Lending Club Case Study analysis

- Sreedhar K R (kr_Sreedhar@yahoo.com)
- Chandeep Singh Arora (chandeeparora19@gmail.com)

Problem statement

Lending Club is an online market-place where 2 types of people come together for business – The borrower who wants to borrow money for various purposes (credit card payment, marriage etc.) and the investor who wants to lend money to earn interest ultimately.

As a company, Lending Club wants to identify the risky loan applicants who have higher chances of defaulting so that credit loss to the company is reduced as much as possible. There is a need to identify some consumer and loan attributes to meet this purpose.

Assumptions

Below assumptions have been made :

1. Only Fully paid and Charged Off customers will be considered from the data set as Current ones are the people who are already paying the loan and won’t be adding a lot of value to the analysis.

2. All the numeric values will be considered up to 2 decimal places.

Discussion on outliers

Identify the outliers

- Describe can be used to identif outliers
- funded_amnt_inv - has most of the records between 5000 and 14000. Zero and 35000 are two extremes and indicates outliers
- int_rate has most of the values between 8 to 14. 24% rate of interest is too high compared to average and is an outlier
- inq_last_6mths - Most of the records lie between 0 and 1, few with 2 and 3. 8 is an outlier
- annual_inc has out liers at the maximum value 6000000

In [6]:
```
data.describe()
```

Out[6]:

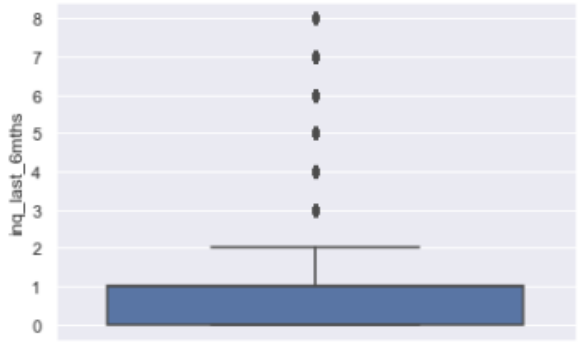
	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	annual_inc	dti	delinq_2yrs	inq_last_6mths
count	38577.000000	38577.000000	38577.000000	38577.000000	38577.000000	38577.000000	3.857700e+04	38577.000000	38577.000000	38577.000000
mean	11047.025430	10784.058506	10222.481134	41.898437	11.932219	322.466318	6.877797e+04	13.272727	0.146668	0.871737
std	7348.441646	7090.306027	7022.720648	10.333136	3.691327	208.639215	6.421868e+04	6.673044	0.492271	1.071546
min	500.000000	500.000000	0.000000	36.000000	5.420000	15.690000	4.000000e+03	0.000000	0.000000	0.000000
25%	5300.000000	5200.000000	5000.000000	36.000000	8.940000	165.740000	4.000000e+04	8.130000	0.000000	0.000000
50%	9600.000000	9550.000000	8733.440000	36.000000	11.710000	277.860000	5.886800e+04	13.370000	0.000000	1.000000
75%	15000.000000	15000.000000	14000.000000	36.000000	14.380000	425.550000	8.200000e+04	18.560000	0.000000	1.000000
max	35000.000000	35000.000000	35000.000000	60.000000	24.400000	1305.190000	6.000000e+06	29.990000	11.000000	8.000000

Discussion on outliers

Identifying and removing outliers

In [7]:
```
# Below plot indicates the precense of outliers for inq_last_6mths variable
sns.boxplot(y=data['inq_last_6mths'])
```

Out[7]: <AxesSubplot:ylabel='inq_last_6mths'>



In [8]:
```
# Below plot indicates the precense of outliers for annual_inc variable
sns.boxplot(y=data['annual_inc'])
```

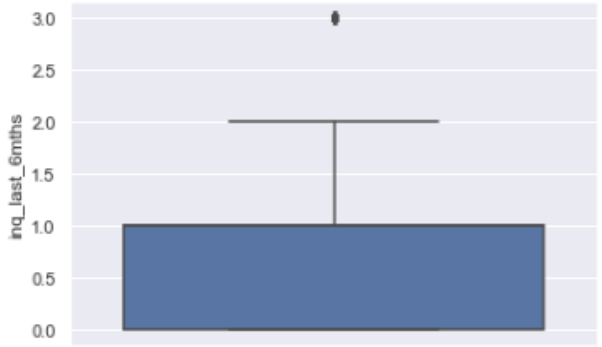
Out[8]: <AxesSubplot:ylabel='annual_inc'>

Discussion on outliers

In [9]:
```
# Removing outliers in annual_inc and inq_last_6mths
data = data[data["annual_inc"] < data["annual_inc"].quantile(0.99)]
data = data[data.inq_last_6mths<4]
```

