

## Importing Packages and Files

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib
```

```
In [2]: Bank_Churners_infile = 'C:/Users/matth/OneDrive/Documents/Grad School/Data Analytics/Spr
Churn_Train_infile = 'C:/Users/matth/OneDrive/Documents/Grad School/Data Analytics/Spr
Churn_Test_infile = 'C:/Users/matth/OneDrive/Documents/Grad School/Data Analytics/Spr

Bank_Churners = pd.read_csv(Bank_Churners_infile)
Churn_Train = pd.read_csv(Churn_Train_infile)
Churn_Test = pd.read_csv(Churn_Test_infile)
```

## Data Cleaning and Preparation of Test and Train Files

```
In [3]: Churn_Train.columns[0:5]
```

```
Out[3]: Index(['CLIENTNUM', 'Attrition_Flag', 'Customer_Age', 'Gender',
              'Dependent_count'],
              dtype='object')
```

```
In [4]: Churn_Train.shape
```

```
Out[4]: (8102, 21)
```

```
In [5]: Churn_Test.shape
```

```
Out[5]: (2025, 21)
```

```
In [6]: Churn_Train.set_index('CLIENTNUM', inplace = True)
```

```
In [7]: Churn_Train.head()
```

Out[7]:

	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status
CLIENTNUM						
805546308	Existing Customer	65	F	0	High School	Single
717656433	Existing Customer	50	F	2	Graduate	Single
713274183	Existing Customer	45	F	1	Graduate	Single
778819083	Attrited Customer	41	F	2	Doctorate	Married
716399583	Existing Customer	34	M	1	Uneducated	Married

In [8]: Churn\_Test.set\_index('CLIENTNUM', inplace = True)

In [9]: Churn\_Test.head()

Out[9]:

	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status
CLIENTNUM						
709106358	Existing Customer	40	M	3	Uneducated	Married
713061558	Existing Customer	44	M	2	Graduate	Married
709967358	Existing Customer	48	M	4	Post-Graduate	Single
806160108	Existing Customer	61	M	1	High School	Married
708508758	Attrited Customer	62	F	0	Graduate	Married

## Exploratory Analysis

In [10]: *# Minumim Customer Age*  
Bank\_Churners['Customer\_Age'].min()

Out[10]: 26

In [11]: *# Maximum Customer Age*  
Bank\_Churners['Customer\_Age'].max()

Out[11]: 73

```
In [12]: # Lowest Credit Limit Among Clients
Bank_Churners['Credit_Limit'].min()
```

```
Out[12]: 1438.3
```

```
In [13]: # Highest Credit Limit Among Clients
Bank_Churners['Credit_Limit'].max()
```

```
Out[13]: 34516.0
```

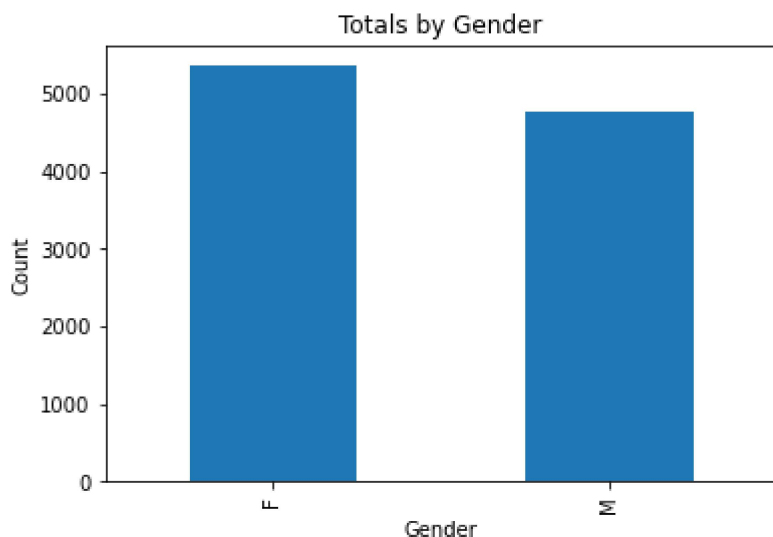
```
In [14]: # Different Card Categories
Bank_Churners['Card_Category'].unique()
```

