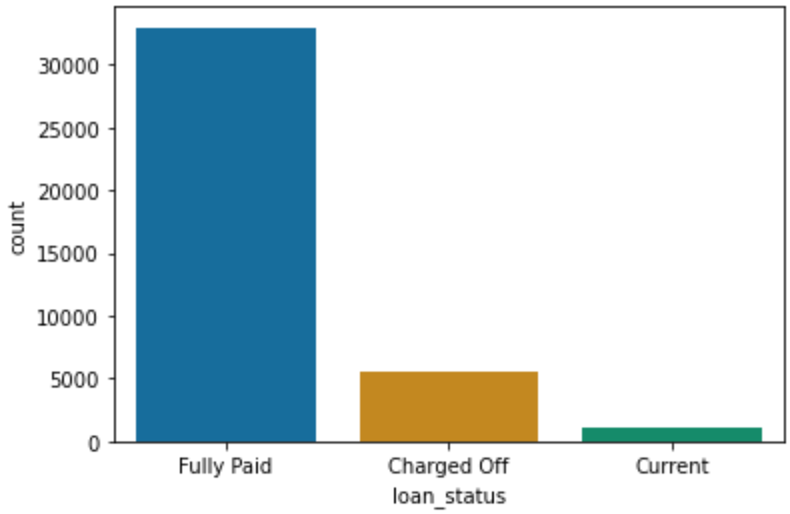
Since we have to look at the status of loan(loan_status), because that field would tell who has paid the loan and who are defaulters.

In [315]:
```
df.loan_status.value_counts()
```

Out[315]:
```
Fully Paid     32950
Charged Off     5627
Current         1140
Name: loan_status, dtype: int64
```

In [316]:
```
sns.countplot(df.loan_status)
```

Out[316]: <AxesSubplot:xlabel='loan_status', ylabel='count'>



Since loan status "Current" doesnt give any infomation for our analysis for approving or rejecting application, so dropping this.

In [317]:
```
#Removing the 'fully paid' records
df.drop(df[df['loan_status']=='Current'].index,inplace=True)
```

In [318]:
```
# Viewing the loan_status after deleting the 'fully paid records'
df.loan_status.unique()
```

Out[318]: array(['Fully Paid', 'Charged Off'], dtype=object)

In [319]:
```
df.shape
```

Out[319]: (38577, 21)

Checking for missing values and handling them.

In [320]:
```python
# Checking for missing values
df.isna().sum()/len(df)
```

Out[320]:
```
loan_amnt              0.000000
funded_amnt_inv        0.000000
term                   0.000000
int_rate               0.000000
installment            0.000000
grade                  0.000000
sub_grade              0.000000
emp_length             0.026778
home_ownership         0.000000
annual_inc             0.000000
verification_status    0.000000
issue_d                0.000000
loan_status            0.000000
purpose                0.000000
dti                    0.000000
earliest_cr_line       0.000000
inq_last_6mths         0.000000
open_acc               0.000000
pub_rec                0.000000
revol_util             0.001296
total_acc              0.000000
dtype: float64
```

In [321]:
```python
# Viewing the unique values of emp_length and their count
df.emp_length.value_counts()
```

Out[321]:
```
10+ years    8488
< 1 year     4508
2 years      4291
3 years      4012
4 years      3342
5 years      3194
1 year       3169
6 years      2168
7 years      1711
8 years      1435
9 years      1226
Name: emp_length, dtype: int64
```

In [322]:
```python
# Viewing the unique values of revol_util and their count
df.revol_util.value_counts()
```

Out[322]:
```
0%        954
0.20%      62
63%        62
40.70%     57
31.20%     57
         ...
77.63%      1
25.74%      1
0.83%       1
47.36%      1
7.28%       1
Name: revol_util, Length: 1088, dtype: int64
```

In [323]:
```python
# Checking 'emp_length' and 'revol_util' columns
print(df.emp_length.isnull().sum())
print(df.revol_util.isnull().sum())
```

```
1033
50
```

In [324]:
```python
# viewing the mode of these features
print(df.emp_length.mode()[0])
print(df.revol_util.mode()[0])
```

```
10+ years
0%
```

The above hypothesis shows us that the mode value has higher frequency than that of the next most frequent value.

- So we can safely assign the value of mode to the null values in the column.
- Also the missing values are in very low percentage. Hence it will not affect the analysis much

In [325]:
```python
# filling the null values with mode value
df.emp_length.fillna(df.emp_length.mode()[0], inplace=True)
df.revol_util.fillna(df.revol_util.mode()[0], inplace=True)
print(df.emp_length.isna().sum())
print(df.revol_util.isna().sum())
```

```
0
0
```

In [326]:
```
# Checking the datatype of the columns
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 38577 entries, 0 to 39716
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   loan_amnt             38577 non-null  int64  
 1   funded_amnt_inv       38577 non-null  float64
 2   term                  38577 non-null  object 
 3   int_rate              38577 non-null  object 
 4   installment           38577 non-null  float64
 5   grade                 38577 non-null  object 
 6   sub_grade             38577 non-null  object 
 7   emp_length            38577 non-null  object 
 8   home_ownership        38577 non-null  object 
 9   annual_inc            38577 non-null  float64
 10  verification_status   38577 non-null  object 
 11  issue_d               38577 non-null  object 
 12  loan_status           38577 non-null  object 
 13  purpose               38577 non-null  object 
 14  dti                   38577 non-null  float64
 15  earliest_cr_line      38577 non-null  object 
 16  inq_last_6mths        38577 non-null  int64  
 17  open_acc              38577 non-null  int64  
 18  pub_rec               38577 non-null  int64  
 19  revol_util            38577 non-null  object 
 20  total_acc             38577 non-null  int64  
dtypes: float64(4), int64(5), object(12)
memory usage: 6.5+ MB
```

Categorical and Numerical Data

- 'revol_util' is object column, but it has numerical data, so we can change it. also removing the '%'.
- Same is the case with 'int_rate'.
- 'emp_length' <1 is assumed as 0 and 10+ years is assumed as 10, so emp_length can be changed to numeric.

In [327]:
```
df.revol_util=pd.to_numeric(df.revol_util.apply(lambda x:x.split
('%')[0]))
df.int_rate=pd.to_numeric(df.int_rate.apply(lambda x:x.split
('%')[0]))
df.emp_length = pd.to_numeric(df.emp_length.apply(lambda x: 0 if
"<" in x else (x.split('+')[0] if "+" in x else x.split()[0])))
```

In [328]:
```
type(df.emp_length[0])
```

Out[328]:
```
numpy.int64
```

In [329]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 38577 entries, 0 to 39716
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   loan_amnt             38577 non-null  int64  
 1   funded_amnt_inv       38577 non-null  float64
 2   term                  38577 non-null  object 
 3   int_rate              38577 non-null  float64
 4   installment           38577 non-null  float64
 5   grade                 38577 non-null  object 
 6   sub_grade             38577 non-null  object 
 7   emp_length            38577 non-null  int64  
 8   home_ownership        38577 non-null  object 
 9   annual_inc            38577 non-null  float64
 10  verification_status   38577 non-null  object 
 11  issue_d               38577 non-null  object 
 12  loan_status           38577 non-null  object 
 13  purpose               38577 non-null  object 
 14  dti                   38577 non-null  float64
 15  earliest_cr_line      38577 non-null  object 
 16  inq_last_6mths        38577 non-null  int64  
 17  open_acc              38577 non-null  int64  
 18  pub_rec               38577 non-null  int64  
 19  revol_util            38577 non-null  float64
 20  total_acc             38577 non-null  int64  
dtypes: float64(6), int64(6), object(9)
memory usage: 7.5+ MB
```

We can see that around all the features which have object dtype are categorical datatypes (except 'issue_d' and 'earliest_cr_line' which are dates), and all the int and float dtypes features are numerical data.

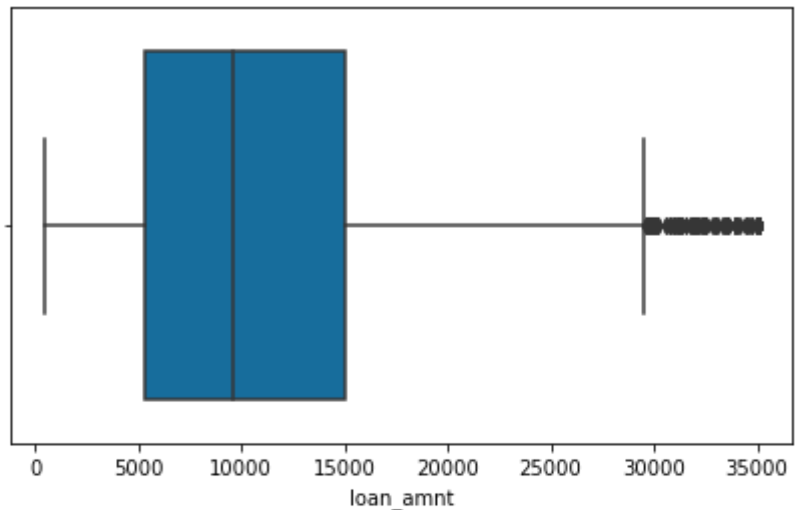
```
- Numerical Features
    - loan_amnt
    - funded_amnt_inv
    - int_rate
    - installment
    - emp_length
    - annual_inc
    - dti
    - revol_util
    - open_acc
    - total_acc
    - inq_last_6mths

- Categorical Features
    - term
    - grade
    - sub_grade
    - home_ownership
    - verification_status
    - loan_status
    - purpose
    - pub_rec

