

```
In [71]: 1 import pandas as pd
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4
```

```
In [72]: 1 df = pd.read_csv('Bank customers.csv')
```

```
In [73]: 1 df.head()
```

Out[73]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Categ
0	768805383	Existing Customer	45	M	3	High School	Married	60K–
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than \$
2	713982108	Existing Customer	51	M	3	Graduate	Married	80K–1.
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than \$
4	709106358	Existing Customer	40	M	3	Uneducated	Married	60K–

```
In [74]: 1 pd.set_option('display.max_columns', None)
2
```

```
In [75]: 1 df.head()
```

Out[75]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Categ
0	768805383	Existing Customer	45	M	3	High School	Married	60K–
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than \$
2	713982108	Existing Customer	51	M	3	Graduate	Married	80K–1.
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than \$
4	709106358	Existing Customer	40	M	3	Uneducated	Married	60K–

START MANIPULATION

```
In [76]: 1 ## CHANGE SOME COLUMN NAMES FOR EASY UNDERSTANDING
```

```
In [77]: 1 df.columns.tolist()
```

```
Out[77]: ['CLIENTNUM',  
          'Attrition_Flag',  
          'Customer_Age',  
          'Gender',  
          'Dependent_count',  
          'Education_Level',  
          'Marital_Status',  
          'Income_Category',  
          'Card_Category',  
          'Months_on_book',  
          'Total_Relationship_Count',  
          'Months_Inactive_12_mon',  
          'Contacts_Count_12_mon',  
          'Credit_Limit',  
          'Total_Revolving_Bal',  
          'Avg_Open_To_Buy',  
          'Total_Amt_Chng_Q4_Q1',  
          'Total_Trans_Amt',  
          'Total_Trans_Ct',  
          'Total_Ct_Chng_Q4_Q1',  
          'Avg_Utilization_Ratio']
```

```
In [78]: 1 df.rename(columns={'Attrition_Flag': 'Attrition_Status'}, inplace=True)  
2 df.rename(columns={'CLIENTNUM': 'Client_Number'}, inplace=True)  
3 df.rename(columns={'Income_Category': 'Income_Per_Annum'}, inplace=True)  
4 df.rename(columns={'Months_Inactive_12_mon': 'Months_Inactive'}, inplace=True)  
5 df.rename(columns={'Contacts_Count_12_mon': 'Contacts_Count'}, inplace=True)  
6
```

```
In [79]: 1 df.columns
```

```
Out[79]: Index(['Client_Number', 'Attrition_Status', 'Customer_Age', 'Gender',  
               'Dependent_count', 'Education_Level', 'Marital_Status',  
               'Income_Per_Annum', 'Card_Category', 'Months_on_book',  
               'Total_Relationship_Count', 'Months_Inactive', 'Contacts_Count',  
               'Credit_Limit', 'Total_Revolving_Bal', 'Avg_Open_To_Buy',  
               'Total_Amt_Chng_Q4_Q1', 'Total_Trans_Amt', 'Total_Trans_Ct',  
               'Total_Ct_Chng_Q4_Q1', 'Avg_Utilization_Ratio'],  
              dtype='object')
```

```
In [80]: 1 ## DROP SOME UNNECESSARY COLUMNS THAT WILL
```

```
In [81]: 1 columns_to_drop = ['Avg_Utilization_Ratio', 'Total_Ct_Chng_Q4_Q1', 'Total_Amt_Chng_Q4_Q1']
```

```
In [82]: 1 df.drop(columns=columns_to_drop, inplace=True)
```

```
In [83]: 1 df.columns
```

```
Out[83]: Index(['Client_Number', 'Attrition_Status', 'Customer_Age', 'Gender',  
               'Dependent_count', 'Education_Level', 'Marital_Status',  
               'Income_Per_Annum', 'Card_Category', 'Months_on_book',  
               'Total_Relationship_Count', 'Months_Inactive', 'Contacts_Count',  
               'Credit_Limit', 'Total_Revolving_Bal', 'Avg_Open_To_Buy',  
               'Total_Trans_Amt', 'Total_Trans_Ct'],  
              dtype='object')
```

```
In [84]: 1 df.shape
```

```
Out[84]: (10127, 18)
```