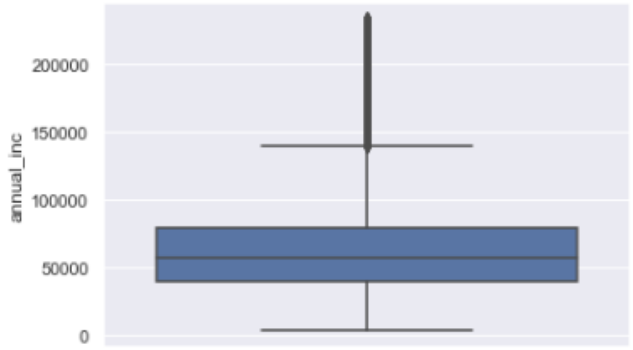
In [10]:
```
# After removing the outliers
sns.boxplot(y=data['inq_last_6mths'])
```

Out[10]: <AxesSubplot:ylabel='inq_last_6mths'>



In [11]:
```
sns.boxplot(y=data['annual_inc'])
```

Out[11]: <AxesSubplot:ylabel='annual_inc'>

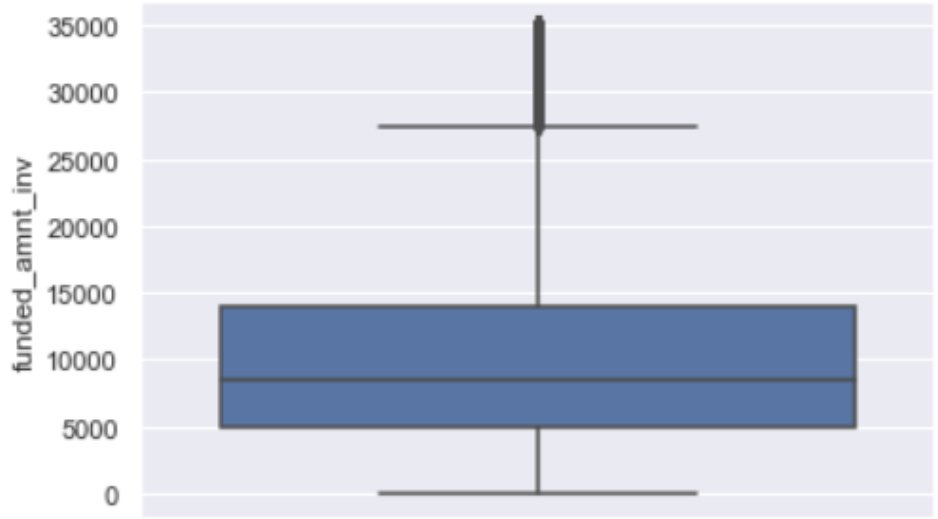


Analysis

Univariate Analysis: funded_amnt_inv

In [56]:
```
# Univariate Analysis on funded_amt_inv
sns.boxplot(y = data['funded_amnt_inv']) #Most of the funded loan amount is falling between 5000- 14000
```

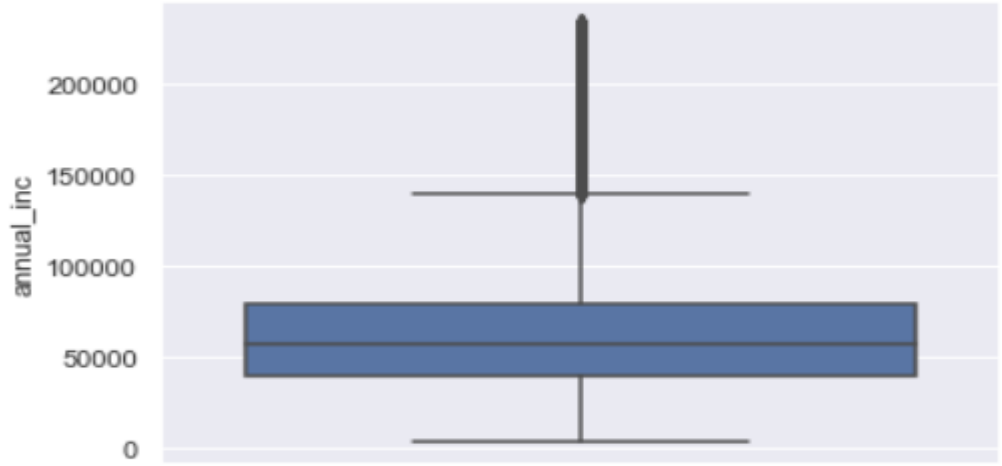
Out[56]: <AxesSubplot:ylabel='funded_amnt_inv'>