- Date Features
    - issue_d
    - earliest_cr_line
```

In [330]:

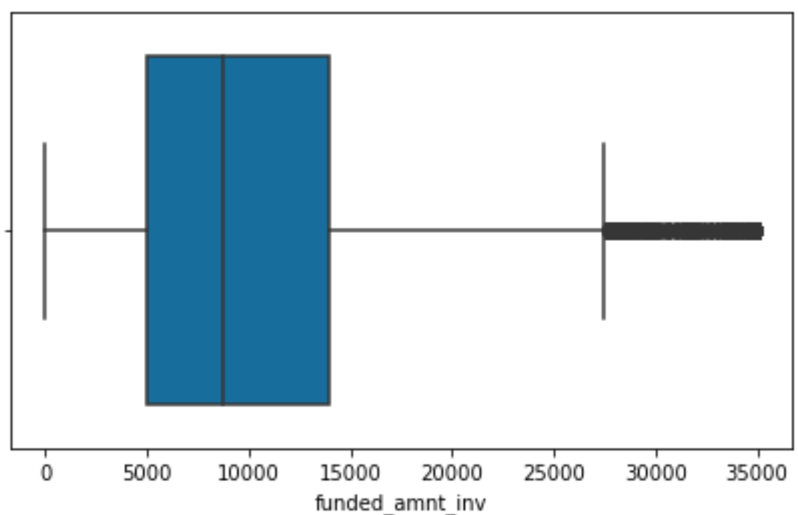
```
# Divideing the dataset into category and numerical datasets
cat_cols=['term','grade','sub_grade','home_ownership','pub_rec','verification_status','loan_status','purpose']
num_cols=['loan_amnt','funded_amnt_inv','int_rate','installment','emp_length','annual_inc','dti','revol_util','open_acc','total_acc','inq_last_6mths']
```

In [331]:
```
# Viewing outliers for numerical features
for i in num_cols:
    print(i.upper())
    plt.subplots(figsize=(7,4))
    sns.boxplot(df[i])
    plt.show()
    print()
```

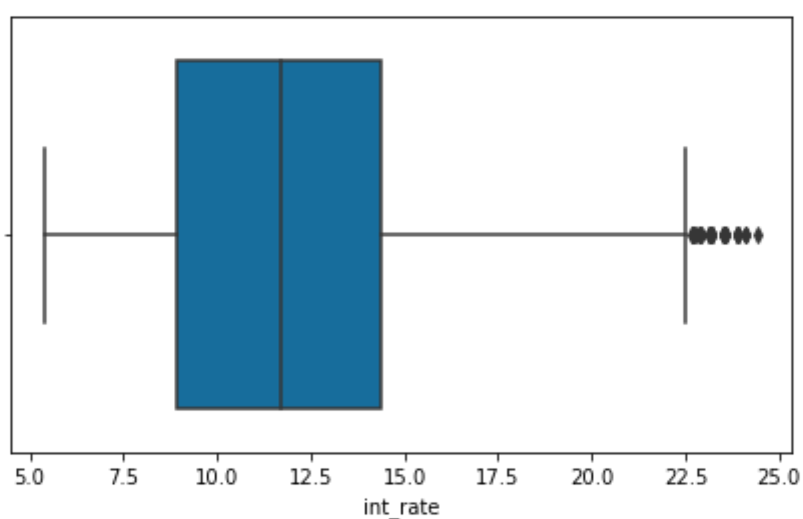
LOAN_AMNT



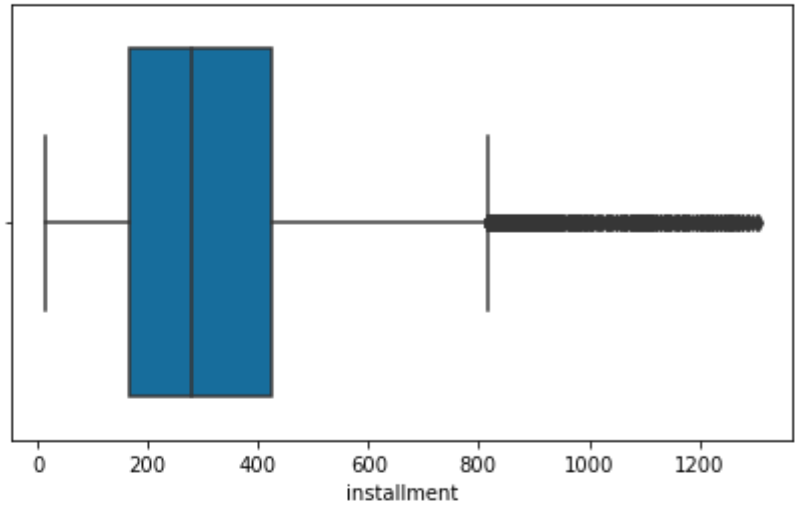
FUNDED_AMNT_INV



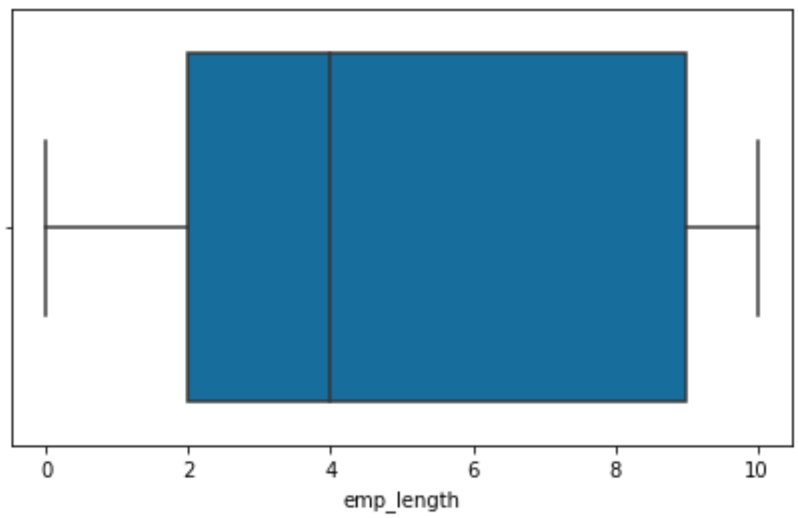
INT_RATE



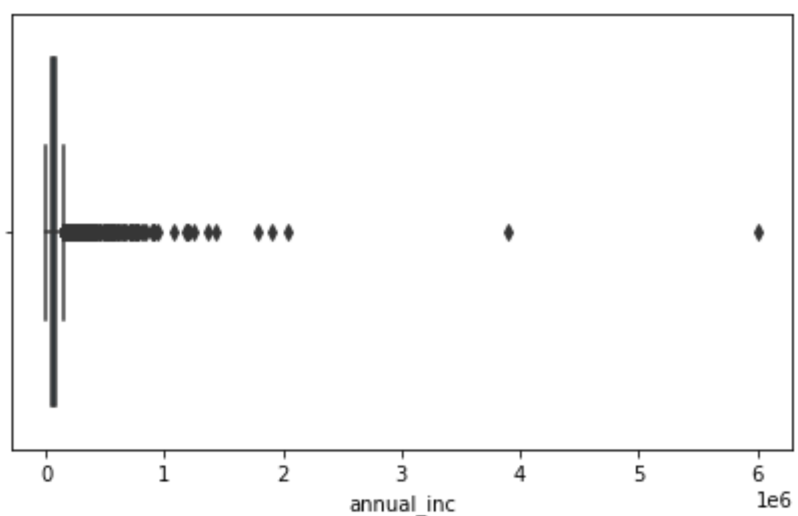
INSTALLMENT



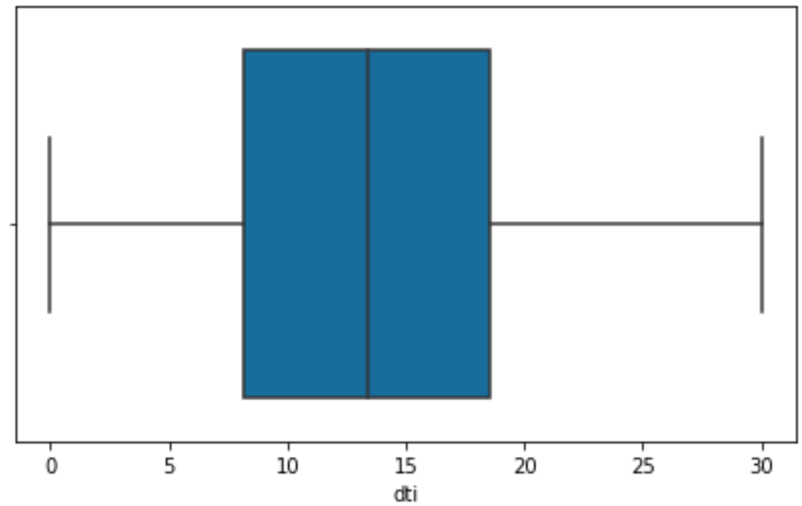
EMP_LENGTH



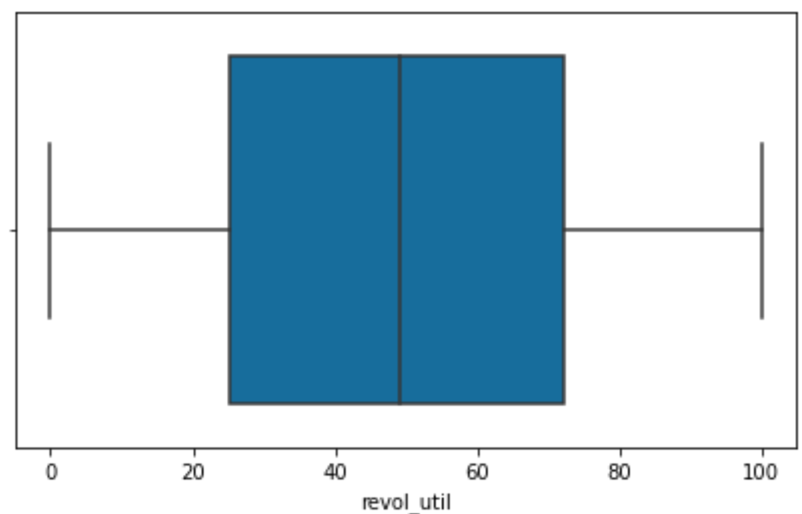
ANNUAL_INC



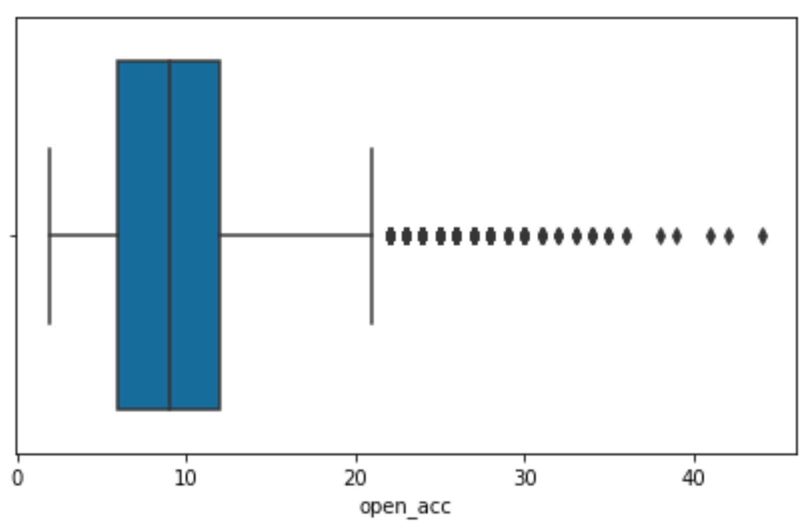
DTI



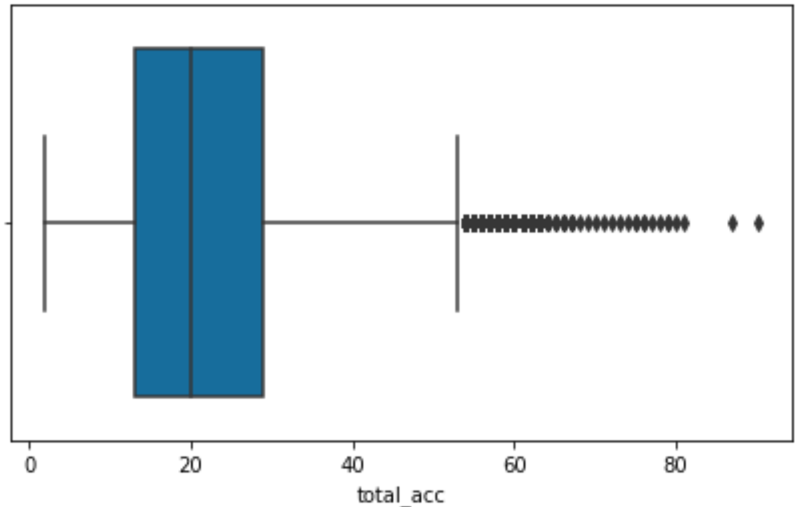
REVOL_UTIL



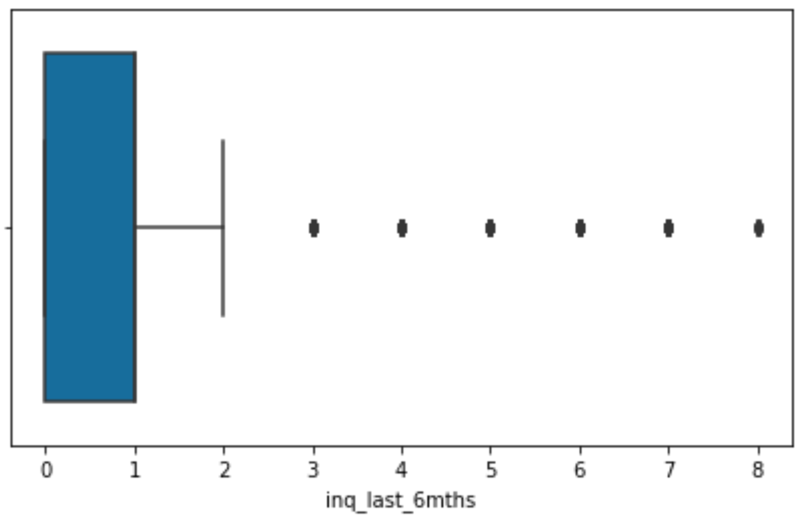
OPEN_ACC



TOTAL_ACC



INQ_LAST_6MTHS



In the above boxplots of numerical features, we see that except annual_inc, others are having some outliers, but they are negligible and more or less they are pretty continuous, so we can do our analysis on them and treat outliers of annual_inc

In [332]:
```
# treating annual_inc
df.annual_inc.quantile([0.25,0.5,0.75,0.90,0.95,0.98,0.99])
```

Out[332]:
```
0.25     40000.0
0.50     58868.0
0.75     82000.0
0.90    115000.0
0.95    140004.0
0.98    187000.0
0.99    234144.0
Name: annual_inc, dtype: float64
```

In [333]:
```
df['annual_inc'].value_counts()
```

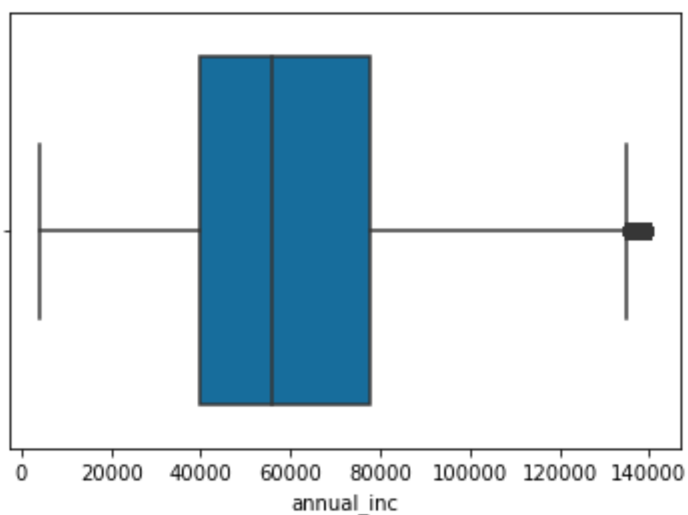
Out[333]:
```
60000.0     1466
50000.0     1029
40000.0      855
45000.0      811
30000.0      808
            ...
80569.0        1
82116.0        1
242400.0       1
133300.0       1
27376.0        1
Name: annual_inc, Length: 5215, dtype: int64
```

Above we see that the general customer is having annual income of under 120,000. But there are outliers which may bias our analysis since they are having very high income, So we must remove these outliers for a good analysis. we can choose a quantile percentage of 95 for removing the outliers.

In [334]:
```
df=df[df.annual_inc<=df.annual_inc.quantile(0.95)]
```

In [335]:
```
sns.boxplot(df.annual_inc)
```

Out[335]: <AxesSubplot:xlabel='annual_inc'>



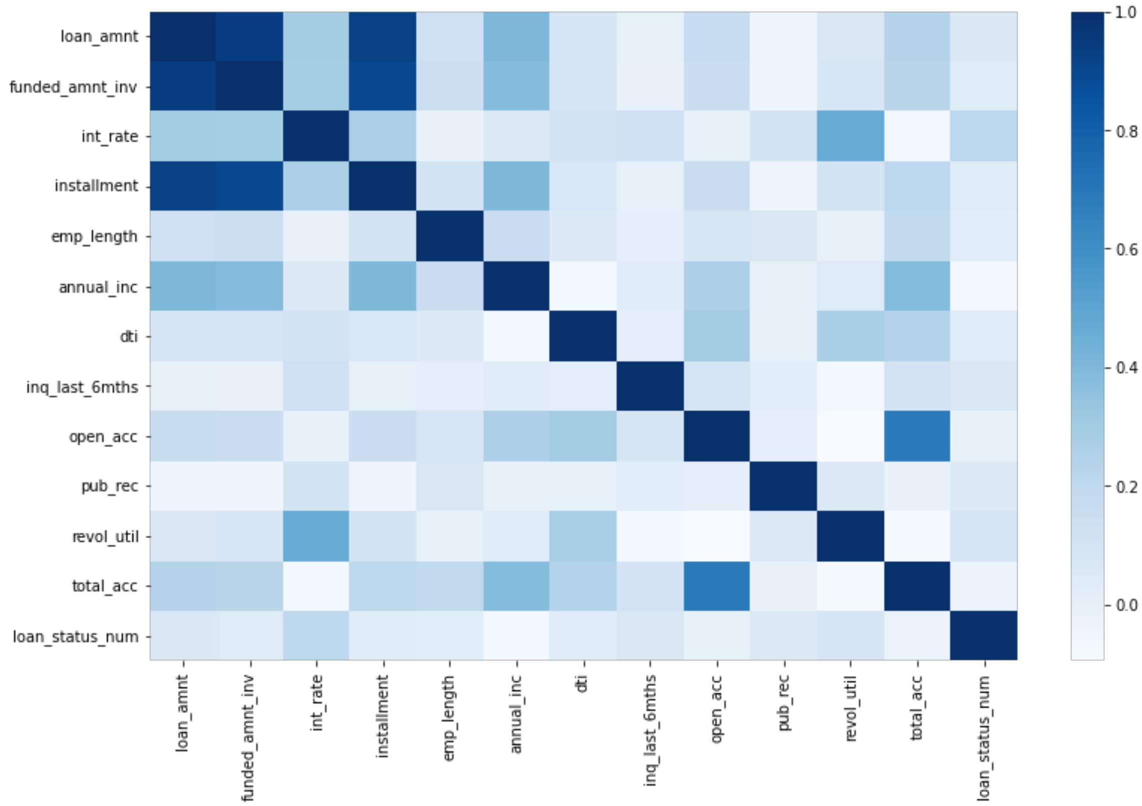
Now our annual_inc looks good for analysis.

We have successfully removed outliers which can bias our analysis. Now lets head towards the *Univariate and Bivariate Analysis* and draw meaningful Observations