In [85]: 1 df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10127 entries, 0 to 10126
Data columns (total 18 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                ---
0   Client_Number                        10127 non-null  int64
1   Attrition_Status                    10127 non-null  object
2   Customer_Age                       10127 non-null  int64
3   Gender                             10127 non-null  object
4   Dependent_count                    10127 non-null  int64
5   Education_Level                    10127 non-null  object
6   Marital_Status                    10127 non-null  object
7   Income_Per_Annum                   10127 non-null  object
8   Card_Category                     10127 non-null  object
9   Months_on_book                     10127 non-null  int64
10  Total_Relationship_Count            10127 non-null  int64
11  Months_Inactive                     10127 non-null  int64
12  Contacts_Count                     10127 non-null  int64
13  Credit_Limit                       10127 non-null  float64
14  Total_Revolving_Bal                10127 non-null  int64
15  Avg_Open_To_Buy                    10127 non-null  float64
16  Total_Trans_Amt                    10127 non-null  int64
17  Total_Trans_Ct                     10127 non-null  int64
dtypes: float64(2), int64(10), object(6)
memory usage: 1.4+ MB
```

In [86]: 1 ## FROM THIS, WE CAN SEE THAT THERE ARE NO NULL VALUES AND ALL THE COLUMNS ARE IN THEIR RIGHT

In [87]: 1 df.head()

Out[87]:

	Client_Number	Attrition_Status	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_F
0	768805383	Existing Customer	45	M	3	High School	Married	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Les:
2	713982108	Existing Customer	51	M	3	Graduate	Married	{
3	769911858	Existing Customer	40	F	4	High School	Unknown	Les:
4	709106358	Existing Customer	40	M	3	Uneducated	Married	

START EDA

In [88]: 1 df.Attrition_Status.unique()

Out[88]: array(['Existing Customer', 'Attrited Customer'], dtype=object)

In [89]: 1 Attrited = df[df.Attrition_Status == 'Attrited Customer']

In [90]: 1 Attrited.shape

Out[90]: (1627, 18)

```
In [91]: 1 Existing = df[df.Attrition_Status == 'Existing Customer']
```

```
In [92]: 1 Existing.shape
```

```
Out[92]: (8500, 18)
```

```
In [93]: 1 # USING INCOME_PER_ANNUM AS A METRIC TO KNOW WHAT COULD CAUSE THE CUSTOMERS ATTRITION
```

```
In [94]: 1 df.Income_Per_Annum.unique()
```

```
Out[94]: array(['$60K - $80K', 'Less than $40K', '$80K - $120K', '$40K - $60K',  
               '$120K +', 'Unknown'], dtype=object)
```

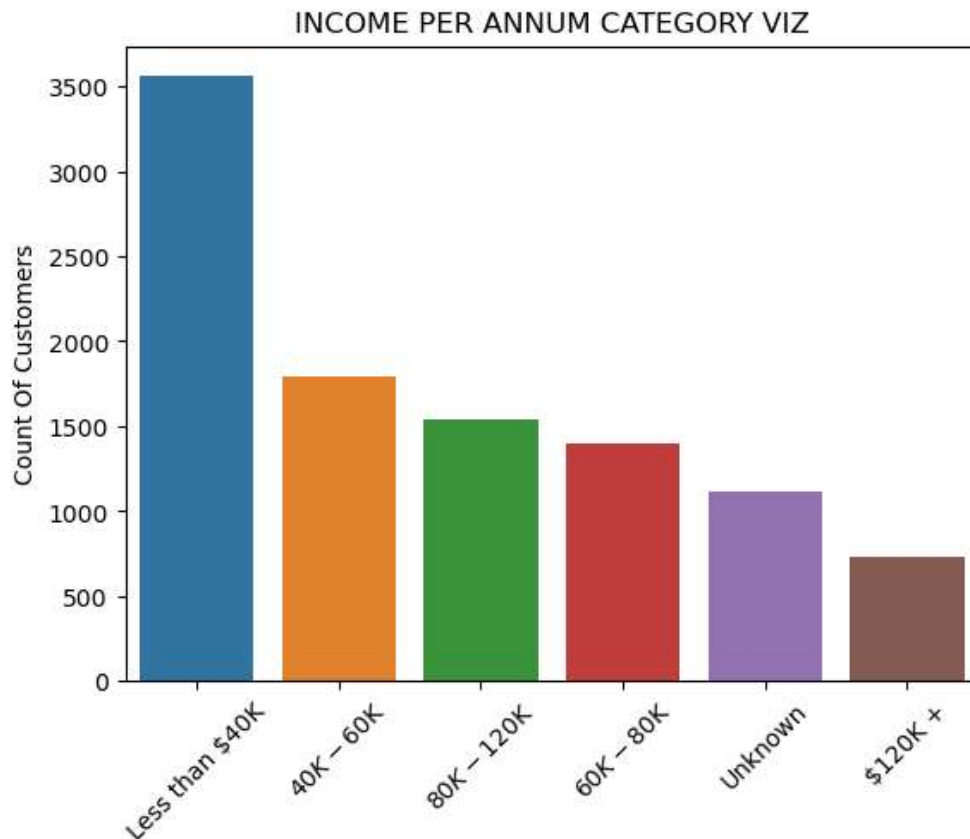
```
In [95]: 1 value_counts = df.Income_Per_Annum.value_counts()
```

