Credit Card Analysis

August 28, 2024

Import Pandas , Matplotlib , seaborn

0.0.1 Import Pandas, Numpy, Seaborn

[3]:
```
import os
import pandas as pd
```

Importing data sets (csv files) namely Credit card & Customers

[4]:
```
cc = pd.read_csv('credit_card.csv')
```

[5]:
```
cust= pd.read_csv('customer.csv')
```

[6]:
```
cust.head()
```

[6]:

	Client_Num	Customer_Age	Gender	Dependent_Count	Education_Level	Marital_Status	state_cd	Zipcode	Car_Owner	House_Owner	Personal_loan	contact	Customer_Job	Income	Cust_Satisfaction_Score
0	708082083	24	F	1	Uneducated	Single	FL	91750	no	yes	no	unknown	Businessman	202326	3
1	708083283	62	F	0	Unknown	Married	NJ	91750	no	no	no	cellular	Selfemployeed	5225	2
2	708084558	32	F	1	Unknown	Married	NJ	91750	yes	no	no	unknown	Selfemployeed	14235	2
3	708085458	38	M	2	Uneducated	Single	NY	91750	no	no	no	cellular	Blue-collar	45683	1
4	708086958	48	M	4	Graduate	Single	TX	91750	yes	yes	no	cellular	Businessman	59279	1

[7]:
```
cc.head()
```

[7]:
```
   Client_Num Card_Category  Annual_Fees  Activation_30_Days  \
0   708082083          Blue          200                   0
1   708083283          Blue          445                   1
2   708084558          Blue          140                   0
3   708085458          Blue          250                   1
4   708086958          Blue          320                   1

   Customer_Acq_Cost Week_Start_Date Week_Num Qtr  current_year  Credit_Limit  \
0                 87      01-01-2023   Week-1  Q1          2023        3544.0
1                108      01-01-2023   Week-1  Q1          2023        3421.0
2                106      01-01-2023   Week-1  Q1          2023        8258.0
3                150      01-01-2023   Week-1  Q1          2023        1438.3
4                106      01-01-2023   Week-1  Q1          2023        3128.0

   Total_Revolving_Bal  Total_Trans_Amt  Total_Trans_Vol  \
0                 1661            15149              111
1                 2517              992               21
2                 1771             1447               23
3                    0             3940               82
4                  749             4369               59

   Avg_Utilization_Ratio Use Chip       Exp Type  Interest_Earned  \
0                  0.469    Chip         Travel          4393.21
1                  0.736   Swipe  Entertainment            69.44
2                  0.214    Chip          Bills           202.58
3                  0.000  Online        Grocery           236.40
4                  0.239   Swipe           Fuel          1004.87

   Delinquent_Acc
0               0
1               0
2               0
3               0
4               1
```

0.1 Data Cleaning

1. to check nan (missing) values if any

[8]:
```
#to check nan values

nan_df = cust[cust.isna().any(axis=1)]

nan_df.head()
```

[8]:
```
Empty DataFrame
Columns: [Client_Num, Customer_Age, Gender, Dependent_Count, Education_Level,
Marital_Status, state_cd, Zipcode, Car_Owner, House_Owner, Personal_loan,
```

```
contact, Customer_Job, Income, Cust_Satisfaction_Score]
Index: []
```

[9]:
```
#to check nan values
nan_df = cc[cc.isna().any(axis=1)]
nan_df.head()
```

[9]:
```
Empty DataFrame
Columns: [Client_Num, Card_Category, Annual_Fees, Activation_30_Days,
Customer_Acq_Cost, Week_Start_Date, Week_Num, Qtr, current_year, Credit_Limit,
Total_Revolving_Bal, Total_Trans_Amt, Total_Trans_Vol, Avg_Utilization_Ratio,
Use Chip, Exp Type, Interest_Earned, Delinquent_Acc]
Index: []
```

[10]:
```
cust.head()
```

[10]:

	Client_Num	Customer_Age	Gender	Dependent_Count	Education_Level	\
0	708082083	24	F	1	Uneducated	
1	708083283	62	F	0	Unknown	
2	708084558	32	F	1	Unknown	
3	708085458	38	M	2	Uneducated	
4	708086958	48	M	4	Graduate	

	Marital_Status	state_cd	Zipcode	Car_Owner	House_Owner	Personal_loan	\
0	Single	FL	91750	no	yes	no	
1	Married	NJ	91750	no	no	no	
2	Married	NJ	91750	yes	no	no	
3	Single	NY	91750	no	no	no	
4	Single	TX	91750	yes	yes	no	

	contact	Customer_Job	Income	Cust_Satisfaction_Score
0	unknown	Businessman	202326	3
1	cellular	Selfemployeed	5225	2
2	unknown	Selfemployeed	14235	2
3	cellular	Blue-collar	45683	1
4	cellular	Businessman	59279	1

2. Convert Columns to correct Data types (string to numeric here)

[11]:
```
#Convert Columns to correct data type

cust['Customer_Age']= pd.to_numeric(cust['Customer_Age'])


cust['Income']= pd.to_numeric(cust['Income'])
```

```
cust['Dependent_Count']= pd.to_numeric(cust['Dependent_Count'])
```

[12]:
```
#Convert Columns  of Cc table to correct data type

cc['Total_Trans_Amt']= pd.to_numeric(cc['Total_Trans_Amt'])

cc['Interest_Earned']= pd.to_numeric(cc['Interest_Earned'])

cc['Annual_Fees']= pd.to_numeric(cc['Annual_Fees'])

cc.head()
```

[12]:

	Client_Num	Card_Category	Annual_Fees	Activation_30_Days	\
0	708082083	Blue	200	0	
1	708083283	Blue	445	1	
2	708084558	Blue	140	0	
3	708085458	Blue	250	1	
4	708086958	Blue	320	1	

	Customer_Acq_Cost	Week_Start_Date	Week_Num	Qtr	current_year	Credit_Limit	\
0	87	01-01-2023	Week-1	Q1	2023	3544.0	
1	108	01-01-2023	Week-1	Q1	2023	3421.0	
2	106	01-01-2023	Week-1	Q1	2023	8258.0	
3	150	01-01-2023	Week-1	Q1	2023	1438.3	
4	106	01-01-2023	Week-1	Q1	2023	3128.0	

	Total_Revolving_Bal	Total_Trans_Amt	Total_Trans_Vol	\
0	1661	15149	111	
1	2517	992	21	
2	1771	1447	23	
3	0	3940	82	
4	749	4369	59	

	Avg_Utilization_Ratio	Use Chip	Exp Type	Interest_Earned	\
0	0.469	Chip	Travel	4393.21	
1	0.736	Swipe	Entertainment	69.44	
2	0.214	Chip	Bills	202.58	
3	0.000	Online	Grocery	236.40	
4	0.239	Swipe	Fuel	1004.87	

	Delinquent_Acc
0	0
1	0
2	0
3	0
4	1

[0]:

0.2 3. Creating new column Age group based on following condition using "apply"

[13]:
```
def val (d):
    if d['Customer_Age'] < 30:
        return "<30"
    elif d['Customer_Age'] >=30 and d['Customer_Age']<40:
        return "30-40"
    elif d['Customer_Age']>=40 and d['Customer_Age']<50:
        return "40-50"
    elif d['Customer_Age'] >=5 and d['Customer_Age']<60:
        return "50-60"
    elif d['Customer_Age'] >=60:
        return ">=60"

cust['Age_Group'] = cust.apply(val, axis =1)



cust.head()
```

[13]:

	Client_Num	Customer_Age	Gender	Dependent_Count	Education_Level	Marital_Status	state_cd	Zipcode	Car_Owner	House_Owner	Personal_loan	contact	Customer_Job	Income	Cust_Satisfaction_Score	Age_Group
0	708082083	24	F	1	Uneducated	Single	FL	91750	no	yes	no	unknown	Businessman	202326	3	<30
1	708083283	62	F	0	Unknown	Married	NJ	91750	no	no	no	cellular	Selfemployeed	5225	2	>=60
2	708084558	32	F	1	Unknown	Married	NJ	91750	yes	no	no	unknown	Selfemployeed	14235	2	30-40
3	708085458	38	M	2	Uneducated	Single	NY	91750	no	no	no	cellular	Blue-collar	45683	1	30-40
4	708086958	48	M	4	Graduate	Single	TX	91750	yes	yes	no	cellular	Businessman	59279	1	40-50

[14]:
```
#creating new column ' Income Group ' based on following condition using "apply"
```

```
def val (d):
    if d['Income'] < 35000:
        return "Low"
    elif d['Income'] >=35000 and d['Income']<70000:
        return "Medium"
    elif d['Income']>=70000:
        return "High"



cust['Income_Group'] = cust.apply(val, axis =1)


cust.head()
```

[14]:

	Client_Num	Customer_Age	Gender	Dependent_Count	Education_Level	Marital_Status	state_cd	Zipcode	Car_Owner	House_Owner	Personal_loan	contact	Customer_Job	Income	Cust_Satisfaction_Score	Age_Group	Income_Group
0	708082083	24	F	1	Uneducated	Single	FL	91750	no	yes	no	unknown	Businessman	202326	3	<30	High
1	708083283	62	F	0	Unknown	Married	NJ	91750	no	no	no	cellular	Selfemployeed	5225	2	>=60	Low
2	708084558	32	F	1	Unknown	Married	NJ	91750	yes	no	no	unknown	Selfemployeed	14235	2	30-40	Low
3	708085458	38	M	2	Uneducated	Single	NY	91750	no	no	no	cellular	Blue-collar	45683	1	30-40	Medium
4	708086958	48	M	4	Graduate	Single	TX	91750	yes	yes	no	cellular	Businessman	59279	1	40-50	Medium

[15]:
```
cc.head()
```

[15]:

	Client_Num	Card_Category	Annual_Fees	Activation_30_Days
0	708082083	Blue	200	0

```
1   708083283          Blue          445                   1
2   708084558          Blue          140                   0
3   708085458          Blue          250                   1
4   708086958          Blue          320                   1

   Customer_Acq_Cost Week_Start_Date Week_Num Qtr  current_year  Credit_Limit  \
0                 87      01-01-2023   Week-1  Q1          2023        3544.0
1                108      01-01-2023   Week-1  Q1          2023        3421.0
2                106      01-01-2023   Week-1  Q1          2023        8258.0
3                150      01-01-2023   Week-1  Q1          2023        1438.3
4                106      01-01-2023   Week-1  Q1          2023        3128.0

   Total_Revolving_Bal  Total_Trans_Amt  Total_Trans_Vol  \
0                 1661            15149              111
1                 2517              992               21
2                 1771             1447               23
3                    0             3940               82
4                  749             4369               59

   Avg_Utilization_Ratio Use Chip       Exp Type  Interest_Earned  \
0                  0.469    Chip         Travel          4393.21
1                  0.736   Swipe  Entertainment            69.44
2                  0.214    Chip          Bills           202.58
3                  0.000  Online        Grocery           236.40
4                  0.239   Swipe           Fuel          1004.87

   Delinquent_Acc
0               0
1               0
2               0
3               0
4               1
```

[0]:

0.3 4.Creating new column "Total Revenue" I. e. Total Revenue generated by bank for its credit card services to the customers.