Univariate Analysis: annual_inc

In [57]:
```
# Univariate Analysis on Annual Income - Quantitative Variables

sns.boxplot(y=data['annual_inc'])


#Most of the annual income falls between 4000- ~70000
```

Out[57]: <AxesSubplot:ylabel='annual_inc'>



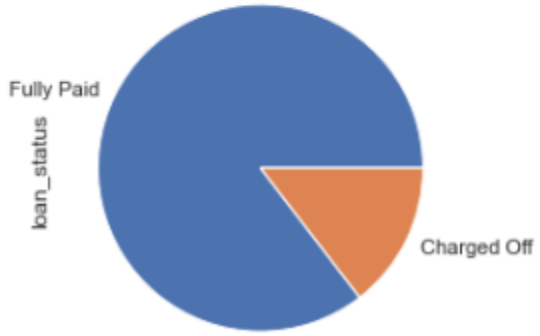
Univariate Analysis: loan_status

Understanding loan status (Fully Paid customers versus Defaulters)

- We see considerable amount of defaulters ratio which would result in loss to the investors

In [17]:
```
# Univariate analysis on loan status

all_loan_status=data['loan_status'].value_counts()
all_loan_status.plot(kind='pie')

# We see considerable amount of defaulters ratio which would result in loss to the investors
```

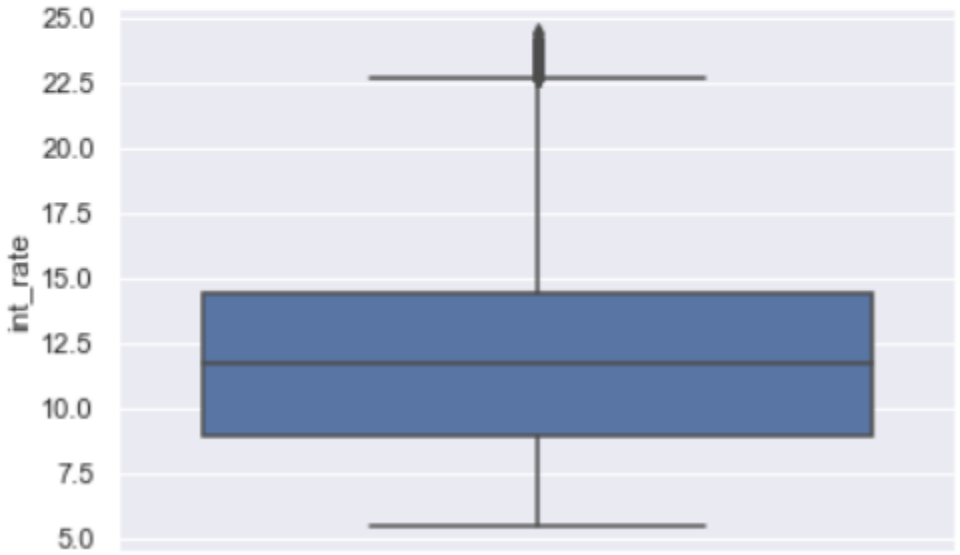
Out[17]: <AxesSubplot:ylabel='loan_status'>



Univariate Analysis: int_rate

In [58]:
```
# Univariate analysis on int_rate
sns.boxplot(y=data['int_rate'])

#maximum interest rate are from ~9% to 15%. Rest are outliers.
```

