Case Study – Lending Club

Business Understanding :

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

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.

Submitted by –
Abhishek Paul

Background:

To find out to whom bank can issue loans, depends on certain factors like

- What is the income range of the individual?
- What is the period of income (work experience) for the individual?
- What is the rate of interest over which loans are being currently issued?
- What is the duration or term of the loan taken?
- Details about the past repayments.

Actions Taken:

To find the result we go though a series of steps

- Sourcing the data from CSV file
- Cleaning the data like, removing certain special characters from those columns like *‘int_rate’ mentioned on the data dictionary*
- *Dropping rows and columns which are empty like ‘il_util,all_util,ing_fi’ are some of the columns of such nature as mentioned in the data dictionary.*
- *Dropping those columns which only have a single value across all its rows. Columns like ‘initial_list_status, policy_code, application_type’ are some of the columns of such nature as mentioned in the data dictionary*
- *Standardizing columns like ‘loan_amnt, funded_amnt, installment, annual_inc’ are some among the others of such nature as mentioned in the data dictionary. Casting these quantitative variable to numerical types which would be used in Univariate and Bivariate Analysis.*
- *Deriving columns like ‘loan_amount_category, annual_income_category and interest_rate_category’ which would be further utilized during Bivariate Analysis.*

Bivariate Analysis

Required Column Derivation

In [659]:

```
# Derived columns
# Creating Loan amount categorise into buckets.
loan['loan_amount_category'] = pd.cut(loan['loan_amnt'], [0, 7000, 14000, 21000, 28000, 35000],
                                      labels=['0-7000', '7000-14000', '14000-21000', '21000-28000', '28000 +'])

# Creating annual incomes categories into buckets.
loan['annual_income_category'] = pd.cut(loan['annual_inc'], [0, 20000, 40000, 60000, 80000,1000000],
                                        labels=['0-20000', '20000-40000', '40000-60000', '60000-80000', '80000 +'])

# Creating intrest rates categories into buckets.
loan['interest_rate_category'] = pd.cut(loan['int_rate'], [0, 10, 12.5, 16, 20],
                                        labels=['0-10', '10-13', '12.5-16', '16 +'])

```

Data Analysis - Lending Club - Univariate Analysis

Univariate Analysis –
(Unordered Categorical Variable)
Distribution of Loan Status

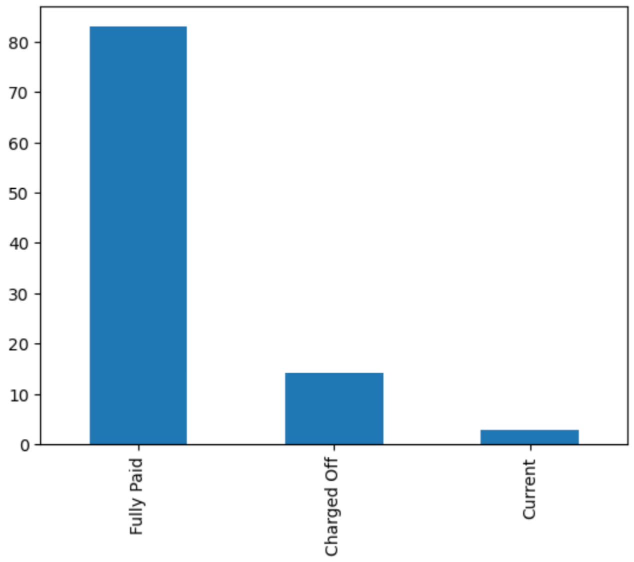
Univariate Analysis for unordered categorical variable

Distribution of Loan Status

In [660]:
```
loan_status = (loan_master.loan_status.value_counts()*100)/len(loan_master)
loan_status.plot.bar()

#Observation
# It is seen that most number of loans are in the fully paid status.
# Number of current loans are the least amost other categories like Fully Paid and Charged.
```

Out[660]: <Axes: >



Observation -

It is seen that most number of loans are in the fully paid status.

Number of current loans are the least amost other categories like Fully Paid and Charged.

