

#Import the needed modules

```
import pandas as pd
import matplotlib.pyplot as plt
```

#Settings to produce nice plots

```
plt.style.use('ggplot')
```

#Import the customer churn data in csv format

```
da1=pd.read_csv('C:/Users/pc/Documents/BICHALLENGE/Telecom+Customer+Churn/telecom_customer_
_churn.csv')
```

#Import the customer population data in csv format

```
da2=pd.read_csv('C:/Users/pc/Documents/BICHALLENGE/Telecom+Customer+Churn/telecom_zipcode_
population.csv')
```

#Print the head of customer churn data

```
da1.head()
   Customer ID  Gender  ...  Churn Category  Churn Reason
0  0002-ORFBO  Female  ...                NaN            NaN
1  0003-MKNFE   Male   ...                NaN            NaN
2  0004-TLHLJ   Male   ...  Competitor  Competitor had better devices
3  0011-IGKFF   Male   ...  Dissatisfaction  Product dissatisfaction
4  0013-EXCHZ  Female  ...  Dissatisfaction  Network reliability

[5 rows x 38 columns]
```

#Print the head of customer city population data

```
da2.head()
   Zip Code  Population
0    90001      54492
1    90002      44586
2    90003      58198
3    90004      67852
4    90005      43019
```

#Print the columns of the customer churn data

```
da1.columns
```

```
Index(['Customer ID', 'Gender', 'Age', 'Married', 'Number of Dependents',
      'City', 'Zip Code', 'Latitude', 'Longitude', 'Number of Referrals',
      'Tenure in Months', 'Offer', 'Phone Service',
      'Avg Monthly Long Distance Charges', 'Multiple Lines',
      'Internet Service', 'Internet Type', 'Avg Monthly GB Download',
      'Online Security', 'Online Backup', 'Device Protection Plan',
      'Premium Tech Support', 'Streaming TV', 'Streaming Movies',
      'Streaming Music', 'Unlimited Data', 'Contract', 'Paperless Billing',
      'Payment Method', 'Monthly Charge', 'Total Charges', 'Total Refunds',
      'Total Extra Data Charges', 'Total Long Distance Charges',
      'Total Revenue', 'Customer Status', 'Churn Category', 'Churn Reason'],
      dtype='object')
```

#Print the columns of the customer city population

da2.columns

```
Index(['Zip Code', 'Population'], dtype='object')
```

#Merge the customer churn data, customer city population data and print the output

data=da1.merge(da2,on='Zip Code')

print(data)

```
   Customer ID  Gender  ...  Churn Reason  Population
0    0002-ORFBO  Female  ...             NaN         4498
1    5183-SNMJQ   Male   ...             NaN         4498
2    6847-KJLTS  Female  ...             NaN         4498
3    8788-DOXSU   Male   ...             NaN         4498
4    0003-MKNFE   Male   ...             NaN        31297
...         ...     ...   ...         ...         ...
7038  9396-ZSFLL  Female  ...             NaN        17409
7039  9281-OFDMF   Male   ...             NaN          301
7040  9494-MRNYX   Male   ...             NaN          301
7041  9565-JSNFM   Male   ...  Lack of affordable download/upload speed  301
7042  9972-EWRJS  Female  ...             NaN          301

[7043 rows x 39 columns]
```

#Display the data type for each column

data.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7043 entries, 0 to 7042
Data columns (total 39 columns):
#   Column                                Non-Null Count  Dtype  19  Online Backup                        5517 non-null  object
#   Column                                Non-Null Count  Dtype  20  Device Protection Plan               5517 non-null  object
#   Column                                Non-Null Count  Dtype  21  Premium Tech Support                5517 non-null  object
#   Column                                Non-Null Count  Dtype  22  Streaming TV                        5517 non-null  object
#   Column                                Non-Null Count  Dtype  23  Streaming Movies                    5517 non-null  object
#   Column                                Non-Null Count  Dtype  24  Streaming Music                     5517 non-null  object
#   Column                                Non-Null Count  Dtype  25  Unlimited Data                      5517 non-null  object
#   Column                                Non-Null Count  Dtype  26  Contract                            7043 non-null  object
#   Column                                Non-Null Count  Dtype  27  Paperless Billing                   7043 non-null  object
#   Column                                Non-Null Count  Dtype  28  Payment Method                     7043 non-null  object
#   Column                                Non-Null Count  Dtype  29  Monthly Charge                     7043 non-null  float64
#   Column                                Non-Null Count  Dtype  30  Total Charges                      7043 non-null  float64
#   Column                                Non-Null Count  Dtype  31  Total Refunds                      7043 non-null  float64
#   Column                                Non-Null Count  Dtype  32  Total Extra Data Charges            7043 non-null  int64
#   Column                                Non-Null Count  Dtype  33  Total Long Distance Charges         7043 non-null  float64
#   Column                                Non-Null Count  Dtype  34  Total Revenue                      7043 non-null  float64
#   Column                                Non-Null Count  Dtype  35  Customer Status                    7043 non-null  object
#   Column                                Non-Null Count  Dtype  36  Churn Category                     1869 non-null  object
0   Customer ID                        7043 non-null  object
1   Gender                             7043 non-null  object
2   Age                                7043 non-null  int64
3   Married                           7043 non-null  object
4   Number of Dependents               7043 non-null  int64
5   City                               7043 non-null  object
6   Zip Code                           7043 non-null  int64
7   Latitude                           7043 non-null  float64
8   Longitude                           7043 non-null  float64
9   Number of Referrals                7043 non-null  int64
10  Tenure in Months                   7043 non-null  int64
11  Offer                              7043 non-null  object
12  Phone Service                      7043 non-null  object
13  Avg Monthly Long Distance Charges  6361 non-null  float64
14  Multiple Lines                     6361 non-null  object
15  Internet Service                   7043 non-null  object
16  Internet Type                      5517 non-null  object
17  Avg Monthly GB Download            5517 non-null  float64
18  Online Security                    5517 non-null  object
```

#Display the total missing value for each column