Out[58]: <AxesSubplot:ylabel='int_rate'>

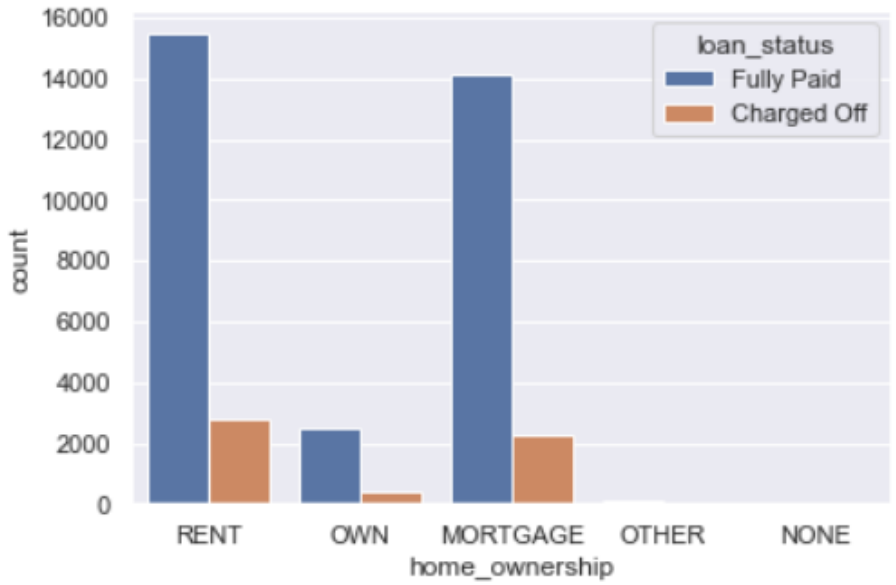


home_ownership vs loan_status

home_ownership variable analysis

In [27]:
```
# Analysis of home_ownership and loan_status
sns.countplot(x="home_ownership",data=data,hue='loan_status')

# We see that people having their own home take less loans and less like to be defaulted. However people already on
# mortgage and rented houses tend to default more. But the loan applicants are higher too.
```

Out[27]:
```
<AxesSubplot:xlabel='home_ownership', ylabel='count'>
```



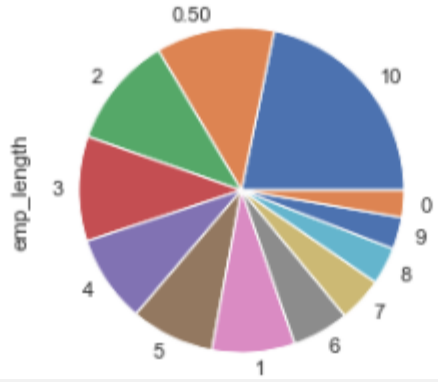
Univariate Analysis: emp_length

emp_length variable analysis

- maximum number of people who are taking loan are 10 years of experience and above

In [16]:
```
# Univariate Analysis on Emp_Length

all_emp_lengths=data['emp_length'].value_counts()
all_emp_lengths.plot(kind='pie')
# maximum number of people who are taking loan are 10 years of experience and above
```

Out[16]: <AxesSubplot:ylabel='emp_length'>



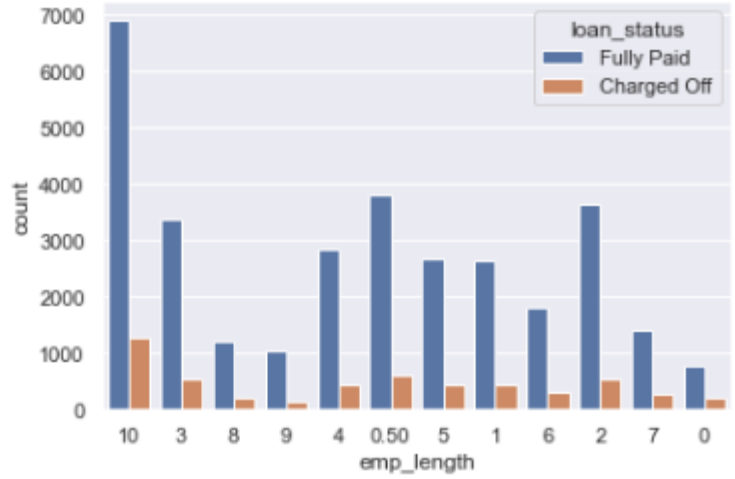
loan_status vs employee length

Employee years of experience (emp_length) with Loan Status (loan_status)

- So we can see that 10 years of experience people tend to pay the loan on time and do not get charged off a lot.
- We also notice that employees with less than two years of experience tend to be more defaulters.

In [18]:
```
# Analysis of emp_length and loan_status
sns.countplot(x="emp_length",data=data,hue='loan_status')

# So we can see that 10 years of experience people tend to pay the loan on time and do not get charged off a lot.
# We also notice that employees with less than two years of experience tend to be more defaulters.
```