Data Analysis - Lending Club - Univariate Analysis

Univariate Analysis –
(Unordered Categorical Variable)
Distribution of Loan Purpose

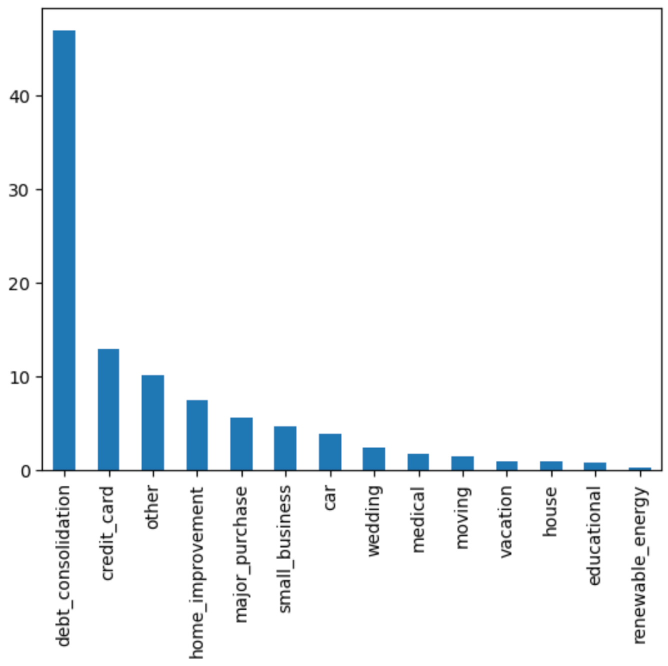
Univariate Analysis for unordered categorical variable

Distribution of Loan Purpose

In [661]:
```
purpose = (loan_master.purpose.value_counts()*100)/len(loan_master)
purpose.plot.bar()

#Observation:
#It is seen that most number of loans are taken for debt_consolidation and least for renewable_energy
```

Out[661]: <Axes: >



Observation -
It is seen that most number of loans are are taken for debt_consolidation and least for renewable_energy

Univariate Analysis – (Quantitative Variables) Distribution of Loan Amount

Univariate Analysis for quantitative variables

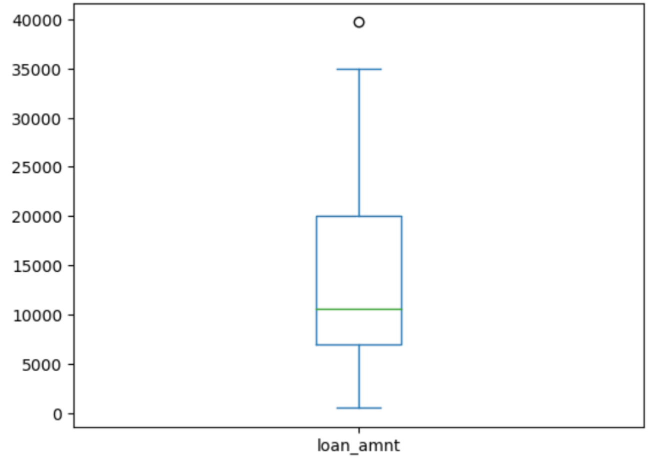
Distribution of Loan Amount

In [596]:
```
loan_amt = loan_master['loan_amnt'].describe()
print(loan_amt)
loan_amt.plot.box()

# It is seen that Minimum Loan amount is 500 whereas Maximum Loan amount is 35000.
# 50% of Loans are amounting in upper range 10000 - 35000
# Another 50% of Loans are amounting in lower range 500 - 10000
```

```
count    39717.000000
mean     11219.443815
std       7456.670694
min        500.000000
25%       5500.000000
50%      10000.000000
75%      15000.000000
max      35000.000000
Name: loan_amnt, dtype: float64
```

Out[596]: `<Axes: >`



Observation -

It is seen that Minimum Loan amount is 500 whereas Maximum Loan amount is 35000.