```
In [96]: 1 df.Income_Per_Annum.value_counts()
```

```
Out[96]: Less than $40K      3561  
         $40K - $60K       1790  
         $80K - $120K     1535  
         $60K - $80K       1402  
         Unknown          1112  
         $120K +           727  
         Name: Income_Per_Annum, dtype: int64
```

```
In [97]: 1 # WE CAN SEE THAT MOST OF THE CUSTOMERS EARN LESS THAN $40,000 PER ANNUM, WE HAVE TO PREPARE
```

```
In [98]: 1 sns.barplot(x=value_counts.index, y=value_counts.values)
2 plt.title('INCOME PER ANNUM CATEGORY VIZ')
3 plt.ylabel('Count Of Customers')
4 plt.xticks(rotation=45)
5 plt.show()
```



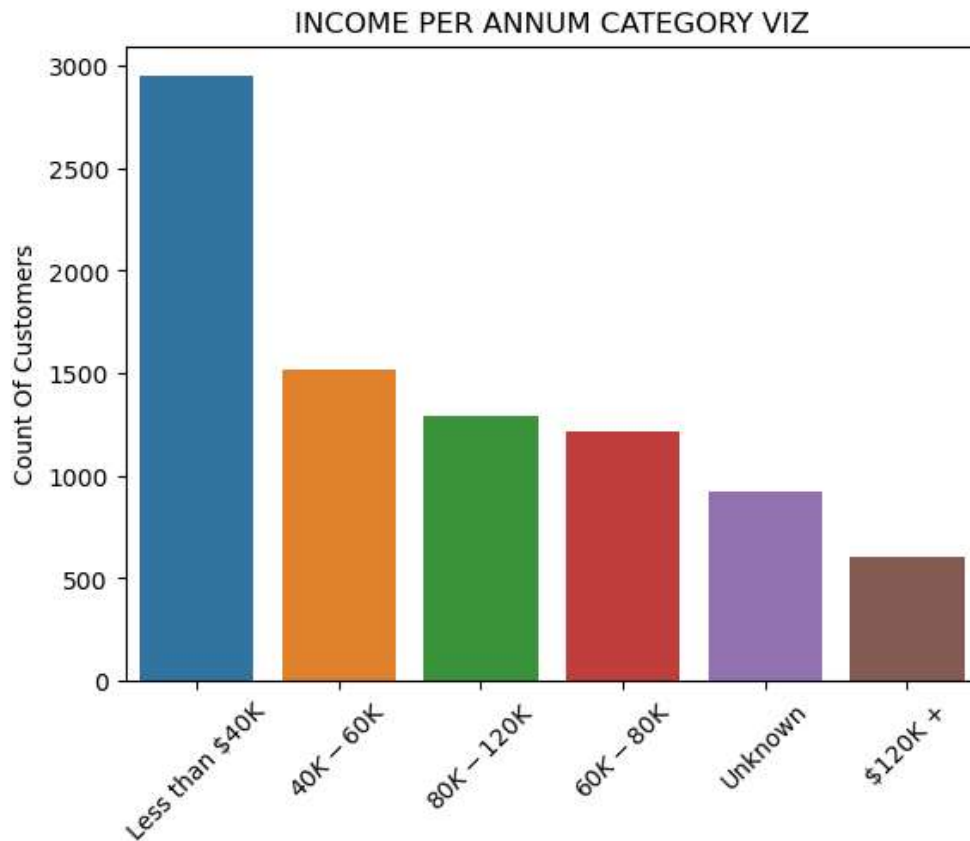
```
In [99]: 1 # AFTER CHECKING THE INCOME CATEGORY OF THE MAIN DATA FRAME, WE CAN SEE THAT MOST OF OUR CUS
2 # HOWEVER, WE HAVE TO CHECK THE INCOME CATEGORY OF OUR EXISTING AND ATTRITED CUSTOMERS TO GE
```

