

Ques-1: Import Necessary Libraries

```
In [14]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
matplotlib.interactive = True
sns.set(color_codes =True)
from warnings import filterwarnings
filterwarnings('ignore')

In [21]: df=pd.read_csv('BankChurners (3).csv')

In [3]: df

Out[3]:
```

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_Limit
0	768805383	Existing Customer	45	M	3	High School	Married	60K-80K	Blue	39	1	3	12691.0
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than 540K	Blue	44	1	2	8256.0
2	713982108	Existing Customer	51	M	3	Graduate	Married	80K-120K	Blue	36	1	0	3418.0
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than 540K	Blue	34	4	1	3313.0
4	709106358	Existing Customer	40	M	3	Uneducated	Married	60K-80K	Blue	21	1	0	4716.0
...
10122	772668833	Existing Customer	50	M	2	Graduate	Single	40K-60K	Blue	40	2	3	4003.0
10123	710638233	Attrited Customer	41	M	2	Unknown	Divorced	40K-60K	Blue	25	2	3	4277.0
10124	71650063	Attrited Customer	44	F	1	High School	Married	Less than 540K	Blue	36	3	4	5409.0
10125	717406983	Attrited Customer	30	M	2	Graduate	Unknown	40K-60K	Blue	36	3	3	5281.0
10126	714337233	Attrited Customer	43	F	2	Graduate	Married	Less than 540K	Silver	25	2	4	10388.0

10127 rows x 21 columns

Ques-2: Display a sample of five rows of the data frame

```
In [4]: df1=df.head(5)

In [5]: df1

Out[5]:
```

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_Limit
0	768805383	Existing Customer	45	M	3	High School	Married	60K-80K	Blue	39	1	3	12691.0
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than 540K	Blue	44	1	2	8256.0
2	713982108	Existing Customer	51	M	3	Graduate	Married	80K-120K	Blue	36	1	0	3418.0
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than 540K	Blue	34	4	1	3313.0
4	709106358	Existing Customer	40	M	3	Uneducated	Married	60K-80K	Blue	21	1	0	4716.0

5 rows x 21 columns

Ques-3: Check the shape of data

```
In [6]: df.shape

Out[6]: (10127, 21)
```

Ques-4: Check the missing values

```
In [7]: df.isnull().sum()

Out[7]:
```

CLIENTNUM	0
Attrition_Flag	0
Customer_Age	0
Gender	0
Dependent_count	0
Education_Level	0
Marital_Status	0
Income_Category	0
Card_Category	0
Months_on_book	0
Total_Relationship_Count	0
Months_Inactive_12_mon	0
Contacts_Count_12_mon	0
Credit_Limit	0
Total_Revolving_Bal	0
Avg_Open_To_Buy	0
Total_Amt_Chng_Q4_Q1	0
Total_Trans_Amt	0
Total_Trans_Ct	0
Total_Ct_Chng_Q4_Q1	0
Avg_Utilization_Ratio	0
dtype:	int64

Percentage of missing values

```
In [8]: df.isnull().sum()/len(df)

Out[8]:
```

CLIENTNUM	0.0
Attrition_Flag	0.0
Customer_Age	0.0
Gender	0.0
Dependent_count	0.0
Education_Level	0.0
Marital_Status	0.0
Income_Category	0.0
Card_Category	0.0
Months_on_book	0.0
Total_Relationship_Count	0.0
Months_Inactive_12_mon	0.0
Contacts_Count_12_mon	0.0
Credit_Limit	0.0
Total_Revolving_Bal	0.0
Avg_Open_To_Buy	0.0
Total_Amt_Chng_Q4_Q1	0.0
Total_Trans_Amt	0.0
Total_Trans_Ct	0.0
Total_Ct_Chng_Q4_Q1	0.0
Avg_Utilization_Ratio	0.0
dtype:	float64

Ques-5: Check the Duplicate values

```
In [9]: duplicate = df[df.duplicated()]

In [10]: duplicate

Out[10]:
```

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_book	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_Limit
--	-----------	----------------	--------------	--------	-----------------	-----------------	----------------	-----------------	---------------	----------------	------------------------	-----------------------	--------------