50% of Loans are amounting in upper range 10000 - 35000

Another 50% of Loans are amounting in lower range 500 - 10000

Univariate Analysis – (Quantitative Variables)
Distribution of Loan Interest Rate

Univariate Analysis for quantitative variables

Distribution of Loan Interest Rate

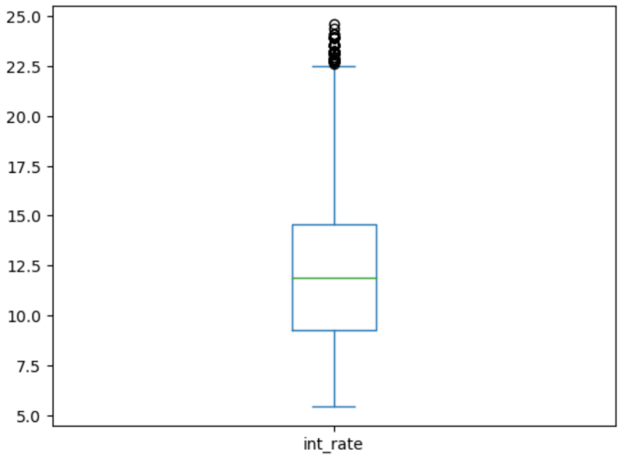
In [597]:
```
print(loan_master['int_rate'].describe())

interest_rate.plot.box()

##It is seen from the box plot that the upper half of the distribution i.e. 0% to 50% is between 5% to 12%
##Whereas for the lower half of the box plot the distribution i.e. 50% to 100% is between 12% to 25%.
## So we can infer that most loans were issued where interest rates were between 12% to 25%.
```

```
count    39717.000000
mean        12.021177
std          3.724825
min          5.420000
25%          9.250000
50%         11.860000
75%         14.590000
max         24.590000
Name: int_rate, dtype: float64
```

Out[597]: <Axes: >



Observation -

It is seen from the box plot that the upper half of the distribution i.e. 0% to 50% is between 5% to 12%
Whereas for the lower half of the box plot the distribution i.e. 50% to 100% is between 12% to 25%.
So we can infer that most loans were issued where interest rates were between 12% to 25%.

Univariate Analysis – (Quantitative Variables) Distribution of Loan Repayment

Univariate Analysis for quantitative variables

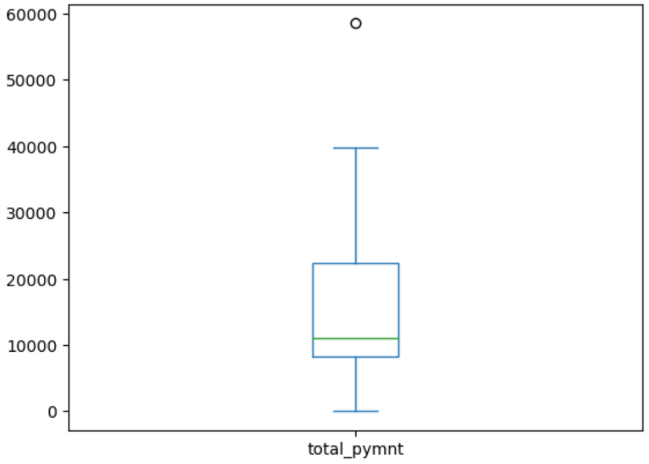
Distribution of Loan Repayement

In [598]:
```
loan_repay = loan_master['total_pymnt'].describe()
print(loan_repay)
loan_repay.plot.box()

##It is seen that Minimum Loan Repayement amount is 0 whereas Maximum Loan Repayement amount is 58563.
##50% of Loans Repayements are amounting in upper range 9899 - 35000
##Another 50% of Loans Repayements are amounting in lower range 0 - 9899
```

```
count    39717.000000
mean     12153.596544
std       9042.040766
min          0.000000
25%       5576.930000
50%       9899.640319
75%      16534.433040
max      58563.679930
Name: total_pymnt, dtype: float64
```

Out[598]: <Axes: >



Observation -

It is seen that Minimum Loan Repayement amount is 0 whereas Maximum Loan Repayement amount is 58563.