```
In [100]: 1 Existing.Income_Per_Annum.value_counts()
```

```
Out[100]: Less than $40K    2949
$40K - $60K    1519
$80K - $120K    1293
$60K - $80K    1213
Unknown        925
$120K +        601
Name: Income_Per_Annum, dtype: int64
```

```
In [101]: 1 val_counts = Existing.Income_Per_Annum.value_counts()
```

```
In [102]: 1 sns.barplot(x=val_counts.index, y=val_counts.values)
2 plt.title('INCOME PER ANNUM CATEGORY VIZ')
3 plt.ylabel('Count Of Customers')
4 plt.xticks(rotation=45)
5 plt.show()
```

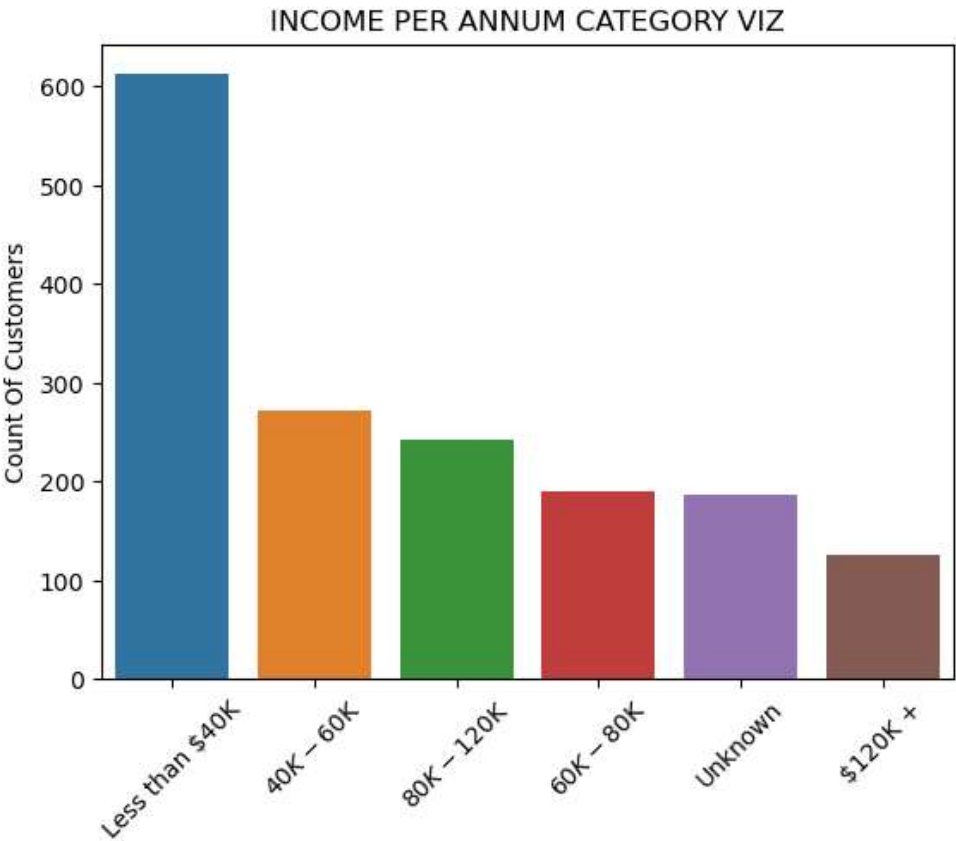


```
In [103]: 1 Attrited.Income_Per_Annum.value_counts()
```

```
Out[103]: Less than $40K      612
$40K - $60K      271
$80K - $120K     242
$60K - $80K      189
Unknown          187
$120K +          126
Name: Income_Per_Annum, dtype: int64
```

```
In [104]: 1 v_counts = Attrited.Income_Per_Annum.value_counts()
```

```
In [105]: 1 sns.barplot(x=v_counts.index, y=v_counts.values)
2 plt.title('INCOME PER ANNUM CATEGORY VIZ')
3 plt.ylabel('Count Of Customers')
4 plt.xticks(rotation=45)
5 plt.show()
```