```
data.isna().sum()
```

```
Customer ID      0 Premium Tech Support      1526
Gender           0 Streaming TV          1526
Age              0 Streaming Movies       1526
Married          0 Streaming Music        1526
Number of Dependents  0 Unlimited Data      1526
City             0 Contract              0
Zip Code         0 Paperless Billing        0
Latitude         0 Payment Method          0
Longitude        0 Monthly Charge          0
Number of Referrals  0 Total Charges            0
Tenure in Months  0 Total Refunds             0
Offer            0 Total Extra Data Charges  0
Phone Service    0 Total Long Distance Charges  0
Avg Monthly Long Distance Charges  682 Total Revenue              0
Multiple Lines   682 Customer Status          0
Internet Service 0 Churn Category          5174
Internet Type    1526 Churn Reason            5174
Avg Monthly GB Download 1526 Population              0
Online Security  1526
Online Backup     1526
Device Protection Plan 1526
dtype: int64
```

#Fill the missing values with 'Nil' and print the data

```
df=data.fillna('Nil')
```

```
print(df)
```

```
Customer ID  Gender  ...  Churn Reason  Population
0  0002-ORFBO  Female  ...  Nil  4498
1  5183-SNMJQ  Male    ...  Nil  4498
2  6847-KJLTS  Female  ...  Nil  4498
3  8788-DOXSU  Male    ...  Nil  4498
4  0003-MKNFE  Male    ...  Nil  31297
...  ...  ...  ...  ...
7038  9396-ZSFL  Female  ...  Nil  17409
7039  9281-OFDM  Male    ...  Nil  301
7040  9494-MRNY  Male    ...  Nil  301
7041  9565-JSNF  Male    ...  Lack of affordable download/upload speed  301
7042  9972-EWRJ  Female  ...  Nil  301

[7043 rows x 39 columns]
```

#Display the summary statistics of the data

```
df.describe()
```

```
Age  Number of Dependents  ...  Total Revenue  Population
count  7043.000000  7043.000000  ...  7043.000000  7043.000000
mean    46.509726    0.468692  ...  3034.379056  22139.603294
std     16.750352    0.962802  ...  2865.204542  21152.392837
min     19.000000    0.000000  ...    21.360000    11.000000
25%     32.000000    0.000000  ...    605.610000   2344.000000
50%     46.000000    0.000000  ...   2108.640000  17554.000000
75%     60.000000    0.000000  ...   4801.145000  36125.000000
max     80.000000    9.000000  ...  11979.340000 105285.000000

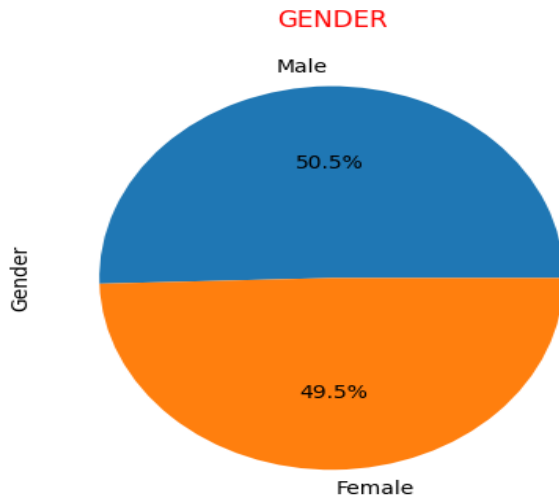
[8 rows x 14 columns]
```

#Count each gender of the data and print