50% of Loans Repayements are amounting in upper range 9899 - 35000

Another 50% of Loans Repayements are amounting in lower range 0 - 9899

Univariate Analysis – (Quantitative Variables) Distribution of Annual Income

Univariate Analysis for quantitative variables

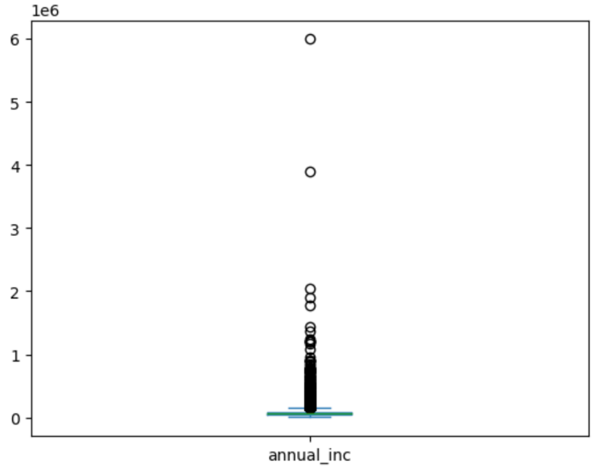
Distribution of Annual_Income

In [599]:
```python
print(loan_master['annual_inc'].describe())

loan_master.annual_inc.plot.box()
```

```
count    3.971700e+04
mean     6.896893e+04
std      6.379377e+04
min      4.000000e+03
25%      4.040400e+04
50%      5.900000e+04
75%      8.230000e+04
max      6.000000e+06
Name: annual_inc, dtype: float64
```

Out[599]: <Axes: >

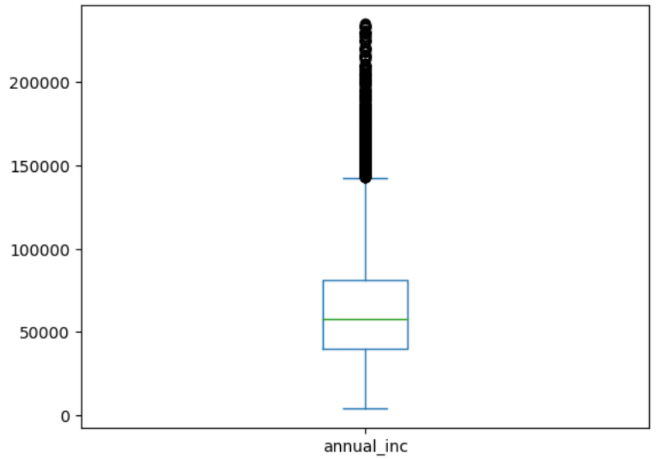


After Outlier treatment

In [600]:
```python
loan = loan_master[loan_master["annual_inc"] < loan_master["annual_inc"].quantile(0.99)]
print(loan['annual_inc'].describe())
loan['annual_inc'].plot.box()
# Observation:
# So the tangible Annual Income ranges between 40K to 81K.
# The average Income is 58k
```

```
count     39319.000000
mean      65524.215803
std       35215.885297
min        4000.000000
25%       40000.000000
50%       58000.000000
75%       81000.000000
max      234996.000000
Name: annual_inc, dtype: float64
```

Out[600]: <Axes: >



Observation –

So the tangible Annual Income ranges between 40000 to 81000.
The average Income is 58000.

Data Analysis - Lending Club - Univariate Analysis

Univariate Analysis –
(Ordered Categorical Variables)
Distribution of Payement Term

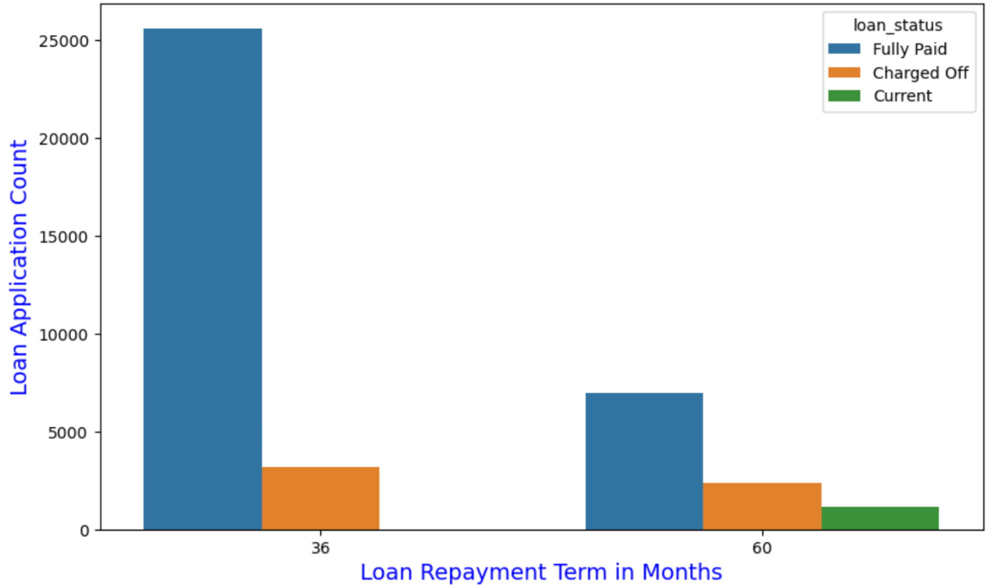
Univariate Analysis for ordered categorical variable