```
In [106]: 1 # WE CAN SEE FROM THE ABOVE THAT BOTH OUR EXISTING AND ATTRITED CUSTOMERS MAKE LESS THAN $40K
2 # WHAT THIS MEANS IS THAT THEIR ANNUAL INCOME HAS NOTHING TO DO WITH WHAT COULD MAKE SOME CU
```

```
In [107]: 1 df.head()
```

Out[107]:

	Client_Number	Attrition_Status	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_F
0	768805383	Existing Customer	45	M	3	High School	Married	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Les
2	713982108	Existing Customer	51	M	3	Graduate	Married	{
3	769911858	Existing Customer	40	F	4	High School	Unknown	Les
4	709106358	Existing Customer	40	M	3	Uneducated	Married	

```
In [108]: 1 # NEXT WE HAVE TO EXPLORE THE AGE BRACKET OF OUR CUSTOMERS
```

```
In [109]: 1 df_age = df.Customer_Age.value_counts().sort_index()
```

```
In [110]: 1 # CHECK FOR THE EXISTING CUSTOMERS
```

```
In [111]: 1 Ex_age = Existing.Customer_Age.value_counts().sort_index()
```

```
In [112]: 1 Attrited.Customer_Age.value_counts().sort_index()
```

```
Out[112]: 26      6
          27      3
          28      1
          29      7
          30     15
          31     13
          32     17
          33     20
          34     19
          35     21
          36     24
          37     37
          38     47
          39     48
          40     64
          41     76
          42     62
          43     85
          44     84
          45     79
          46     82
          47     76
          48     85
          49     79
          50     71
          51     58
          52     58
          53     59
          54     69
          55     51
          56     43
          57     33
          58     24
          59     40
          60     13
          61     17
          62     17
          63      8
          64      5
          65      9
          66      1
          68      1
          Name: Customer_Age, dtype: int64
```

```
In [113]: 1 # FROM THIS WE CAN SEE THAT CUSTOMERS THAT CHURNED THE BANK ARE BETWEEN THE AGE BRACKET OF 26-68
          2 # WHAT THIS MEANS IS THESE INDIVIDUALS ARE MATURED AND POSSIBLY WITH A FAMILY AND MOST OF WH.
```


In [114]: 1 df.head()

Out[114]:

	Client_Number	Attrition_Status	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_F
0	768805383	Existing Customer	45	M	3	High School	Married	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Les:
2	713982108	Existing Customer	51	M	3	Graduate	Married	{
3	769911858	Existing Customer	40	F	4	High School	Unknown	Les:
4	709106358	Existing Customer	40	M	3	Uneducated	Married	

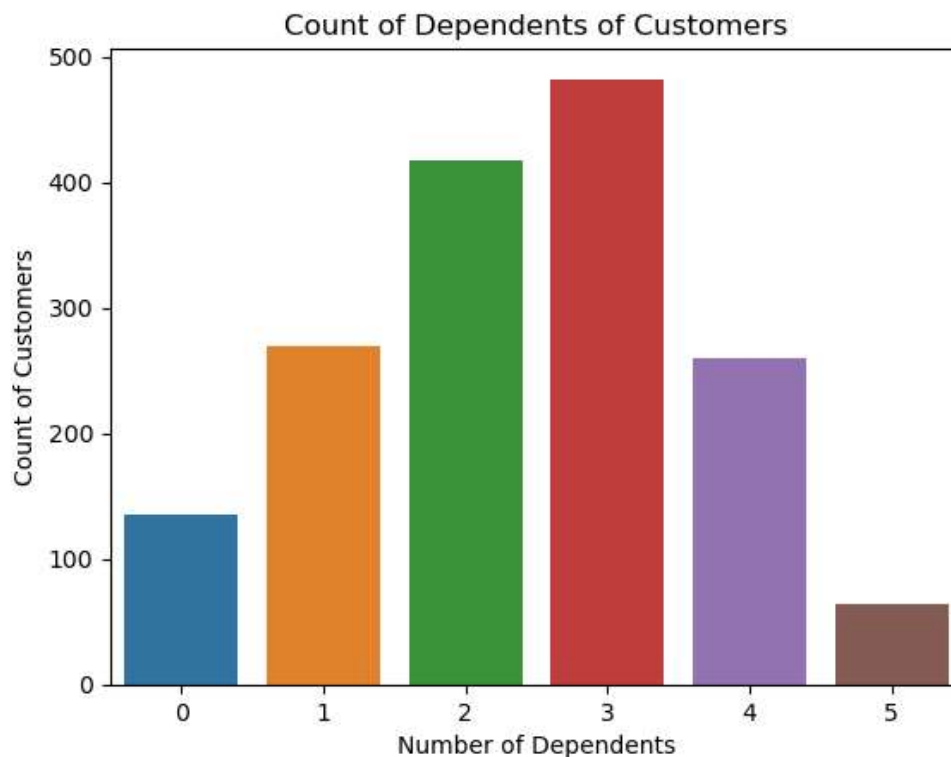
In [115]: 1 # Next is to know the level of dependents the attrited customers have.

In [116]: 1 Attrited.Dependent_count.unique()

Out[116]: array([0, 1, 2, 4, 5, 3], dtype=int64)

In [117]: 1 att = Attrited.Dependent_count.value_counts()

In [118]: 1 sns.barplot(x= att.index, y = att.values)
 2 plt.xlabel('Number of Dependents')
 3 plt.ylabel('Count of Customers')
 4 plt.title('Count of Dependents of Customers')
 5 plt.show()