```
Out[14]: array(['Blue', 'Gold', 'Silver', 'Platinum'], dtype=object)
```

## Single Variable Analysis

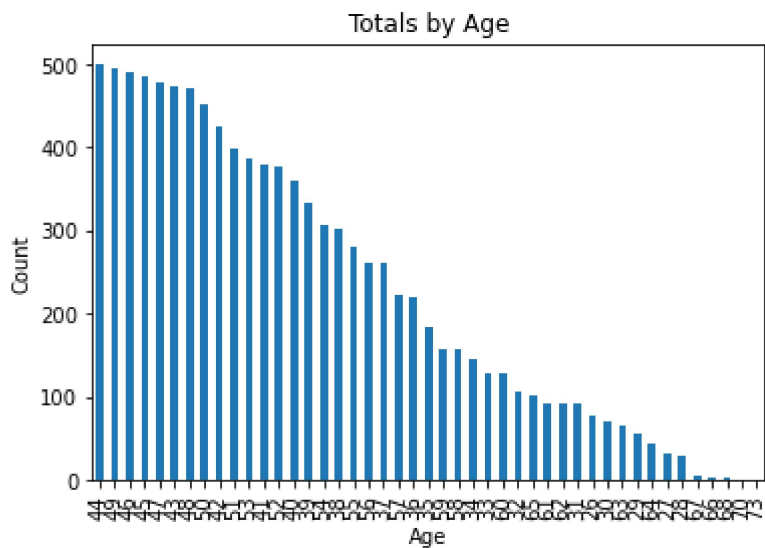
```
In [15]: Bank_Churners['Gender'].value_counts().plot(kind='bar')
plt.xlabel("Gender")
plt.ylabel("Count")
plt.title("Totals by Gender")
```

```
Out[15]: Text(0.5, 1.0, 'Totals by Gender')
```



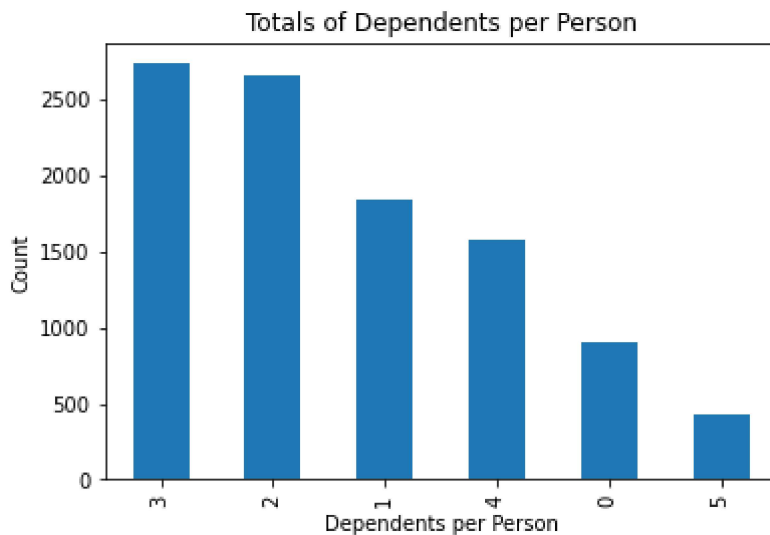
```
In [16]: Bank_Churners['Customer_Age'].value_counts().plot(kind="bar")
plt.xlabel("Age")
plt.ylabel("Count")
plt.title("Totals by Age")
```

```
Out[16]: Text(0.5, 1.0, 'Totals by Age')
```



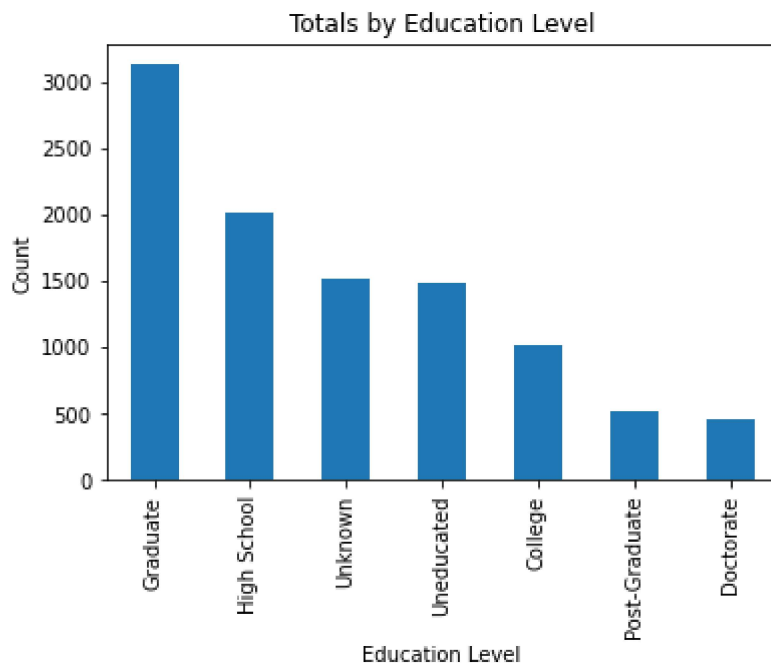
```
In [17]: Bank_Churners['Dependent_count'].value_counts().plot(kind="bar")
plt.xlabel("Dependents per Person")
plt.ylabel("Count")
plt.title("Totals of Dependents per Person")
```