Distribution of Payment Term

In [601]:

```
plt.figure(figsize=(10,6))
ax = sns.countplot(x="term",data=loan,hue='loan_status')
ax.set_xlabel('Loan Repayment Term in Months',fontsize=14,color = 'b')
ax.set_ylabel('Loan Application Count',fontsize=14,color = 'b')
plt.show()

# Observation :
# It is seen that there are more number of loans with teure of 36 Months than those loans
# which are having a tenure of 60 months.
# Plot is also showing that that there are more number of current loans with tenure of 60 Months.
# Below plot shows that those who had taken loan to repay in 60 months had more % of number of applicants getting
# charged off as compared to applicants who had taken loan for 36 months.
```



Observation -

It is seen that there are more number of loans with teure of 36 Months than those loans
which are having a tenure of 60 months.
Plot is also showing that that there are more number of current loans with tenure of 60 Months.
Below plot shows that those who had taken loan to repay in 60 months had more % of number of applicants getting charged off as compared to applicants who had taken loan for 36 months.

upGrad

Data Analysis - Lending Club - Bivariate Analysis

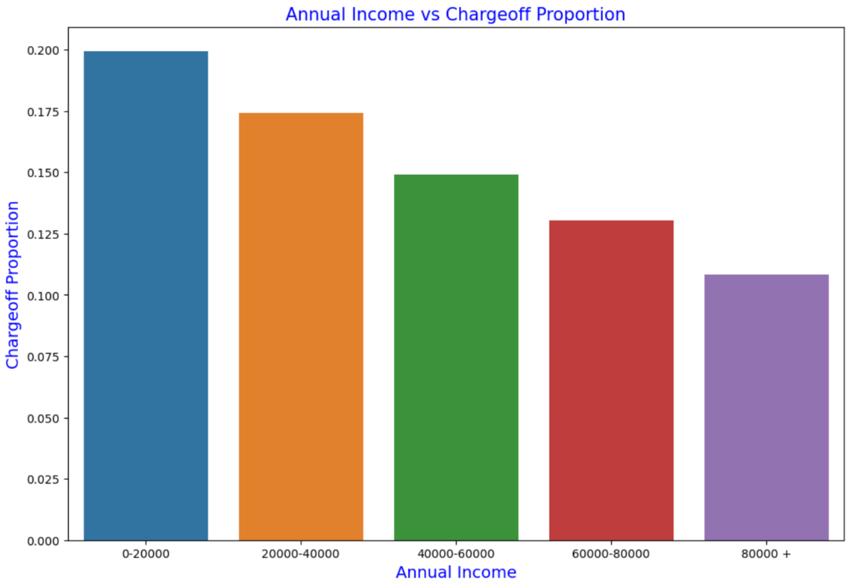
Bivariate Analysis –
Annual Income vs Chargeoff Proportion

Bivariate Analysis for Annual Income vs Chargeoff Proportion

In [662]:
```python
out = loan.groupby(['annual_income_category', 'loan_status']).loan_status.count().unstack().fillna(0).reset_index()
out['total'] = out['Charged Off'] + out['Current'] + out['Fully Paid']
out['chargeoff_proportion'] = out['Charged Off'] / out['total']
out.sort_values('chargeoff_proportion', ascending=False)
print(out)
fig, ax1 = plt.subplots(figsize=(12, 8),facecolor='w')
ax1.set_title('Annual Income vs Chargeoff Proportion',fontsize=15,color = 'b')
ax1=sns.barplot(x='annual_income_category', y='chargeoff_proportion', data=out)
ax1.set_ylabel('Chargeoff Proportion',fontsize=14,color = 'b')
ax1.set_xlabel('Annual Income',fontsize=14,color='b')
plt.show()

# Observation:
# Chargeoff Proportion is decreasing with annual income inscrease
# Income range 80000+  has less chances of getting charged off.
# Income range 0-20000 has high chances of getting charged off.
```

loan_status	annual_income_category	Charged Off	Current	Fully Paid	total	chargeoff_proportion
0	0-20000	237	9	943	1189	0.199327
1	20000-40000	1514	170	7004	8688	0.174263
2	40000-60000	1729	345	9534	11608	0.148949
3	60000-80000	1024	240	6597	7861	0.130263
4	80000 +	1080	362	8531	9973	0.108292



Observation -

Chargeoff Proportion is decreasing with annual income inscrease
Income range 80000+ has less chances of getting charged off.
Income range 0-20000 has high chances of getting charged off.

upGrad

Data Analysis - Lending Club - Bivariate Analysis

Bivariate Analysis –
Interest Rate vs Chargeoff Proportion

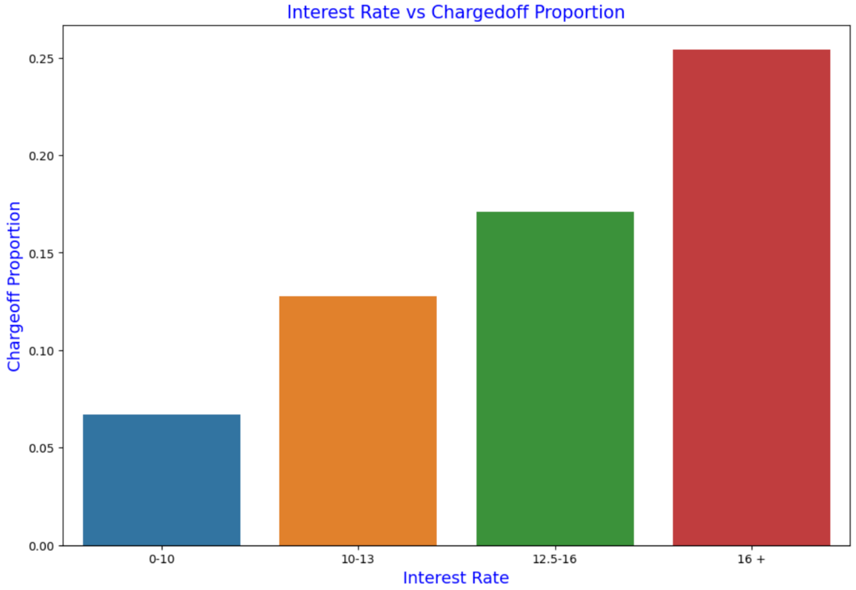
Bivariate Analysis for Interest Rate vs Chargeoff Proportion

In [663]:

```python
out = loan.groupby(['interest_rate_category', 'loan_status']).loan_status.count().unstack().fillna(0).reset_index()
out['total'] = out['Charged Off'] + out['Current'] + out['Fully Paid']
out['chargeoff_proportion'] = out['Charged Off'] / out['total']
out.sort_values('chargeoff_proportion', ascending=False)
print(out)
fig, ax1 = plt.subplots(figsize=(12, 8),facecolor='w')
ax1.set_title('Interest Rate vs Chargedoff Proportion',fontsize=15,color='b')
ax1=sns.barplot(x='interest_rate_category', y='chargeoff_proportion', data=out)
ax1.set_xlabel('Interest Rate',fontsize=14,color='b')
ax1.set_ylabel('Chargeoff Proportion',fontsize=14,color = 'b')
plt.show()

# Observation:
# Charged off proportion is increasing with higher intrest rates.
# Interest of rate more than 16% has good chnaces of charged off as compared to other category intrest rates.
# interest of rate less than 10% has less chances of charged off.
```

loan_status	interest_rate_category	Charged Off	Current	Fully Paid	total	chargeoff_proportion
0	0-10	825	77	11403	12305	0.067046
1	10-13	1224	269	8083	9576	0.127820
2	12.5-16	1995	329	9354	11678	0.170834
3	16 +	1250	351	3317	4918	0.254168