Out[18]: <AxesSubplot:xlabel='emp_length', ylabel='count'>

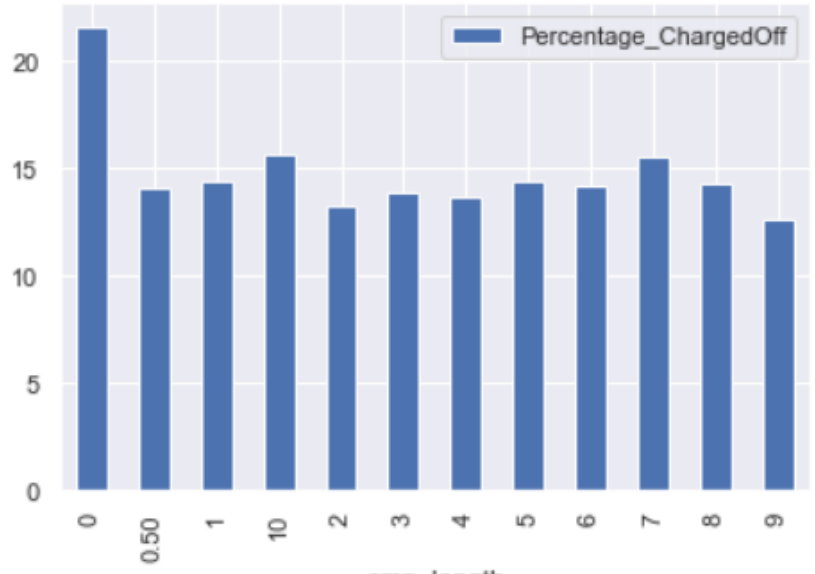


Bivariate Analysis: Charged off percentage vs emp length

Bivariate Analysis : Charged off percentage vs emp length

In [49]:
```
emp_length_grpby = data.groupby(['emp_length', 'loan_status'],as_index=True).loan_status.count().unstack().reset_index()

totalLoanStatus = emp_length_grpby['Charged Off'] + emp_length_grpby['Fully Paid']
emp_length_grpby['Percentage_ChargedOff'] = (emp_length_grpby['Charged Off'] / totalLoanStatus)*100


emp_length_grpby.plot(kind='bar',x='emp_length',y='Percentage_ChargedOff')

# This proves the hypothesis that people with less than 2years of experience tend to default more than others.
```

Out[49]: <AxesSubplot:xlabel='emp_length'>



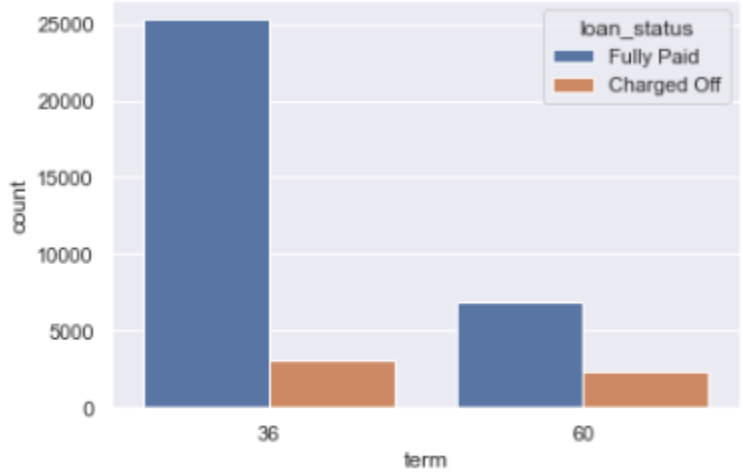
term vs loan_status

Analysis of loan term affecting the number of defaulters ratio

- people taking loan for 60 months have more chances of being charged off.

In [19]:
```
#Univariate analysis: Loan repayment term

sns.countplot(x="term",data=data,hue='loan_status')

# people taking loan for 60 months have more chances of being charged off.
```

Out[19]:
```
<AxesSubplot:xlabel='term', ylabel='count'>
```



Univariate Analysis: Verification status

Verification Status field analysis

In [63]:
```
#univariate analysis: Verification status.
all_verification_status=data['verification_status'].value_counts()
print(all_verification_status)
sns.countplot(x="verification_status",data=data,hue='loan_status')

#looks like verification status does not really matter a lot.
```

```
Not Verified       16364
Verified           11831
Source Verified     9439
Name: verification_status, dtype: int64
```

Out[63]: <AxesSubplot:xlabel='verification_status', ylabel='count'>

Univariate Analysis: Purpose

Analysis on variable 'purpose' on the loan

- debt_consolidation and credit card are the major purposes. We can also see how many are charged off for these 2 purposes.

In [64]:
```
# Univariate Analysis - Purpose of loan

all_loan_purpose=data['purpose'].value_counts()
all_loan_purpose.plot(kind='pie')

# debt_consolidation and credit card are the major purposes. We can also see how many are charged off for these 2 purposes.
```