```
Out[17]: Text(0.5, 1.0, 'Totals of Dependents per Person')
```



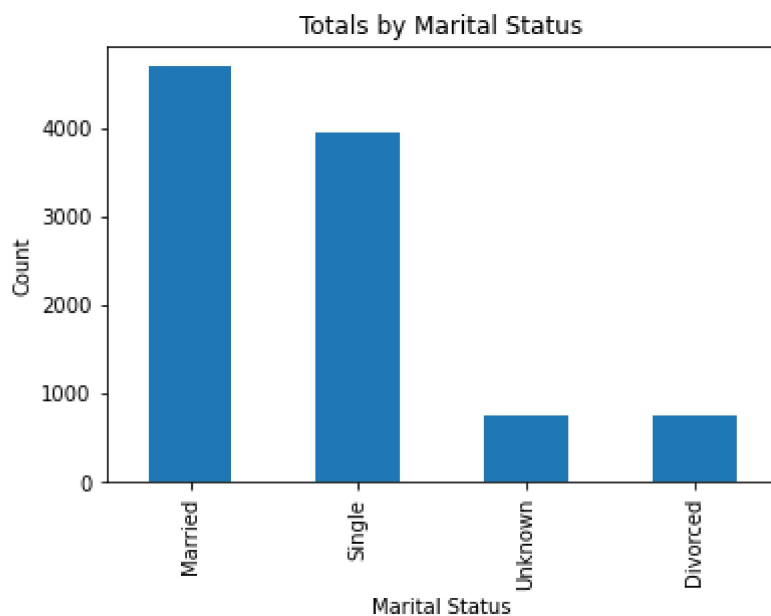
```
In [18]: Bank_Churners['Education_Level'].value_counts().plot(kind="bar")
plt.xlabel("Education Level")
plt.ylabel("Count")
plt.title("Totals by Education Level")
```

```
Out[18]: Text(0.5, 1.0, 'Totals by Education Level')
```



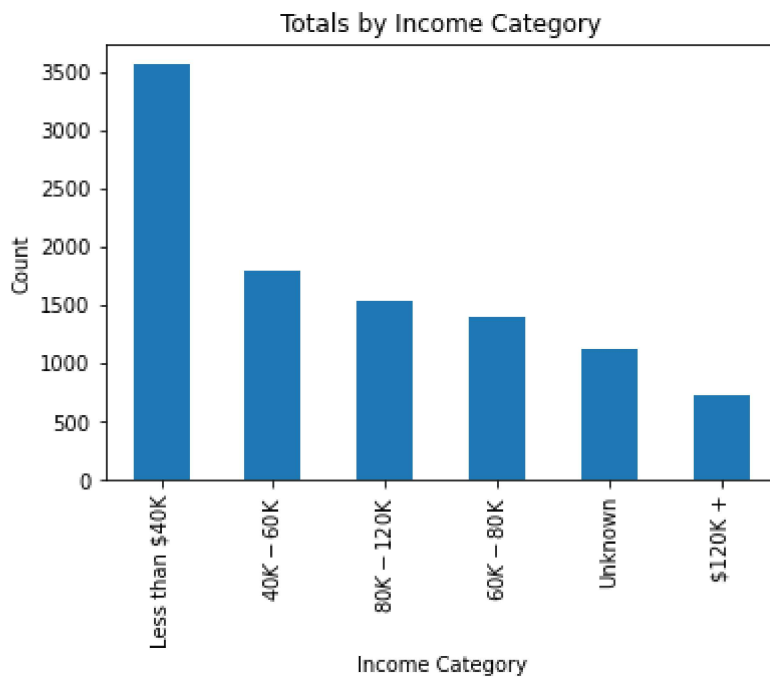
```
In [19]: Bank_Churners['Marital_Status'].value_counts().plot(kind="bar")
plt.xlabel("Marital Status")
plt.ylabel("Count")
plt.title("Totals by Marital Status")
```