0 rows x 21 columns

Ques-6: Check the distribution

```
In [11]: mean = df['Customer_Age'].mean()

In [12]: mean

Out[12]: 46.32596938413745

In [13]: median = df['Customer_Age'].median()

In [14]: median

Out[14]: 46.0

In [15]: std = df['Customer_Age'].std()

In [16]: std

Out[16]: 8.016814932549028
```

Ques-7: Plot a BOXPLOT

```
In [17]: ax = df1.boxplot(column='Customer_Age',by='Gender',figsize=(8,4))

Out[17]:
```

Text(0, 0.5, 'Customer_Age')



Plot Piechart

```
In [18]: ax=df1['Customer_Age']
az=df1['Gender']

In [19]: plt.pie(ax,labels = az,autopct='%1.1f%%',shadow=True)

Out[19]:
```

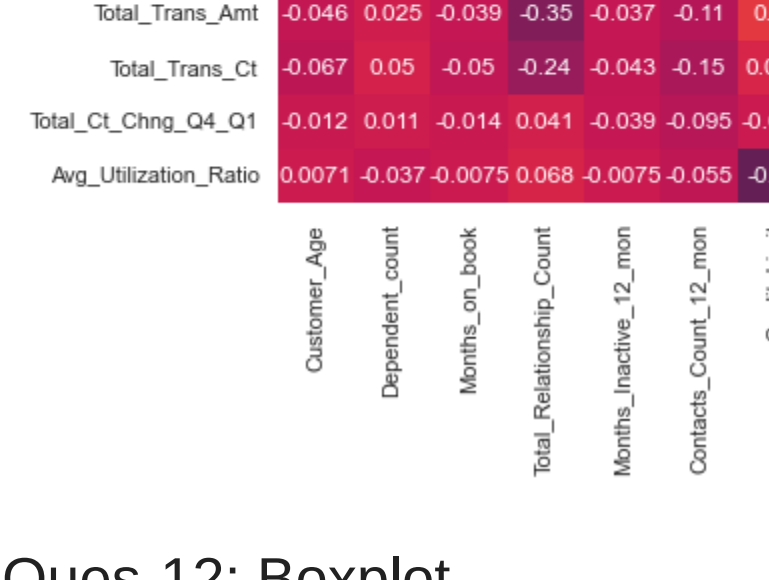


Ques-8: Plot a Boxplot

```
In [20]: ab = df1.boxplot(column='Total_Revolving_Bal',by='Attrition_Flag',figsize=(8,4))

Out[20]:
```

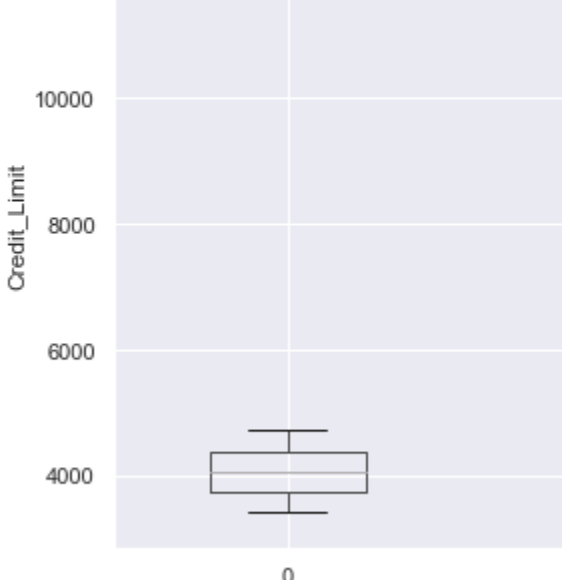
Text(0.5, 1.0, 'Attrition_Flag')



```
In [21]: sns.boxplot(x = df1['Card_Category'],
y = df1['Total_Revolving_Bal'],
hue = df1['Attrition_Flag'])

Out[21]:
```

<AxesSubplot:xlabel='Total_Revolving_Bal', ylabel='Card_Category'>




Ques-11: Plot Heatmap

```
In [22]: plt.figure(figsize=(12, 7))

sns.heatmap(df.drop(['CLIENTNUM','Attrition_Flag'],axis=1).corr(), annot = True, vmin = -1, vmax = 1)

plt.show()
```