Out[64]: <AxesSubplot:ylabel='purpose'>

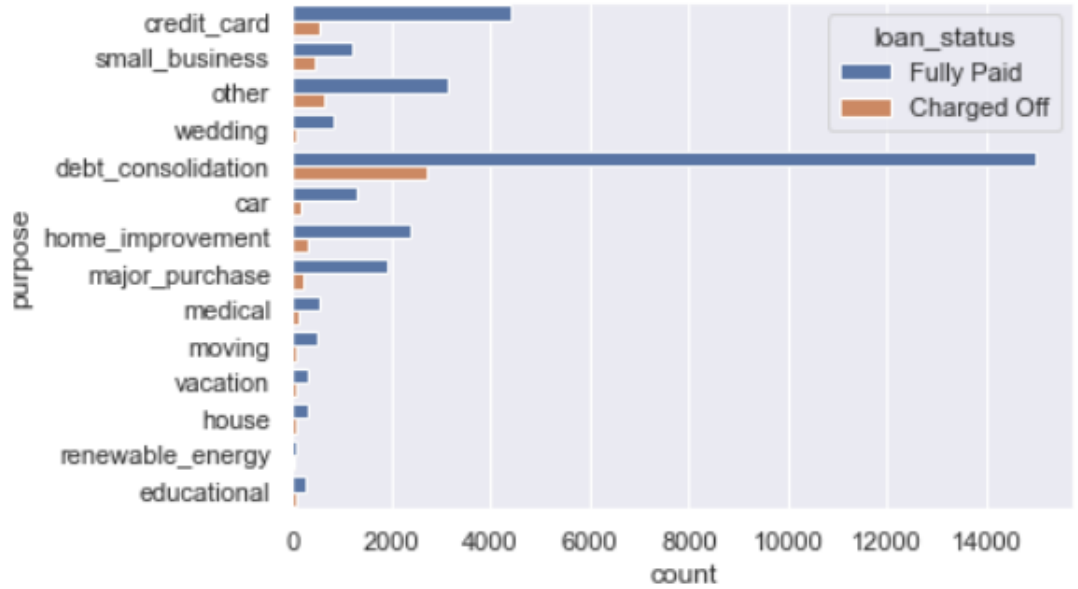
Purpose vs loan status

Purpose of the loan compared with loan status

- debt_consolidation,credit card and other is the purpose where the charged off numbers are extremely high.

In [65]:
```
# Checking how many are charged majorly for credit card and debt_consolidation
sns.countplot(y="purpose",data=data,hue='loan_status')
# debt_consolidation,credit card and other is the purpose where the charged off numbers are extremely high.
```

Out[65]:
```
<AxesSubplot:xlabel='count', ylabel='purpose'>
```



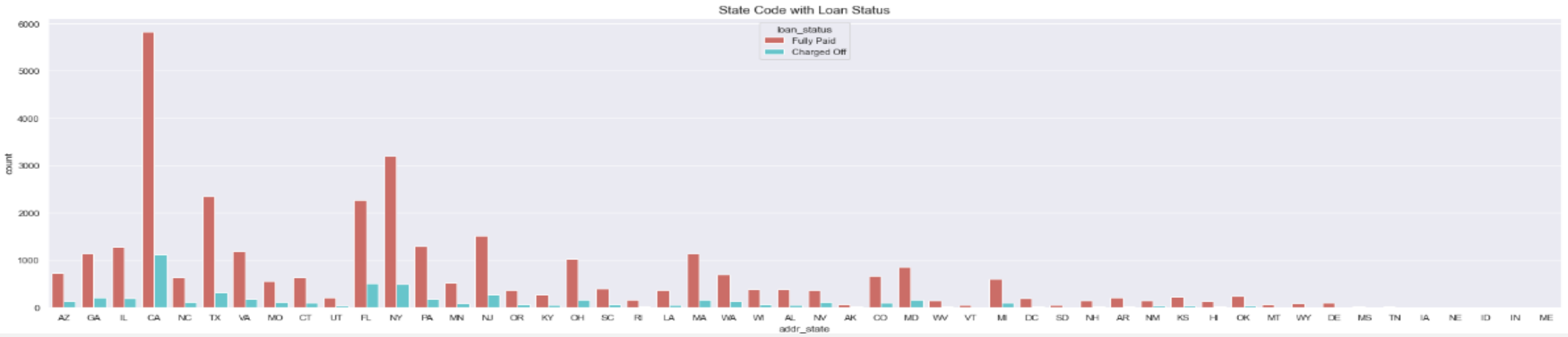
loan_status vs addr_state

Analysis of good customers and defaulters with respect to state code (addr_state)

- The graph indicates fully paid customers with defaulters for each state code.
- We see FL NJ, MD etc have higher ratio of defaulters w.r.t combined borrowers when compared to other states

In [110]:

```
df_good = data[data["loan_status"] == 'Fully Paid']
df_bad = data[data["loan_status"] == 'Charged Off']

fig, ax = plt.subplots( figsize=(30,8))
plt.subplots_adjust(hspace = 0.4, top = 0.8)

g2 = sns.countplot(x="addr_state",data=data,
                   palette="hls", ax=ax,
                   hue = "loan_status")
g2.set_title("State Code with Loan Status", fontsize=15)
g2.set_xlabel("addr_state")
plt.show()

# The graph indicates fully paid customers with defaulters for each state code.
# We see FL NJ, MD etc have higher ratio of defaulters w.r.t combined borrowers when compared to other states
```



loan_status vs inq_last_6mnths

Analysing borrowers with number of inquiries in last 6 months variable (inq_last_6mths)