```
Out[19]: Text(0.5, 1.0, 'Totals by Marital Status')
```



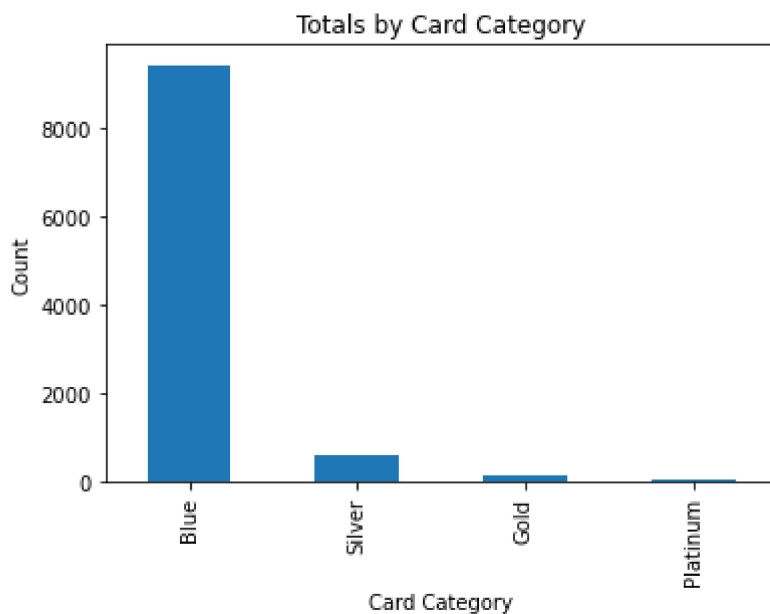
```
In [20]: Bank_Churners['Income_Category'].value_counts().plot(kind="bar")
plt.xlabel("Income Category")
plt.ylabel("Count")
plt.title("Totals by Income Category")
```

```
Out[20]: Text(0.5, 1.0, 'Totals by Income Category')
```



```
In [21]: Bank_Churners['Card_Category'].value_counts().plot(kind="bar")
plt.xlabel("Card Category")
plt.ylabel("Count")
plt.title("Totals by Card Category")
```

```
Out[21]: Text(0.5, 1.0, 'Totals by Card Category')
```



## Multivariable Analysis

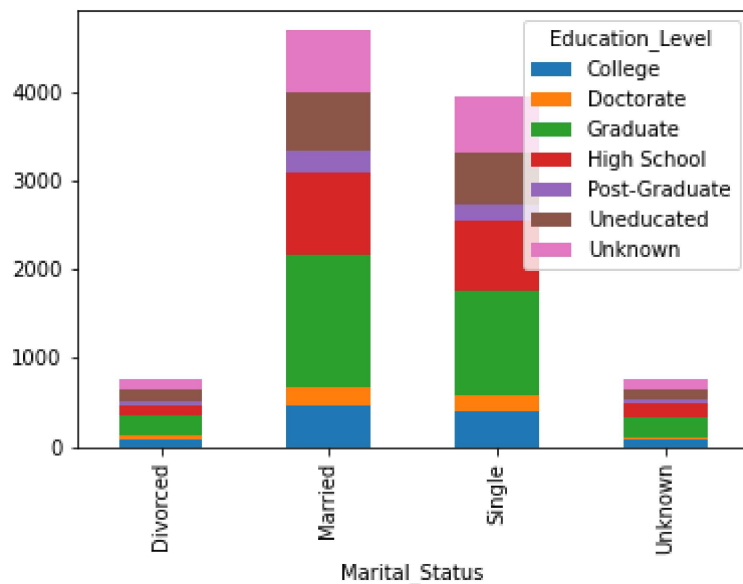
```
In [22]: agg_gender = Bank_Churners.groupby(['Marital_Status', 'Education_Level'])['Gender'].count()
agg_gender.head(5)
```

```
Out[22]:
```

Education_Level	College	Doctorate	Graduate	High School	Post-Graduate	Uneducated	Unknown
<b>Marital_Status</b>							
<b>Divorced</b>	86	36	225	128	41	136	96
<b>Married</b>	467	205	1479	949	243	656	688
<b>Single</b>	386	182	1197	782	189	586	621
<b>Unknown</b>	74	28	227	154	43	109	114

```
In [23]: agg_gender.plot(kind="bar",stacked=True)
```

```
Out[23]: <AxesSubplot:xlabel='Marital_Status'>
```

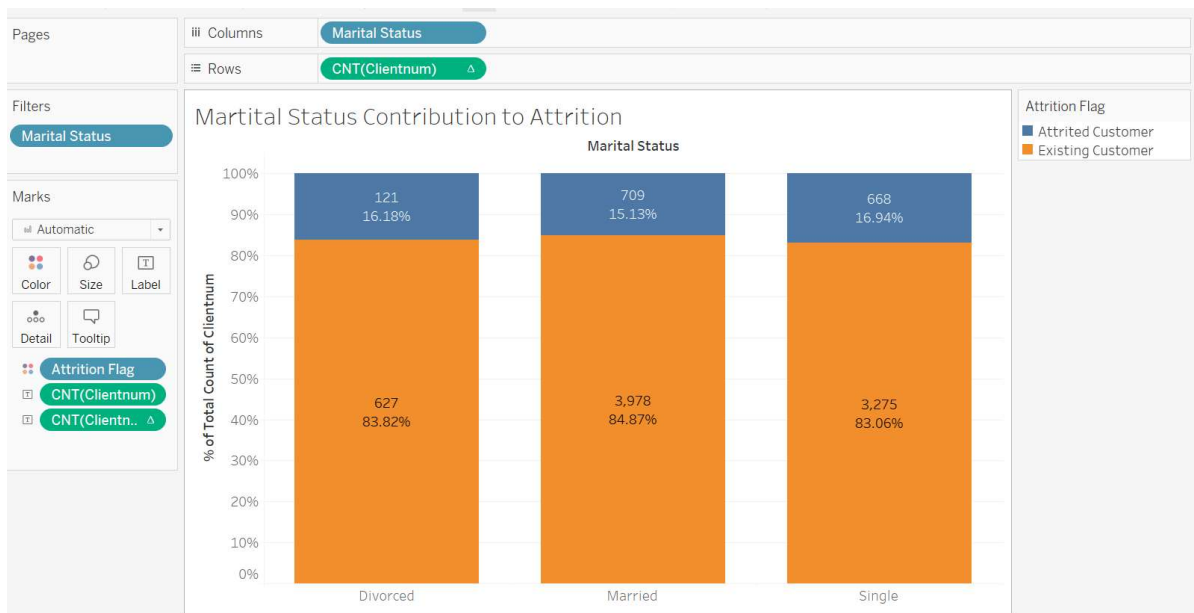


## Primary Drivers of Credit Card Churn

```
In [24]: # Churn due to Marital Status
marital_attrition = Bank_Churners.groupby(['Marital_Status','Attrition_Flag'])['Attri
marital_attrition.head(5)
```

```
Out[24]:
```

Attrition_Flag	Attrited Customer	Existing Customer
<b>Marital_Status</b>		
<b>Divorced</b>	121	627
<b>Married</b>	709	3978
<b>Single</b>	668	3275
<b>Unknown</b>	129	620

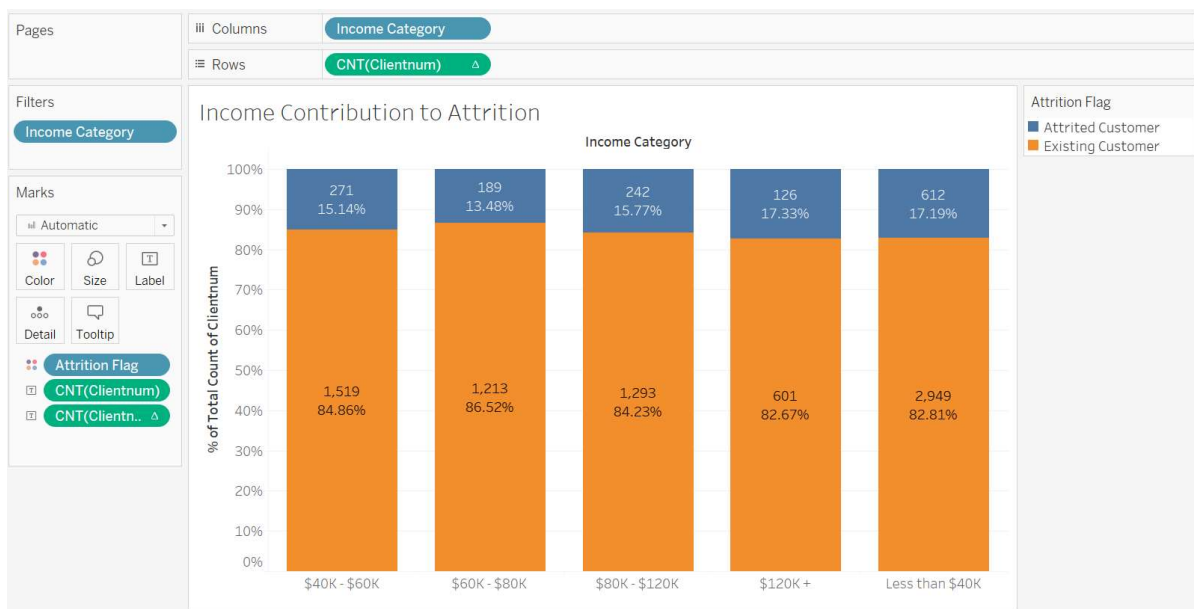


In [25]: *# Churn due to Income*

```
marital_income = Bank_Churners.groupby(['Income_Category', 'Attrition_Flag'])['Attrition_Flag'].value_counts().unstack()
marital_income.head(5)
```

Out[25]:

	Attrition_Flag	Attributed Customer	Existing Customer
Income_Category			
	\$120K +	126	601
	40K–60K	271	1519
	60K–80K	189	1213
	80K–120K	242	1293
	Less than \$40K	612	2949

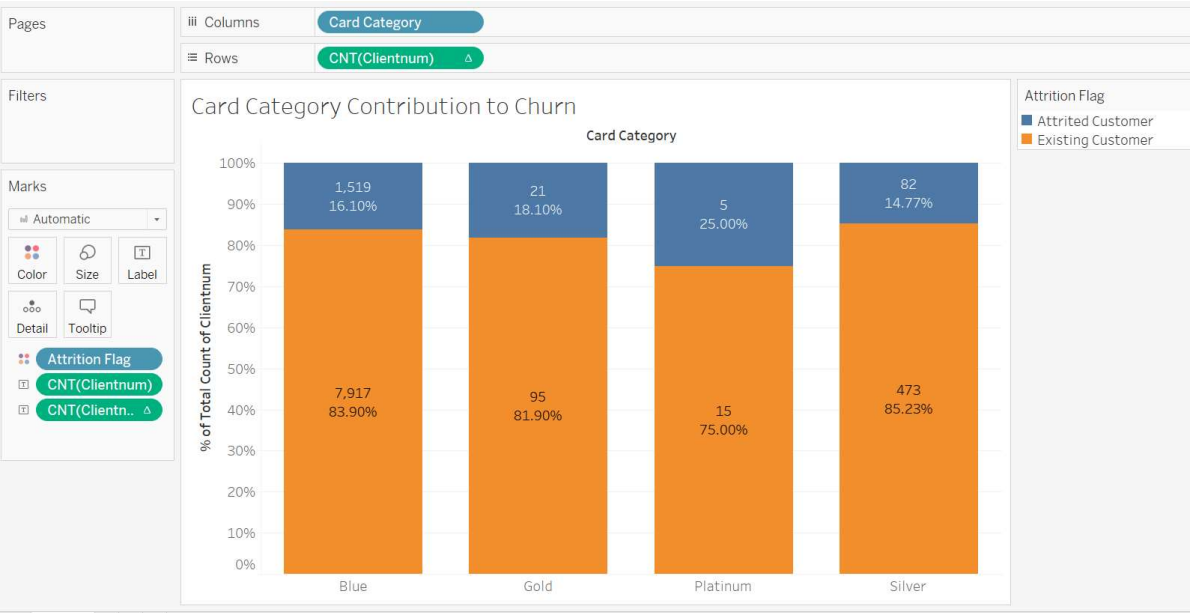




```
In [32]: # Churn due to Card Category
marital_card_category = Bank_Churners.groupby(['Card_Category', 'Attrition_Flag'])['Attrition_Flag'].head(5)
```

Out[32]: Attrition\_Flag Attrited Customer Existing Customer

Marital_Status		
Divorced	121	627
Married	709	3978
Single	668	3275
Unknown	129	620



```
In [ ]: # Categories to one hot encode for our initial model

# Attrition_Flag, Gender, Dependent_Count, Education_Level, Marital_Status, Income_Cat
```