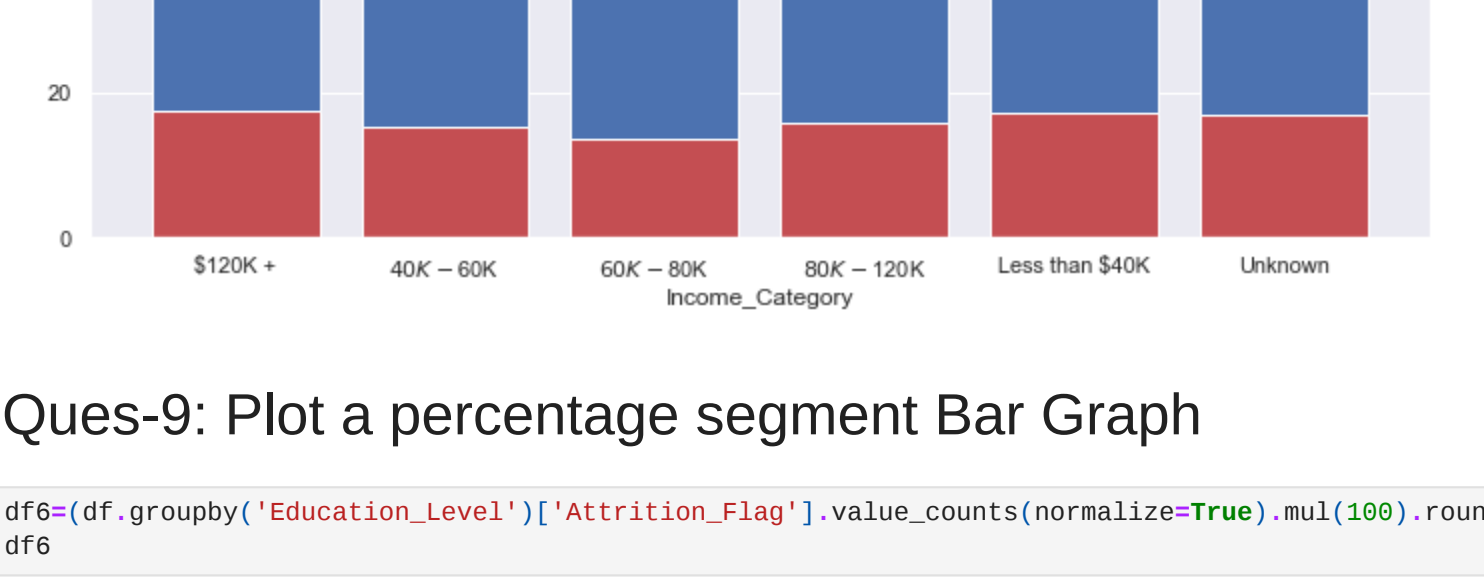
Ques-12: Boxplot

```
In [23]: ax = df1.boxplot(column='Credit_Limit', by='Total_Revolving_Bal', figsize=(12,6))

ax.set_ylabel('Credit_Limit')
ax.set_title('ffff')

Out[23]:
```

Text(0.5, 1.0, 'ffff')



Ques-10: Plot a percentage segment Bar Graph

```
In [24]: df9=(df.groupby('Income_Category')['Attrition_Flag'].value_counts(normalize=True).mul(100).round(2).unstack())

df9

Out[24]:
```

Attrition_Flag	Attrited Customer	Existing Customer
Income_Category		
\$120K +	17.33	82.67
40K-60K	15.14	84.86
60K-80K	13.48	86.52
80K-120K	15.77	84.23
Less than 540K	17.19	82.81
Unknown	16.62	83.38

```
In [25]: plt.figure(figsize=(12, 7))

plt.bar(df9.index, df9['Attrited Customer'],color='r')
plt.bar(df9.index, df9['Existing Customer'],bottom=df9['Attrited Customer'], color='b')
plt.title('Students over 11 Years')
plt.xlabel('Income_Category')
plt.ylabel('Attrition_Flag')
plt.legend(['Attrited Customer','Existing Customer'])

# Show the plot
plt.show()
```