Univariate Analysis

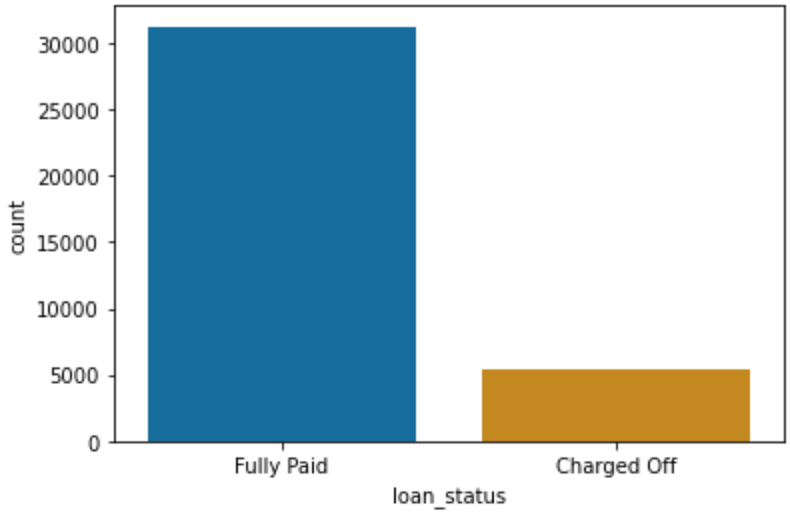
(We mostly have to compare the loan_status field with other fields and draw observations.)

In [336]:
```
# Let watch correlation of loan_status with other fields.
# changing the loan_status to a numeric variable, assign 1 for defa
ulted loans and 0 for paid off ones
df['loan_status_num']=df['loan_status'].apply(lambda x: 1 if x=='Ch
arged Off' else 0)
plt.subplots(figsize=(13,8))
sns.heatmap(df.corr(),cmap='Blues')
plt.show()
```



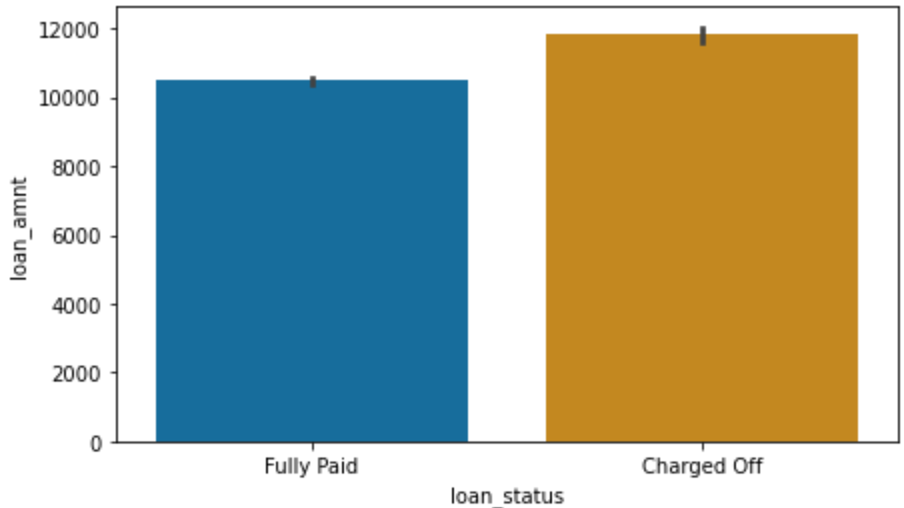
Since we know darker the value heigher the correlation , we can clearly see loan_amnt, funded_amnt, funded_amnt_inv and installment have high correlation. The public records related fields pub_rec and number of accounts related fields open_acc & total_acc are also correlated.

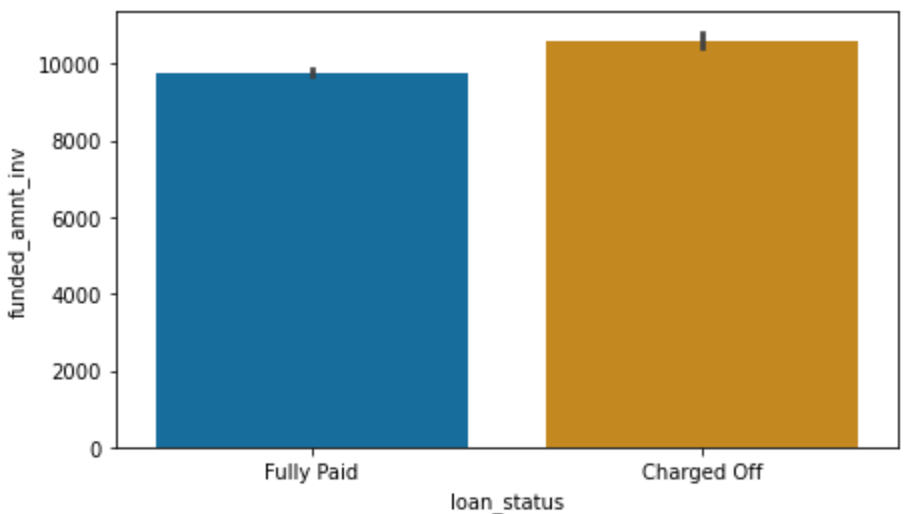
In [337]:
```
sns.countplot(df.loan_status)
plt.show()
```

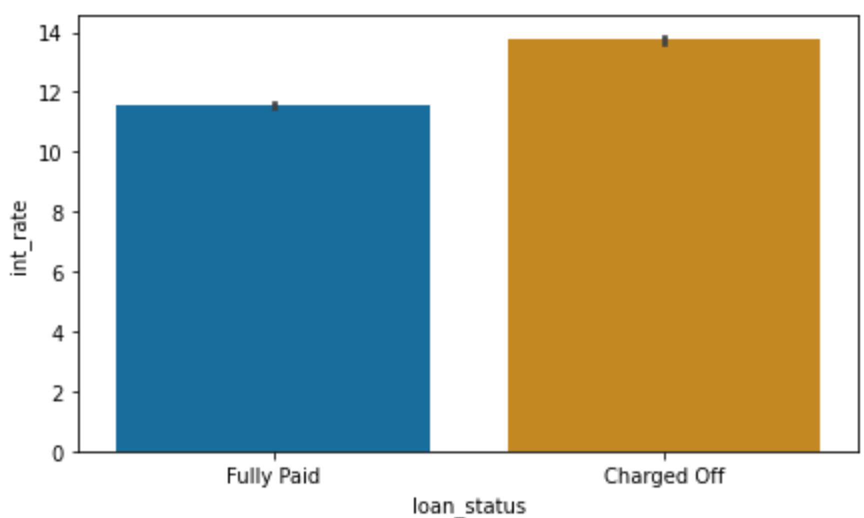


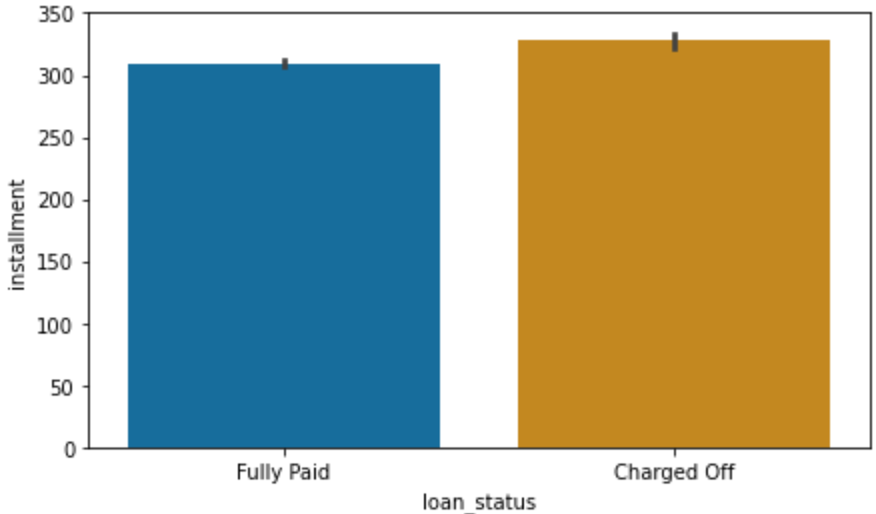
loan_status vs other numerical columns

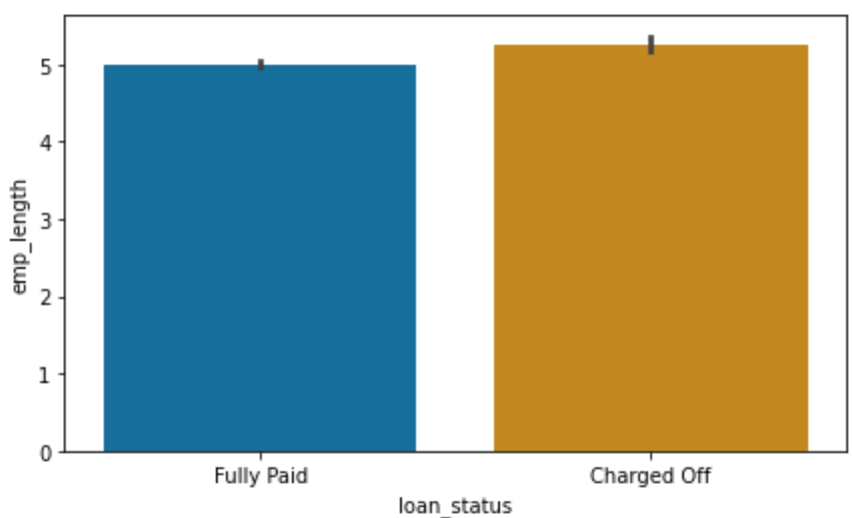
In [338]:
```
for i in num_cols:
    plt.subplots(figsize=(7,4))
    sns.barplot(x=df.loan_status,y=df[i])
    plt.show()
```

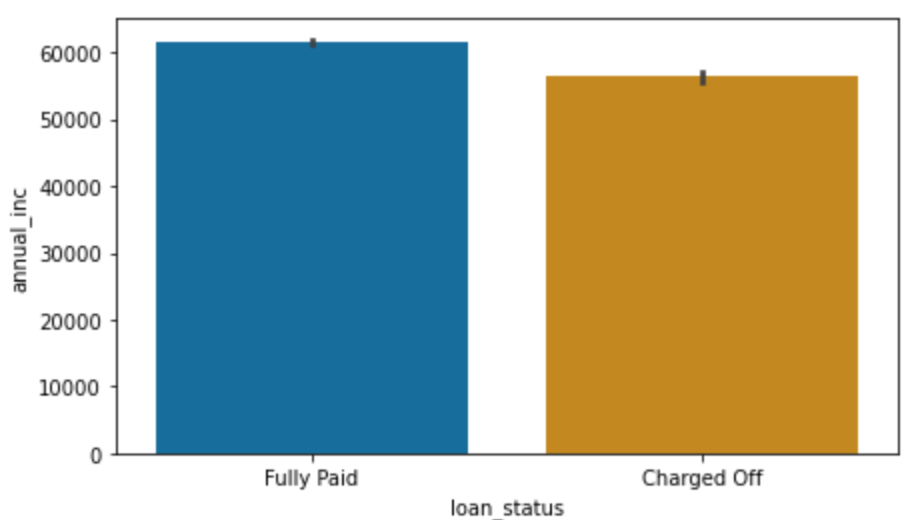

12000
10000
8000
6000
4000
2000
0
loan_amnt
Fully Paid
Charged Off
loan_status

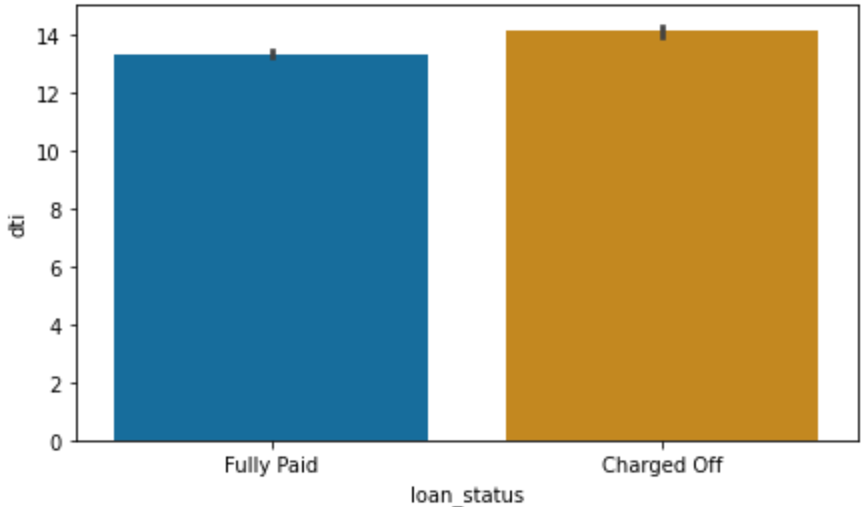

10000
8000
6000
4000
2000
0
funded_amnt_inv
Fully Paid
Charged Off
loan_status

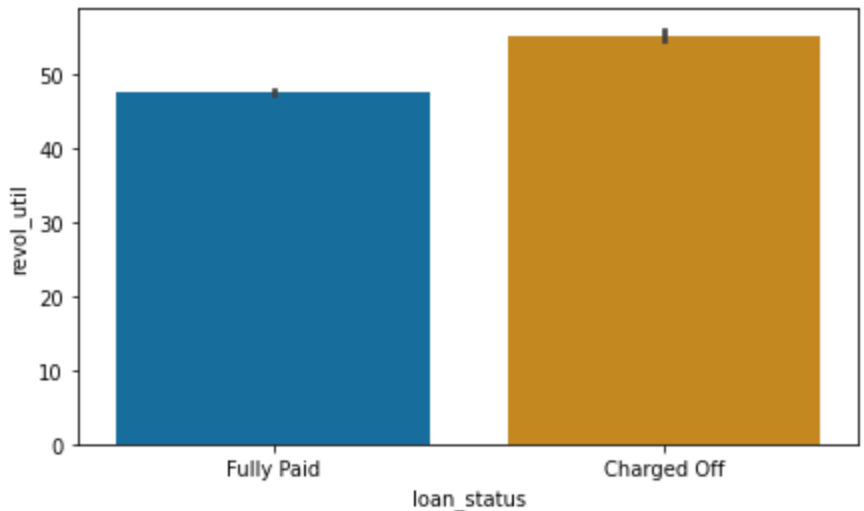

14
12
10
8
6
4
2
0
int_rate
Fully Paid
Charged Off
loan_status

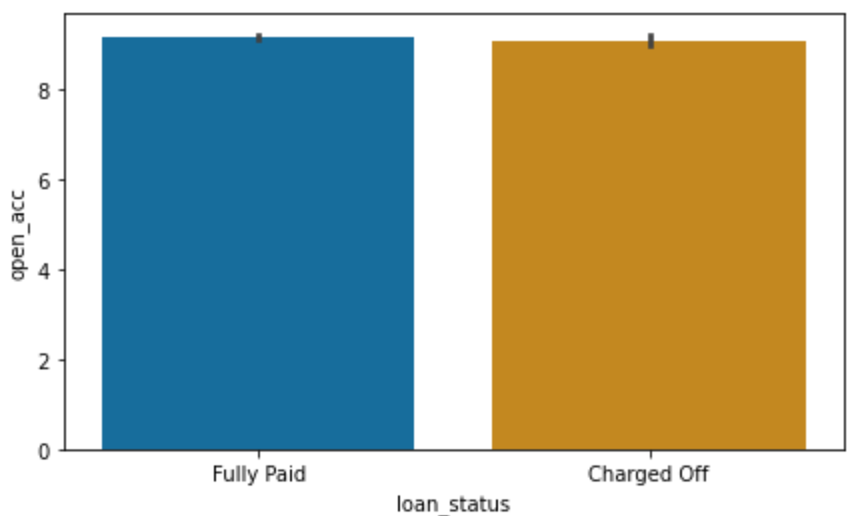

350
300
250
200
150
100
50
0
installment
Fully Paid
Charged Off
loan_status

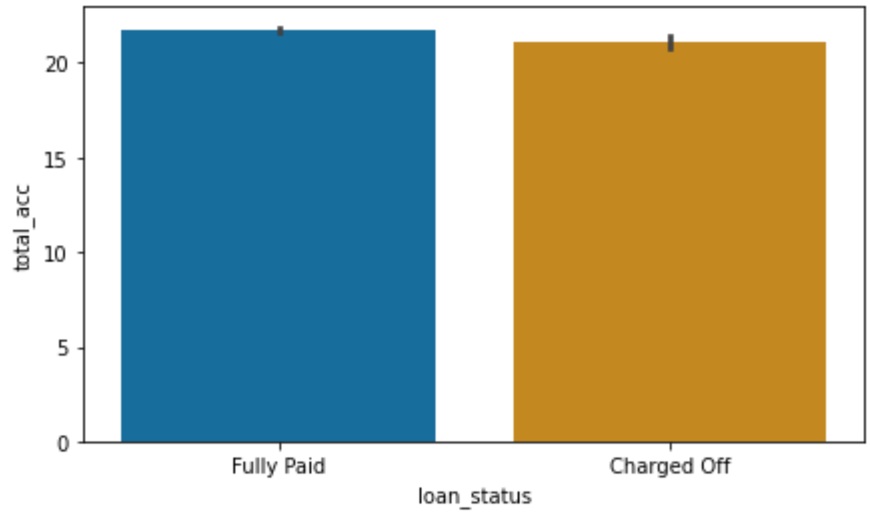

5
4
3
2
1
0
emp_length
Fully Paid
Charged Off
loan_status

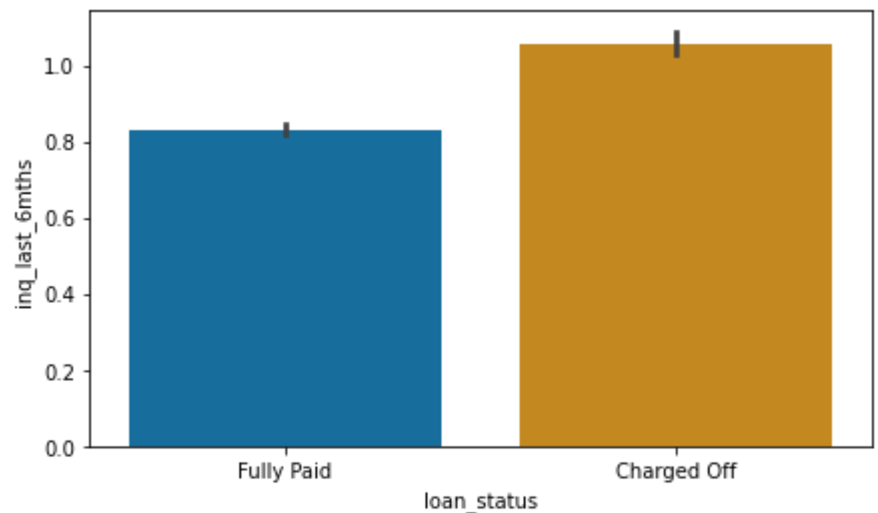

60000
50000
40000
30000
20000
10000
0
annual_inc
Fully Paid
Charged Off
loan_status


dti
Fully Paid
Charged Off
loan_status


revol_util
Fully Paid
Charged Off
loan_status


open_acc
Fully Paid
Charged Off
loan_status



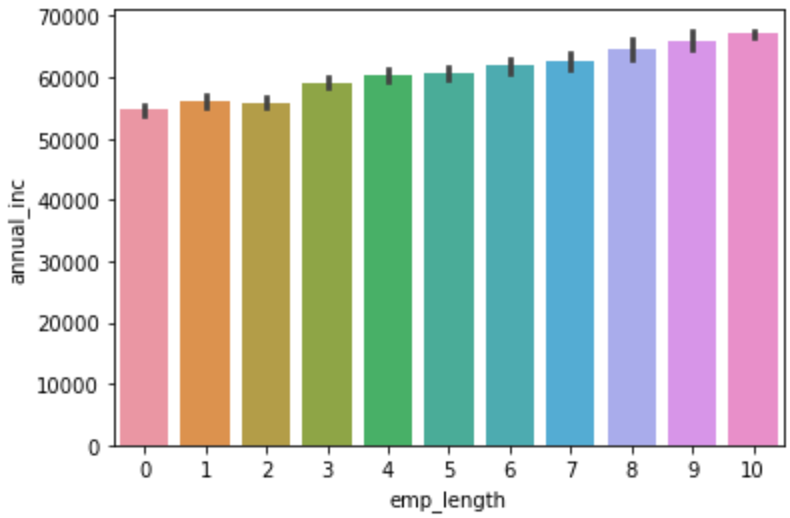


Observations

- We can see that the probability of loan defaulted loans are higher as loan_amounts and the funded amounts increases obviously
- Same is the case with interest rates too, because it must be increasing the total loan and hence people who has around 11% to 16% interest rates default on their loans.