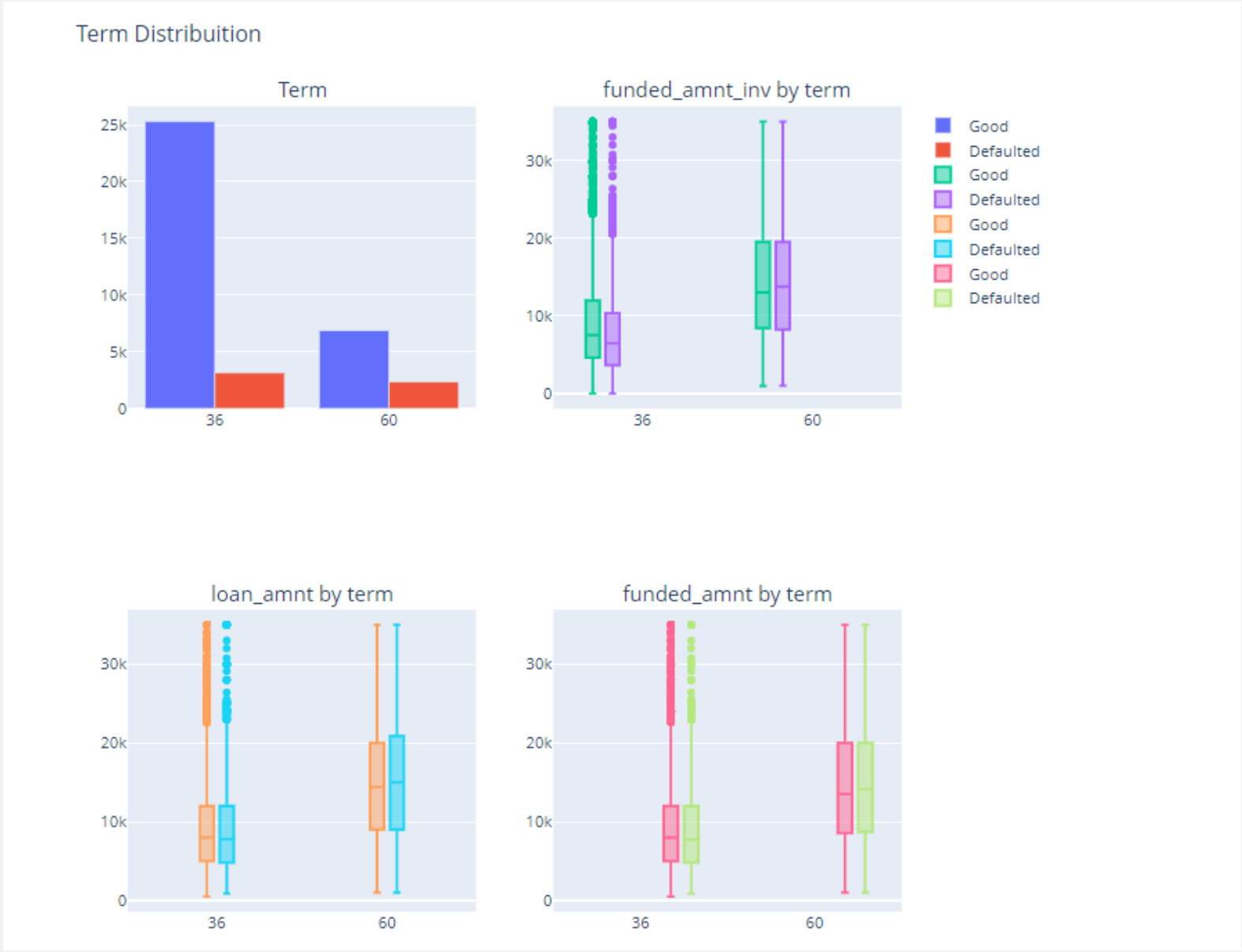
- We see that customers with 2 or 3 inquiries made in last 6 months tend to have more defaulters

In [72]:

```
df_good = data[data["loan_status"] == 'Fully Paid']
df_bad = data[data["loan_status"] == 'Charged Off']

fig, ax = plt.subplots(nrows=2, figsize=(12,8))
plt.subplots_adjust(hspace = 0.4, top = 0.8)

g1 = sns.distplot(df_good["inq_last_6mths"], ax=ax[0],
             color="g")
g1 = sns.distplot(df_bad["inq_last_6mths"], ax=ax[0],
             color='r')
g1.set_title("Inq Last 6 Months Distribuition", fontsize=15)
g1.set_xlabel("Inq Last 6 Months")
g1.set_xlabel("Frequency")

g2 = sns.countplot(x="inq_last_6mths",data=data,
              palette="hls", ax=ax[1],
              hue = "loan_status")
g2.set_title("Inq Last 6 Months with Loan Status", fontsize=15)
g2.set_xlabel("inq_last_6mths")
plt.show()


# We see that customers with 2 or 3 inquiries made in last 6 months tend to have more defaulters
```

loan_status vs inq_last_6mnths

Loan amounts vs Term



loan_amnt vs funded_amnt_inv vs funded_amnt

Analyzing loan amount, approved amount and investors funded amount for each term and comparing with loan status