```
In [119]: 1 # From this we can see that most of the customers that attrited have 3 or 2 dependents that
          2 # What this means is that these customers have other individuals they take care of asides th
          3 # They may have to spend more money on these dependents than would have been if they had none
```

```
In [120]: 1 df.head()
```

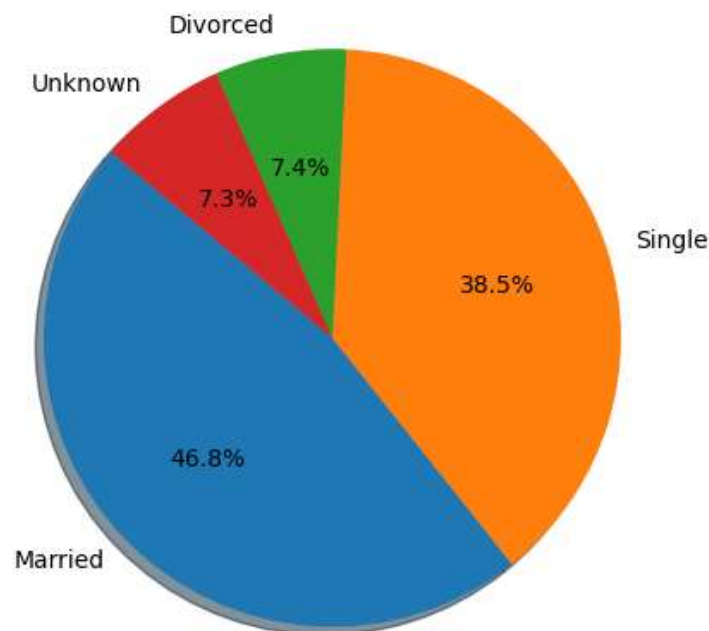
```
Out[120]:
```

	Client_Number	Attrition_Status	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_F
0	768805383	Existing Customer	45	M	3	High School	Married	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Les
2	713982108	Existing Customer	51	M	3	Graduate	Married	
3	769911858	Existing Customer	40	F	4	High School	Unknown	Les
4	709106358	Existing Customer	40	M	3	Uneducated	Married	

```
In [121]: 1 # MAKE AN ANALYSIS ON THE MARITAL STATUS OF BOTH THE EXISTING CUSTOMERS AND THE ATTRITED CUS
          2 # WE WILL DO THIS TO KNOW IF THERE ARE SOME SORT OF INCONSISTENCY BETWEEN BOTH METRICS
```