- Employment length shows an interesting characterstic. It shows more years the people have been employed, more they belong to loan defaulter's category. After analysing the emp_length and annual_income of them below, we see that as the years of employemt increases, the salary is not increases by a significant amount, and hence it might result in this scenario.
- Annual income show an expected result that more the income, more susceptible to pay the loans off.
- dti, revol_util, open_acc, total_acc, shows nothing extraordinary results.
- Inquiry Last 6 months shows the persons who inquired in the last 6 months have high probability of loan defaulters. It maybe due to urgence of money, they take the loans but are unable to pay. It might be an important feature.

Comparing the age of employment and the annual income (its observation is written in above text.)

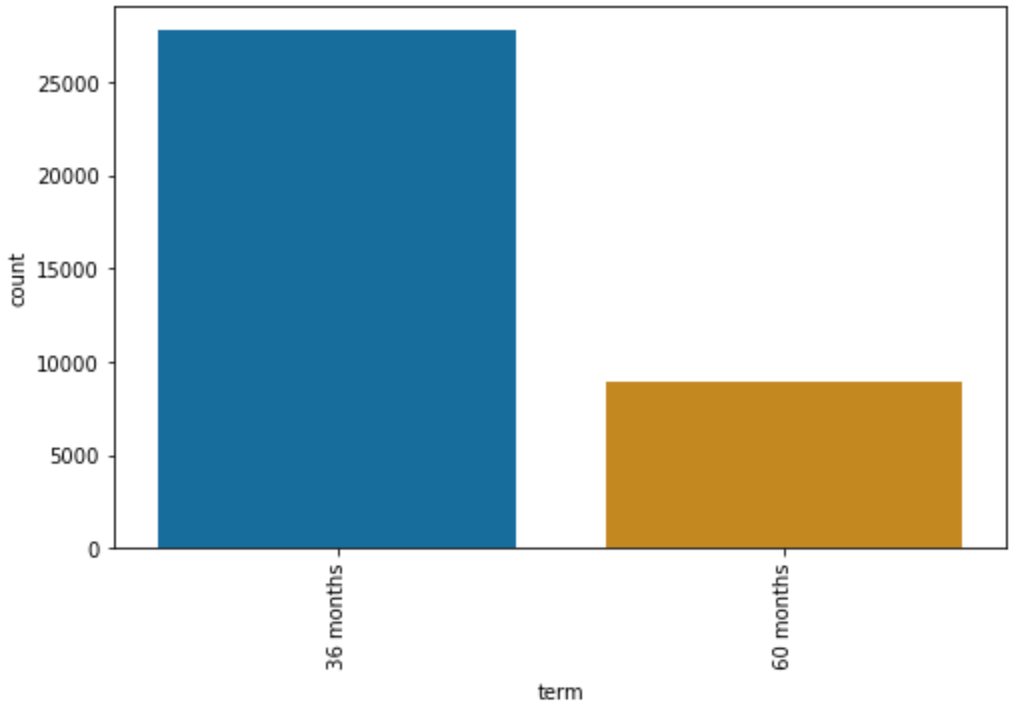
In [339]:
```
sns.barplot(df.emp_length,df.annual_inc)
plt.show()
```

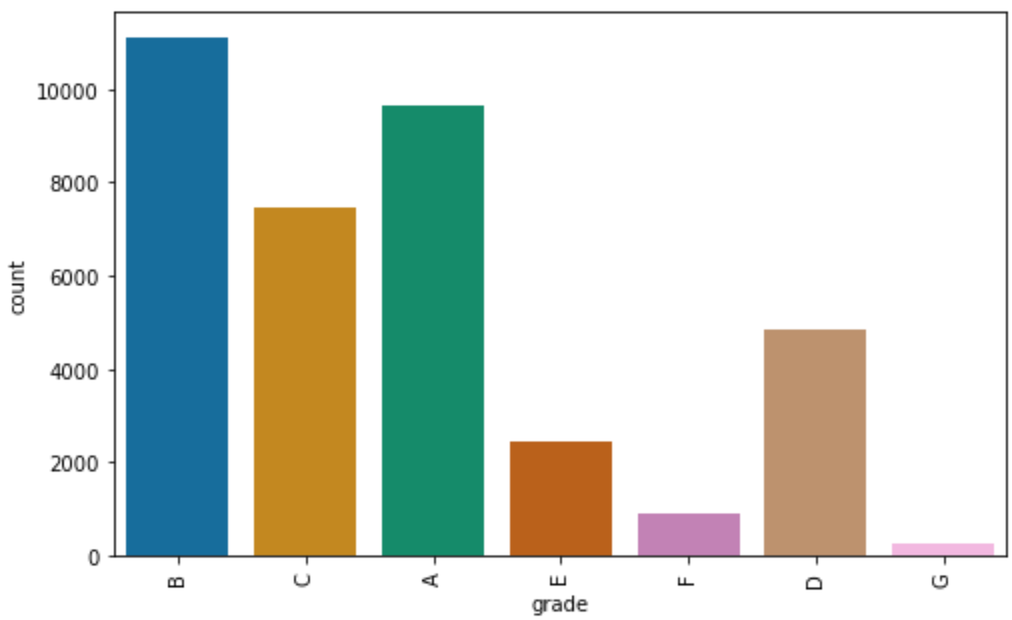


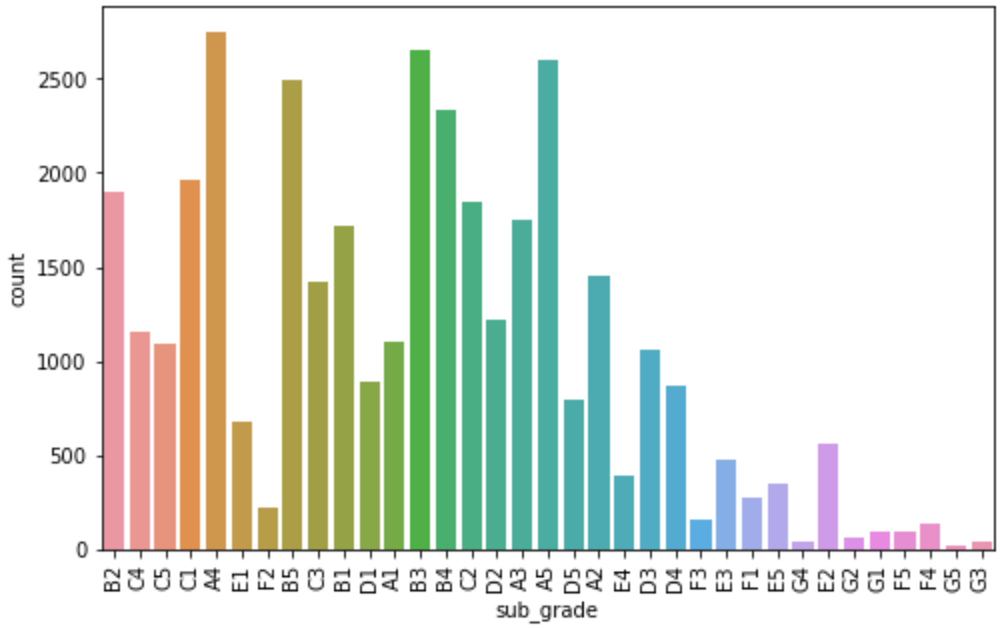
loan_status vs categorical columns

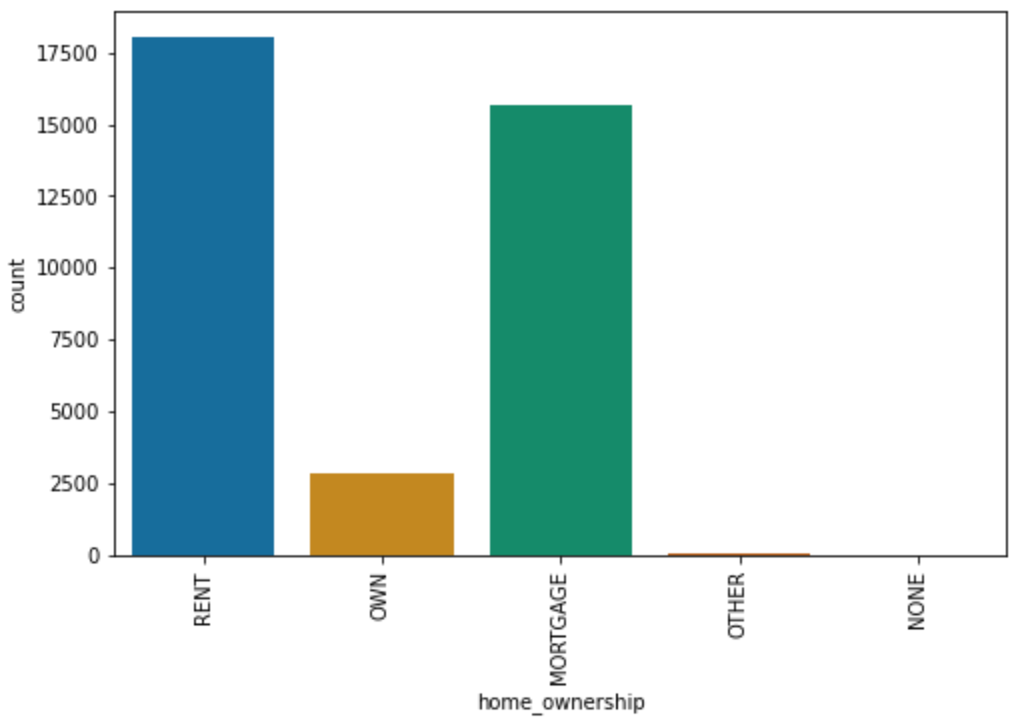
We are analyzing and visualizing only the defaulter data. So subsetting the data while plotting only for 'Charged Off' loan_status for below plots

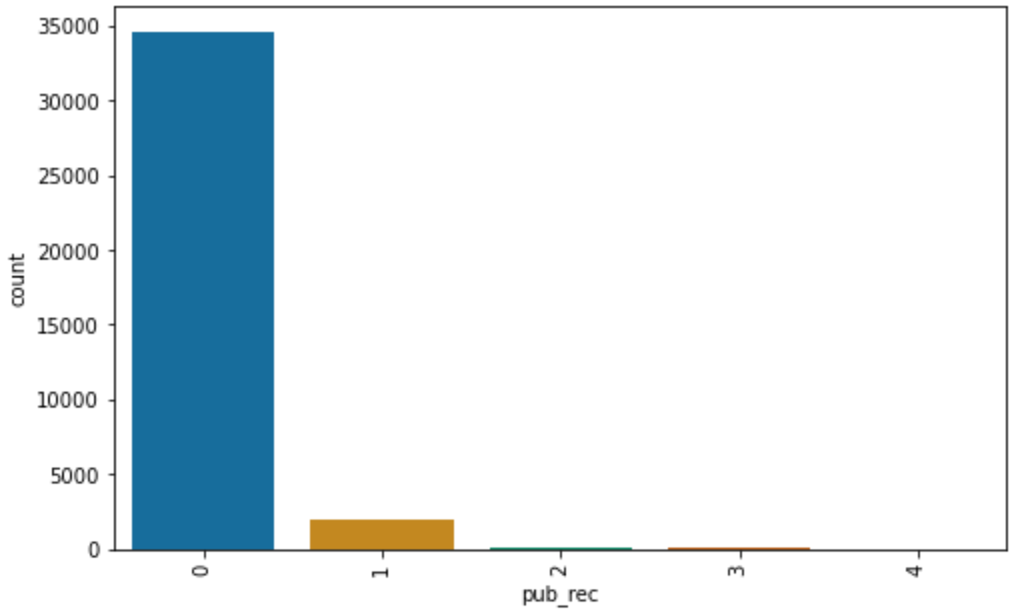
In [340]:
```
for i in cat_cols:
    plt.subplots(figsize=(8,5))
    sns.countplot(df[i],data=df[df['loan_status']=='Charged Off'])
    plt.xticks(rotation=90)
    plt.show()
```

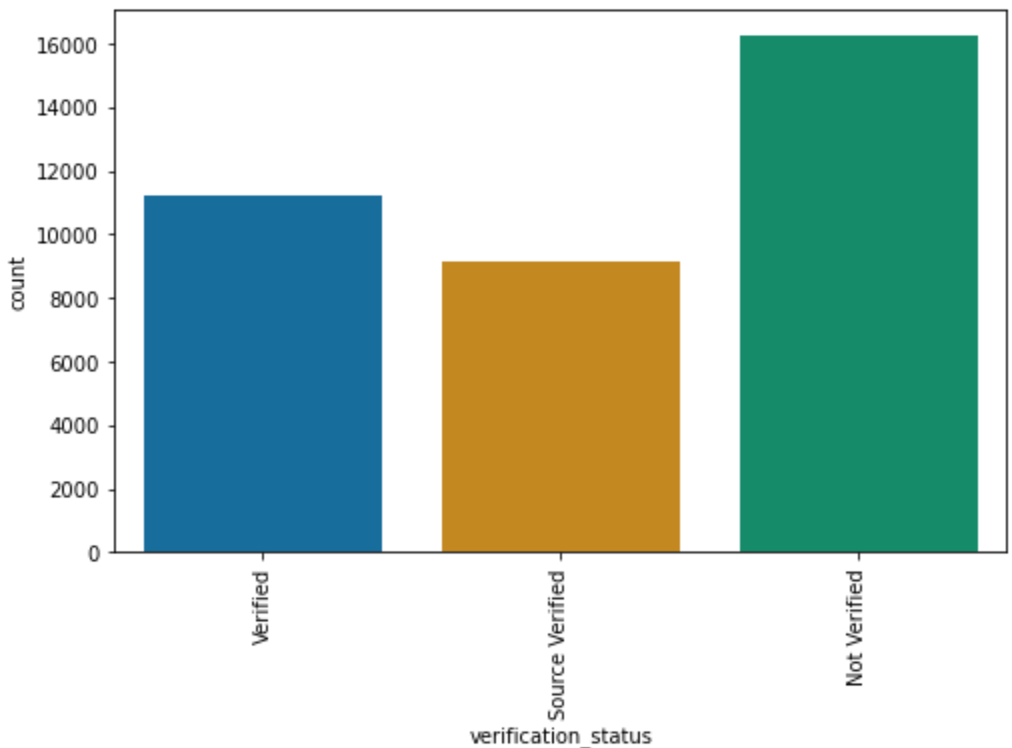

25000
20000
15000
10000
5000
0
count
36 months
60 months
term

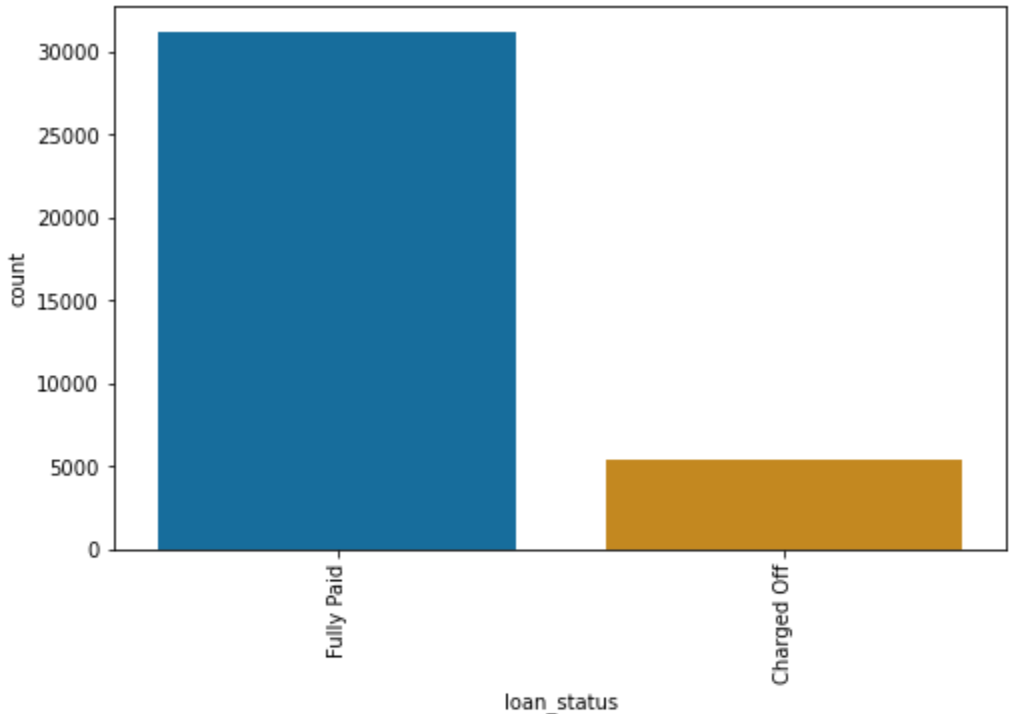

10000
8000
6000
4000
2000
0
count
B
C
A
E
F
D
G
grade

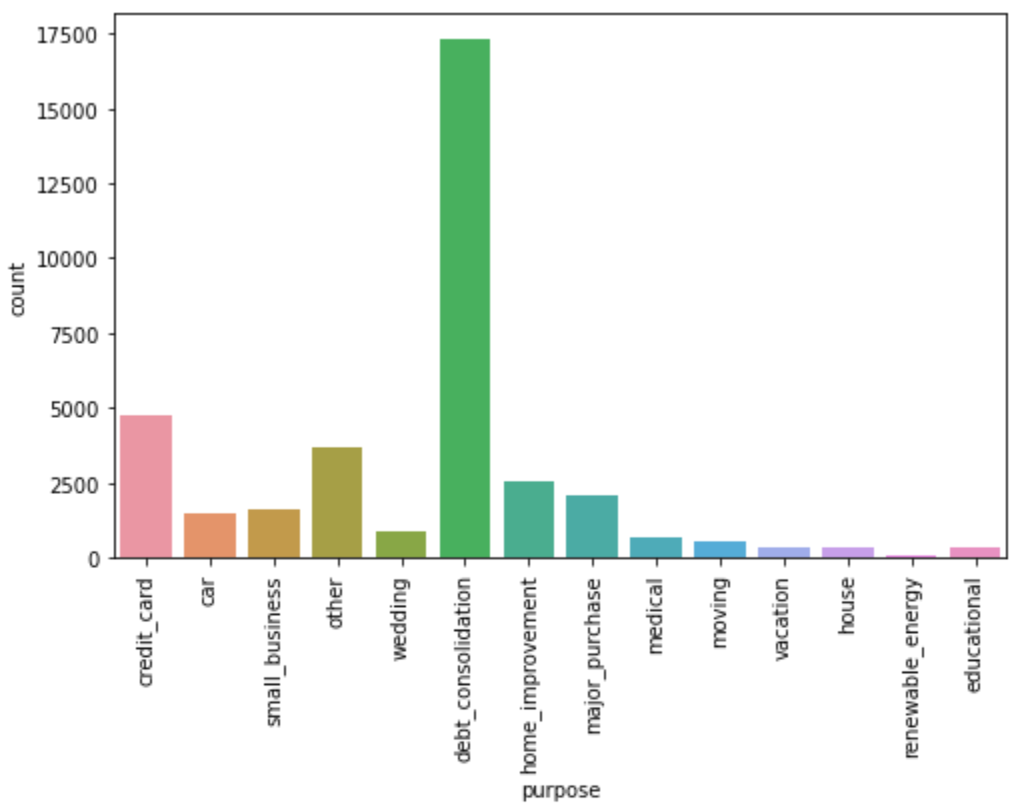

count
sub_grade
0
500
1000
1500
2000
2500
B2
C4
C5
C1
A4
E1
F2
B5
C3
B1
D1
A1
B3
B4
C2
D2
A3
A5
D5
A2
E4
D3
D4
F3
E3
F1
E5
G4
E2
G2
G1
F5
F4
G5
G3


count
home_ownership
0
2500
5000
7500
10000
12500
15000
17500
RENT
OWN
MORTGAGE
OTHER
NONE


35000
30000
25000
20000
15000
10000
5000
0
count
0
1
2
3
4
pub_rec


16000
14000
12000
10000
8000
6000
4000
2000
0
count
Verified
Source Verified
Not Verified
verification_status


count
30000
25000
20000
15000
10000
5000
0
Fully Paid
Charged Off
loan_status


count
17500
15000
12500
10000
7500
5000
2500
0
credit_card
car
small_business
other
wedding
debt_consolidation
home_improvement
major_purchase
medical
moving
vacation
house
renewable_energy
educational
purpose

Observaitons

- 36 months/3 Years loan category is facing significant amount of loan defaulters and compared to 60 months/5 years
- We can clearly see that loan grades A, B,C having highest defaulters. G, F, E and D form grades where default rate is much lower.
- home_ownership shows us that Rent and Morthage have high probability of loan defaulting as compared to rest.