####Total Revenue = Anuual Fees + Interest earned + Total Transaction Amount.

[16]:
```
cc['Total_Revenue']= cc['Annual_Fees'] + cc['Total_Trans_Amt'] +␣
 ↪cc['Interest_Earned']

cc.head()
```

[16]:
```
   Client_Num Card_Category  Annual_Fees  Activation_30_Days  \
0   708082083          Blue          200                   0
```

```
1   708083283          Blue          445                   1
2   708084558          Blue          140                   0
3   708085458          Blue          250                   1
4   708086958          Blue          320                   1

   Customer_Acq_Cost Week_Start_Date Week_Num Qtr  current_year  Credit_Limit  \
0                 87      01-01-2023   Week-1  Q1          2023        3544.0
1                108      01-01-2023   Week-1  Q1          2023        3421.0
2                106      01-01-2023   Week-1  Q1          2023        8258.0
3                150      01-01-2023   Week-1  Q1          2023        1438.3
4                106      01-01-2023   Week-1  Q1          2023        3128.0

   Total_Revolving_Bal  Total_Trans_Amt  Total_Trans_Vol  \
0                 1661            15149              111
1                 2517              992               21
2                 1771             1447               23
3                    0             3940               82
4                  749             4369               59

   Avg_Utilization_Ratio Use Chip       Exp Type  Interest_Earned  \
0                  0.469    Chip         Travel          4393.21
1                  0.736   Swipe  Entertainment            69.44
2                  0.214    Chip          Bills           202.58
3                  0.000  Online        Grocery           236.40
4                  0.239   Swipe           Fuel          1004.87

   Delinquent_Acc  Total_Revenue
0               0       19742.21
1               0        1506.44
2               0        1789.58
3               0        4426.40
4               1        5693.87
```

[17]:
```
cc['Week_No.'] = cc['Week_Num'].apply(lambda x: x.split('-')[1])

cc.head()
```

[17]:
```
   Client_Num Card_Category  Annual_Fees  Activation_30_Days  \
0   708082083          Blue          200                   0
1   708083283          Blue          445                   1
2   708084558          Blue          140                   0
3   708085458          Blue          250                   1
4   708086958          Blue          320                   1

   Customer_Acq_Cost Week_Start_Date Week_Num Qtr  current_year  Credit_Limit  \
0                 87      01-01-2023   Week-1  Q1          2023        3544.0
1                108      01-01-2023   Week-1  Q1          2023        3421.0
```

```
2                106      01-01-2023   Week-1  Q1          2023        8258.0
3                150      01-01-2023   Week-1  Q1          2023        1438.3
4                106      01-01-2023   Week-1  Q1          2023        3128.0

   Total_Revolving_Bal  Total_Trans_Amt  Total_Trans_Vol  \
0                 1661            15149              111
1                 2517              992               21
2                 1771             1447               23
3                    0             3940               82
4                  749             4369               59

   Avg_Utilization_Ratio Use Chip       Exp Type  Interest_Earned  \
0                  0.469    Chip         Travel          4393.21
1                  0.736   Swipe  Entertainment            69.44
2                  0.214    Chip          Bills           202.58
3                  0.000  Online        Grocery           236.40
4                  0.239   Swipe           Fuel          1004.87

   Delinquent_Acc  Total_Revenue Week_No.
0               0       19742.21        1
1               0        1506.44        1
2               0        1789.58        1
3               0        4426.40        1
4               1        5693.87        1
```

[18]:
```
cc['Week_No.'] = pd.to_numeric(cc['Week_No.'])
```

[19]:
```
import matplotlib.pyplot as plt

import seaborn as sns
%matplotlib inline
```

[20]:
```
cust.head()
```

[20]:
```
    Client_Num  Customer_Age Gender  Dependent_Count Education_Level  \
0   708082083            24      F                1      Uneducated
1   708083283            62      F                0         Unknown
2   708084558            32      F                1         Unknown
3   708085458            38      M                2      Uneducated
4   708086958            48      M                4        Graduate

  Marital_Status state_cd  Zipcode Car_Owner House_Owner Personal_loan  \
0         Single       FL    91750        no         yes            no
1        Married       NJ    91750        no          no            no
2        Married       NJ    91750       yes          no            no
3         Single       NY    91750        no          no            no
4         Single       TX    91750       yes         yes            no
```

```
    contact   Customer_Job  Income  Cust_Satisfaction_Score Age_Group  \
0   unknown    Businessman  202326                        3       <30
1  cellular  Selfemployeed    5225                        2      >=60
2   unknown  Selfemployeed   14235                        2     30-40
3  cellular    Blue-collar   45683                        1     30-40
4  cellular    Businessman   59279                        1     40-50

  Income_Group
0         High
1          Low
2          Low
3       Medium
4       Medium
```

[0]:

0.4 Basic Insights

[21]:
```
cust.head()
```

[21]:
```
   Client_Num  Customer_Age Gender  Dependent_Count Education_Level  \
0   708082083            24      F                1      Uneducated
1   708083283            62      F                0         Unknown
2   708084558            32      F                1         Unknown
3   708085458            38      M                2      Uneducated
4   708086958            48      M                4        Graduate

  Marital_Status state_cd  Zipcode Car_Owner House_Owner Personal_loan  \
0         Single       FL    91750        no         yes            no
1        Married       NJ    91750        no          no            no
2        Married       NJ    91750       yes          no            no
3         Single       NY    91750        no          no            no
4         Single       TX    91750       yes         yes            no

    contact   Customer_Job  Income  Cust_Satisfaction_Score Age_Group  \
0   unknown    Businessman  202326                        3       <30
1  cellular  Selfemployeed    5225                        2      >=60
2   unknown  Selfemployeed   14235                        2     30-40
3  cellular    Blue-collar   45683                        1     30-40
4  cellular    Businessman   59279                        1     40-50

  Income_Group
0         High
1          Low
2          Low
3       Medium
```

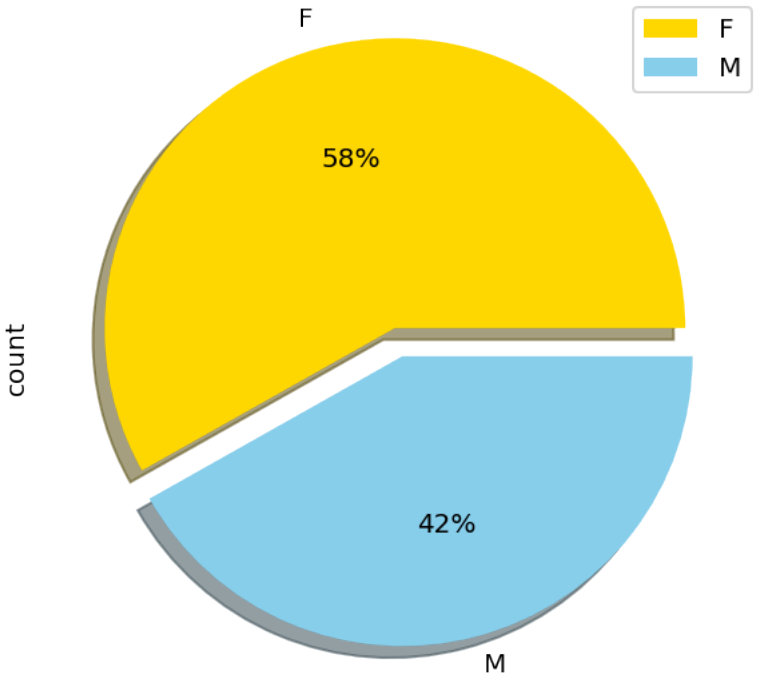
```
4        Medium
```

0.4.1 Gender Distribution

[22]:
```
cust['Gender'].value_counts().plot.pie(figsize=(5,5), autopct='%1.
↪0f%%',legend=True, shadow=True, explode=(0.1,0), colors=['gold', 'skyblue'])
```

[22]:
```
<Axes: ylabel='count'>
```

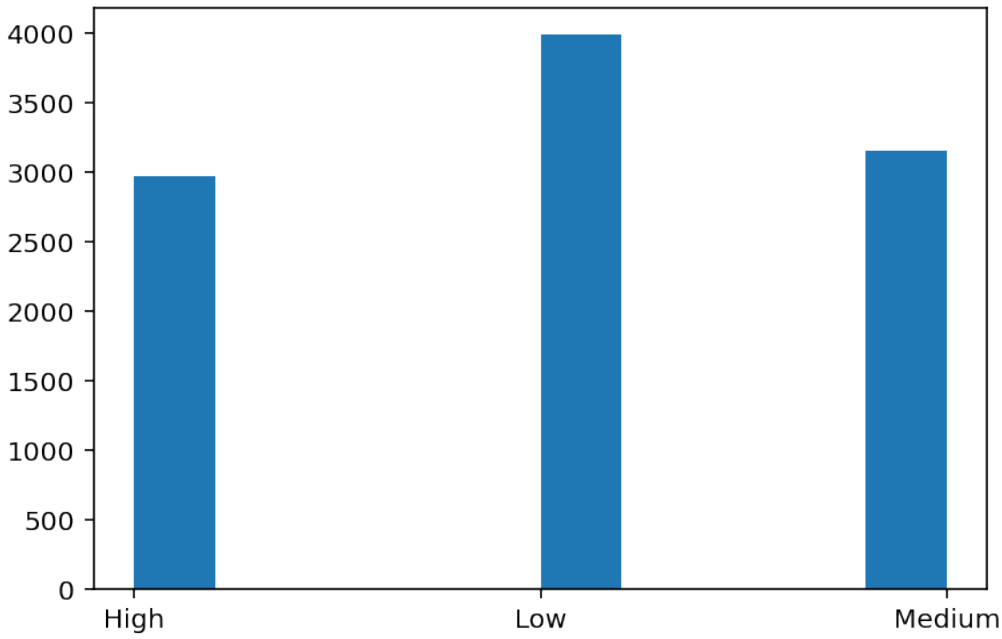
[22]:



0.5 Frequency Distribution of Customers in various Income groups

[23]:
```
plt.hist(cust.Income_Group);
```

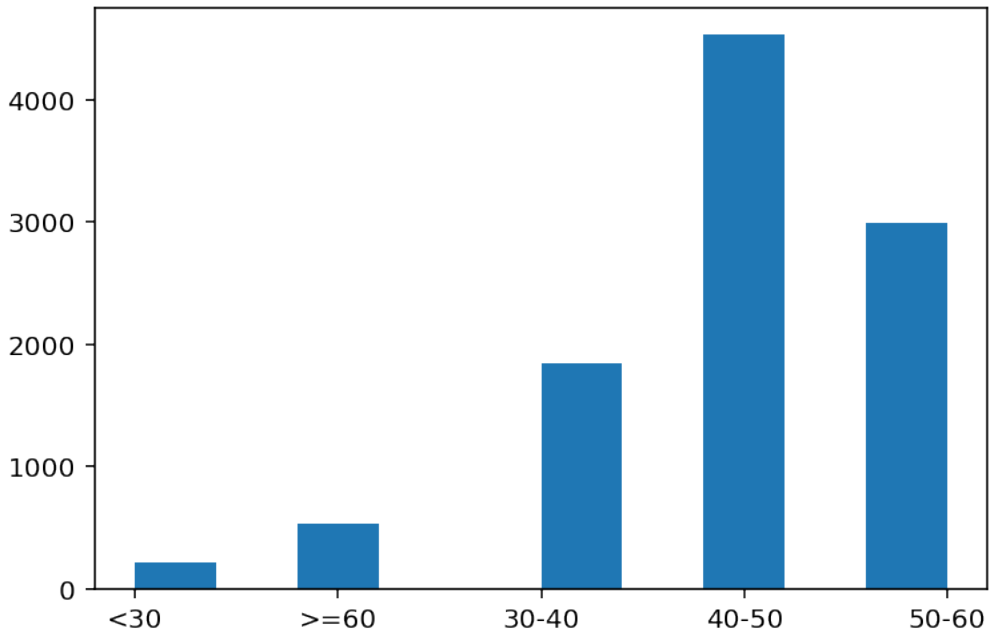
[23]:



0.5.1 2.Frequency Distribution of customers in various Age groups

[24]:
```
plt.hist(cust.Age_Group);
```

[24]:

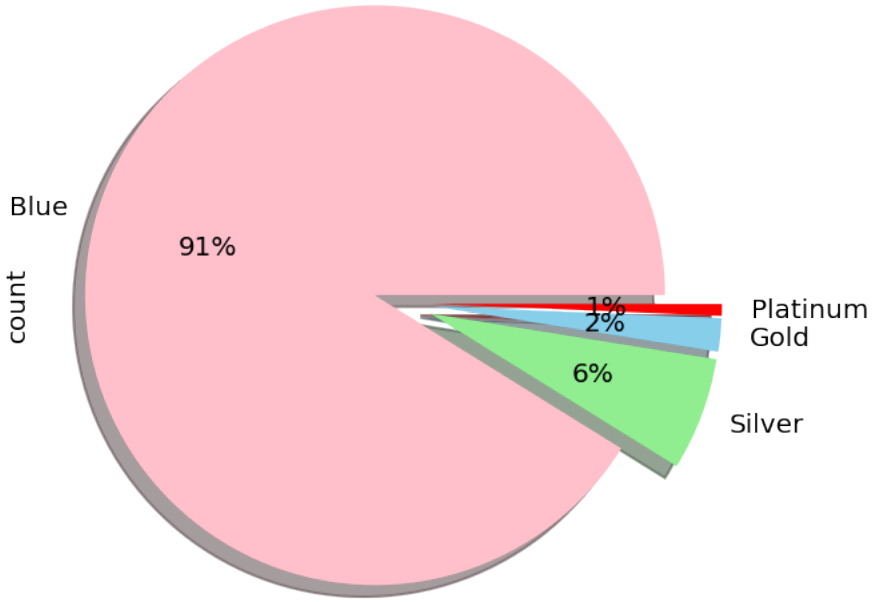


0.5.2 3. Frequency Distribution of Card Categories

[25]:
```
cc['Card_Category'].value_counts().plot.pie(figsize=(5,5), autopct='%1.0f%%',␣
↪shadow=True, explode=(0.1,0.1,0.1,0.1), colors=['pink', 'lightgreen',␣
↪'skyblue','red'])
```

[25]: <Axes: ylabel='count'>

[25]:

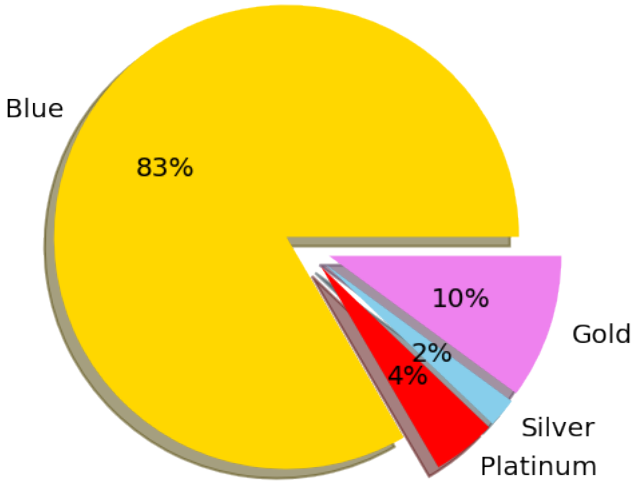


0.5.3 5. Total Revenue as per Card Category

[26]:
```
result=cc.groupby('Card_Category').sum()

category = cc['Card_Category'].unique()

plt.pie(result['Total_Revenue'],labels=category,autopct= '%1.
↪0f%%',colors=['Gold', 'Red','Skyblue','Violet'], explode=(0.1,0.1,0.1,0.
↪1),shadow=True)



#cc['Total_Revenue'].plot.pie(labels=category, autopct='%1.0f%%',␣
↪colors=['Gold', 'Red', 'Skyblue', 'Violet'], explode=(0.1, 0.1, 0.1, 0.1),␣
↪figsize=(7, 7))
```

[26]:
```
([<matplotlib.patches.Wedge at 0x7f4b9ce72bf0>,
  <matplotlib.patches.Wedge at 0x7f4b9cf4bb80>,
  <matplotlib.patches.Wedge at 0x7f4b9ce73880>,
  <matplotlib.patches.Wedge at 0x7f4b9ceac1c0>],
 [Text(-1.0407002504518528, 0.5974470593361816, 'Blue'),
  Text(0.7431925849665428, -0.9421596370311924, 'Platinum'),
  Text(0.9185538928635727, -0.7721779237359591, 'Silver'),
  Text(1.1401084181007262, -0.3743698638724276, 'Gold')],
 [Text(-0.6070751460969142, 0.3485107846127725, '83%'),
  Text(0.43352900789715, -0.5495931216015288, '4%'),
  Text(0.5358231041704173, -0.4504371221793094, '2%'),
  Text(0.6650632438920902, -0.21838242059224944, '10%')])
```

[26]:



[27]:
```
chip_usage = merged.groupby(['Education_Level','Card_Category' 1]).size().
 ↪unstack()
chip_usage.plot(kind='bar', stacked=True, color=['tomato', 'royalblue','gold'])
plt.title('Chip Usage Frequency by Card Category')
plt.xlabel('Card Category')
plt.ylabel('Chip Usage Rate')
plt.show()
```

```
  File "/tmp/ipykernel_449/1564304773.py", line 1
    chip_usage = merged.groupby(['Education_Level','Card_Category' 1]).size().
 ↪unstack()
                                                                    ^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
```

[28]:
```
chip_usage = merged.groupby(['Card_Category','Use Chip']).size().unstack()
chip_usage.plot(kind='bar', stacked=True, color=['tomato', 'royalblue','gold'])
plt.title('Chip Usage Frequency by Card Category')
plt.xlabel('Card Category')
plt.ylabel('Chip Usage Rate')
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_449/87453868.py in <cell line: 1>()
----> 1 chip_usage = merged.groupby(['Card_Category','Use Chip']).size().
 ↪unstack()
      2 chip_usage.plot(kind='bar', stacked=True, color=['tomato',␣
 ↪'royalblue','gold'])
      3 plt.title('Chip Usage Frequency by Card Category')
      4 plt.xlabel('Card Category')
      5 plt.ylabel('Chip Usage Rate')

NameError: name 'merged' is not defined
```

[0]:
```
```

[29]:
```
cc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10108 entries, 0 to 10107
Data columns (total 20 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Client_Num             10108 non-null  int64
 1   Card_Category          10108 non-null  object
 2   Annual_Fees            10108 non-null  int64
 3   Activation_30_Days     10108 non-null  int64
 4   Customer_Acq_Cost      10108 non-null  int64
 5   Week_Start_Date        10108 non-null  object
 6   Week_Num               10108 non-null  object
 7   Qtr                    10108 non-null  object
 8   current_year           10108 non-null  int64
 9   Credit_Limit           10108 non-null  float64
 10  Total_Revolving_Bal    10108 non-null  int64
 11  Total_Trans_Amt        10108 non-null  int64
 12  Total_Trans_Vol        10108 non-null  int64
 13  Avg_Utilization_Ratio  10108 non-null  float64
 14  Use Chip               10108 non-null  object
```

```
 15  Exp Type               10108 non-null  object
 16  Interest_Earned        10108 non-null  float64
 17  Delinquent_Acc         10108 non-null  int64
 18  Total_Revenue          10108 non-null  float64
 19  Week_No.               10108 non-null  int64
dtypes: float64(4), int64(10), object(6)
memory usage: 1.5+ MB
```

[30]:
```
cc.head()
```

[30]:

	Client_Num	Card_Category	Annual_Fees	Activation_30_Days	\
0	708082083	Blue	200	0	
1	708083283	Blue	445	1	
2	708084558	Blue	140	0	
3	708085458	Blue	250	1	
4	708086958	Blue	320	1	

	Customer_Acq_Cost	Week_Start_Date	Week_Num	Qtr	current_year	Credit_Limit	\
0	87	01-01-2023	Week-1	Q1	2023	3544.0	
1	108	01-01-2023	Week-1	Q1	2023	3421.0	
2	106	01-01-2023	Week-1	Q1	2023	8258.0	
3	150	01-01-2023	Week-1	Q1	2023	1438.3	
4	106	01-01-2023	Week-1	Q1	2023	3128.0	

	Total_Revolving_Bal	Total_Trans_Amt	Total_Trans_Vol	\
0	1661	15149	111	
1	2517	992	21	
2	1771	1447	23	
3	0	3940	82	
4	749	4369	59	