Observation -

Charged off proportion is increasing with higher intrest rates.
Interest of rate more than 16% has good chnaces of charged off as compared to other category intrest rates.
interest of rate less than 10% has less chances of charged off.

upGrad

Data Analysis - Lending Club - Bivariate Analysis

Bivariate Analysis –
Employment Length vs Chargeoff Proportion

Bivariate Analysis for Employment Length vs Chargeoff Proportion

In [665]:

```
out = loan.groupby(['emp_length', 'loan_status']).loan_status.count().unstack().fillna(0).reset_index()
out['total'] = out['Charged Off'] + out['Current'] + out['Fully Paid']
out['chargeoff_proportion'] = out['Charged Off'] / out['total']
out.sort_values('chargeoff_proportion', ascending=False)
print(out)
fig, ax1 = plt.subplots(figsize=(12, 8),facecolor='w')
ax1.set_title('Employment Length vs Chargeoff Proportion',fontsize=15,color='b')
ax1=sns.barplot(x='emp_length', y='chargeoff_proportion', data=out)
ax1.set_xlabel('Employment Length',fontsize=14,color='b')
ax1.set_ylabel('Chargeoff Proportion',fontsize=14,color = 'b')
plt.show()

# Observation:
# The Chart clearly show that applicants have almost same chances of getting charged off except for those
# who are having less than one year of working experiance or donot have experience at all.
```

loan_status	emp_length	Charged Off	Current	Fully Paid	total	chargeoff_proportion
0	0	227	42	801	1070	0.212150
1	1	1090	143	6533	7766	0.140355
2	2	561	97	3684	4342	0.129203
3	3	551	82	3426	4059	0.135748
4	4	456	94	2860	3410	0.133724
5	5	456	87	2712	3255	0.140092
6	6	305	58	1846	2209	0.138072
7	7	262	62	1435	1759	0.148948
8	8	203	43	1216	1462	0.138851
9	9	157	32	1058	1247	0.125902
10	10	1316	386	7038	8740	0.150572

Observation -

The Chart clearly show that applicants have almost same chances of getting charged off except for those
who are having less than one year of working experience or don't have experience at all.

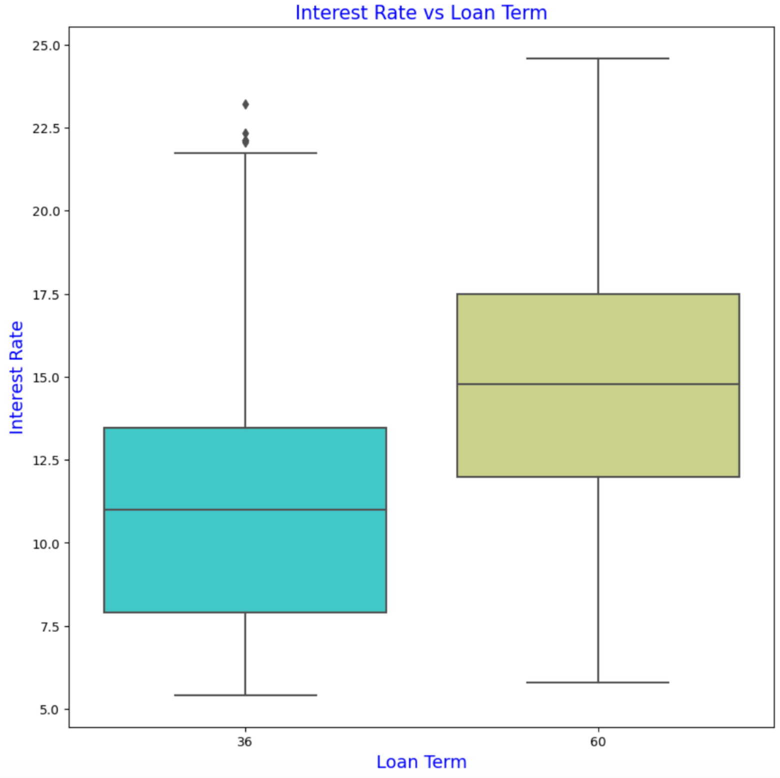
Data Analysis - Lending Club - Bivariate Analysis

Bivariate Analysis –
Interest Rate vs Loan Term

Bivariate Analysis for Interest Rate vs Loan Term

In [652]:
```
plt.figure(figsize=(10,10),facecolor='w')
ax = sns.boxplot(y='int_rate', x='term', data =loan,palette='rainbow')
ax.set_title('Interest Rate vs Loan Term',fontsize=15,color='b')
ax.set_ylabel('Interest Rate',fontsize=14,color = 'b')
ax.set_xlabel('Loan Term',fontsize=14,color = 'b')
plt.show()

# Observations:
# It is seen that 50th percentile of 36 Month Loan Term is lower than 50th percentile of 60 Month Loan Term.
# Loans taken for longer term generally have a higher amount.
```



Observation -

It is seen that 50th percentile of 36 Month Loan Term is lower than 50th percentile of 60 Month Loan Term.
Loans taken for longer term generally have a higher amount.

upGrad

Data Analysis - Lending Club - Bivariate Analysis

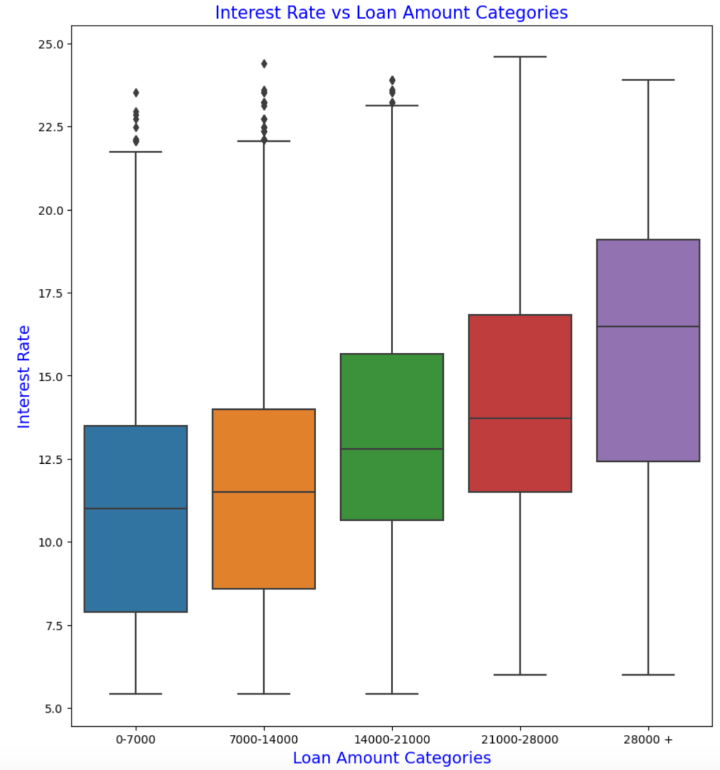
Bivariate Analysis –
Interest Rate vs Loan Amount Categories

Bivariate Analysis for Interest Rate vs Loan Amount Categories

In [649]:

```
plt.figure(figsize=(10,11),facecolor='w')
ax = sns.boxplot(y='int_rate', x='loan_amount_category', data =loan)
ax.set_title('Interest Rate vs Loan Amount Categories',fontsize=15,color='b')
ax.set_ylabel('Interest Rate',fontsize=14,color = 'b')
ax.set_xlabel('Loan Amount Categories',fontsize=14,color = 'b')
plt.show()


# Observations:
# Loan amount has an influence on the inscrease of Interest Rate
# Loans taken with higher amount would also have longer repayment duration
# which again has an effect of having higher interest rate.
```



Observation -

Loan amount has an influence on the increase of Interest Rate
Loans taken with higher amount would also have longer repayment duration which again has an effect of having higher interest rate.

From the analysis we can conclude on the following –

- The consumer finance company can lend amounts to borrowers who have a a higher income range
- Amount can be lend to borrowers who have work experience more than 1 year.
- Loan amounting to lower range can be lend to borrowers as they have higher chance of getting paid off.
- Borrowers with lower loan term like 36 months can be considered more for lending amounts.
- Borrowers who are given loans on lower interest rates have the least chances of getting charged off which is good for the finance company

Thank You