```
In [122]: 1 ex = Existing.Marital_Status.value_counts()
```

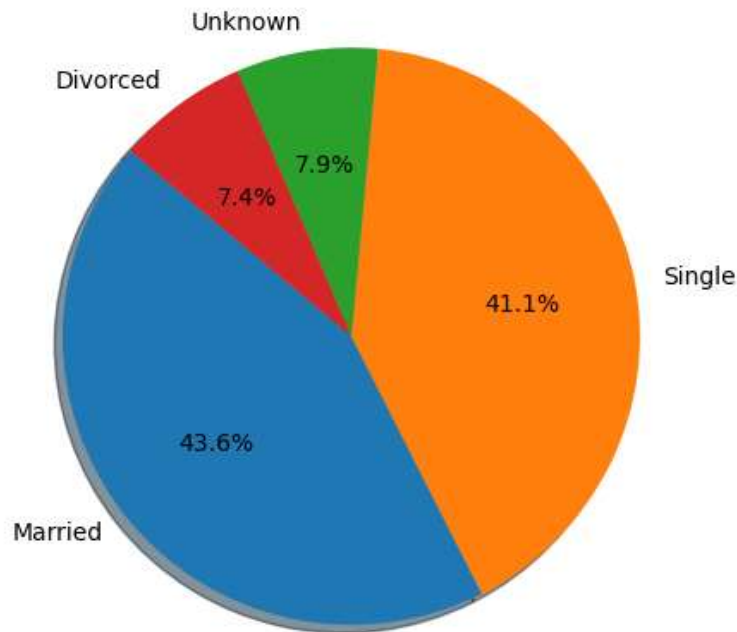
```
In [123]: 1 plt.pie(ex, labels=ex.index, autopct='%1.1f%%', shadow=True, startangle=140)
          2 plt.axis('equal')
          3 plt.show()
          4
```



```
In [124]: 1 # NOW THE ANALYSIS FOR THE ATTRITED CUSTOMERS
```

```
In [125]: 1 vc = Attrited.Marital_Status.value_counts()
```

```
In [126]: 1 plt.pie(vc, labels=vc.index, autopct='%1.1f%%', shadow=True, startangle=140)
2 plt.axis('equal')
3 plt.show()
4
```



```
In [127]: 1 # JUDGING FROM THIS PIE CHART DISTRIBUTION, WE CAN SEE THAT THE CUSTOMERS THAT CHURNED THE B
2 # DISTRIBUTED 50/50 BECAUSE THE DIFFERENCE BETWEEN THE MARRIED AND SINGLE IS VERY SLIM, SO W
3 # USING THIS METRIC
4 # BESIDES, MOST OF THE CLIENTS THAT ARE EXISTING ARE MARRIED, SO WE CANT SAY MARRIAGE AFFECTE
```

```
In [128]: 1 df.head()
```

Out[128]:

	Client_Number	Attrition_Status	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_F
0	768805383	Existing Customer	45	M	3	High School	Married	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Les
2	713982108	Existing Customer	51	M	3	Graduate	Married	
3	769911858	Existing Customer	40	F	4	High School	Unknown	Les
4	709106358	Existing Customer	40	M	3	Uneducated	Married	

```
In [129]: 1 credit_report = Attrited[['Credit_Limit', 'Total_Revolving_Bal', 'Avg_Open_To_Buy', 'Total_T
```

```
In [130]: 1 pd.set_option('display.max_rows', None)
```



```
In [140]: 1 ## Most of our existing customers owe us too because of the 8500 existing customers, only 15
          2 ## Leaving 6923 oweing the bank and are still existing.
```