Ques-9: Plot a percentage segment Bar Graph

```
In [26]: df6=(df.groupby('Education_Level')['Attrition_Flag'].value_counts(normalize=True).mul(100).round(2).unstack())

df6

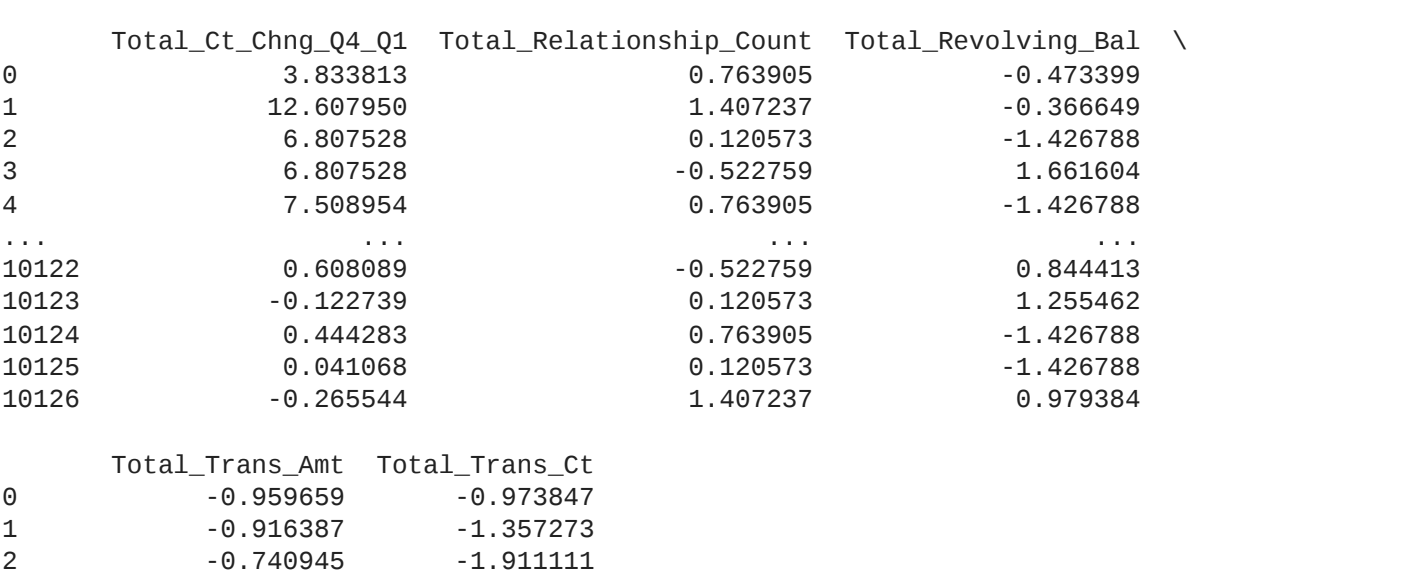
Out[26]:
```

Attrition_Flag	Attrited Customer	Existing Customer
Education_Level		
College	15.20	84.80
Doctorate	21.06	78.94
Graduate	15.57	84.43
High School	15.20	84.80
Post-Graduate	17.83	82.17
Uneducated	15.94	84.06
Unknown	16.85	83.15

```
In [27]: # Plot the data using bar() method
plt.figure(figsize=(12, 7))

plt.bar(df6.index, df6['Attrited Customer'],color='r')
plt.bar(df6.index, df6['Existing Customer'],bottom=df6['Attrited Customer'], color='b')
plt.title('Students over 11 Years')
plt.xlabel('Education_Level')
plt.ylabel('Number of Students')
plt.legend(['Attrited Customer','Existing Customer'])

# Show the plot
plt.show()
```



Ques-13: Map the Attrition_Flag values to 0 and 1

```
In [28]: df['Attrition_Flag'] = df['Attrition_Flag'].map({'Existing Customer': 0, 'Attrited Customer': 1})

df_standardized = (df - df.mean()) / df.std()

print(df_standardized)
```