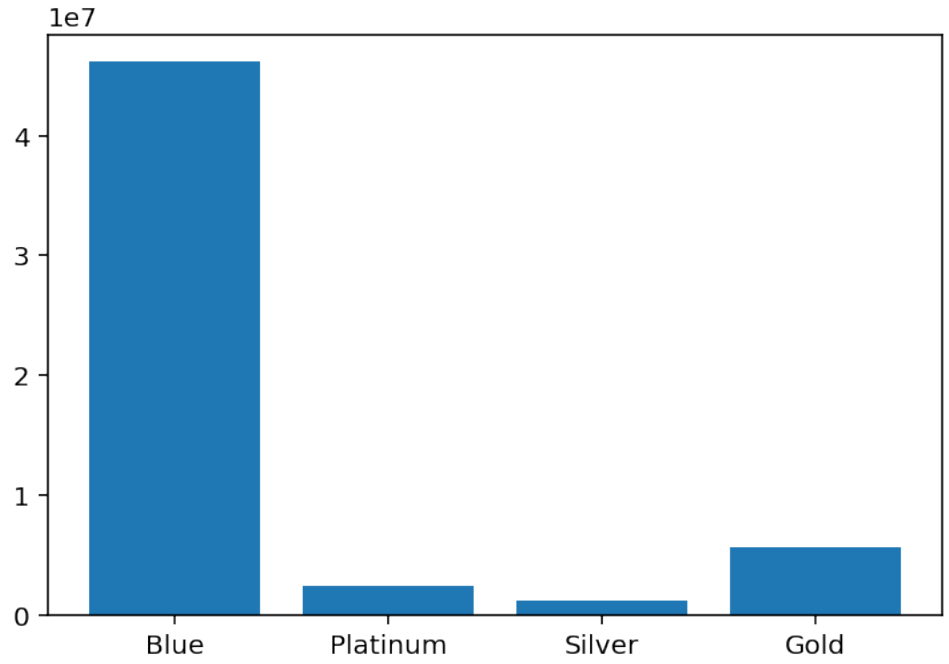
	Avg_Utilization_Ratio	Use Chip	Exp Type	Interest_Earned	\
0	0.469	Chip	Travel	4393.21	
1	0.736	Swipe	Entertainment	69.44	
2	0.214	Chip	Bills	202.58	
3	0.000	Online	Grocery	236.40	
4	0.239	Swipe	Fuel	1004.87	

	Delinquent_Acc	Total_Revenue	Week_No.
0	0	19742.21	1
1	0	1506.44	1
2	0	1789.58	1
3	0	4426.40	1
4	1	5693.87	1

[31]:
```
result=cc.groupby('Card_Category').sum()
```

```
category = cc['Card_Category'].unique()

plt.bar(category, result['Total_Revenue'])

plt.xticks(category)

plt.show()
```

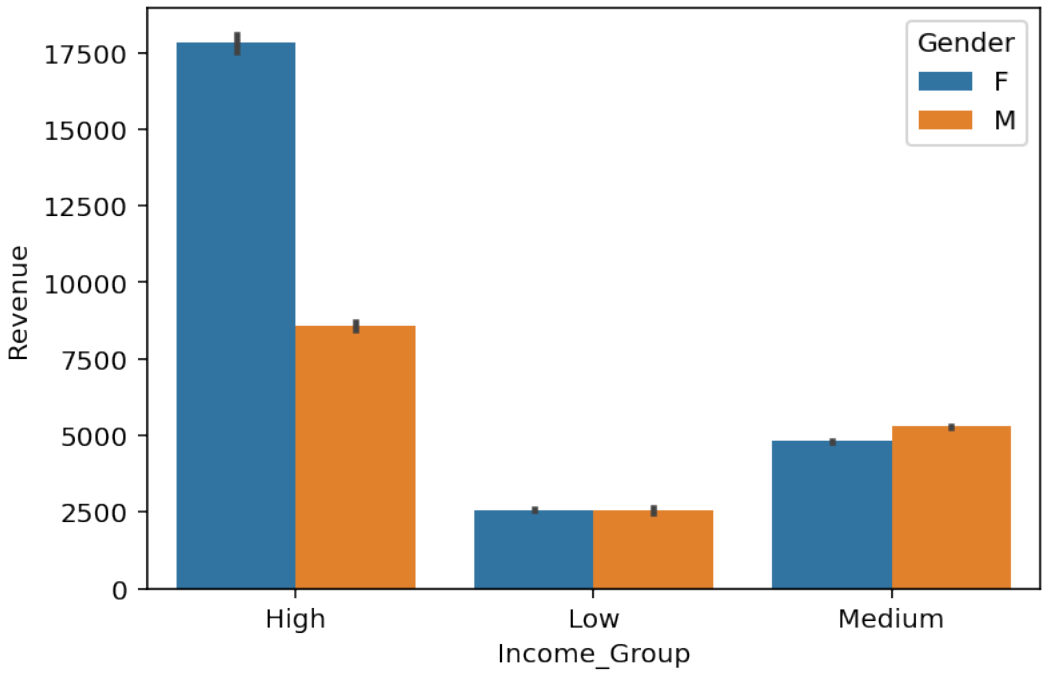
[31]:



[32]:
```
cust['Revenue'] = cc['Total_Revenue']
```

[33]:
```
sns.barplot(x=cust.Income_Group, y=cust.Revenue, hue=cust.Gender);
```

[33]:



[34]:
```
cust.head()
```

[34]:

	Client_Num	Customer_Age	Gender	Dependent_Count	Education_Level
0	708082083	24	F	1	Uneducated
1	708083283	62	F	0	Unknown
2	708084558	32	F	1	Unknown
3	708085458	38	M	2	Uneducated
4	708086958	48	M	4	Graduate

	Marital_Status	state_cd	Zipcode	Car_Owner	House_Owner	Personal_loan
0	Single	FL	91750	no	yes	no
1	Married	NJ	91750	no	no	no
2	Married	NJ	91750	yes	no	no
3	Single	NY	91750	no	no	no
4	Single	TX	91750	yes	yes	no

	contact	Customer_Job	Income	Cust_Satisfaction_Score	Age_Group
0	unknown	Businessman	202326	3	<30
1	cellular	Selfemployeed	5225	2	>=60
2	unknown	Selfemployeed	14235	2	30-40
3	cellular	Blue-collar	45683	1	30-40
4	cellular	Businessman	59279	1	40-50

	Income_Group	Revenue
0	High	19742.21

```
1          Low   1506.44
2          Low   1789.58
3       Medium   4426.40
4       Medium   5693.87
```

[35]:
```
cc.head()
```

[35]:
```
   Client_Num Card_Category  Annual_Fees  Activation_30_Days  \
0   708082083          Blue          200                   0
1   708083283          Blue          445                   1
2   708084558          Blue          140                   0
3   708085458          Blue          250                   1
4   708086958          Blue          320                   1

   Customer_Acq_Cost Week_Start_Date Week_Num Qtr  current_year  Credit_Limit  \
0                 87      01-01-2023   Week-1  Q1          2023        3544.0
1                108      01-01-2023   Week-1  Q1          2023        3421.0
2                106      01-01-2023   Week-1  Q1          2023        8258.0
3                150      01-01-2023   Week-1  Q1          2023        1438.3
4                106      01-01-2023   Week-1  Q1          2023        3128.0

   Total_Revolving_Bal  Total_Trans_Amt  Total_Trans_Vol  \
0                 1661            15149              111
1                 2517              992               21
2                 1771             1447               23
3                    0             3940               82
4                  749             4369               59

   Avg_Utilization_Ratio Use Chip       Exp Type  Interest_Earned  \
0                  0.469    Chip         Travel          4393.21
1                  0.736   Swipe  Entertainment            69.44
2                  0.214    Chip          Bills           202.58
3                  0.000  Online        Grocery           236.40
4                  0.239   Swipe           Fuel          1004.87

   Delinquent_Acc  Total_Revenue  Week_No.
0               0       19742.21         1
1               0        1506.44         1
2               0        1789.58         1
3               0        4426.40         1
4               1        5693.87         1
```

[36]:
```
w= cc['Qtr'].unique
w
```

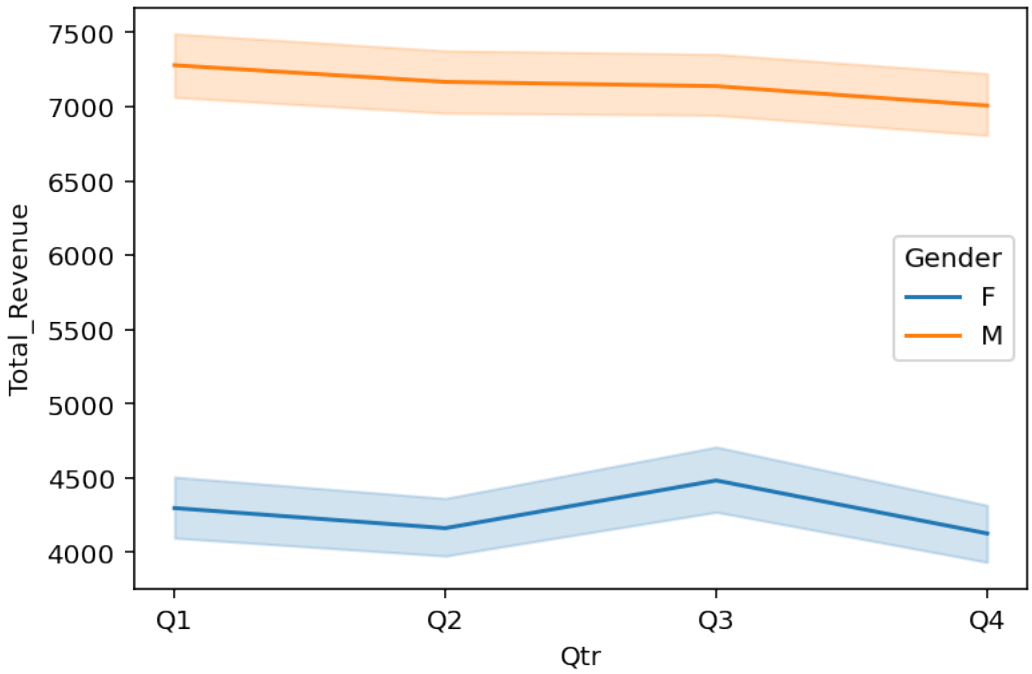
[36]:
```
<bound method Series.unique of 0        Q1
1        Q1
```

```
2        Q1
3        Q1
4        Q1
         ..
10103    Q4
10104    Q4
10105    Q4
10106    Q4
10107    Q4
Name: Qtr, Length: 10108, dtype: object>
```

[37]:
```
sns.lineplot(x= cc.Qtr,y= cc.Total_Revenue, hue= cust. Gender)
```

[37]:
```
<Axes: xlabel='Qtr', ylabel='Total_Revenue'>
```

[37]:



[38]:
```
cc['current_year'].unique
```

[38]:
```
<bound method Series.unique of 0        2023
1        2023
2        2023
3        2023
4        2023
         …
10103    2023
10104    2023
```

```
10105    2023
10106    2023
10107    2023
Name: current_year, Length: 10108, dtype: int64>
```

[39]:
```
cust.head()
```

[39]:
```
    Client_Num  Customer_Age Gender  Dependent_Count Education_Level  \
0   708082083            24      F                1      Uneducated
1   708083283            62      F                0         Unknown
2   708084558            32      F                1         Unknown
3   708085458            38      M                2      Uneducated
4   708086958            48      M                4        Graduate

  Marital_Status state_cd  Zipcode Car_Owner House_Owner Personal_loan  \
0         Single       FL    91750        no         yes            no
1        Married       NJ    91750        no          no            no
2        Married       NJ    91750       yes          no            no
3         Single       NY    91750        no          no            no
4         Single       TX    91750       yes         yes            no

    contact   Customer_Job  Income  Cust_Satisfaction_Score Age_Group  \
0   unknown    Businessman  202326                         3       <30
1  cellular  Selfemployeed    5225                         2      >=60
2   unknown  Selfemployeed   14235                         2     30-40
3  cellular    Blue-collar   45683                         1     30-40
4  cellular    Businessman   59279                         1     40-50

  Income_Group   Revenue
0         High  19742.21
1          Low   1506.44
2          Low   1789.58
3       Medium   4426.40
4       Medium   5693.87
```