```
In [ ]: 1
```

THEORY OF ANALYSIS

- INITIALLY, WHAT WE ARE INTERESTED IN, OR THE BOTTOM LINE OF THIS ANALYSIS IS TO KNOW WHAT COULD BE A PRINCIPAL REASON AS TO WHY SOME OF OUR CUSTOMERS STOP USING OUR CREDIT CARD SERVICE.

-THERE ARE VARIOUS REASONS THAT CAN CAUSE SUCH DECISION TO CHURN THE BANK SOME OF WHICH COULD RANGE FROM SOCIAL REASONS TO TRANSACTIONAL REASONS (ECONOMICAL).

- WE KNOW FOR A FACT THAT THE ONLY LOGICAL REASON FOR THEM CHURNING THE BANK HAS TO DO WITH A SOCIOECONOMIC CAUSE, THEREFORE, WE HAVE TO MAKE A SOCIOECONOMICAL ANALYSIS TO FIND OUT WHY.

> FIRSTLY, JUST LIKE ANY GOOD ANALYSIS, THE DATA CLEANING HAS TO BE DONE.

> AN ANALYSIS WAS DONE ON THE INCOME PER ANNUM OF ALL CUSTOMERS AND WE FOUND OUT THAT MOST OF OUR CUSTOMERS MAKE LESS THAN 40,000 DOLLARS, THIS SAME ANALYSIS APPLIED TO BOTH EXISTING AND ATTRITED CUSTOMERS WHICH MEANS THEIR ANNUAL REVENUE HAS LITTLE OR NOTHING TO DO WITH THEIR DECISION TO CHURN.

> ANOTHER METRIC WE EXPLORED WAS TO KNOW THE AGE BRACKET OF THE CUSTOMERS THAT ATTRITED AND WE FOUND OUT THAT MOST OF THEM ARE BETWEEN THE AGE OF 37-59, THIS GIVES US AN INSIGHT THAT THESE ARE MATURED INDIVIDUALS WHO MOST LIKELY HAVE A FAMILY THEY TAKE CARE OF WHICH BREEDS A NEED TO EXTENSIVELY MAKE USE OF THEIR CREDIT CARDS.

> SPEAKING OF FAMILIES, WE MADE AN ANALYSIS TO KNOW THE RATIO OF CUSTOMERS WITH FAMILIES AND ONES THAT ARE STILL SINGLE AND ALSO THEIR NUMBER OF DEPENDENTS, THE RESULT CAME OUT AS MOST OF OUR EXISTING CUSTOMERS ARE MARRIED AND WITH 2-3 DEPENDENTS WHILE CUSTOMERS THAT ATTRITED HAVE THE DISTRIBUTION AT 50/50 WITH SOME OF THEM HAVING 3 DEPENDENTS OR 2-3, WHAT THIS MEANS IS THAT WE CAN NOT USE THEIR SOCIAL LIFE AS A PRINCIPAL REASON FOR THE ATTRITED CUSTOMERS CHURING THE CREDIT CARD SERVICE. WE ONLY KNOW THEY HAVE A REASON TO REALLY USE THE SERVICE.

> WHAT WE ARE LEFT WITH IS TO ENGAGE IN A TRANSACTIONAL ANALYSIS WHICH BRINGS US TO THE ECONOMICAL PART OF THE ANALYSIS. THE INSIGHT WE BROUGHT OUT IS THAT OF THE 1600+ CUSTOMERS THAT CHURNED THE BANK CREDIT CARD SERVICE, 890+ OF THEM DO NOT OWE THE ORGANIZATION AS THEIR TOTAL REVOLVING BALANCE OR CURRENT ACCOUNT INDICATES 0 WHICH MEANS A LITTLE MORE THAN HALF OF THEM DO NOT OWE THE BANK, HOWEVER, OF OUR EXISTING CUSTOMERS OR THE ACTIVE CUSTOMERS, JUST 1700+ OF THE 8500 OF THEM DO NOT OWE THE ORGANIZATION BECAUSE THEIR TOTAL REVOLVING BALANCE OR CURRENT BALANCE INDICATES SO, WHAT THIS MEANS PRIMARILY IS THAT, OUR EXISTING CUSTOMERS ARE STILL WITH US BECAUSE THEY ARE STILL ACTIVELY INVOLVED IN USING THE CREDIT CARD TO NAVIGATE THEIR LIVES AND THE ONES THAT LEFT ARE NOT REALLY "A FAN" OF USING CREDIT CARDS POSSIBLY BECAUSE THEY DO NOT WANT TO ENTERTAIN OWING THE BANK AND THE ONES THAT EVEN OWE THE BANK LEFT BECAUSE THEY OWE THE BANK.

SOLUTION: THE PRIMARY SOLUTION IS TO REORIENTATE OUR CUSTOMERS ON THE NEED TO ONLY USE THEIR CREDIT CARDS WHEN EXTREMELY NECESSARY AND ALSO TO PAY UP ON THEIR CURRENT BALANCE. REORIENTATION IS A VERY VITAL TECHNIC THAT WORKS PSYCHOLOGICALLY.

ALSO THERE IS NEED TO CONTACT OUR ATTRITED CUSTOMERS AND REMIND THEM OF HOW WELL WE ARE WILLING TO SERVE THEM MORE IF THEY CHOOSE TO COME BACK TO USING OUR SERVICE, WE ALSO HAVE TO OFFER INCENTIVES THAT WILL BE OF VALUE TO BOTH PARTIES. IF THESE CHURNING CUSTOMERS ARE BACK IN THE SYSTEM, OUR REORIENTATION TECHNIC WILL ALSO PROPEL THEM TO MAKE VALUEABLE CHOICES AND ALSO PAY UP THEIR CURRENT BALANCE.

ANALYSIS BY: GABRIEL STEVEN OLUWAPELUMI