- When the number of derogatory public records is 0, then higher chance of default.
- When verificaiton status is not verified, higher chance of default, but verified and source verified have significant defaulters too.
- When the purpose of the debt is debt consolidation(debt taken to pay other debts or liabilities), then it has higher chance of default.

For more insights about the data, we can divide the numerical data into groups and then analyze them again with only charged_off loans (defaulter loans)

In [341]:
```
# creating new columns by deriving categorical columns from numeric
al columns by dividing them into bins
df['int_rate_groups'] = pd.cut(df['int_rate'], bins=5,precision =0,
labels=['5%-9%','9%-13%','13%-17%','17%-21%','21%-24%'])
df['open_acc_groups'] = pd.cut(df['open_acc'],bins = 5,precision =
0,labels=['2-10','10-19','19-27','27-36','36-44'])
df['revol_util_groups'] = pd.cut(df['revol_util'], bins=5,precision
=0,labels=['0-20','20-40','40-60','60-80','80-100'])
df['total_acc_groups'] = pd.cut(df['total_acc'], bins=5,precision =
0,labels=['2-20','20-37','37-55','55-74','74-90'])
df['annual_inc_groups'] = pd.cut(df['annual_inc'], bins=5,precision
=0,labels =['3k-31k','31k-58k','58k-85k','85k-112k','112k-140k'])
df['installment_groups'] = pd.cut(df['installment'], bins=10,precis
ion =0,labels=['14-145','145-274','274-403','403-531','531-660','66
0-789','789-918','918-1047','1047-1176','1176-1305'])
df['funded_amnt_inv_group'] = pd.cut(df['funded_amnt_inv'], bins=7,
labels=['0-5k','5k-10k','10k-15k','15k-20k','20k-25k','25k-30k','30
k-35k'])
df['loan_amnt_groups'] = pd.cut(df['loan_amnt'], bins=7,precision =
0,labels=['0-5k','5k-10k','10k-15k','15k-20k','20k-25k','25k-30k','
30k-35k'])
df['dti_groups'] = pd.cut(df['dti'], bins=5,precision =0,labels=['0
-6','6-12','12-18','18-24','24-30'])
```

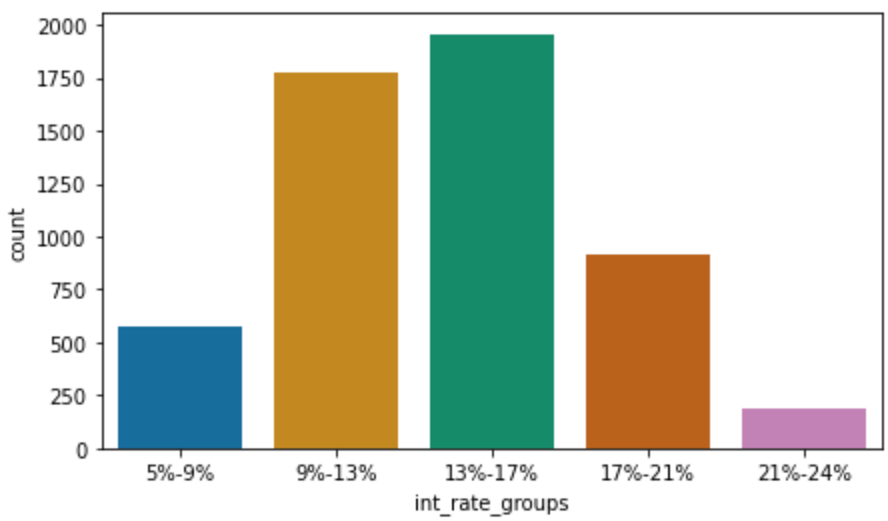
In [342]:
```
# Viewing the new df
df.head()
```

Out[342]:

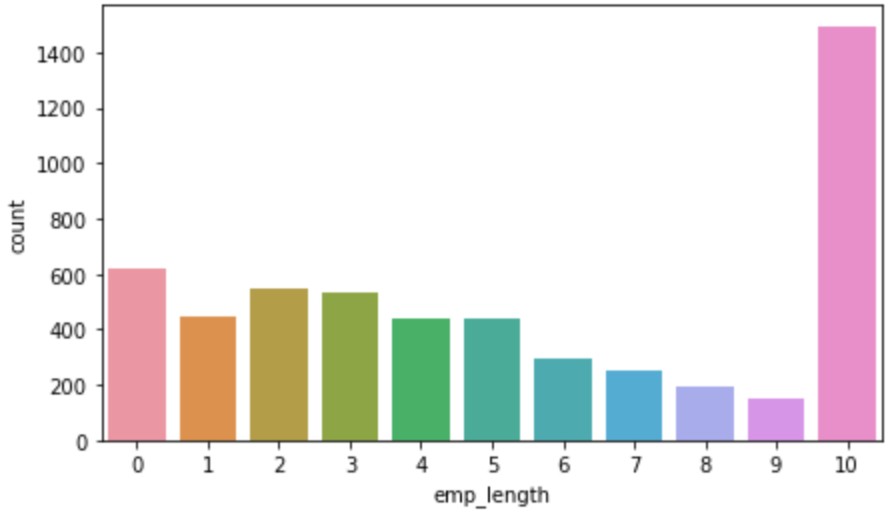
	loan_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	emp_length
0	5000	4975.0	36 months	10.65	162.87	B	B2	10
1	2500	2500.0	60 months	15.27	59.83	C	C4	0
2	2400	2400.0	36 months	15.96	84.33	C	C5	10
3	10000	10000.0	36 months	13.49	339.31	C	C1	10
5	5000	5000.0	36 months	7.90	156.46	A	A4	3

5 rows × 31 columns

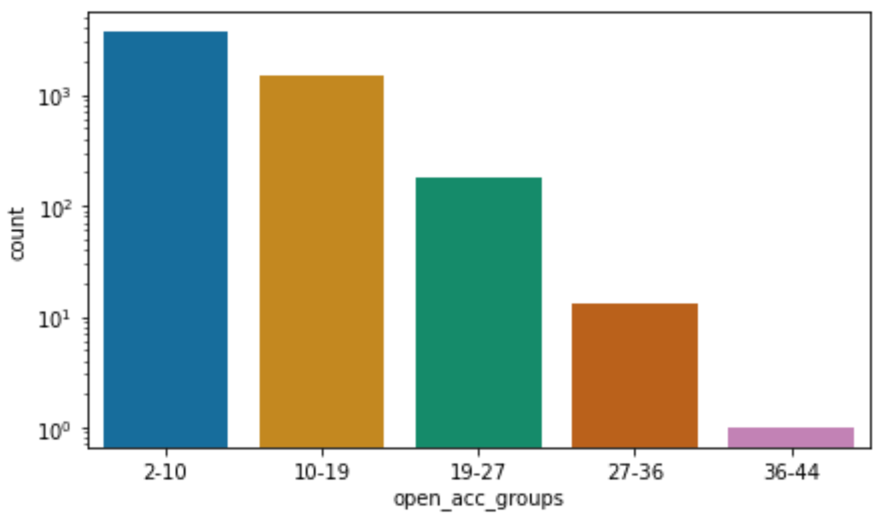
In [343]:
```
# Analyzing the interest rates groups with the defaulter's. It show
s 9% to 17% are having high chance of defaulter
plt.subplots(figsize = (7,4))
sns.countplot(x='int_rate_groups', data=df[df.loan_status == 'Charg
ed Off'])