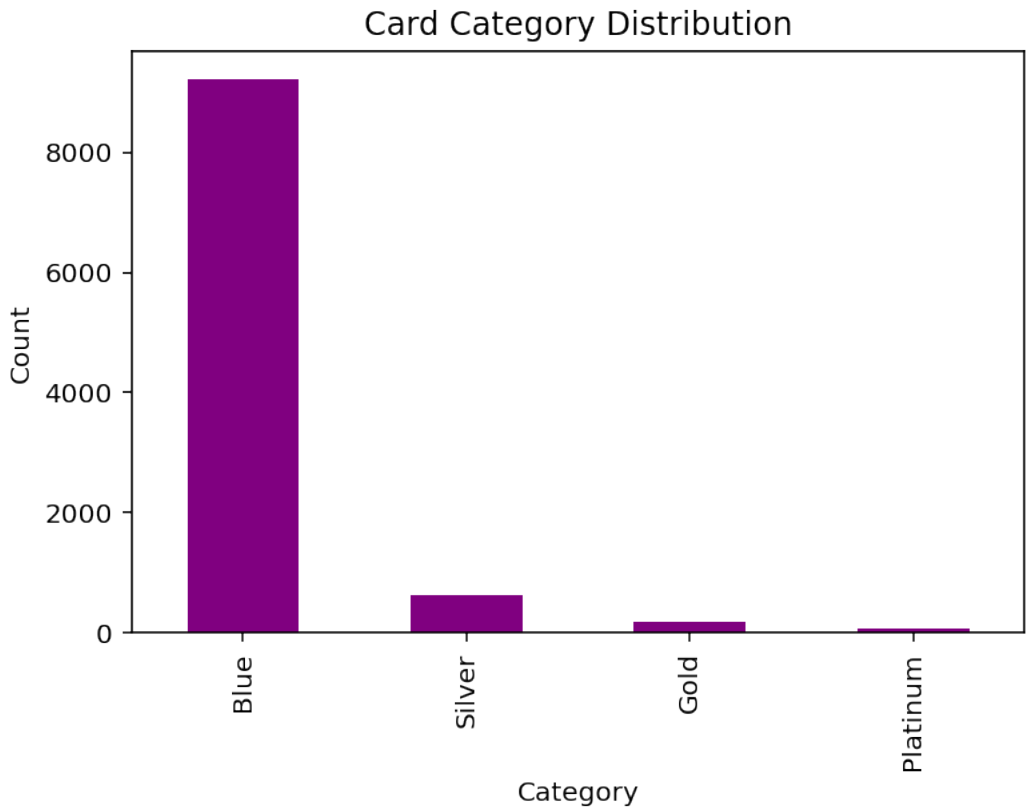
0.6 Some More Insights

#Analysis

0.7 1.Card Category distribution

[40]:
```
cc['Card_Category'].value_counts().plot(kind='bar', color='Purple')
plt.title('Card Category Distribution')
plt.xlabel('Category')
plt.ylabel('Count')
plt.show()
```

[40]:

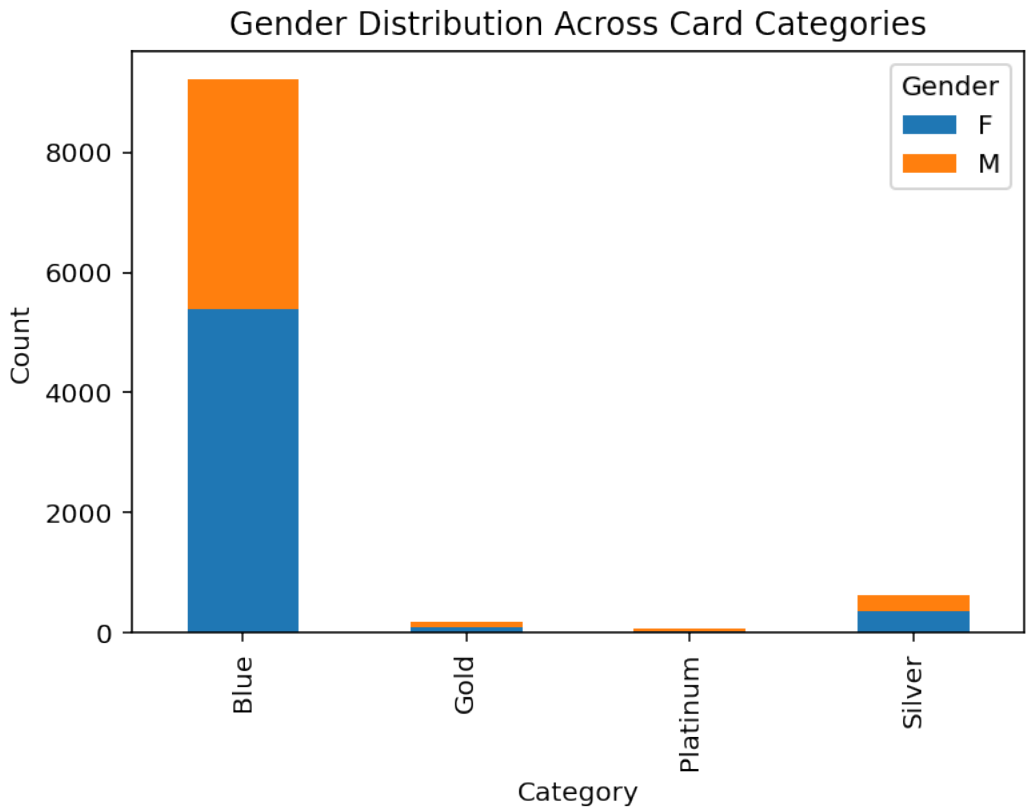


0.8 2.Gender Distribution Across Card Categories

[41]:
```
cust['Card_Category'] = cc['Card_Category']
```

[42]:
```
cust.groupby(['Card_Category','Gender']).size().unstack().plot(kind='bar',␣
↪stacked=True)
plt.title('Gender Distribution Across Card Categories')
plt.xlabel('Category')
plt.ylabel('Count')
plt.show()
```

[42]:



0.9 3. Activation 30 days with card category & Gender

[0]:

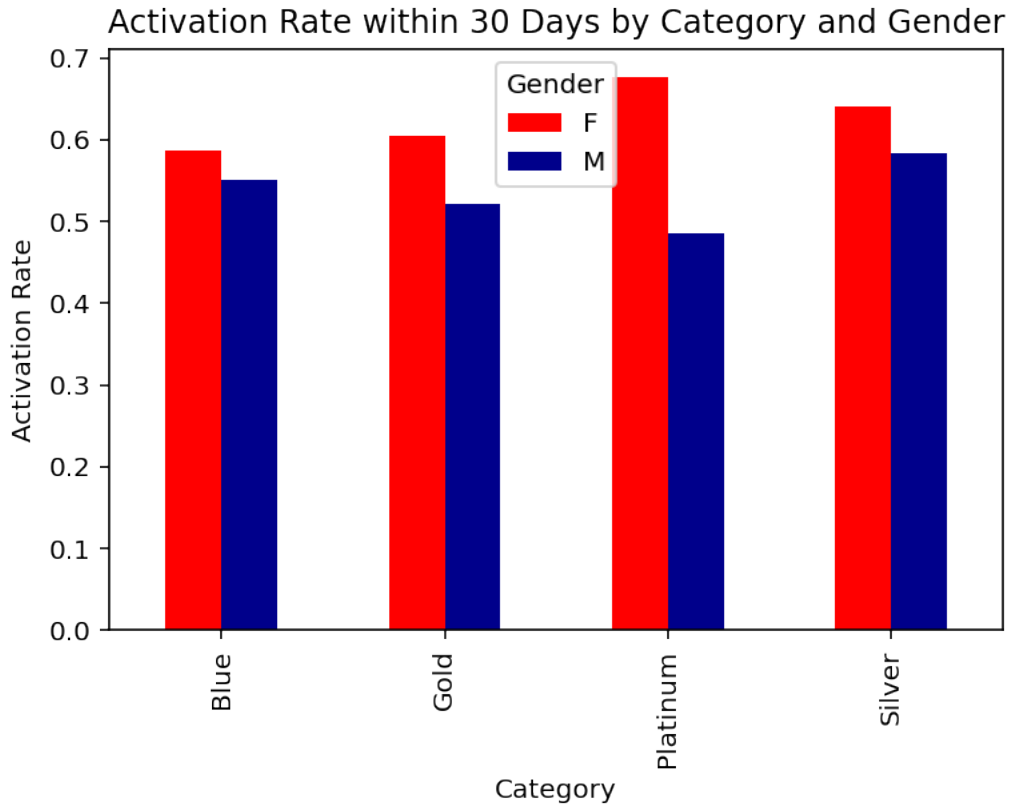
0.10 Merge two data frames cust & cc

[43]:
```
import pandas as pd
import matplotlib.pyplot as plt
# Assuming you have the following DataFrames:
# cust = pd.read_csv('customer_details.csv')  # Customer details DataFrame
# cc = pd.read_csv('credit_card_details.csv')  # Credit card details DataFrame
# Merge the two DataFrames on Client ID

merged = pd.merge(cc, cust[['Client_Num', 'Gender']], on='Client_Num')
```

[44]:
```
activation_rate = merged.groupby(['Card_Category',␣
↪'Gender'])['Activation_30_Days'].mean().unstack()
activation_rate.plot(kind='bar', color=['Red', 'darkblue'])
plt.title('Activation Rate within 30 Days by Category and Gender')
plt.xlabel('Category')
```

```
plt.ylabel('Activation Rate')
plt.show()
```