```
gender=df['Gender'].value_counts()
print(gender)
Male      3555
Female    3488
Name: Gender, dtype: int64
```

#Plot a pie chart for the gender

```
gender.plot(kind='pie',autopct='%1.1f%%')
<AxesSubplot:ylabel='Gender'>
plt.title('GENDER',color='r')
Text(0.5, 1.0, 'GENDER')
```



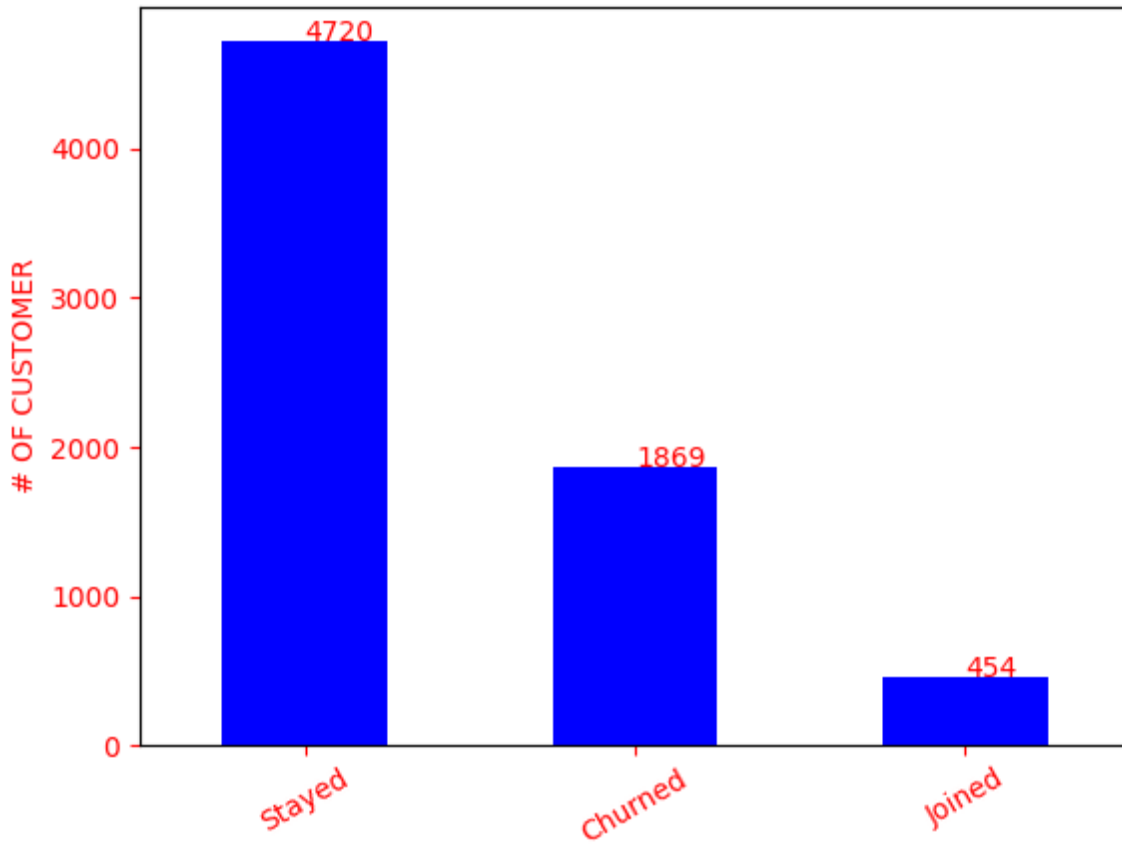
#Count the total number of customer status and print

```
status=df['Customer Status'].value_counts()
print(status)
Stayed      4720
Churned     1869
Joined       454
Name: Customer Status, dtype: int64
```

#plot a bar chart for the customer status

```
status.plot(kind='bar',color='b')
<AxesSubplot:>
plt.title('TELECOM CUSTOMER STATUS',color='r')
Text(0.5, 1.0, 'TELECOM CUSTOMER STATUS')
plt.xlabel('STATUS',color='r')
Text(0.5, 0, 'STATUS')
plt.ylabel('# OF CUSTOMER',color='r')
Text(0, 0.5, '# OF CUSTOMER')
plt.tick_params('x',color='r')
plt.tick_params('y',color='r')
plt.xticks(rotation=30)
(array([0, 1, 2]), [Text(0, 0, 'Stayed'), Text(1, 0, 'Churned'), Text(2, 0, 'Joined')])
plt.text(0,4720,4720,color='r')
Text(0, 4720, '4720')
plt.text(1,1869,1869,color='r')
Text(1, 1869, '1869')
plt.text(2,454,454,color='r')
Text(2, 454, '454')
plt.show()
```

TELECOM CUSTOMER STATUS



#What are the reasons for customer churn

```
reasons=df['Churn Reason'].value_counts()
```

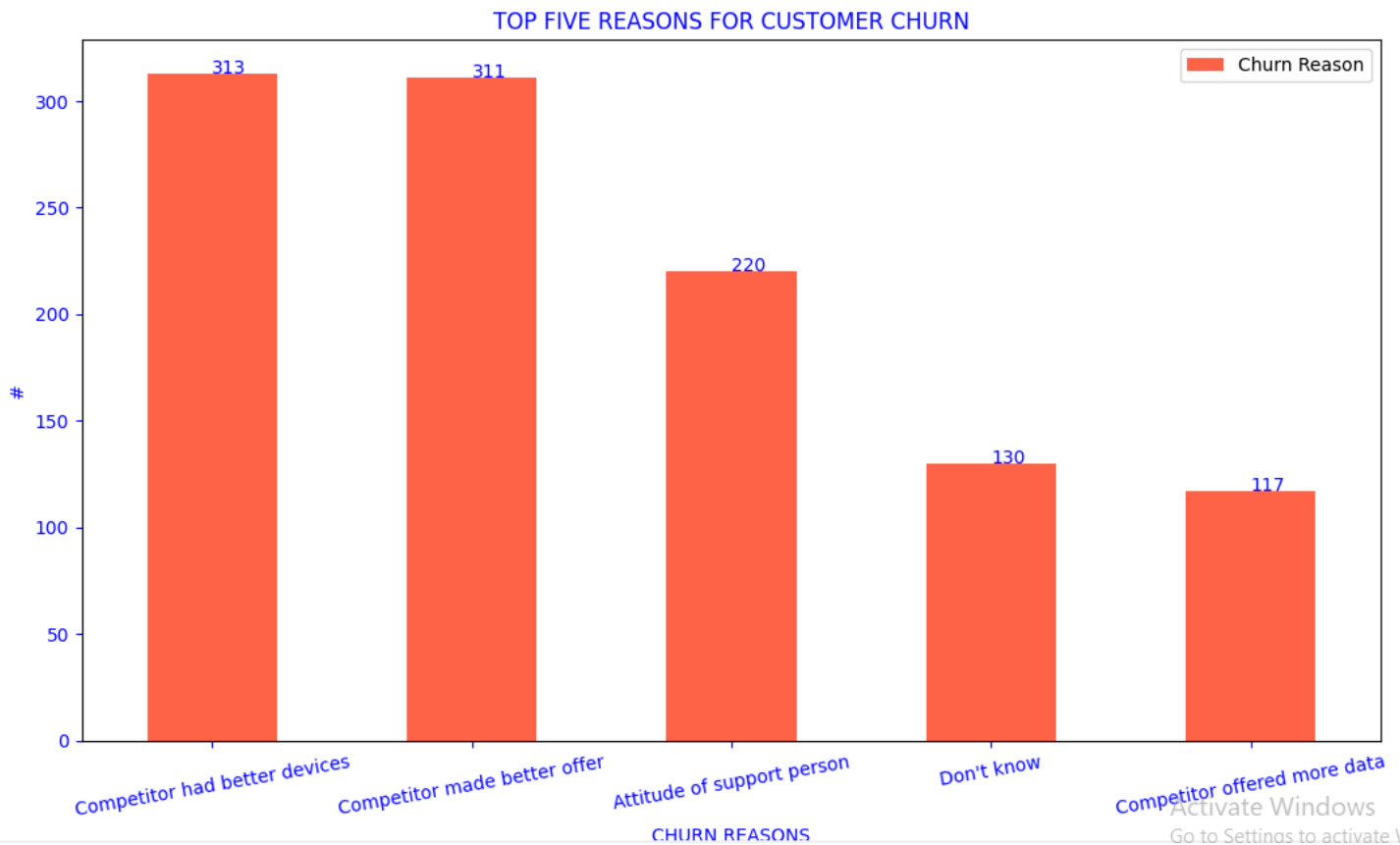
reasons

Competitor had better devices	313
Competitor made better offer	311
Attitude of support person	220
Don't know	130
Competitor offered more data	117
Competitor offered higher download speeds	100
Attitude of service provider	94
Price too high	78
Product dissatisfaction	77
Network reliability	72
Long distance charges	64
Service dissatisfaction	63
Moved	46
Extra data charges	39
Limited range of services	37
Poor expertise of online support	31
Lack of affordable download/upload speed	30
Lack of self-service on Website	29
Poor expertise of phone support	12
Deceased	6
Name: Churn Reason, dtype: int64	

#plot the top five reasons for customer churn on a bar chart

```
reasons.plot(kind='bar',color='tomato')
<AxesSubplot:xlabel='index'>
plt.title('TOP FIVE REASONS FOR CUSTOMER CHURN',color='b')
Text(0.5, 1.0, 'TOP FIVE REASONS FOR CUSTOMER CHURN')
plt.xlabel('CHURN REASONS',color='b')
Text(0.5, 0, 'CHURN REASONS')
plt.ylabel('#',color='b')
Text(0, 0.5, '#')
plt.tick_params('x',colors='b')
plt.tick_params('y',colors='b')
plt.xticks(rotation=20)
(array([0, 1, 2, 3, 4]), [Text(0, 0, 'Competitor had better devices'), Text(1, 0, 'Competitor made better offer'), Text(2, 0, 'Attitude of support person'), Text(3, 0, "Don't know"), Text(4, 0, 'Competitor offered more data')])
plt.text(0,313,313,color='b')
Text(0, 313, '313')
plt.text(1,311,311,color='b')
Text(1, 311, '311')
plt.text(2,220,220,color='b')
Text(2, 220, '220')
plt.text(3,130,130,color='b')
Text(3, 130, '130')
plt.text(4,117,117,color='b')
Text(4, 117, '117')
plt.show()
```

Activate Windows
Go to Settings to activate Windows.



Activate Windows
Go to Settings to activate Windows.

#Display the customer status by each gender

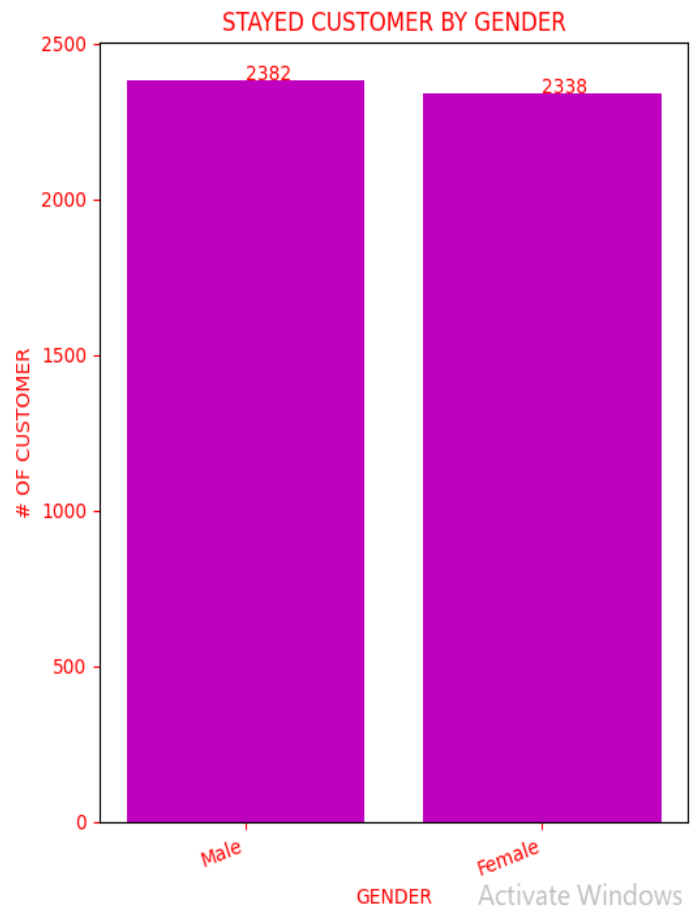
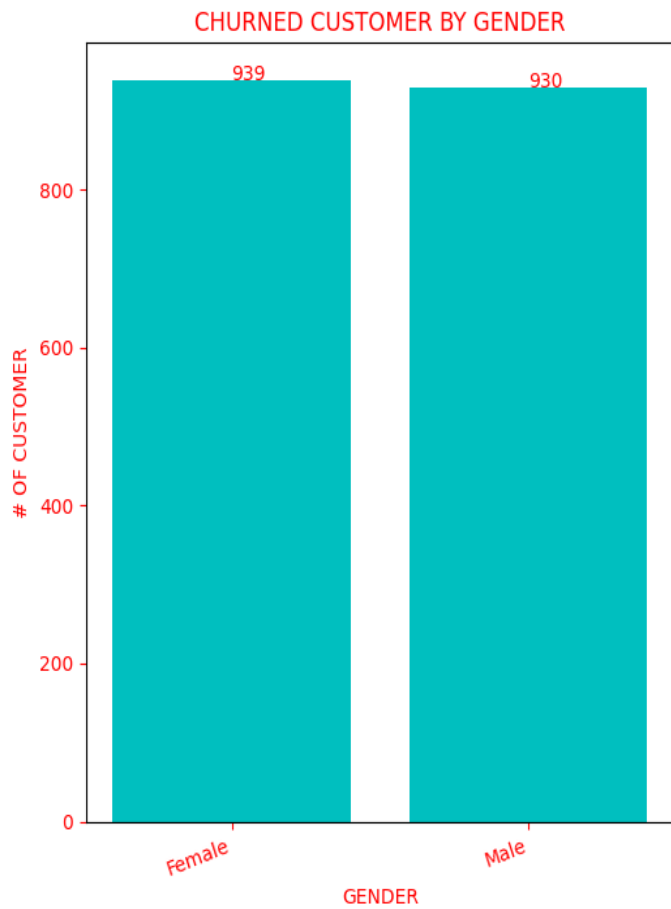
```
churned
  index  Gender
0  Female    939
1   Male    930
stayed
  index  Gender
0   Male   2382
1  Female   2338
joined
  index  Gender
0   Male    243
1  Female    211
```

#plot a bar chart for churned customer and stayed customer status

```
fig,ax=plt.subplots(1,2)
ax[0].bar(churned['index'],churned['Gender'],color='c')
<BarContainer object of 2 artists>
ax[0].set_title('CHURNED CUSTOMER BY GENDER',color='r')
Text(0.5, 1.0, 'CHURNED CUSTOMER BY GENDER')
ax[0].set_xlabel('GENDER',color='r')
Text(0.5, 0, 'GENDER')
ax[0].set_ylabel('# OF CUSTOMER',color='r')
Text(0, 0.5, '# OF CUSTOMER')
ax[0].tick_params('x',colors='r')
ax[0].tick_params('y',colors='r')
ax[0].set_xticklabels(churned['index'],rotation=20,horizontalalignment='right')

Warning (from warnings module):
  File "<pyshell#58>", line 1
UserWarning: FixedFormatter should only be used together with FixedLocator
[Text(0, 0, 'Female'), Text(1, 0, 'Male')]
ax[0].text(0,939,939,color='r')
Text(0, 939, '939')
ax[0].text(1,930,930,color='r')
Text(1, 930, '930')
ax[1].bar(stayed['index'],stayed['Gender'],color='m')
<BarContainer object of 2 artists>
ax[1].set_title('STAYED CUSTOMER BY GENDER',color='r')
Text(0.5, 1.0, 'STAYED CUSTOMER BY GENDER')
ax[1].set_xlabel('GENDER',color='r')
Text(0.5, 0, 'GENDER')
ax[1].set_ylabel('# OF CUSTOMER',color='r')
Text(0, 0.5, '# OF CUSTOMER')
ax[1].tick_params('x',colors='r')
ax[1].tick_params('y',colors='r')
ax[1].set_xticklabels(stayed['index'],rotation=20,horizontalalignment='right')

Warning (from warnings module):
  File "<pyshell#67>", line 1
UserWarning: FixedFormatter should only be used together with FixedLocator
[Text(0, 0, 'Male'), Text(1, 0, 'Female')]
ax[1].text(0,2382,2382,color='r')
Text(0, 2382, '2382')
ax[1].text(1,2338,2338,color='r')
Text(1, 2338, '2338')
plt.show()
```



#Display the offer given to churned customer and stayed customer

```
offer_churned
None      1051
Offer E    426
Offer D    161
Offer B    101
Offer C     95
Offer A     35
Name: Offer, dtype: int64
offer_stayed
None      2547
Offer B    723
Offer A    485
Offer D    441
Offer C    320
Offer E    204
Name: Offer, dtype: int64
```


#plot a bar chart for offer given to churned and stayed customer

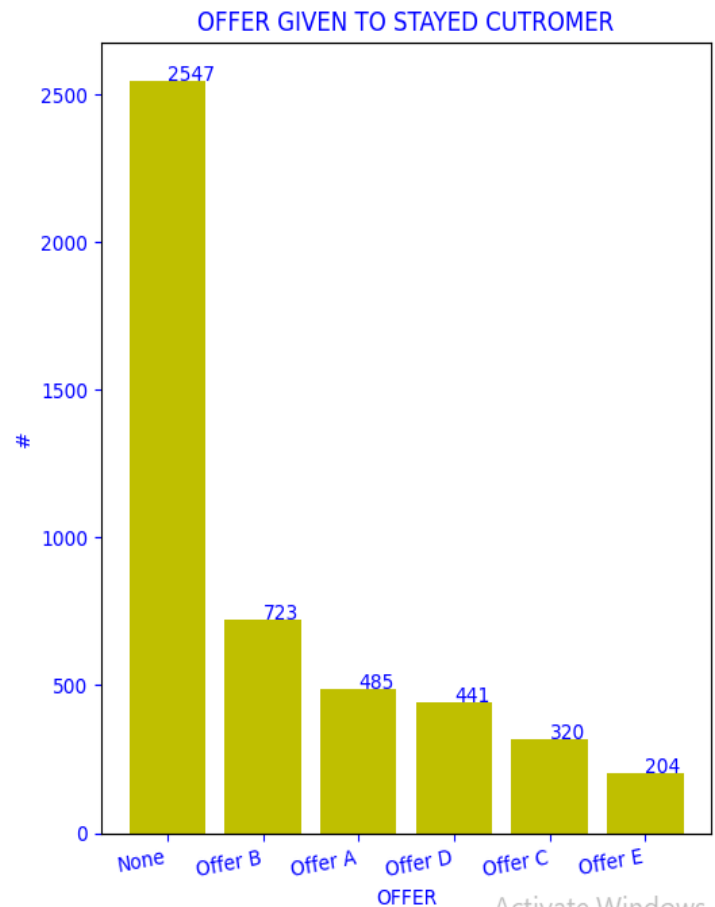
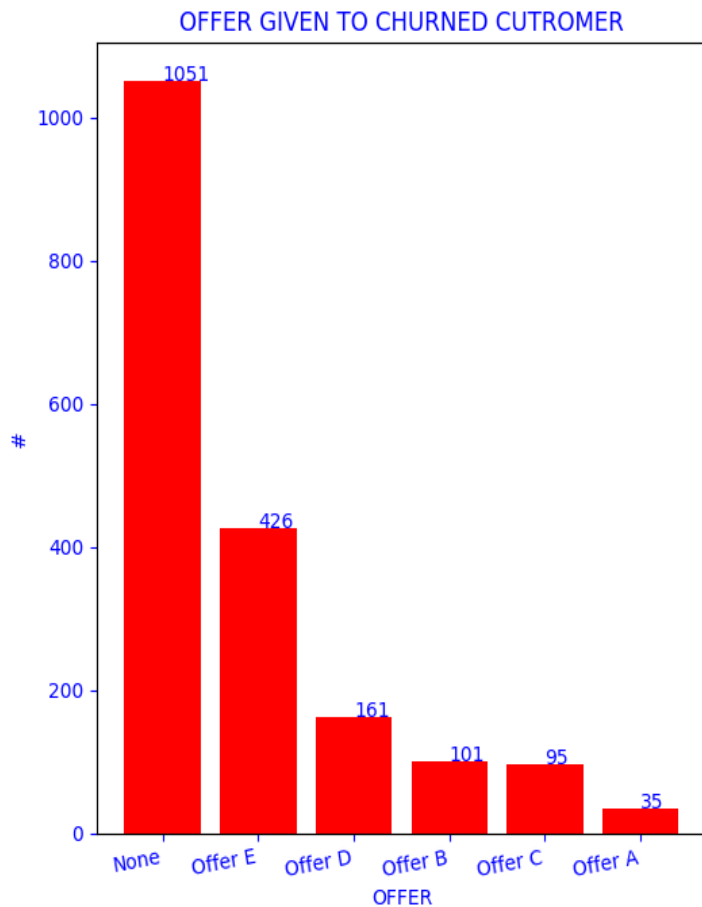
```
fig,ax=plt.subplots(1,2)
ax[0].bar(offer_churned['index'],offer_churned['Offer'],color='r')
<BarContainer object of 6 artists>
ax[0].set_title('OFFER GIVEN TO CHURNED CUTROMER',color='b')
Text(0.5, 1.0, 'OFFER GIVEN TO CHURNED CUTROMER')
ax[0].set_xlabel('OFFER',color='b')
Text(0.5, 0, 'OFFER')
ax[0].set_ylabel('#',color='b')
Text(0, 0.5, '#')
ax[0].tick_params('x',colors='b')
ax[0].tick_params('y',colors='b')
ax[0].set_xticklabels(offer_churned['index'],rotation=10,horizontalalignment='right')

Warning (from warnings module):
  File "<pyshell#167>", line 1
UserWarning: FixedFormatter should only be used together with FixedLocator
[Text(0, 0, 'None'), Text(1, 0, 'Offer E'), Text(2, 0, 'Offer D'), Text(3, 0, 'Offer B'), Text(4, 0, 'Offer C'), Text(5, 0, 'Offer A')]
ax[0].text(0,1051,1051,color='b')
Text(0, 1051, '1051')
ax[0].text(1,426,426,color='b')
Text(1, 426, '426')
ax[0].text(2,161,161,color='b')
Text(2, 161, '161')
ax[0].text(3,101,101,color='b')
Text(3, 101, '101')
ax[0].text(4,95,95,color='b')
Text(4, 95, '95')
ax[0].text(5,35,35,color='b')
Text(5, 35, '35')
ax[1].bar(offer_stayed['index'],offer_stayed['Offer'],color='y')
<BarContainer object of 6 artists>
ax[1].set_title('OFFER GIVEN TO STAYED CUTROMER',color='b')
Text(0.5, 1.0, 'OFFER GIVEN TO STAYED CUTROMER')
ax[1].set_xlabel('OFFER',color='b')
Text(0.5, 0, 'OFFER')
ax[1].set_ylabel('#',color='b')
Text(0, 0.5, '#')
ax[1].tick_params('x',colors='b')
ax[1].tick_params('y',colors='b')
ax[1].set_xticklabels(offer_stayed['index'],rotation=10,horizontalalignment='right')
```

Activate Windows

```
Warning (from warnings module):
  File "<pyshell#180>", line 1
UserWarning: FixedFormatter should only be used together with FixedLocator
[Text(0, 0, 'None'), Text(1, 0, 'Offer B'), Text(2, 0, 'Offer A'), Text(3, 0, 'Offer D'), Text(4, 0, 'Offer C'), Text(5, 0, 'Offer E')]
ax[1].text(0,2547,2547,color='b')
Text(0, 2547, '2547')
ax[1].text(1,723,723,color='b')
Text(1, 723, '723')
ax[1].text(2,485,485,color='b')
Text(2, 485, '485')
ax[1].text(3,441,441,color='b')
Text(3, 441, '441')
ax[1].text(4,320,320,color='b')
Text(4, 320, '320')
ax[1].text(5,204,204,color='b')
Text(5, 204, '204')
plt.show()
```

Activate Windows
Go to Settings to act

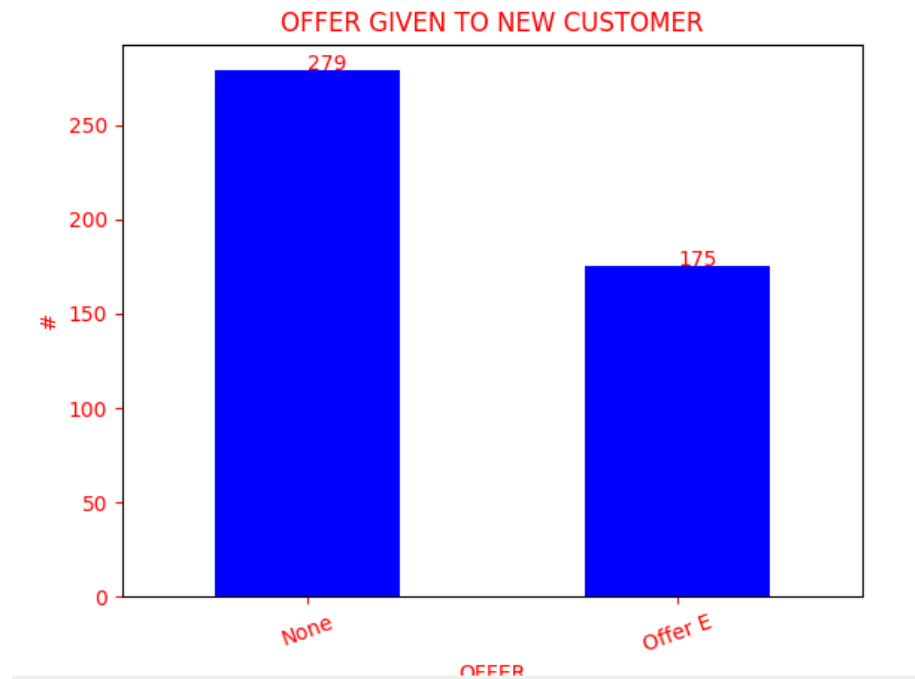


#Display offer given to the new customers

```
joined
None      279
Offer E   175
Name: Offer, dtype: int64
```

#Plot a bar chart for the new customer offer

```
joined.plot(kind='bar',color='b')
<AxesSubplot:>
plt.title('OFFER GIVEN TO NEW CUSTOMER',color='r')
Text(0.5, 1.0, 'OFFER GIVEN TO NEW CUSTOMER')
plt.xlabel('OFFER',color='r')
Text(0.5, 0, 'OFFER')
plt.ylabel('#',color='r')
Text(0, 0.5, '#')
plt.tick_params('x',colors='r')
plt.tick_params('y',colors='r')
plt.text(0,279,279,color='r')
Text(0, 279, '279')
plt.text(1,175,175,color='r')
Text(1, 175, '175')
plt.xticks(rotation=20)
(array([0, 1]), [Text(0, 0, 'None'), Text(1, 0, 'Offer E')])
plt.show()
```



RECOMMENDATION

According to the data provided, I will recommend that the organization (TELECOM PLC) provide a better and reliable device to their existing and potential customers because about 313 churned customers are not satisfied with the organization device which is one of the major reasons for customer churn.

Secondly, the organization should look into offer been given to their customers generally, about 311 churned customers rated their offers has one of the reasons for not engaging and buying the organization device. Also, the existing and new customers that has no offer are 2826 customers while customers with offers are 2348 customers. As it stands, there is probability that the organization may experience more customer churn due to this reason.

Thirdly, the organization should organize a retreat/meeting to educate their support person (sales rep) on how to engage customers.