plt.show()
```



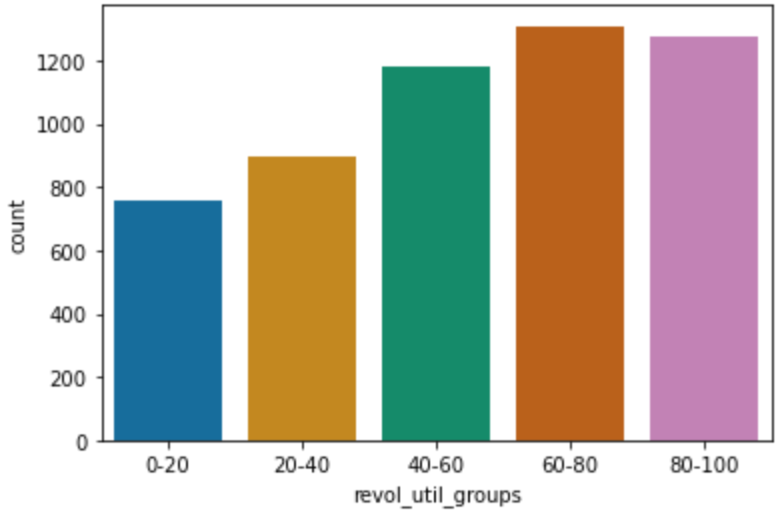
In [344]:
```
# Analyzing years of employemt
plt.subplots(figsize = (7,4))
sns.countplot(x='emp_length', data=df[df.loan_status == 'Charged Of
f'])
plt.show()
```



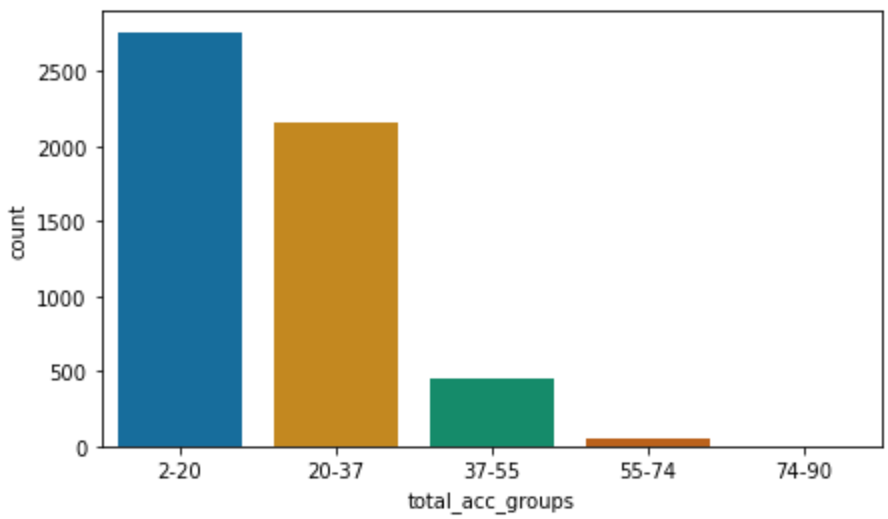
In [345]:
```
# Analysing open_account_groups
fig,ax=plt.subplots(figsize = (7,4))
ax.set_yscale('log')
sns.countplot(x='open_acc_groups', data=df[df.loan_status == 'Charg
ed Off'])
plt.show()
```



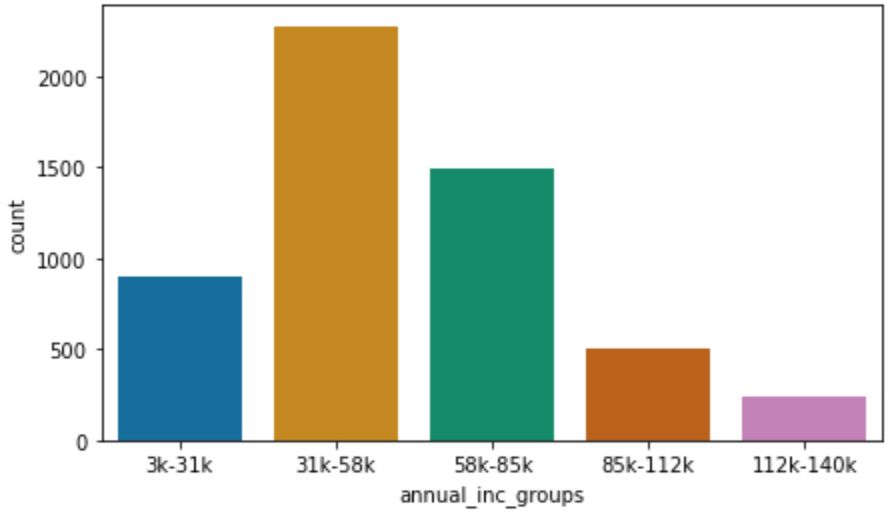
In [346]:
```
# Analysing revol_util_groups
sns.countplot(x='revol_util_groups', data=df[df.loan_status == 'Cha
rged Off'])
plt.show()
```



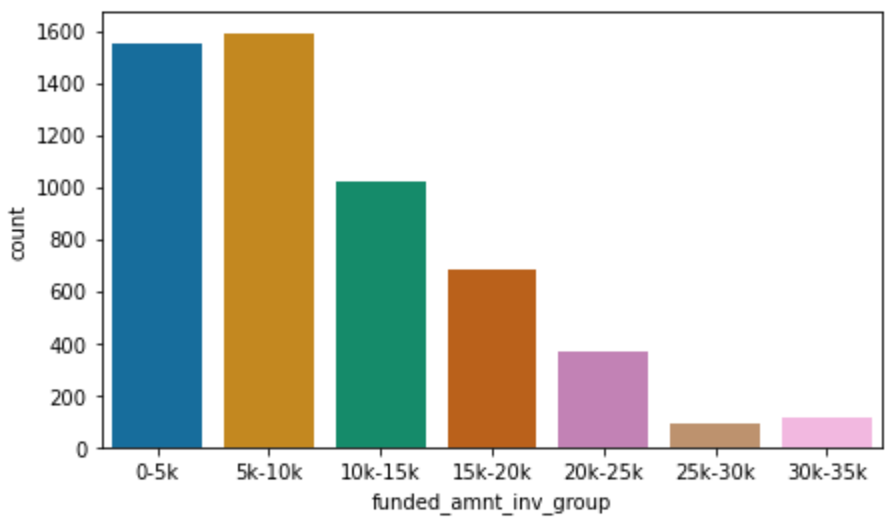
In [347]:
```
# Analysing total_Acc_groups
plt.subplots(figsize = (7,4))
ax.set_yscale('log')
sns.countplot(x='total_acc_groups', data=df[df.loan_status == 'Char
ged Off'])
plt.show()
```



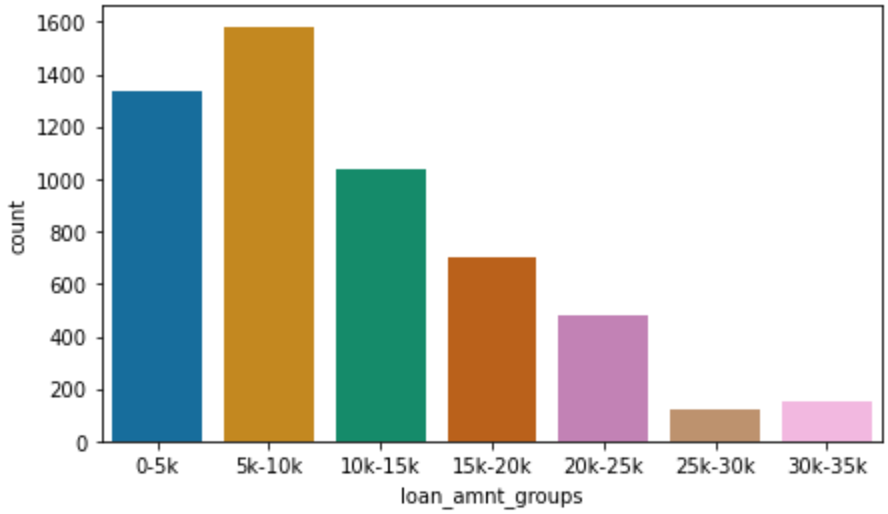
In [348]:
```
# Analysing the annual_income groups
plt.subplots(figsize = (7,4))
sns.countplot(x='annual_inc_groups', data=df[df.loan_status == 'Cha
rged Off'])
plt.show()
```



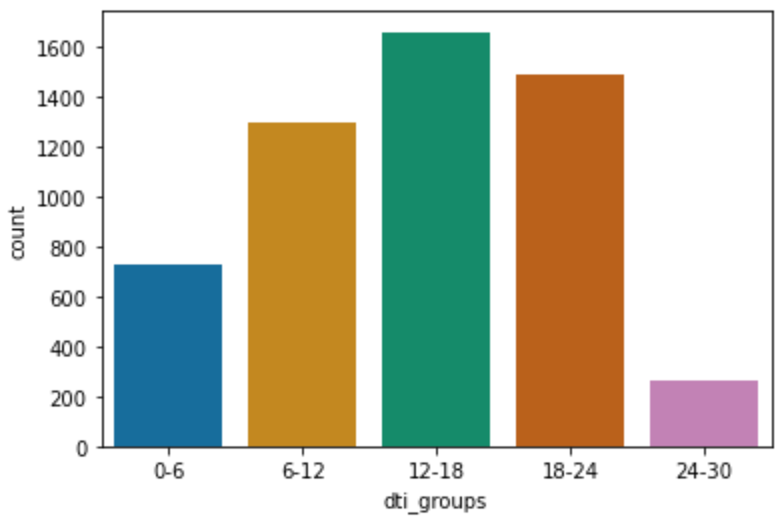
In [349]:
```
# Analysing the funded amount
plt.subplots(figsize = (7,4))
ax.set_yscale('log')
sns.countplot(x='funded_amnt_inv_group', data=df[df['loan_status']=
='Charged Off'])
plt.show()
```



In [350]:
```
# Analysing the loan amount
plt.subplots(figsize = (7,4))
ax.set_yscale('log')
sns.countplot(x='loan_amnt_groups', data=df[df['loan_status']=='Charged Off'])
plt.show()
```



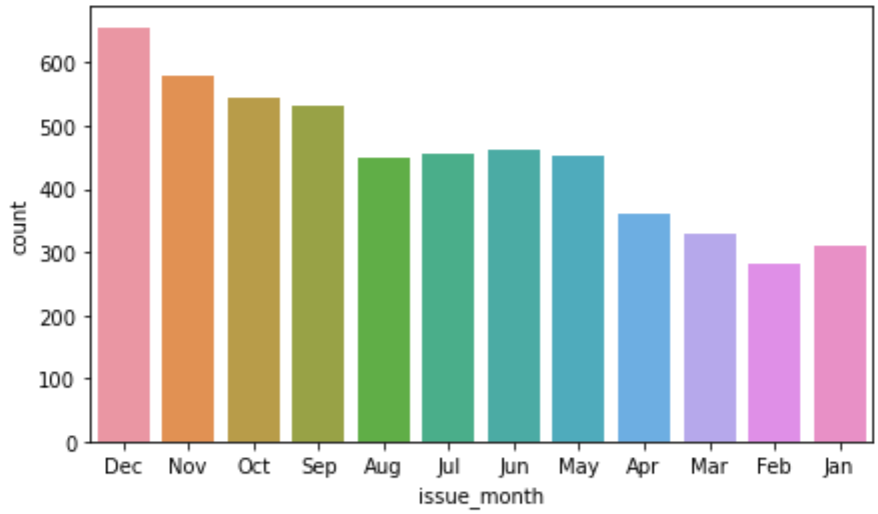
In [351]:
```
# analysing the dti
sns.countplot(x='dti_groups', data=df[df['loan_status']=='Charged Off'])
plt.show()
```



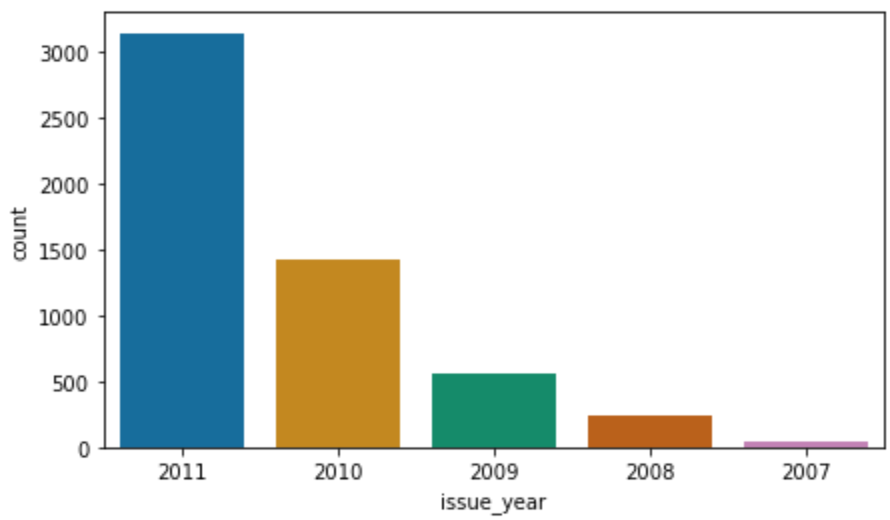
Analysing by Month and Year

In [352]:
```
## Extracting month and year
df_month_year = df['issue_d'].str.partition("-", True)
df['issue_month']=df_month_year[0]
df['issue_year']='20' + df_month_year[2]
```

In [353]:
```
plt.figure(figsize=(7,4))
sns.countplot(x='issue_month', data=df[df['loan_status']=='Charged Off'])
plt.show()
```



In [354]:
```
plt.figure(figsize=(7,4))
sns.countplot(x='issue_year', data=df[df['loan_status']=='Charged Off'])
plt.show()
```



The above analysis shows us that maximum number of defaults occured when the loan was issued in December.

Loan issued in the year 2011 also performed poorly as compared to other years. (maybe due to financial issues)

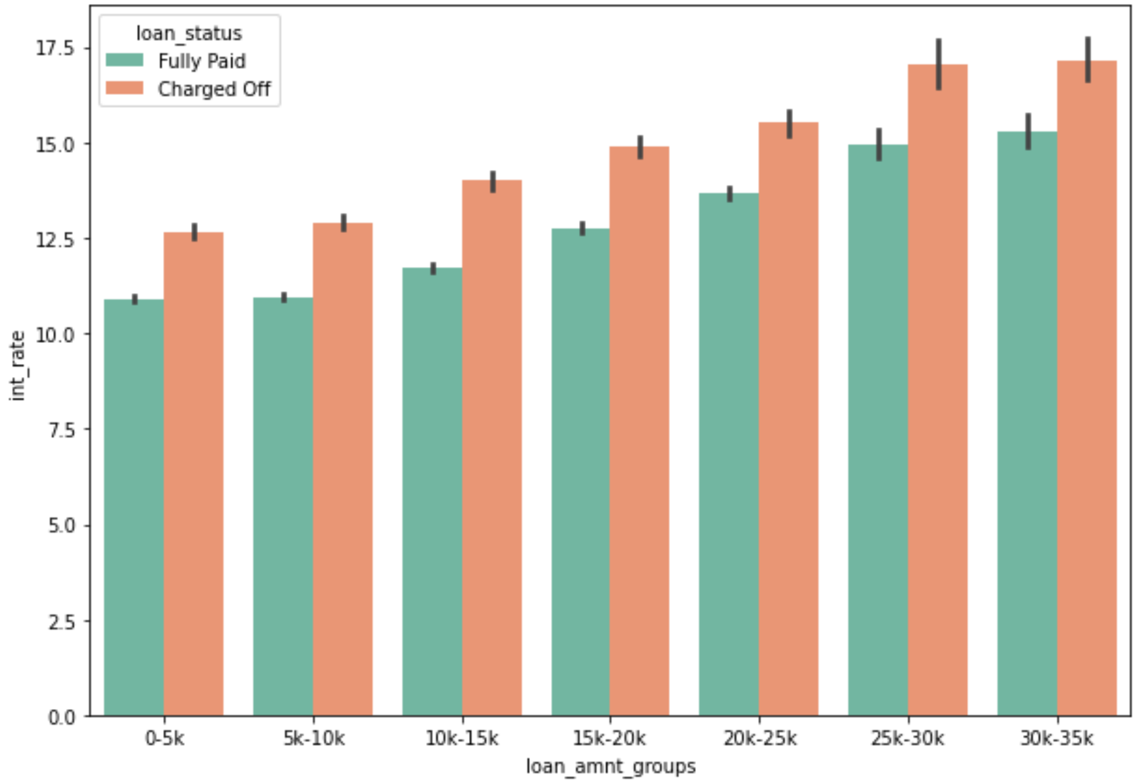
Based on the above observations, we can say that there is more chance of defaulting in following cases :

- Borrower who use the loan to clear other debts
- When funded amount by investor is between 5000-10000
- Borrower having house_ownership as 'RENT'
- Borrower with employement length of 10
- Borrower who have an income of range 31201 - 58402
- Borrower who have 20-37 open_acc
- Borrower who receive interest at the rate of 13-17%
- Loan amount is between 5429 - 10357
- Dti is between 12-18
- Grade is 'B'
- And a total grade of 'B5' level.
- When the purpose is 'debt_consolidation'
- When monthly installments are between 145-274
- When the number of derogatory public records is 0
- When the no of enquiries in last 6 months is 0
- When the loan status is Not verified
- Term of 36 months
- In the last months of the years
- Loans taken in 2011 (maybe due to some financial issues at that time)

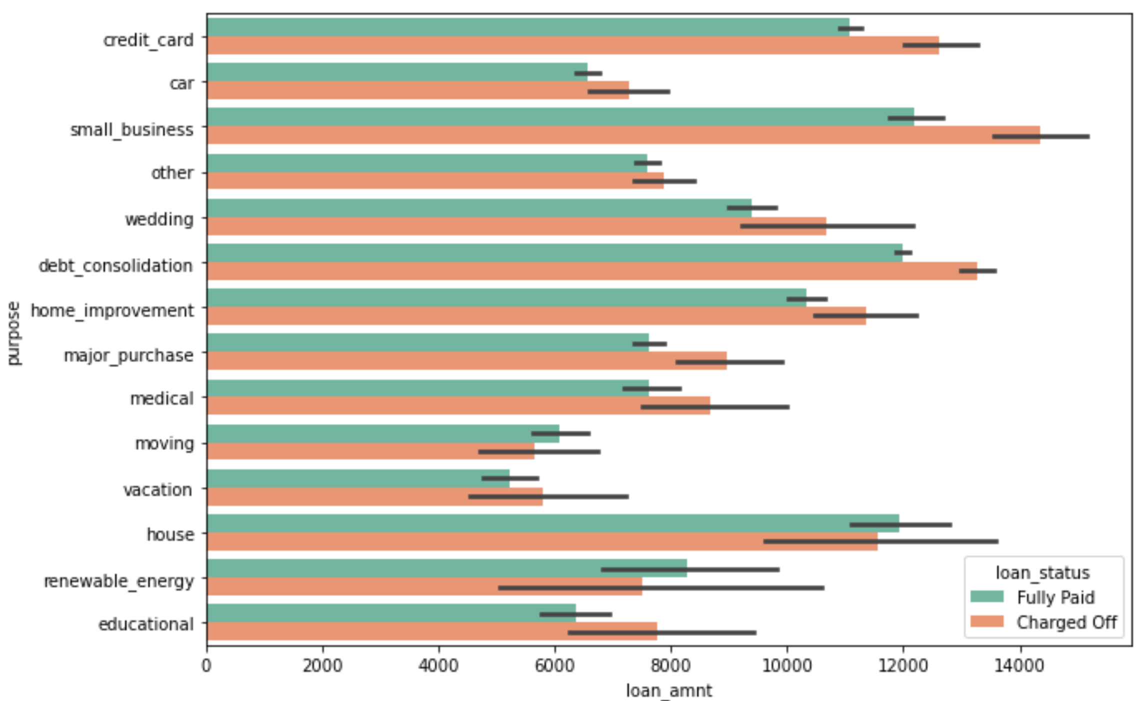
Bivariate

Loan Amount with other columns

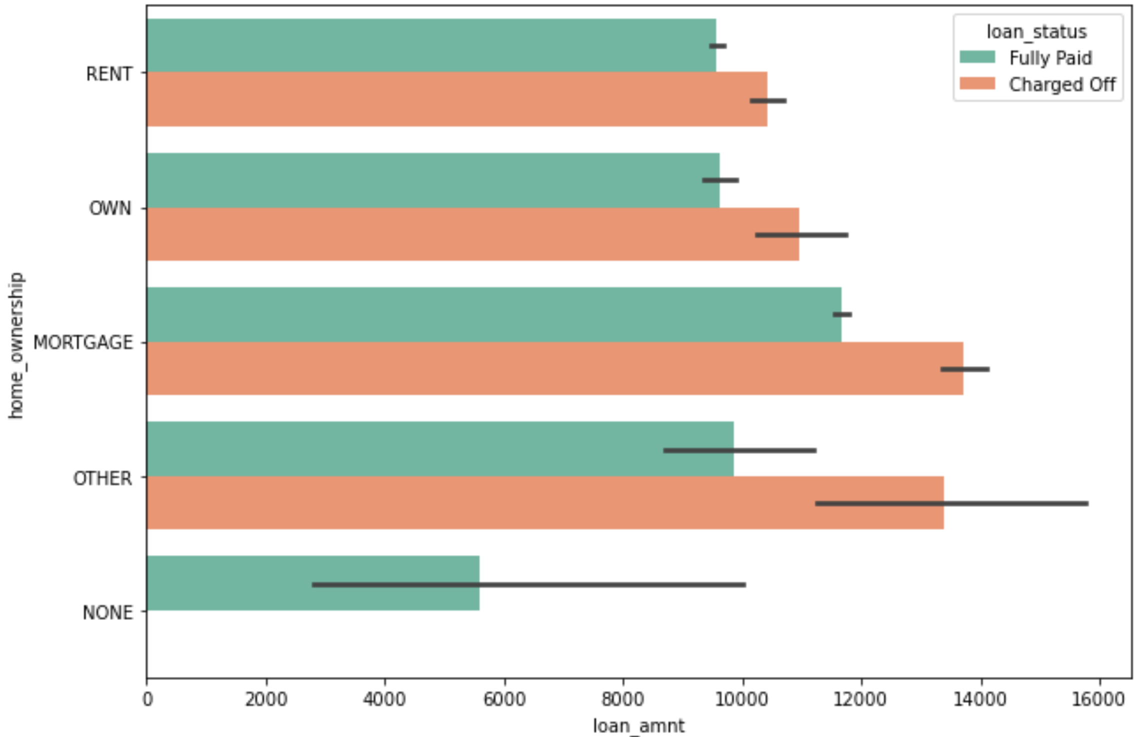
In [355]:
```
plt.figure(figsize=(10,7))
sns.barplot(data =df,x='loan_amnt_groups', y='int_rate', hue ='loan_status',palette="Set2")
plt.show()
```



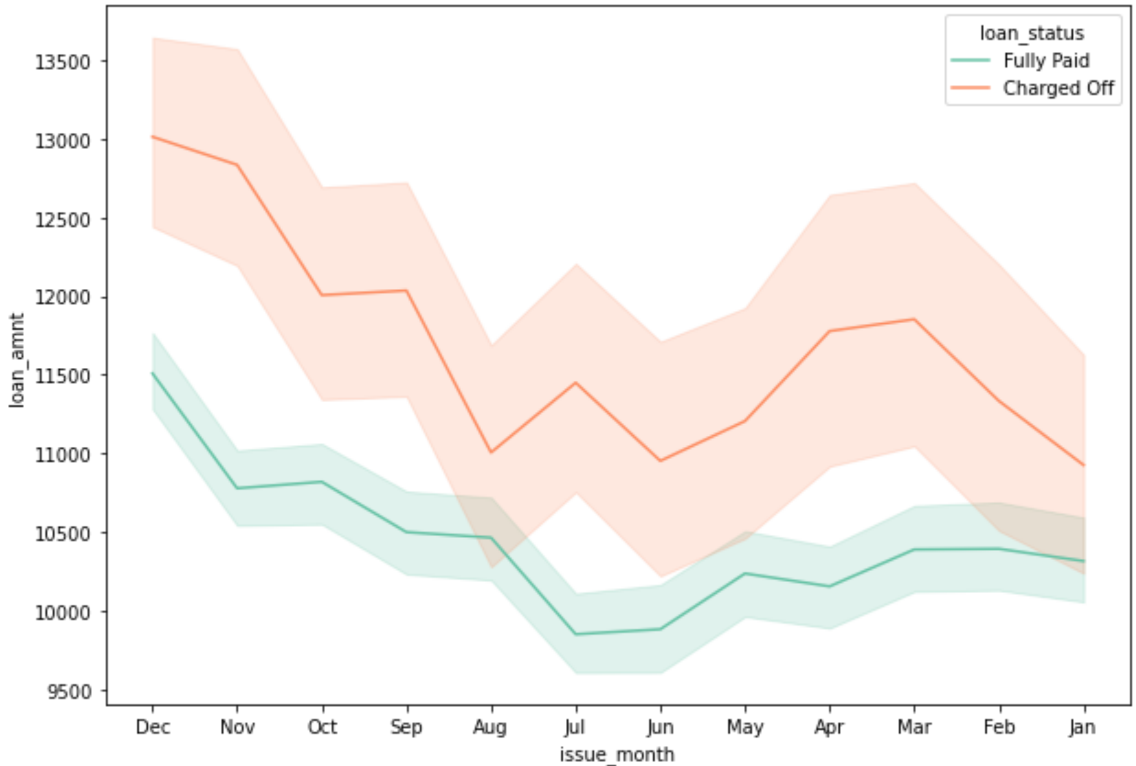
In [356]:
```
plt.figure(figsize=(10,7))
sns.barplot(data =df,x='loan_amnt', y='purpose', hue ='loan_status',palette="Set2")
plt.show()
```



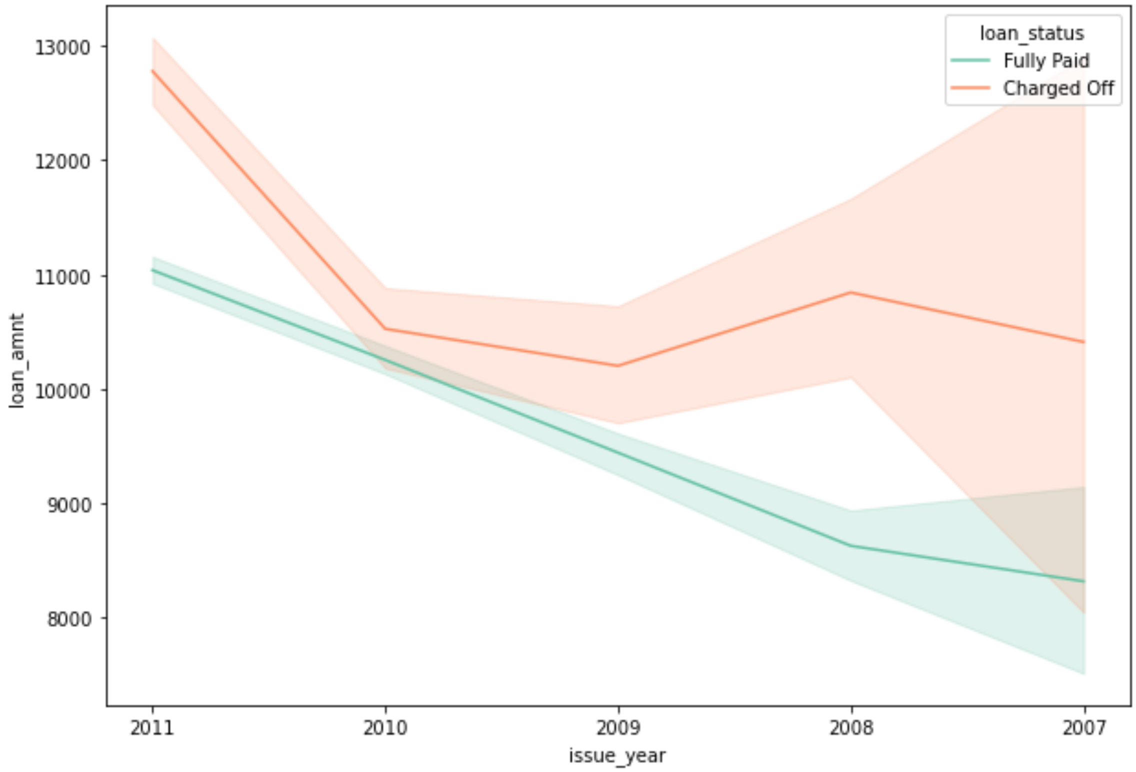
In [357]:
```
plt.figure(figsize=(10,7))
sns.barplot(data =df,x='loan_amnt', y='home_ownership', hue ='loan_
status',palette="Set2")
plt.show()
```



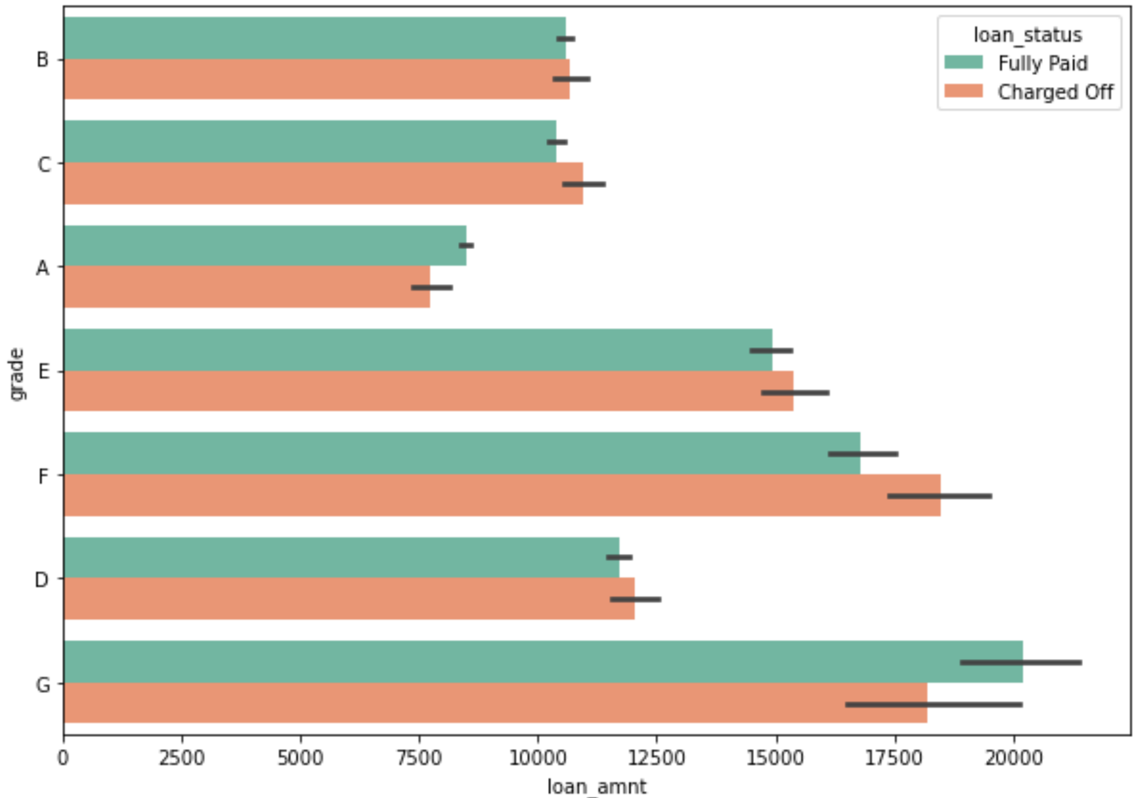
In [358]:
```
plt.figure(figsize=(10,7))
sns.lineplot(data =df,y='loan_amnt', x='issue_month', hue ='loan_status',palette="Set2")
plt.show()
```



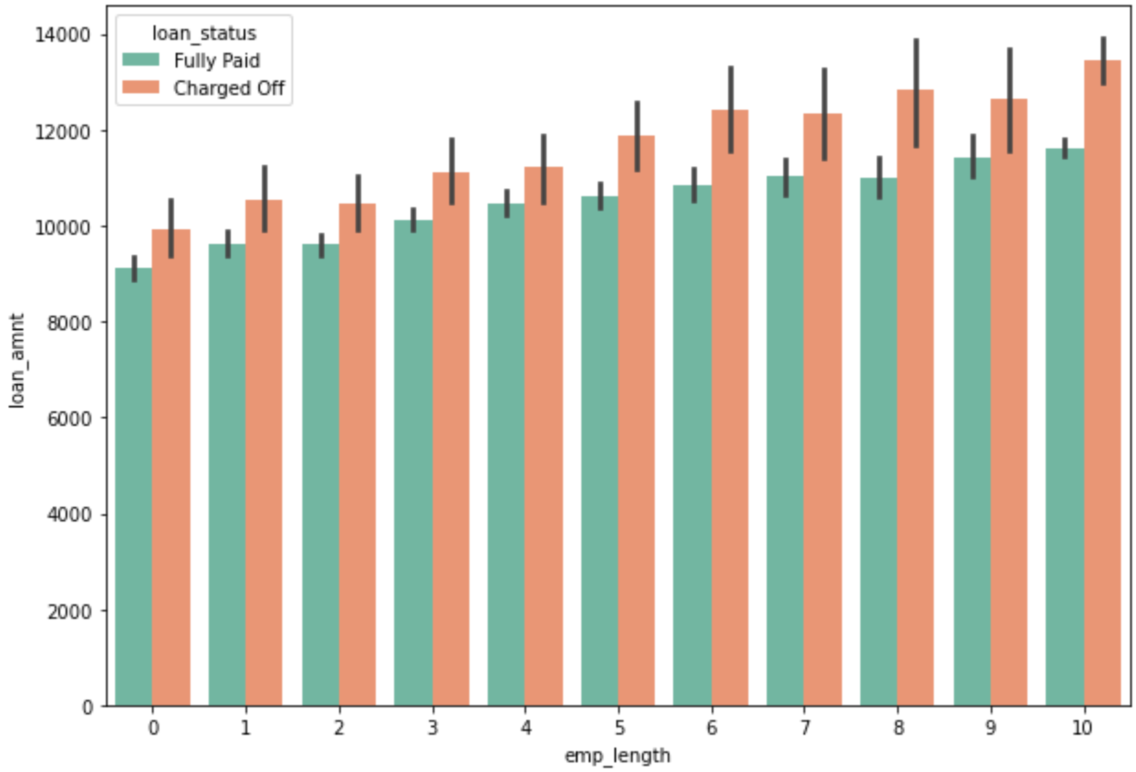
In [359]:
```
plt.figure(figsize=(10,7))
sns.lineplot(data =df,y='loan_amnt', x='issue_year', hue ='loan_status',palette="Set2")
plt.show()
```



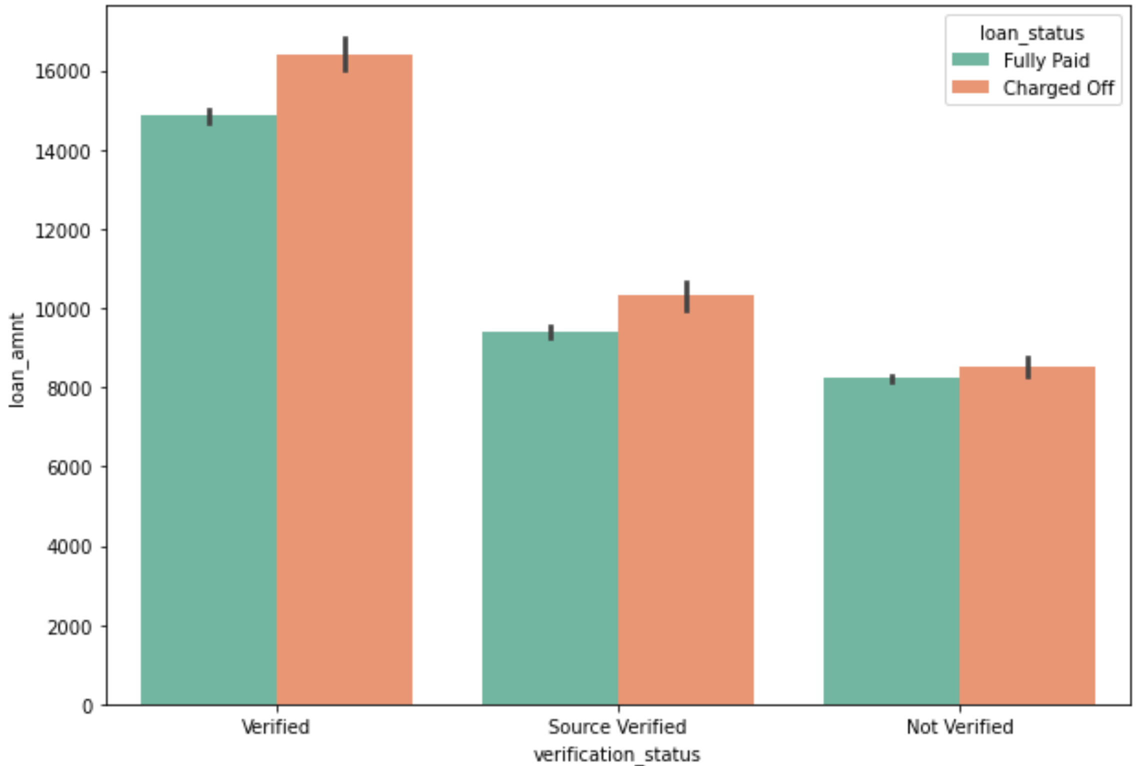
In [360]:
```
plt.figure(figsize=(10,7))
sns.barplot(data =df,x='loan_amnt', y='grade', hue ='loan_status',palette="Set2")
plt.show()
```



In [361]:
```
plt.figure(figsize=(10,7))
sns.barplot(data =df,y='loan_amnt', x='emp_length', hue ='loan_status',palette="Set2")
plt.show()
```

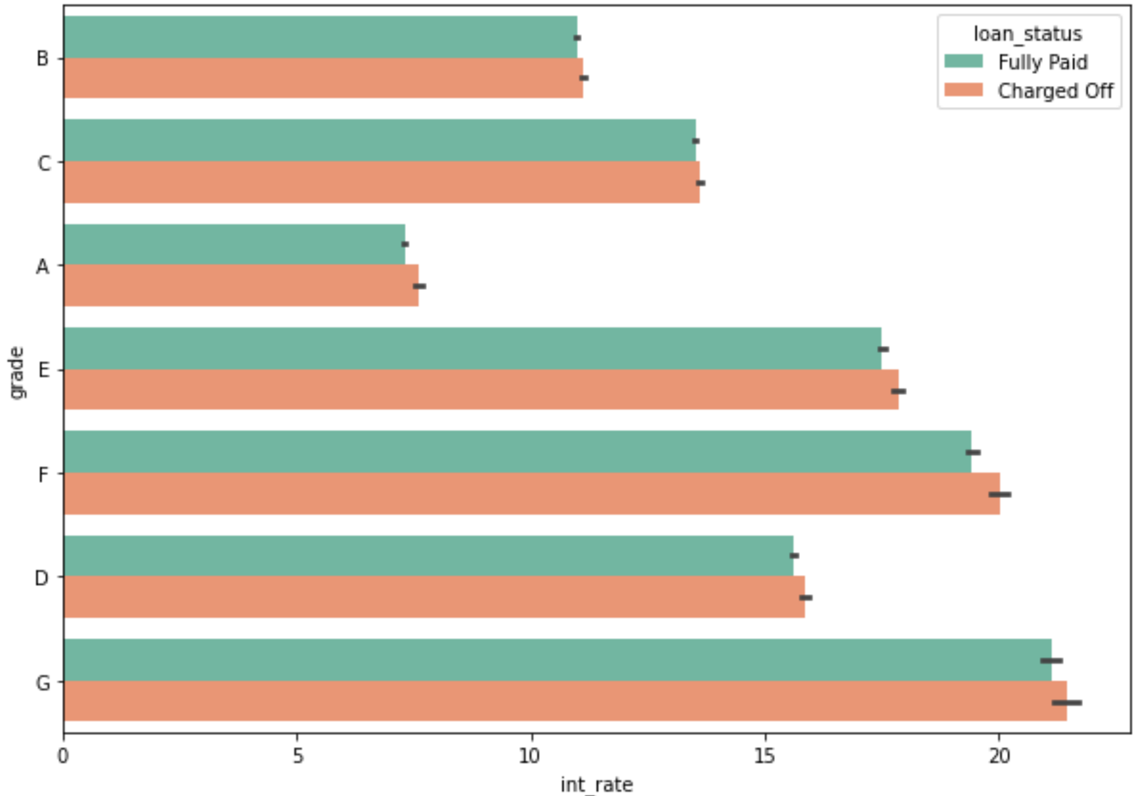


In [362]:
```
plt.figure(figsize=(10,7))
sns.barplot(data =df,y='loan_amnt', x='verification_status', hue ='
loan_status',palette="Set2")
plt.show()
```



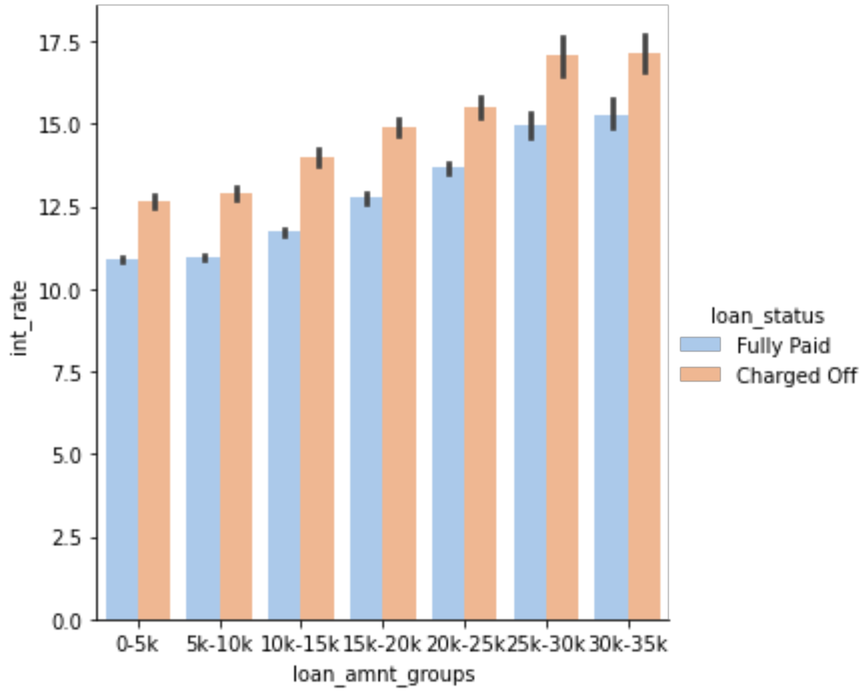
Longer years of work is approving higher amount of loans.. Verified loan applicaions have higher loan amounts. which shows lenders are verifying the applications before sanctioning a significant amount of loan.

In [363]:
```
plt.figure(figsize=(10,7))
sns.barplot(data =df,x='int_rate', y='grade', hue ='loan_status',pa
lette="Set2")
plt.show()
```

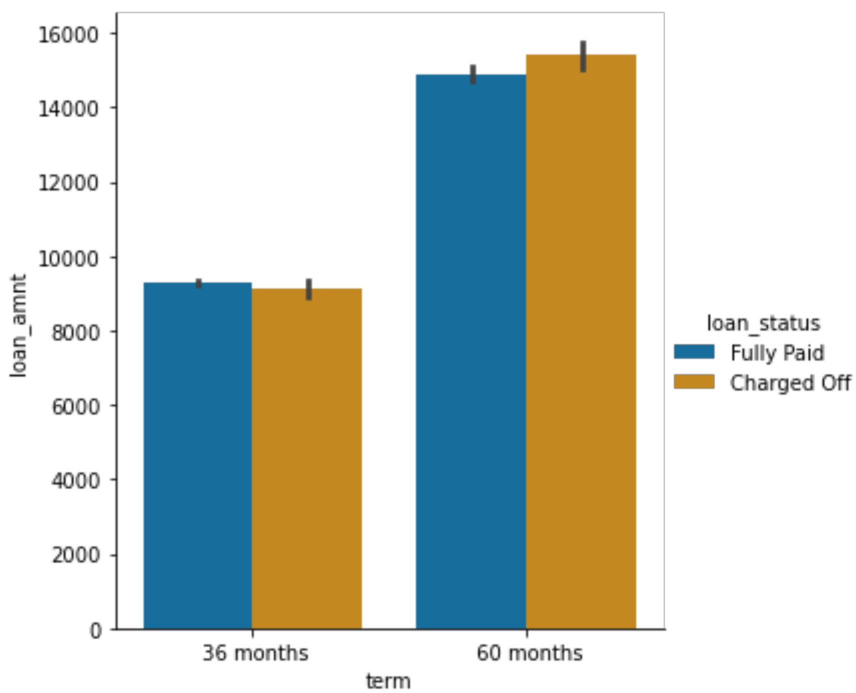


In [364]:
```
plt.tight_layout()
sns.catplot(data =df,y ='int_rate', x ='loan_amnt_groups', hue ='lo
an_status',palette="pastel",kind = 'bar')
plt.show()
```

<Figure size 432x288 with 0 Axes>



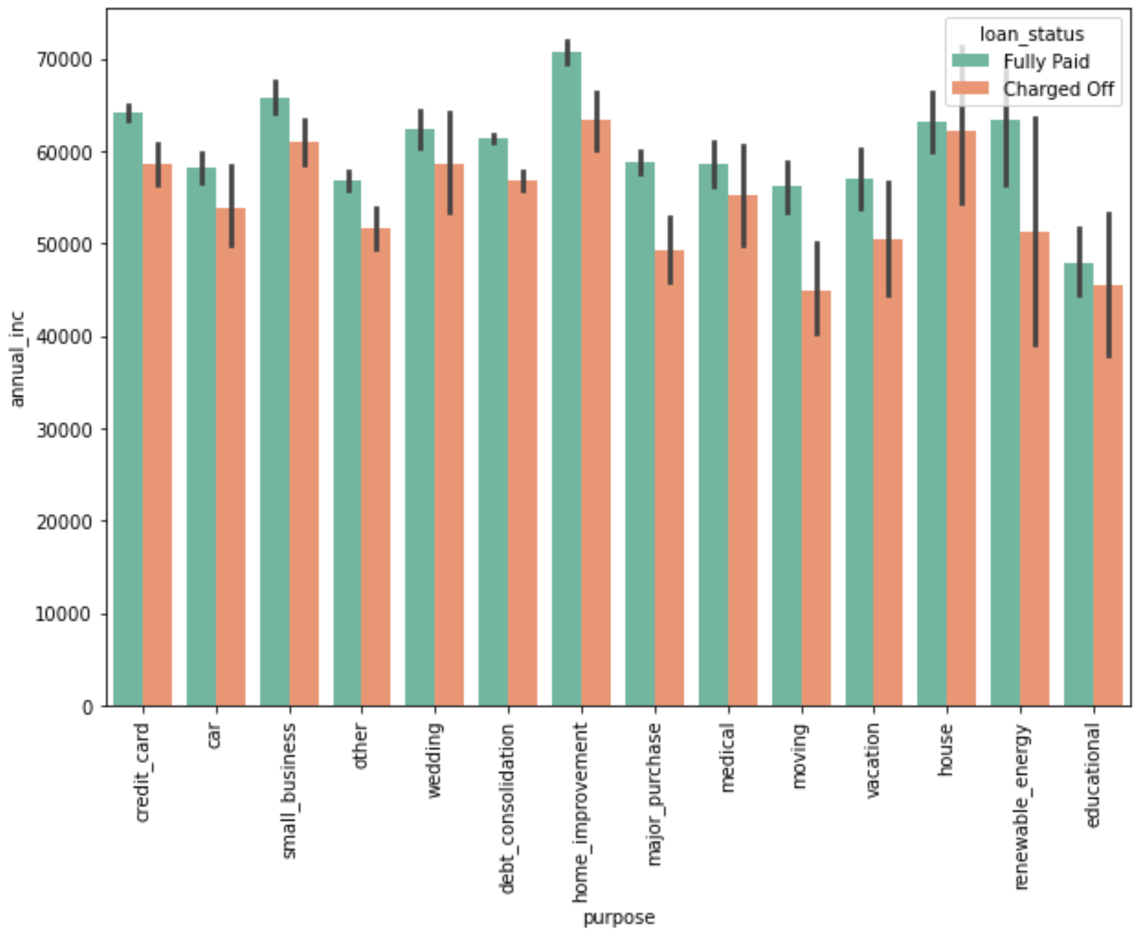
In [365]:
```
sns.catplot(x = 'term', y = 'loan_amnt', data = df,hue = 'loan_stat
us', kind = 'bar')
plt.show()
```



Interest Rate is pretty high for defaulter loans. It is a good point to make.

Annual Income vs Other

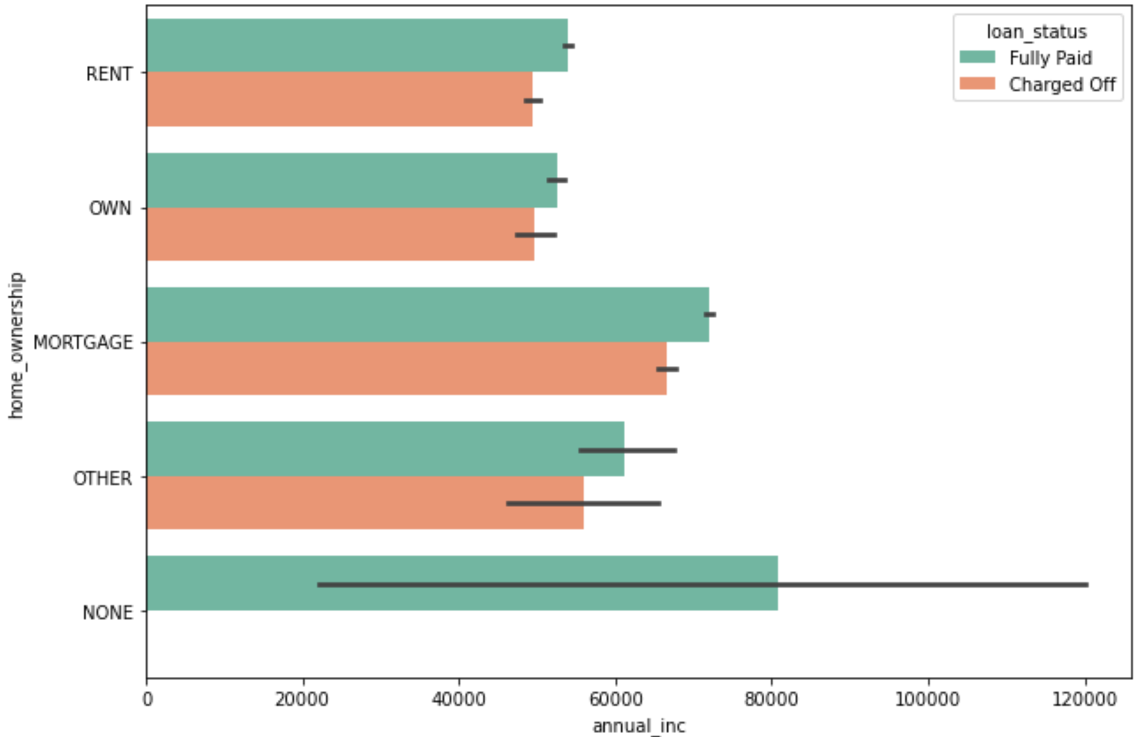
In [366]:
```
# Annual income vs loan purpose
plt.figure(figsize=(10,7))
sns.barplot(data =df,x='purpose', y='annual_inc', hue ='loan_status
',palette="Set2")
plt.xticks(rotation=90)
plt.show()
```



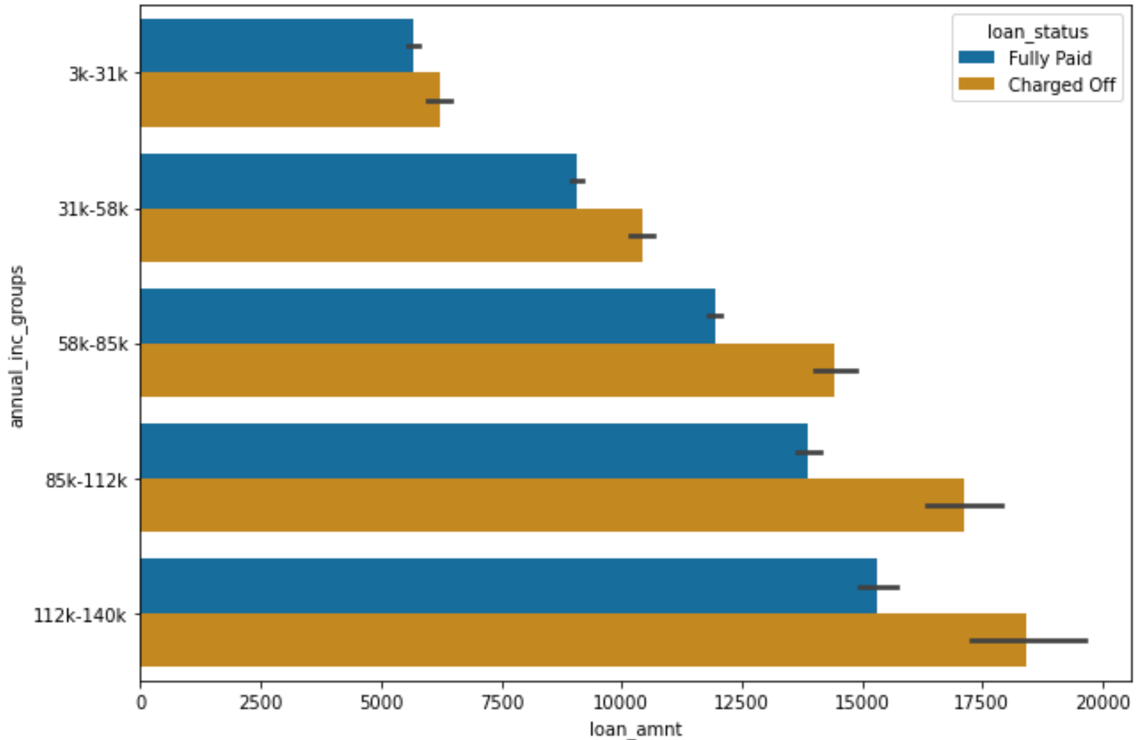
Since highest number of defaulters is for "debt_consolation", the annual income of those who applied isn't the highest.

- Borrowers with more salary mostly applied loans for "home_improvment", "house", "renewable_energy" and "small_businesses"

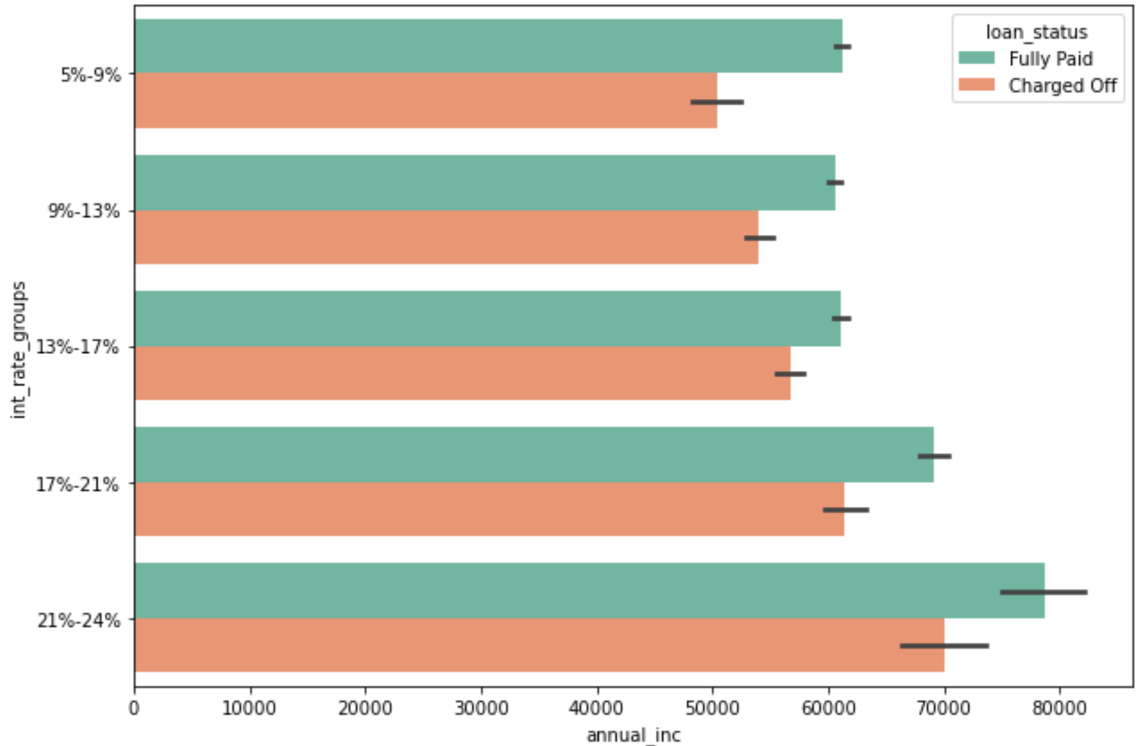
In [367]:
```
# Analysing the anuual income with home ownership
plt.figure(figsize=(10,7))
sns.barplot(data =df,x='annual_inc', y='home_ownership', hue ='loan_status',palette="Set2")
plt.show()
```



In [368]:
```
# Analysing the loan amount and annual income
plt.figure(figsize=(10,7))
sns.barplot(x = "loan_amnt", y = "annual_inc_groups", hue = 'loan_status', data = df)
plt.show()
```



In [369]:
```
# Analysing the annual income anf interest rate
plt.figure(figsize=(10,7))
sns.barplot(data =df,x='annual_inc', y='int_rate_groups', hue ='loan_status',palette="Set2")
plt.show()
```



Observations

The above analysis with respect to the charged off loans. There is a more probability of defaulting when :

- Borrower taking loan for 'home improvement' and have income of 60k -70k
- When grade is F and loan amount is between 15k-20k
- When the loan is verified and loan amount is above 16k
- When employment length is 10yrs and loan amount is 12k-14k
- For grade G and interest rate above 20%
- Borrower who receive interest at the rate of 21-24% and have an income of 70k-80k
- Borrower whose home ownership is 'MORTGAGE and have income of 60-70k
- Borrower whose home ownership is 'MORTGAGE and have loan of 14-16k
- Applicants who have taken a loan in the range 30k - 35k and are charged interest rate of 15-17.5 %
- Borrower who have taken a loan for small business and the loan amount is greater than 14k

We have arrived at our end of EDA. Thank You

Note : Above observations are based on my analysis.