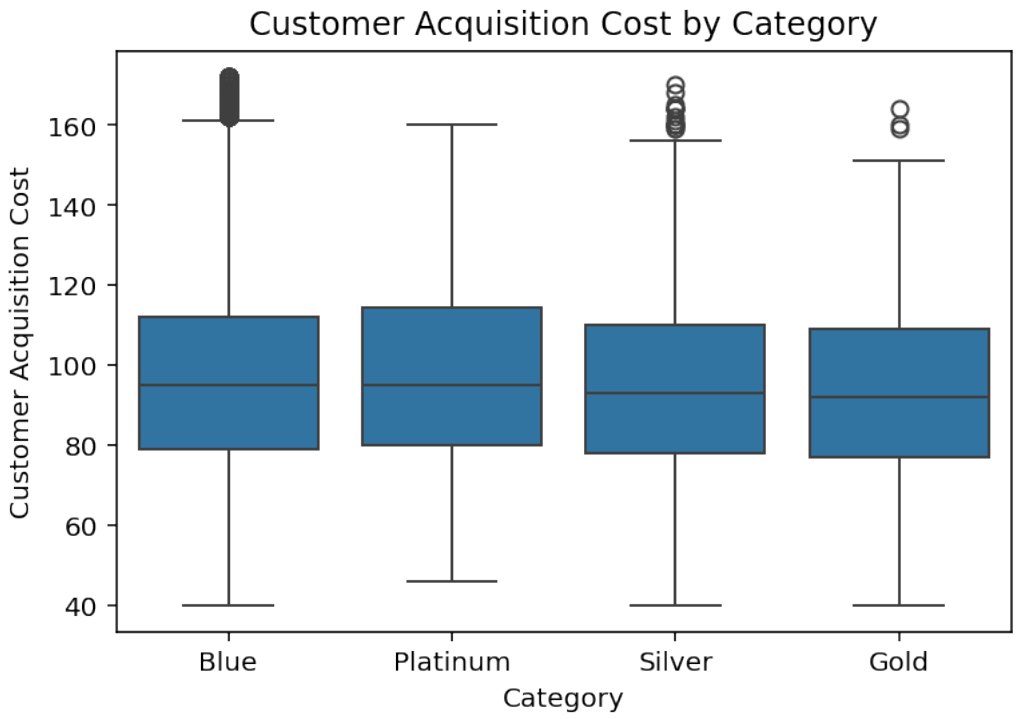
[44]:



0.11 4. Customer Acquisition Cost by Card Category.

[45]:
```
sns.boxplot(x='Card_Category', y='Customer_Acq_Cost', data=cc)
plt.title('Customer Acquisition Cost by Category')
plt.xlabel('Category')
plt.ylabel('Customer Acquisition Cost')
plt.show()
```

[45]:

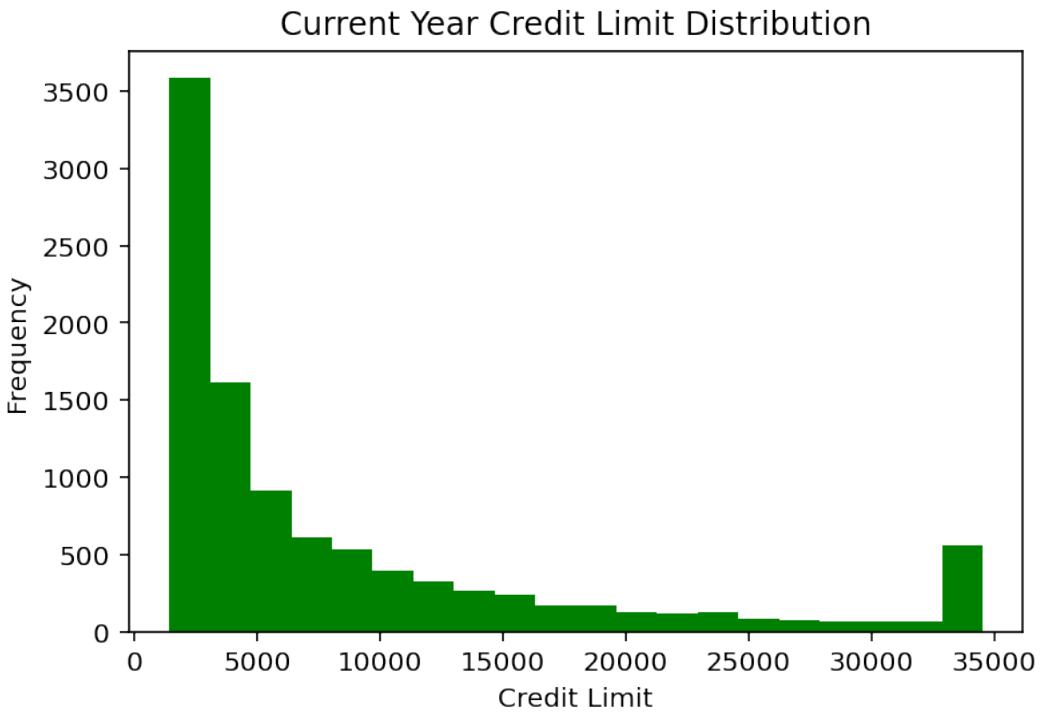


[0]:

0.12 5. Current Year Credit Limit Distribution

[46]:
```
cc['Credit_Limit'].plot(kind='hist', bins=20, color='green')
plt.title('Current Year Credit Limit Distribution')
plt.xlabel('Credit Limit')
plt.ylabel('Frequency')
plt.show()
```

[46]:

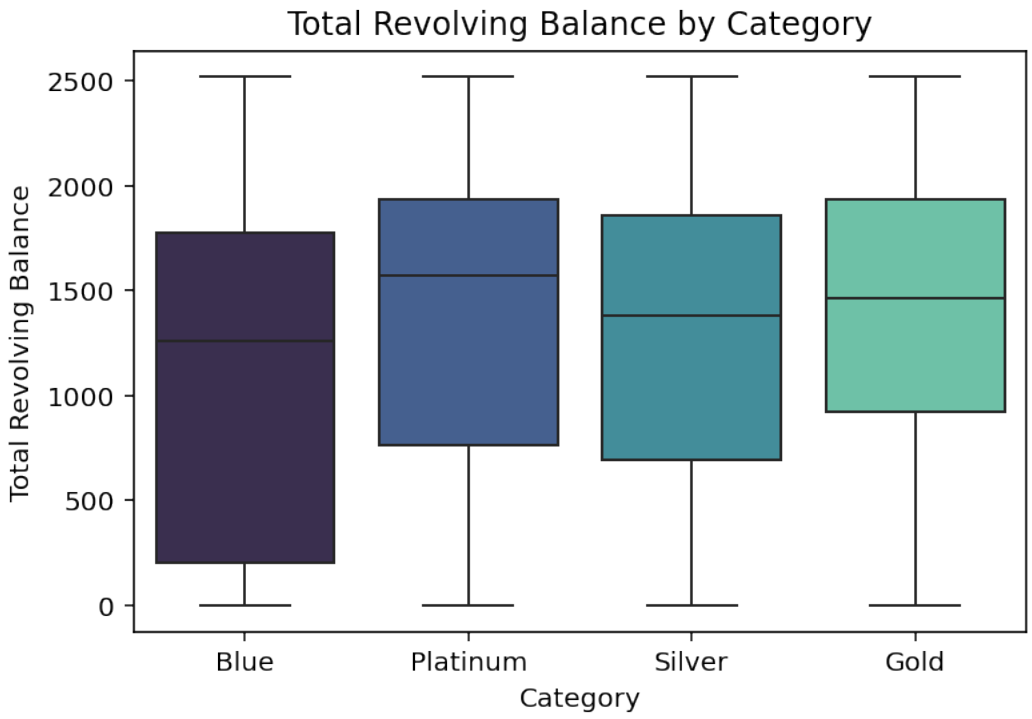


0.13 6. Total Revolving balance by category

[47]:
```
sns.boxplot(x='Card_Category', y='Total_Revolving_Bal', data=cc, palette="mako")
plt.title('Total Revolving Balance by Category')
plt.xlabel('Category')
plt.ylabel('Total Revolving Balance')
plt.show()
```

```
/tmp/ipykernel_449/1479599055.py:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(x='Card_Category', y='Total_Revolving_Bal', data=cc,
palette="mako")
```

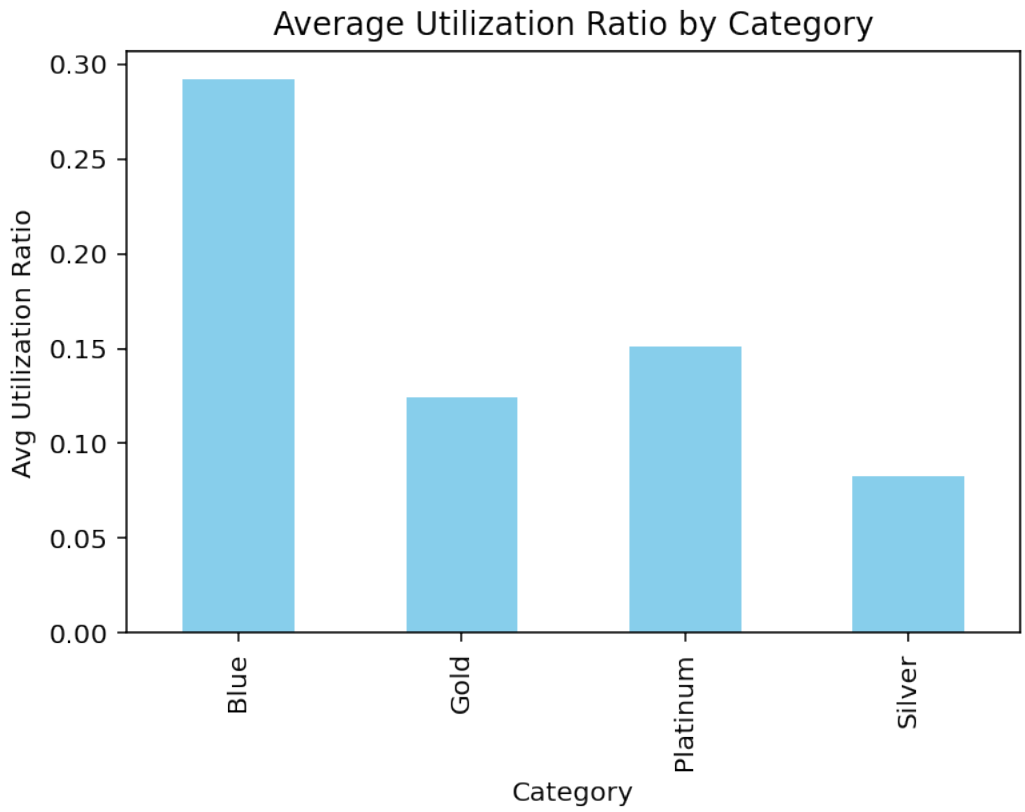
[47]:



0.13.1 7. Average Utilization Ratio by card category

[48]:
```
cc.groupby('Card_Category')['Avg_Utilization_Ratio'].mean().plot(kind='bar',␣
↪color='skyblue')
plt.title('Average Utilization Ratio by Category')
plt.xlabel('Category')
plt.ylabel('Avg Utilization Ratio')
plt.show()
```