- In all three types of amount, we see more chances of Charged off for 60 term loans

In [25]:

```
#First plot
tr0 = go.Bar(
    x = data[data["loan_status"]== 'Fully Paid']["term"].value_counts().index.values,
    y = data[data["loan_status"]== 'Fully Paid']["term"].value_counts().values,
    name='Good'
)

#First plot 2
tr1 = go.Bar(
    x = data[data["loan_status"]== 'Charged Off']["term"].value_counts().index.values,
    y = data[data["loan_status"]== 'Charged Off']["term"].value_counts().values,
    name="Defaulted"
)

#Second plot
tr2 = go.Box(
    x = data[data["loan_status"]== 'Fully Paid']["term"],
    y = data[data["loan_status"]== 'Fully Paid']["funded_amnt_inv"],
    name=tr0.name
)
```

loan_amnt vs funded_amnt_inv vs funded_amnt

```
#Second plot 2
tr3 = go.Box(
    x = data[data["loan_status"]== 'Charged Off']["term"],
    y = data[data["loan_status"]== 'Charged Off']["funded_amnt_inv"],
    name=tr1.name
)


#Third plot
tr4 = go.Box(
    x = data[data["loan_status"]== 'Fully Paid']["term"],
    y = data[data["loan_status"]== 'Fully Paid']["loan_amnt"],
    name=tr0.name
)

#Third plot 2
tr5 = go.Box(
    x = data[data["loan_status"]== 'Charged Off']["term"],
    y = data[data["loan_status"]== 'Charged Off']["loan_amnt"],
    name=tr1.name
)

#Fourth plot
tr6 = go.Box(
    x = data[data["loan_status"]== 'Fully Paid']["term"],
    y = data[data["loan_status"]== 'Fully Paid']["funded_amnt"],
    name=tr0.name
)
```

```
#Fourth plot 2
tr7 = go.Box(
    x = data[data["loan_status"]== 'Charged Off']["term"],
    y = data[data["loan_status"]== 'Charged Off']["funded_amnt"],
    name=tr1.name
)

fig = make_subplots(rows=2, cols=2,
                          subplot_titles=('Term', 'funded_amnt_inv by term','loan_amnt by term','funded_amnt by term'))

fig.append_trace(tr0, 1, 1)
fig.append_trace(tr1, 1, 1)
fig.append_trace(tr2, 1, 2)
fig.append_trace(tr3, 1, 2)
fig.append_trace(tr4, 2, 1)
fig.append_trace(tr5, 2, 1)
fig.append_trace(tr6, 2, 2)
fig.append_trace(tr7, 2, 2)

fig['layout'].update(height=800, width=800, title='Term Distribuition', boxmode='group')
py.iplot(fig, filename='term-subplot')

# In all three types of amount, we see more chances of Charged off for 60 term loans
```

Recommendations – Based on analysis

- People with 10 or more years of experience are less likely to be defaulted because of higher annual income and can be trusted to give a loan.
- On the same lines the company should avoid giving loans to people having less then 2 years of experience as they tend to default more. Since raising interest might not help, the company can avoid giving loans to these bracket of people
- People taking loan for 60 or more months have higher chances of being charged off. In this case, the company can raise the interest rate for giving loans for 60 or more months.
- People having their own homes and taking loans should be highly trusted. However, we do understand from data that such proportion of people are less. The company should be cautious about giving loans to people on mortgaged homes and interest rate should be higher for these people.
- Debt consolidation and credit card payment are the 2 areas where people taking loans default the most. The company should be aware before giving loans for these 2 purposes. Charging higher interest rate might be helpful in this area.
- People from FL, NJ, MD have higher rate of defaulters as compared to the combined defaulters of other states. The company needs to be cautious before giving loans to people from these states. Charging higher interest rates from people of these states can be a helpful measure as they constitute a good number of people who take loans.
- People making higher inquiries to take loans tend to default more against people making less enquiries. This can happen because a person making more enquiries for the loan is still trying to figure out a way to pay back and can try to negotiate for a lower interest rate over calls. Company needs to be cautious about these bracket of people.