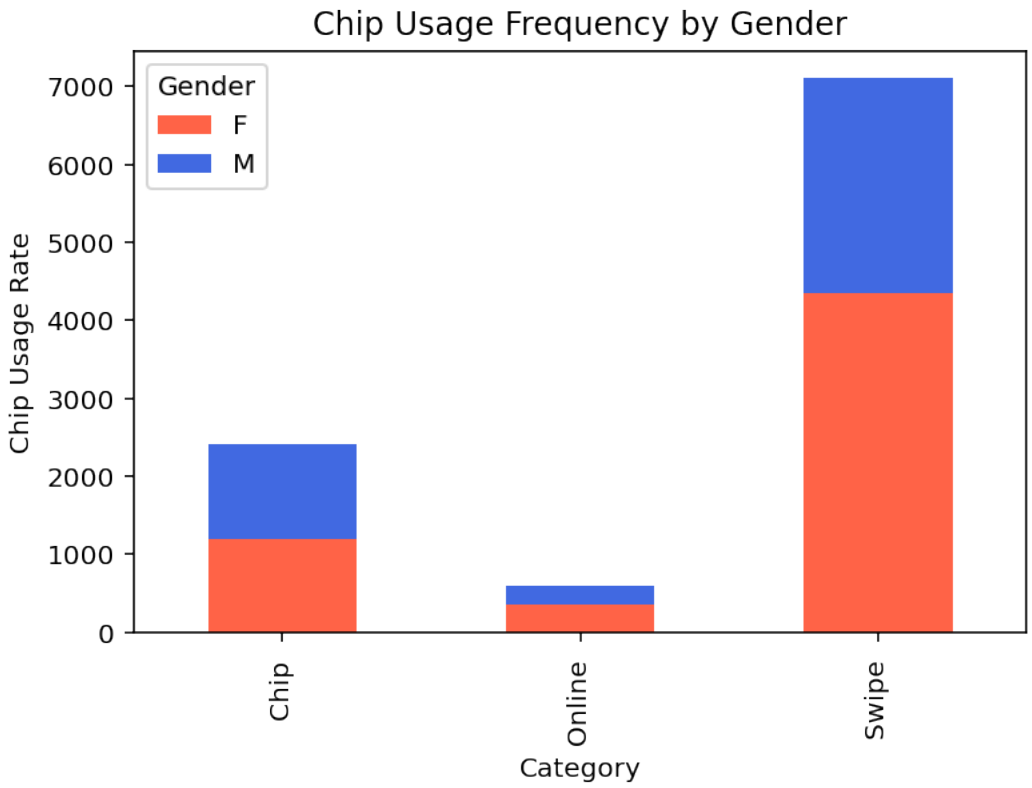
[48]:



0.13.2 8. Chip Usage Rate by Category & Gender

[49]:
```
chip_usage = merged.groupby(['Use Chip', 'Gender']).size().unstack()
chip_usage.plot(kind='bar', stacked=True, color=['tomato', 'royalblue'])
plt.title('Chip Usage Frequency by Gender')
plt.xlabel('Category')
plt.ylabel('Chip Usage Rate')
plt.show()
```

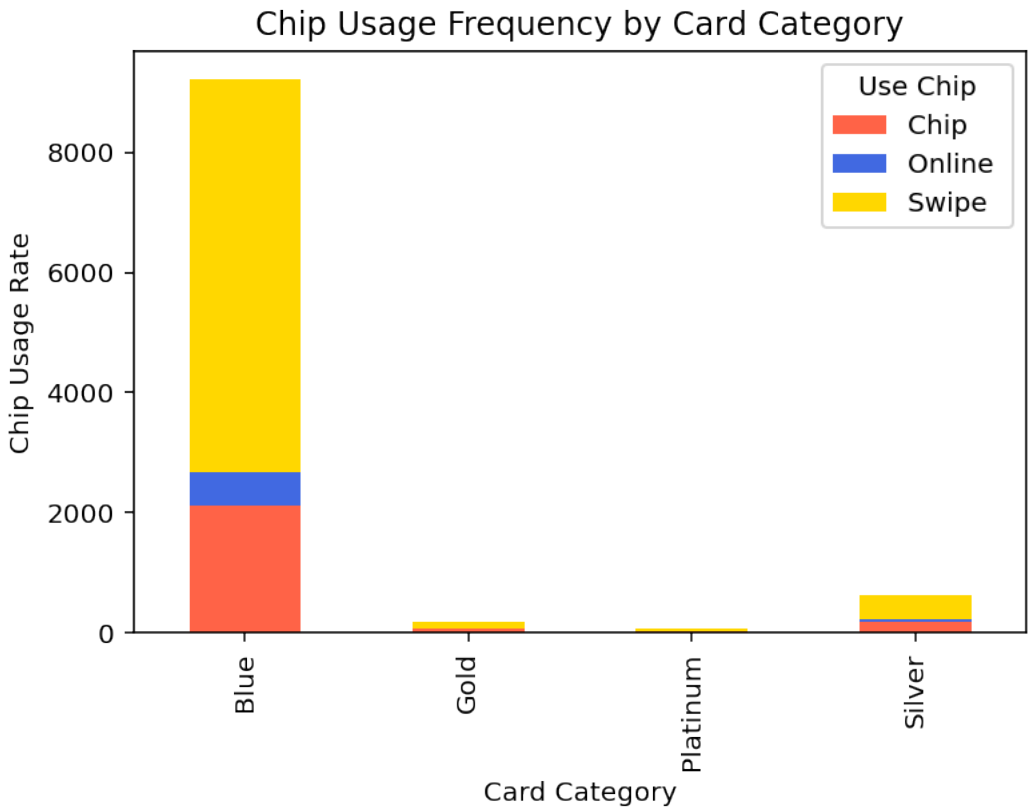
[49]:



0.13.3 9.Chip usage frequency as per Card Category

[50]:
```
chip_usage = merged.groupby(['Card_Category','Use Chip']).size().unstack()
chip_usage.plot(kind='bar', stacked=True, color=['tomato', 'royalblue','gold'])
plt.title('Chip Usage Frequency by Card Category')
plt.xlabel('Card Category')
plt.ylabel('Chip Usage Rate')
plt.show()
```

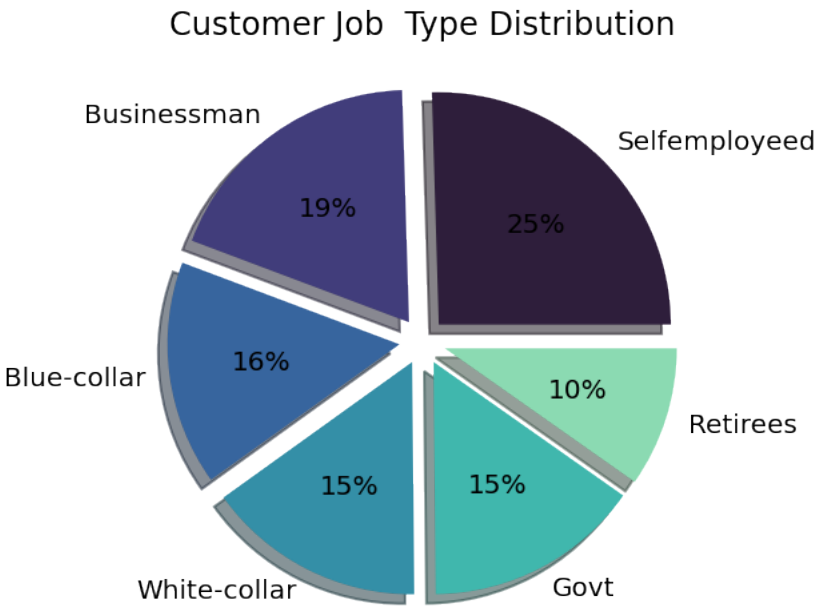
[50]:



0.13.4 10.1 Customer Job Type Distribution

[51]:
```
cust['Customer_Job'].value_counts().plot(kind='pie', autopct='%1.0f%%',␣
↪colors=sns.color_palette('mako'),shadow= True, explode=(0.1,0.1,0.1,0.1,0.
↪1,0.1))
plt.title('Customer Job  Type Distribution')
plt.ylabel('')
plt.show()
```

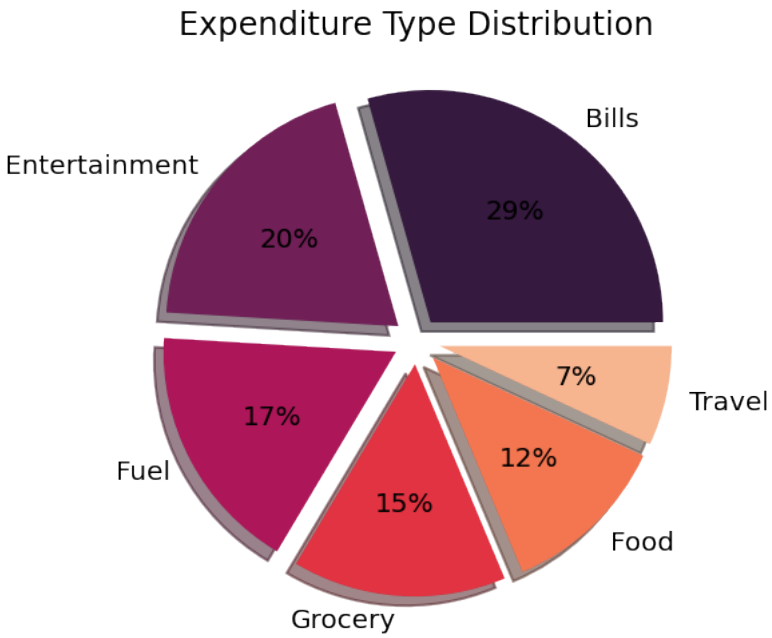
[51]:



0.13.5 10.2 Expenditure Type DisDistribution

[52]:
```
cc['Exp Type'].value_counts().plot(kind='pie', autopct='%1.0f%%', colors=sns.
↪color_palette('rocket'),shadow= True, explode=(0.1,0.1,0.1,0.1,0.1,0.1))
plt.title('Expenditure Type Distribution')
plt.ylabel('')
plt.show()
```

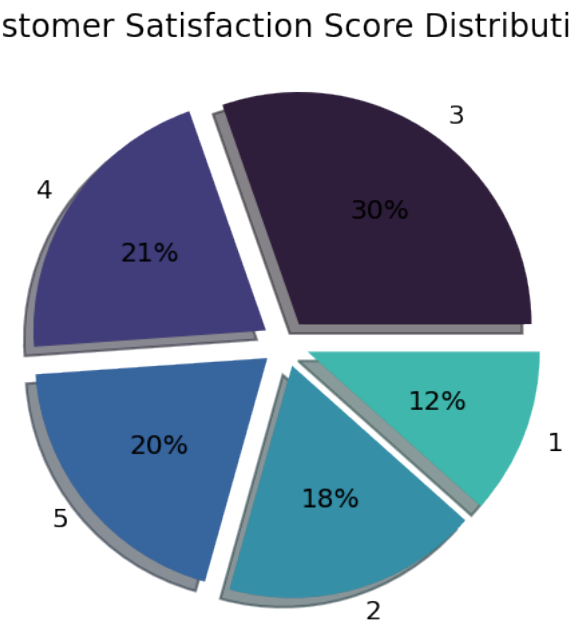
[52]:



0.13.6 10.3 Customer Satisfaction

[53]:
```
cust['Cust_Satisfaction_Score'].value_counts().plot(kind='pie', autopct='%1.
↪0f%%', colors=sns.color_palette('mako'), shadow=True, explode=(0.1,0.1,0.1,0.
↪1,0.1))
plt.title('Customer Satisfaction Score Distribution')
plt.ylabel('')
plt.show()
#cust['Cust_Satisfaction _Score'].value_counts().plot(kind='pie', autopct='%1.
↪0f%%', colors=sns.color_palette('rocket'),shadow= True, explode=(0.1,0.1,0.
↪1,0.1,0.1))
```

[53]:



[54]:
```
cc.head()
```

[54]:

	Client_Num	Card_Category	Annual_Fees	Activation_30_Days	\
0	708082083	Blue	200	0	
1	708083283	Blue	445	1	
2	708084558	Blue	140	0	
3	708085458	Blue	250	1	
4	708086958	Blue	320	1	

	Customer_Acq_Cost	Week_Start_Date	Week_Num	Qtr	current_year	Credit_Limit	\
0	87	01-01-2023	Week-1	Q1	2023	3544.0	
1	108	01-01-2023	Week-1	Q1	2023	3421.0	
2	106	01-01-2023	Week-1	Q1	2023	8258.0	
3	150	01-01-2023	Week-1	Q1	2023	1438.3	
4	106	01-01-2023	Week-1	Q1	2023	3128.0	

	Total_Revolving_Bal	Total_Trans_Amt	Total_Trans_Vol	\
0	1661	15149	111	
1	2517	992	21	
2	1771	1447	23	
3	0	3940	82	
4	749	4369	59	

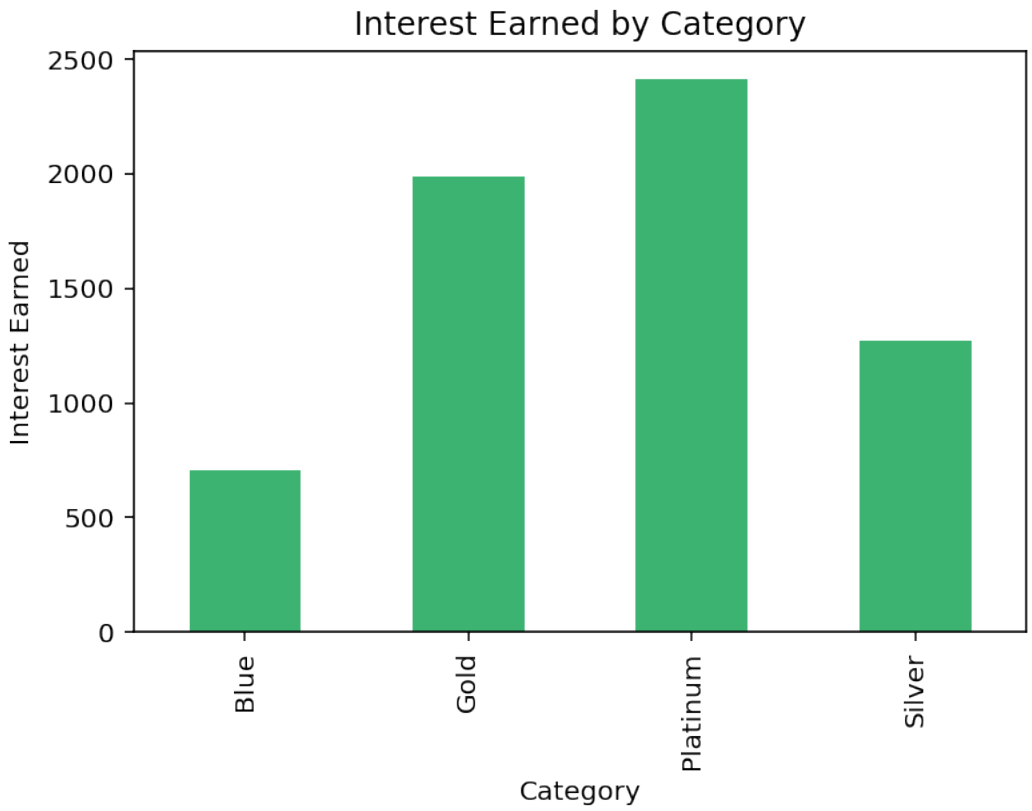
	Avg_Utilization_Ratio	Use Chip	Exp Type	Interest_Earned	\
0	0.469	Chip	Travel	4393.21	
1	0.736	Swipe	Entertainment	69.44	

```
2                  0.214    Chip           Bills          202.58
3                  0.000  Online         Grocery          236.40
4                  0.239   Swipe            Fuel         1004.87

   Delinquent_Acc  Total_Revenue  Week_No.
0               0       19742.21         1
1               0        1506.44         1
2               0        1789.58         1
3               0        4426.40         1
4               1        5693.87         1
```

0.13.7 11. Interest Earned by Category

[55]:
```
cc.groupby('Card_Category')['Interest_Earned'].mean().plot(kind='bar',␣
↪color='mediumseagreen')
plt.title('Interest Earned by Category')
plt.xlabel('Category')
plt.ylabel('Interest Earned')
plt.show()
```

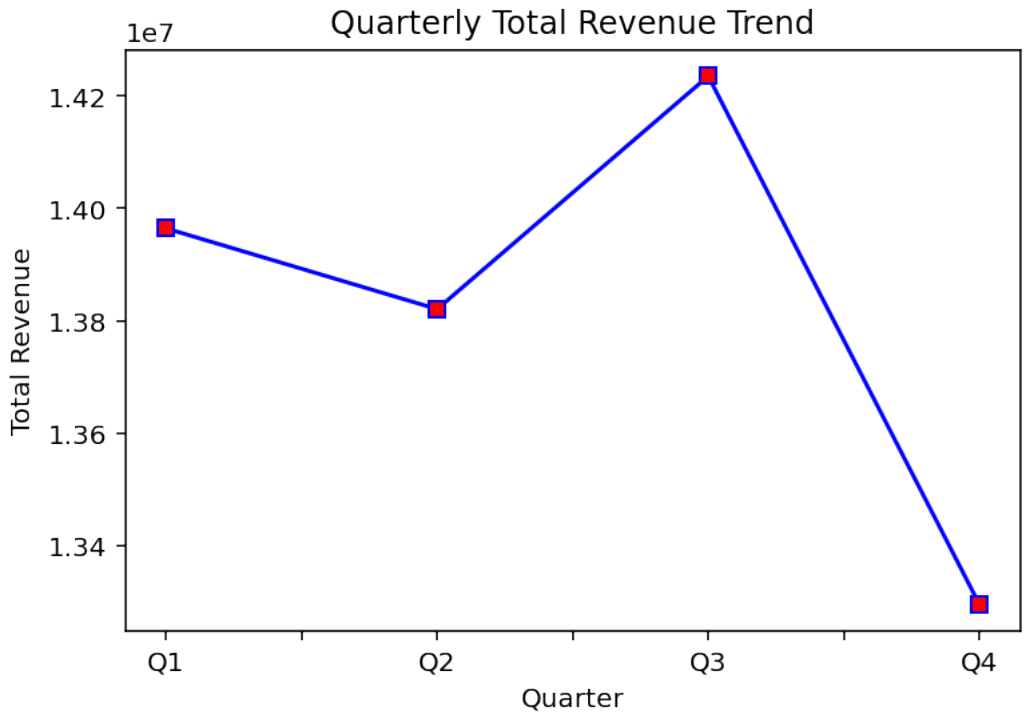
[55]:



0.13.8 12. Quarterly Total Revenue Trend

[56]:
```
cc.groupby('Qtr')['Total_Revenue'].sum().plot(kind='line',marker='s',color='blue', mfc='red')
plt.title('Quarterly Total Revenue Trend')
plt.xlabel('Quarter')
plt.ylabel('Total Revenue')
plt.show()
```

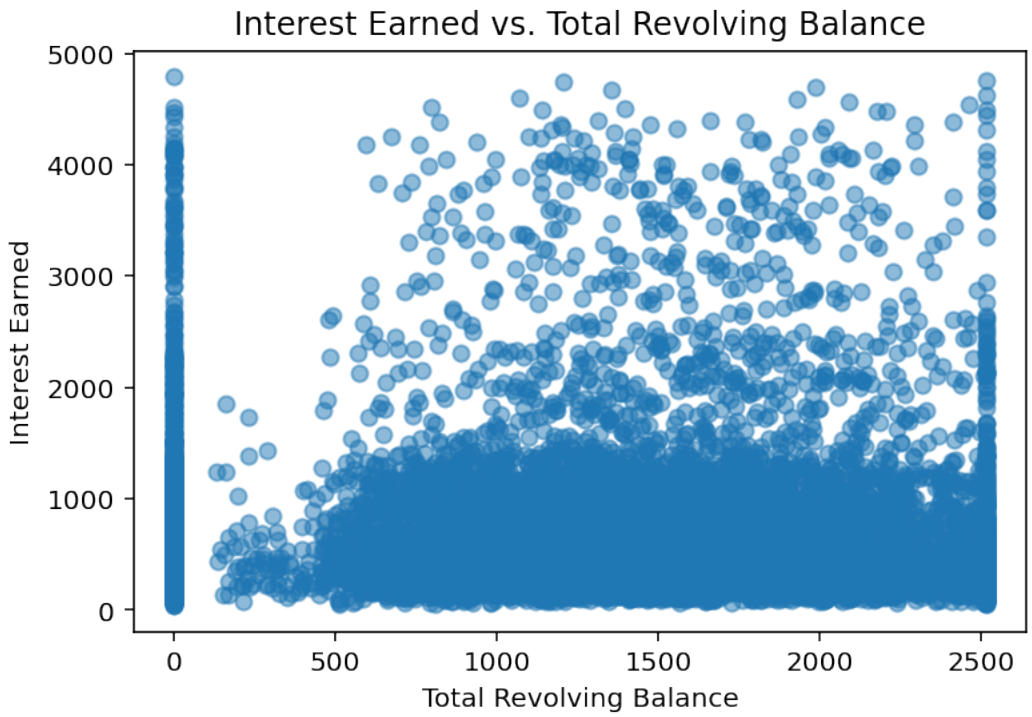
[56]:



0.13.9 13. Interest Earned vs. Total Revolving Balance

[57]:
```
plt.scatter(cc['Total_Revolving_Bal'], cc['Interest_Earned'], alpha=0.5)
plt.title('Interest Earned vs. Total Revolving Balance')
plt.xlabel('Total Revolving Balance')
plt.ylabel('Interest Earned')
plt.show()
```

[57]:


Interest Earned vs. Total Revolving Balance
Interest Earned
Total Revolving Balance
0
1000
2000
3000
4000
5000
0
500
1000
1500
2000
2500

[0]: