CHURN ANALYSIS

#Importing libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

#Reading the file

```
df= pd.read_csv('Customer Churn.csv')
df

      customerID  gender  SeniorCitizen Partner Dependents  tenure  \
0     7590-VHVEG  Female              0     Yes         No       1
1     5575-GNVDE    Male              0      No         No      34
2     3668-QPYBK    Male              0      No         No       2
3     7795-CFOCW    Male              0      No         No      45
4     9237-HQITU  Female              0      No         No       2
...          ...     ...            ...     ...        ...     ...
7038  6840-RESVB    Male              0     Yes        Yes      24
7039  2234-XADUH  Female              0     Yes        Yes      72
7040  4801-JZAZL  Female              0     Yes        Yes      11
7041  8361-LTMKD    Male              1     Yes         No       4
7042  3186-AJIEK    Male              0      No         No      66

     PhoneService     MultipleLines InternetService
OnlineSecurity  ...  \
0              No  No phone service             DSL
No  ...
1             Yes                No             DSL
Yes  ...
2             Yes                No             DSL
Yes  ...
3              No  No phone service             DSL
Yes  ...
4             Yes                No     Fiber optic
No  ...
...           ...               ...             ...            ...  ..
.
7038          Yes               Yes             DSL
Yes  ...
7039          Yes               Yes     Fiber optic
No  ...
7040           No  No phone service             DSL
Yes  ...
7041          Yes               Yes     Fiber optic
No  ...
```

```
7042          Yes                No     Fiber optic
Yes  ...

     DeviceProtection TechSupport StreamingTV StreamingMovies
Contract  \
0                  No          No          No              No  Month-
to-month
1                 Yes          No          No              No
One year
2                  No          No          No              No  Month-
to-month
3                 Yes         Yes          No              No
One year
4                  No          No          No              No  Month-
to-month
...               ...         ...         ...             ...
...
7038              Yes         Yes         Yes             Yes
One year
7039              Yes          No         Yes             Yes
One year
7040               No          No          No              No  Month-
to-month
7041               No          No          No              No  Month-
to-month
7042              Yes         Yes         Yes             Yes
Two year

     PaperlessBilling              PaymentMethod MonthlyCharges
TotalCharges  \
0                 Yes           Electronic check          29.85
29.85
1                  No               Mailed check          56.95
1889.5
2                 Yes               Mailed check          53.85
108.15
3                  No  Bank transfer (automatic)          42.30
1840.75
4                 Yes           Electronic check          70.70
151.65
...               ...                        ...            ...
...
7038              Yes               Mailed check          84.80
1990.5
7039              Yes    Credit card (automatic)         103.20
7362.9
7040              Yes           Electronic check          29.60
346.45
7041              Yes               Mailed check          74.40
```

```
306.6
7042              Yes  Bank transfer (automatic)         105.65
6844.5

     Churn
0       No
1       No
2      Yes
3       No
4      Yes
...    ...
7038    No
7039    No
7040    No
7041   Yes
7042    No

[7043 rows x 21 columns]
```

#To get to know about data

#head() will display 5 rows

```
df.head()

   customerID  gender  SeniorCitizen Partner Dependents  tenure
PhoneService  \
0  7590-VHVEG  Female              0     Yes         No       1
No
1  5575-GNVDE    Male              0      No         No      34
Yes
2  3668-QPYBK    Male              0      No         No       2
Yes
3  7795-CFOCW    Male              0      No         No      45
No
4  9237-HQITU  Female              0      No         No       2
Yes

      MultipleLines InternetService OnlineSecurity  ...
DeviceProtection  \
0  No phone service             DSL             No  ...
No
1                No             DSL            Yes  ...
Yes
2                No             DSL            Yes  ...
No
3  No phone service             DSL            Yes  ...
Yes
4                No     Fiber optic             No  ...
No
```

```
  TechSupport StreamingTV StreamingMovies        Contract
PaperlessBilling  \
0          No          No              No  Month-to-month
Yes
1          No          No              No        One year
No
2          No          No              No  Month-to-month
Yes
3         Yes          No              No        One year
No
4          No          No              No  Month-to-month
Yes

               PaymentMethod MonthlyCharges  TotalCharges Churn
0           Electronic check          29.85         29.85    No
1               Mailed check          56.95        1889.5    No
2               Mailed check          53.85        108.15   Yes
3  Bank transfer (automatic)          42.30       1840.75    No
4           Electronic check          70.70        151.65   Yes

[5 rows x 21 columns]
```

#To get information from data about columns

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object 
 20  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

#From above output we saw that column 'TotalCharges' have 'object' Dtype... So we need to inspect it from csv file

#We notice that TotalCharge column has some blanks values where tenure column has 0 values

#so we need to rplace it with 0

#Replacing blanks with '0' as tenure is '0' and no TotalCharges is recorded

```
df["TotalCharges"]=df["TotalCharges"].replace(" ","0")
```

#Changing the datatype from object to float

```
df["TotalCharges"]=df["TotalCharges"].astype("float")
```

#Again checking for information(checking whether the Dtype for TotalCharge has changed or not

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   SeniorCitizen     7043 non-null   int64  
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            7043 non-null   int64  
 6   PhoneService      7043 non-null   object 
 7   MultipleLines     7043 non-null   object 
 8   InternetService   7043 non-null   object 
 9   OnlineSecurity    7043 non-null   object 
 10  OnlineBackup      7043 non-null   object 
 11  DeviceProtection  7043 non-null   object 
 12  TechSupport       7043 non-null   object 
 13  StreamingTV       7043 non-null   object 
 14  StreamingMovies   7043 non-null   object 
 15  Contract          7043 non-null   object 
 16  PaperlessBilling  7043 non-null   object 
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   float64
 20  Churn             7043 non-null   object 
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB
```

#Now we are checking for null values in dataset

```
df.isnull()

      customerID  gender  SeniorCitizen  Partner  Dependents 
tenure  \
0          False   False          False    False       False   False

1          False   False          False    False       False   False

2          False   False          False    False       False   False

3          False   False          False    False       False   False

4          False   False          False    False       False   False

...          ...     ...            ...      ...         ...     ...

7038       False   False          False    False       False   False

7039       False   False          False    False       False   False

7040       False   False          False    False       False   False

7041       False   False          False    False       False   False

7042       False   False          False    False       False   False


      PhoneService  MultipleLines  InternetService 
OnlineSecurity  ...  \
0            False          False            False 
False  ...
1            False          False            False 
False  ...
2            False          False            False 
False  ...
3            False          False            False 
False  ...
4            False          False            False 
False  ...
...            ...            ...              ...             ...  ..
.
```

```
7038         False          False            False
False  ...
7039         False          False            False
False  ...
7040         False          False            False
False  ...
7041         False          False            False
False  ...
7042         False          False            False
False  ...

      DeviceProtection  TechSupport  StreamingTV  StreamingMovies
Contract  \
0                False        False        False            False
False
1                False        False        False            False
False
2                False        False        False            False
False
3                False        False        False            False
False
4                False        False        False            False
False
...                ...          ...          ...              ...
...
7038             False        False        False            False
False
7039             False        False        False            False
False
7040             False        False        False            False
False
7041             False        False        False            False
False
7042             False        False        False            False
False

      PaperlessBilling  PaymentMethod  MonthlyCharges  TotalCharges
Churn
0                False          False           False         False
False
1                False          False           False         False
False
2                False          False           False         False
False
3                False          False           False         False
False
4                False          False           False         False
False
...                ...            ...             ...           ...
```

```
...
7038              False          False          False         False
False
7039              False          False          False         False
False
7040              False          False          False         False
False
7041              False          False          False         False
False
7042              False          False          False         False
False

[7043 rows x 21 columns]
```

#It gives us values in True AND False, we need the total number of null values if available

```
df.isnull().sum()

customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

#Above code gives us null values based on columns, but we want total number of null value in entire dataset

```
df.isnull().sum().sum()

np.int64(0)

print(df.isnull().sum().sum())
```

```
0
```

#We want to know about aggregation functions

```
df.describe()
```

```
       SeniorCitizen       tenure  MonthlyCharges  TotalCharges
count    7043.000000  7043.000000     7043.000000   7043.000000
mean        0.162147    32.371149       64.761692   2279.734304
std         0.368612    24.559481       30.090047   2266.794470
min         0.000000     0.000000       18.250000      0.000000
25%         0.000000     9.000000       35.500000    398.550000
50%         0.000000    29.000000       70.350000   1394.550000
75%         0.000000    55.000000       89.850000   3786.600000
max         1.000000    72.000000      118.750000   8684.800000
```

#Now we are trying to find out if there is any duplicate value availbale

```
df.duplicated()
```

```
0       False
1       False
2       False
3       False
4       False
        ...
7038    False
7039    False
7040    False
7041    False
7042    False
Length: 7043, dtype: bool
```

#It gives us a value in True/False, so we wanted a total duplicated values available if any

```
df.duplicated().sum()
```

```
np.int64(0)
```

```
print(df.duplicated().sum())
```

```
0
```

#We have to check for duplicated value based on Unique column as well

```
df['customerID'].duplicated().sum()
```

```
np.int64(0)
```

```
print(df['customerID'].duplicated().sum())

0
```

#In dataset we noticed that SeniorCitizen column has 1 or 0....which is not so good to be read so

#Convert 0 & 1 from SeniorCitizen column to yes/no to make it easier to understand

#For that we define a function named convert

```
def convert(value):
    if value == 1:
        return 'yes'
    else:
        return 'no'

df['SeniorCitizen']=df['SeniorCitizen'].apply(convert)
```

#Checking whether SeniorCitizen column values have changed or not

```
df.head()

   customerID  gender SeniorCitizen Partner Dependents  tenure 
PhoneService  \
0  7590-VHVEG  Female            no     Yes         No       1 
No   
1  5575-GNVDE    Male            no      No         No      34 
Yes   
2  3668-QPYBK    Male            no      No         No       2 
Yes   
3  7795-CFOCW    Male            no      No         No      45 
No   
4  9237-HQITU  Female            no      No         No       2 
Yes   

      MultipleLines InternetService OnlineSecurity  ... 
DeviceProtection  \
0  No phone service             DSL             No  ...  
No   
1                No             DSL            Yes  ...  
Yes   
2                No             DSL            Yes  ...  
No   
3  No phone service             DSL            Yes  ...  
Yes   
4                No     Fiber optic             No  ...  
No   

  TechSupport StreamingTV StreamingMovies        Contract 
PaperlessBilling  \
0          No          No                No  Month-to-month
Yes
1          No          No                No        One year
No
2          No          No                No  Month-to-month
Yes
3         Yes          No                No        One year
No
4          No          No                No  Month-to-month
Yes

               PaymentMethod MonthlyCharges  TotalCharges  Churn
0           Electronic check          29.85         29.85     No
1               Mailed check          56.95       1889.50     No
2               Mailed check          53.85        108.15    Yes
3  Bank transfer (automatic)          42.30       1840.75     No
4           Electronic check          70.70        151.65    Yes

[5 rows x 21 columns]
```

```
df.head(30)
```

```
    customerID  gender SeniorCitizen Partner Dependents  tenure
PhoneService  \
0   7590-VHVEG  Female            no     Yes         No       1
No
1   5575-GNVDE    Male            no      No         No      34
Yes
2   3668-QPYBK    Male            no      No         No       2
Yes
3   7795-CFOCW    Male            no      No         No      45
No
4   9237-HQITU  Female            no      No         No       2
Yes
5   9305-CDSKC  Female            no      No         No       8
Yes
6   1452-KIOVK    Male            no      No        Yes      22
Yes
7   6713-OKOMC  Female            no      No         No      10
No
8   7892-POOKP  Female            no     Yes         No      28
Yes
9   6388-TABGU    Male            no      No        Yes      62
Yes
10  9763-GRSKD    Male            no     Yes        Yes      13
Yes
11  7469-LKBCI    Male            no      No         No      16
Yes
12  8091-TTVAX    Male            no     Yes         No      58
Yes
```

```
13  0280-XJGEX    Male             no      No          No      49
Yes
14  5129-JLPIS    Male             no      No          No      25
Yes
15  3655-SNQYZ  Female             no     Yes         Yes      69
Yes
16  8191-XWSZG  Female             no      No          No      52
Yes
17  9959-WOFKT    Male             no      No         Yes      71
Yes
18  4190-MFLUW  Female             no     Yes         Yes      10
Yes
19  4183-MYFRB  Female             no      No          No      21
Yes
20  8779-QRDMV    Male            yes      No          No       1
No
21  1680-VDCWW    Male             no     Yes          No      12
Yes
22  1066-JKSGK    Male             no      No          No       1
Yes
23  3638-WEABW  Female             no     Yes          No      58
Yes
24  6322-HRPFA    Male             no     Yes         Yes      49
Yes
25  6865-JZNKO  Female             no      No          No      30
Yes
26  6467-CHFZW    Male             no     Yes         Yes      47
Yes
27  8665-UTDHZ    Male             no     Yes         Yes       1
No
28  5248-YGIJN    Male             no     Yes          No      72
Yes
29  8773-HHUOZ  Female             no      No         Yes      17
Yes

       MultipleLines InternetService       OnlineSecurity  ...  \
0   No phone service             DSL                   No  ...
1                 No             DSL                  Yes  ...
2                 No             DSL                  Yes  ...
3   No phone service             DSL                  Yes  ...
4                 No     Fiber optic                   No  ...
5                Yes     Fiber optic                   No  ...
6                Yes     Fiber optic                   No  ...
7   No phone service             DSL                  Yes  ...
8                Yes     Fiber optic                   No  ...
9                 No             DSL                  Yes  ...
10                No             DSL                  Yes  ...
11                No              No  No internet service  ...
12               Yes     Fiber optic                   No  ...
```

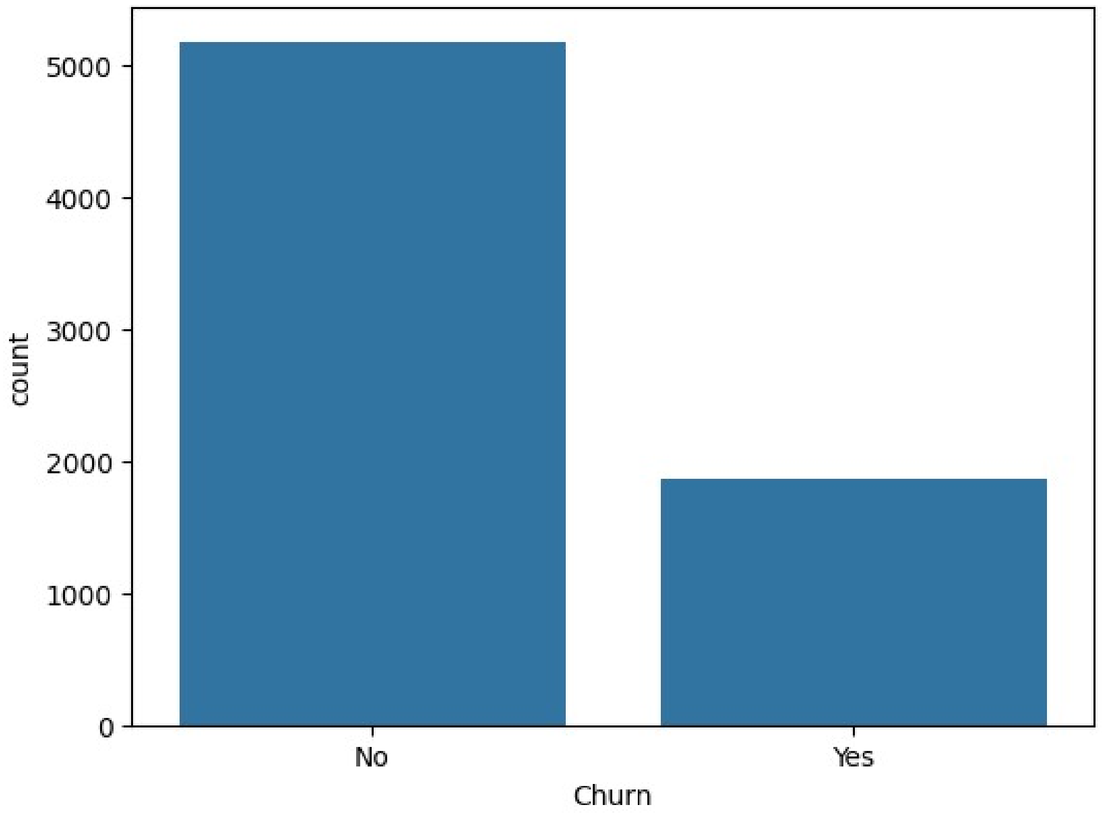
```
13                Yes     Fiber optic                   No  ...
14                 No     Fiber optic                  Yes  ...
15                Yes     Fiber optic                  Yes  ...
16                 No              No  No internet service  ...
17                Yes     Fiber optic                  Yes  ...
18                 No             DSL                   No  ...
19                 No     Fiber optic                   No  ...
20   No phone service             DSL                   No  ...
21                 No              No  No internet service  ...
22                 No              No  No internet service  ...
23                Yes             DSL                   No  ...
24                 No             DSL                  Yes  ...
25                 No             DSL                  Yes  ...
26                Yes     Fiber optic                   No  ...
27   No phone service             DSL                   No  ...
28                Yes             DSL                  Yes  ...
29                 No             DSL                   No  ...

       DeviceProtection          TechSupport          StreamingTV  \
0                    No                   No                   No
1                   Yes                   No                   No
2                    No                   No                   No
3                   Yes                  Yes                   No
4                    No                   No                   No
5                   Yes                   No                  Yes
6                    No                   No                  Yes
7                    No                   No                   No
8                   Yes                  Yes                  Yes
9                    No                   No                   No
10                   No                   No                   No
11  No internet service  No internet service  No internet service
12                  Yes                   No                  Yes
13                  Yes                   No                  Yes
14                  Yes                  Yes                  Yes
15                  Yes                  Yes                  Yes
16  No internet service  No internet service  No internet service
17                  Yes                   No                  Yes
18                  Yes                  Yes                   No
19                  Yes                   No                   No
20                  Yes                   No                   No
21  No internet service  No internet service  No internet service
22  No internet service  No internet service  No internet service
23                   No                  Yes                   No
24                   No                  Yes                   No
25                   No                   No                   No
26                   No                   No                  Yes
27                   No                   No                   No
28                  Yes                  Yes                  Yes
29                   No                   No                  Yes
```

```
        StreamingMovies        Contract PaperlessBilling  \
0                    No  Month-to-month              Yes
1                    No        One year               No
2                    No  Month-to-month              Yes
3                    No        One year               No
4                    No  Month-to-month              Yes
5                   Yes  Month-to-month              Yes
6                    No  Month-to-month              Yes
7                    No  Month-to-month               No
8                   Yes  Month-to-month              Yes
9                    No        One year               No
10                   No  Month-to-month              Yes
11  No internet service        Two year               No
12                  Yes        One year               No
13                  Yes  Month-to-month              Yes
14                  Yes  Month-to-month              Yes
15                  Yes        Two year               No
16  No internet service        One year               No
17                  Yes        Two year               No
18                   No  Month-to-month               No
19                  Yes  Month-to-month              Yes
20                  Yes  Month-to-month              Yes
21  No internet service        One year               No
22  No internet service  Month-to-month               No
23                   No        Two year              Yes
24                   No  Month-to-month               No
25                   No  Month-to-month              Yes
26                  Yes  Month-to-month              Yes
27                   No  Month-to-month               No
28                  Yes        Two year              Yes
29                  Yes  Month-to-month              Yes

                PaymentMethod MonthlyCharges  TotalCharges  Churn
0            Electronic check          29.85         29.85     No
1                Mailed check          56.95       1889.50     No
2                Mailed check          53.85        108.15    Yes
3   Bank transfer (automatic)          42.30       1840.75     No
4            Electronic check          70.70        151.65    Yes
5            Electronic check          99.65        820.50    Yes
6     Credit card (automatic)          89.10       1949.40     No
7                Mailed check          29.75        301.90     No
8            Electronic check         104.80       3046.05    Yes
9   Bank transfer (automatic)          56.15       3487.95     No
10               Mailed check          49.95        587.45     No
11    Credit card (automatic)          18.95        326.80     No
12    Credit card (automatic)         100.35       5681.10     No
13  Bank transfer (automatic)         103.70       5036.30    Yes
14           Electronic check         105.50       2686.05     No
```

```
15    Credit card (automatic)          113.25        7895.15     No
16               Mailed check           20.65        1022.95     No
17  Bank transfer (automatic)          106.70        7382.25     No
18    Credit card (automatic)           55.20         528.35    Yes
19           Electronic check           90.05        1862.90     No
20           Electronic check           39.65          39.65    Yes
21  Bank transfer (automatic)           19.80         202.25     No
22               Mailed check           20.15          20.15    Yes
23    Credit card (automatic)           59.90        3505.10     No
24    Credit card (automatic)           59.60        2970.30     No
25  Bank transfer (automatic)           55.30        1530.60     No
26           Electronic check           99.35        4749.15    Yes
27           Electronic check           30.20          30.20    Yes
28    Credit card (automatic)           90.25        6369.45     No
29               Mailed check           64.70        1093.10    Yes

[30 rows x 21 columns]
```

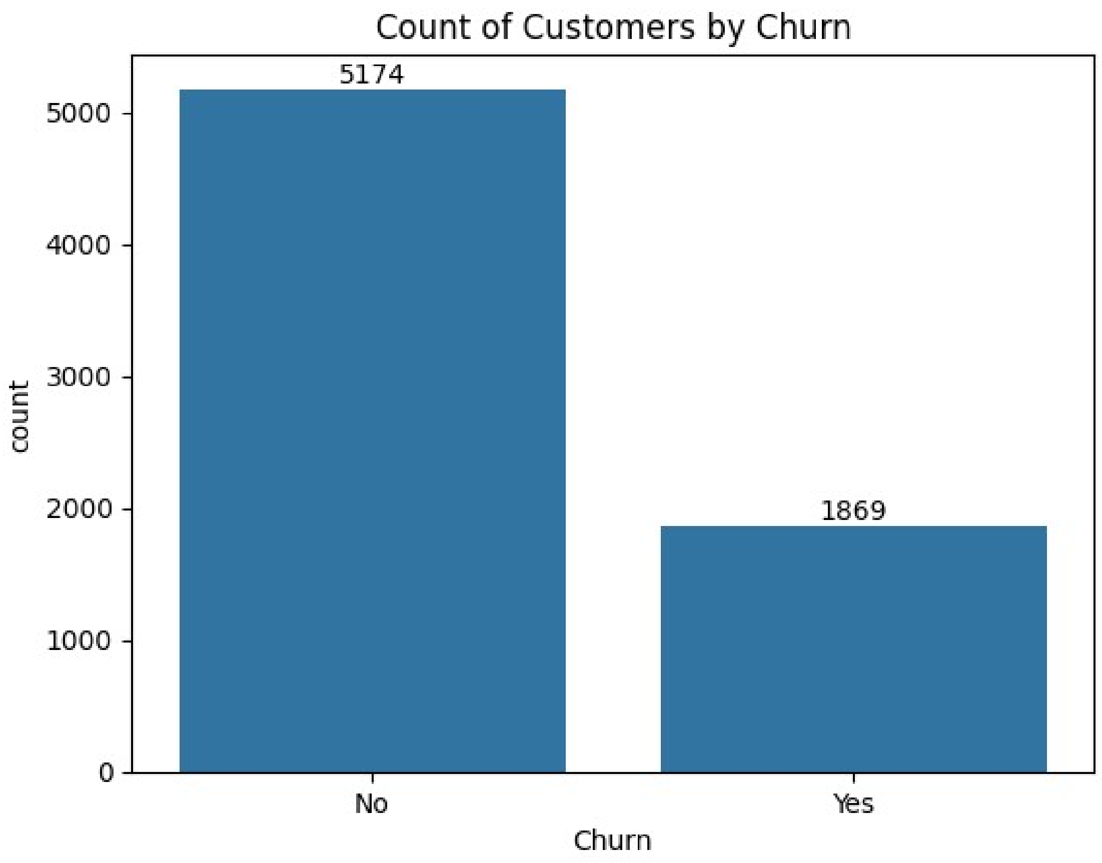
#Countplot to see how many customers has churned out or not

```
sns.countplot(x=df['Churn'], data=df)
plt.show()
```



#Modify the above code so that labels and title are displayed

```
ax=sns.countplot(x='Churn', data=df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Churn")
plt.show()
```

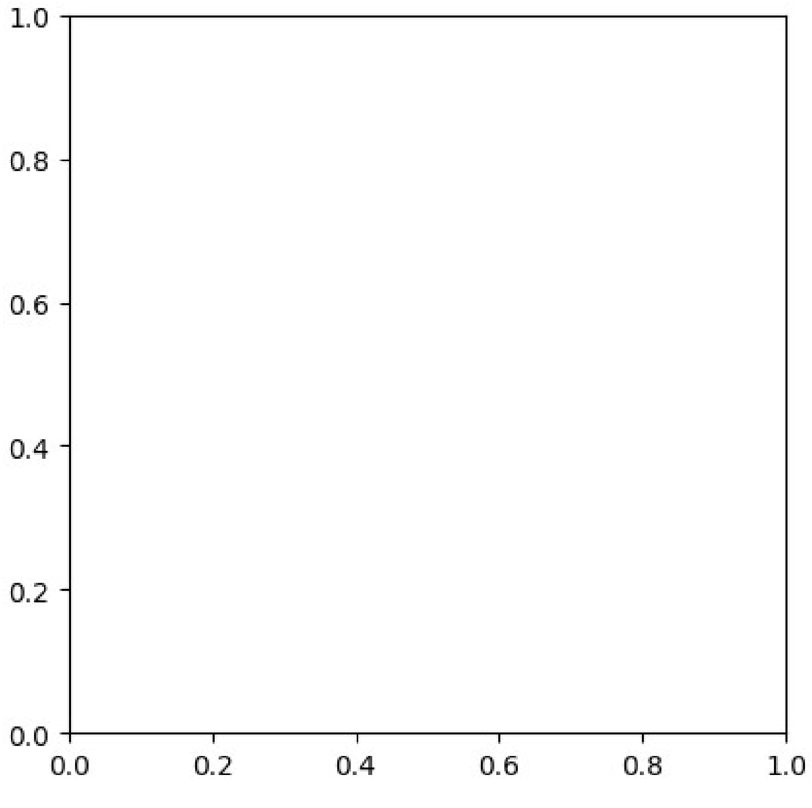


#Now we want to see above values in terms of percentage

```
plt.pie(df['Churn'])
plt.show()

----------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
Cell In[23], line 1
----> 1 plt.pie(df['Churn'])
      2 plt.show()

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\
matplotlib\pyplot.py:3762, in pie(x, explode, labels, colors, autopct,
pctdistance, shadow, labeldistance, startangle, radius, counterclock,
wedgeprops, textprops, center, frame, rotatelabels, normalize, hatch,
data)
```

```
   3739 @_copy_docstring_and_deprecators(Axes.pie)
   3740 def pie(
   3741     x: ArrayLike,
   (...)
   3760     data=None,
   3761 ) -> tuple[list[Wedge], list[Text]] | tuple[list[Wedge],
list[Text], list[Text]]:
-> 3762     return gca().pie(
   3763         x,
   3764         explode=explode,
   3765         labels=labels,
   3766         colors=colors,
   3767         autopct=autopct,
   3768         pctdistance=pctdistance,
   3769         shadow=shadow,
   3770         labeldistance=labeldistance,
   3771         startangle=startangle,
   3772         radius=radius,
   3773         counterclock=counterclock,
   3774         wedgeprops=wedgeprops,
   3775         textprops=textprops,
   3776         center=center,
   3777         frame=frame,
   3778         rotatelabels=rotatelabels,
   3779         normalize=normalize,
   3780         hatch=hatch,
   3781         **({"data": data} if data is not None else {}),
   3782     )

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\
matplotlib\__init__.py:1473, in _preprocess_data.<locals>.inner(ax,
data, *args, **kwargs)
   1470 @functools.wraps(func)
   1471 def inner(ax, *args, data=None, **kwargs):
   1472     if data is None:
-> 1473         return func(
   1474             ax,
   1475             *map(sanitize_sequence, args),
   1476             **{k: sanitize_sequence(v) for k, v in
kwargs.items()})
   1478     bound = new_sig.bind(ax, *args, **kwargs)
   1479     auto_label = (bound.arguments.get(label_namer)
   1480                   or bound.kwargs.get(label_namer))

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\
matplotlib\axes\_axes.py:3280, in Axes.pie(self, x, explode, labels,
colors, autopct, pctdistance, shadow, labeldistance, startangle,
radius, counterclock, wedgeprops, textprops, center, frame,
rotatelabels, normalize, hatch)
```

```
   3277 self.set_aspect('equal')
   3278 # The use of float32 is "historical", but can't be changed
without
   3279 # regenerating the test baselines.
-> 3280 x = np.asarray(x, np.float32)
   3281 if x.ndim > 1:
   3282     raise ValueError("x must be 1D")

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\
pandas\core\series.py:1031, in Series.__array__(self, dtype, copy)
    981 """
    982 Return the values as a NumPy array.
    983 
   (...)
   1028       dtype='datetime64[ns]')
   1029 """
   1030 values = self._values
-> 1031 arr = np.asarray(values, dtype=dtype)
   1032 if using_copy_on_write() and astype_is_view(values.dtype,
arr.dtype):
   1033     arr = arr.view()

ValueError: could not convert string to float: 'No'
```

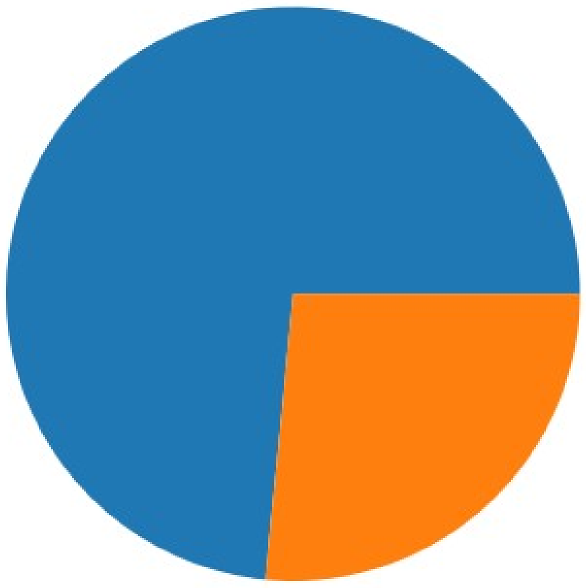


#To solve this issue We do groupby on Churn column and find aggregation count

```
gb= df.groupby("Churn").agg({'Churn': "count"})
```

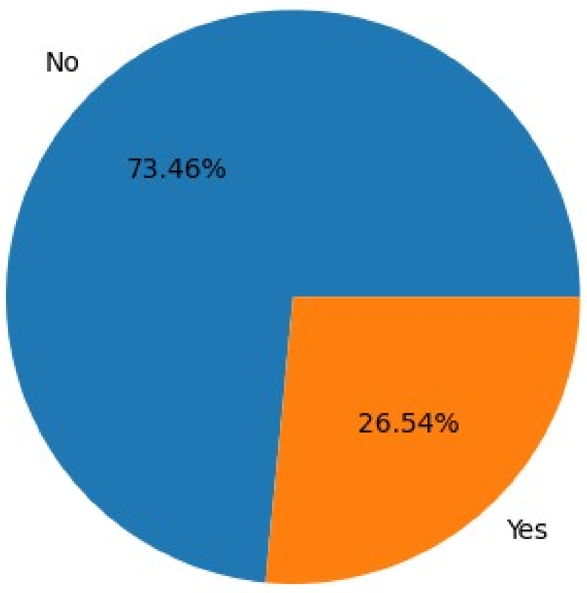
#Now plot this as pie chart

```
plt.pie(gb['Churn'])
plt.show()
```



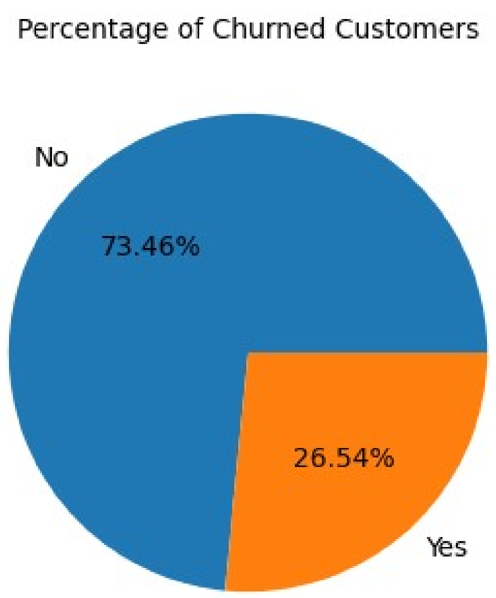
#Modify the above code to give labels to chart

```
gb= df.groupby("Churn").agg({'Churn': "count"})
plt.pie(gb['Churn'], labels=gb.index, autopct="%1.2f%%")
plt.show()
```



#To change the chart size and add title --Modify the code as below

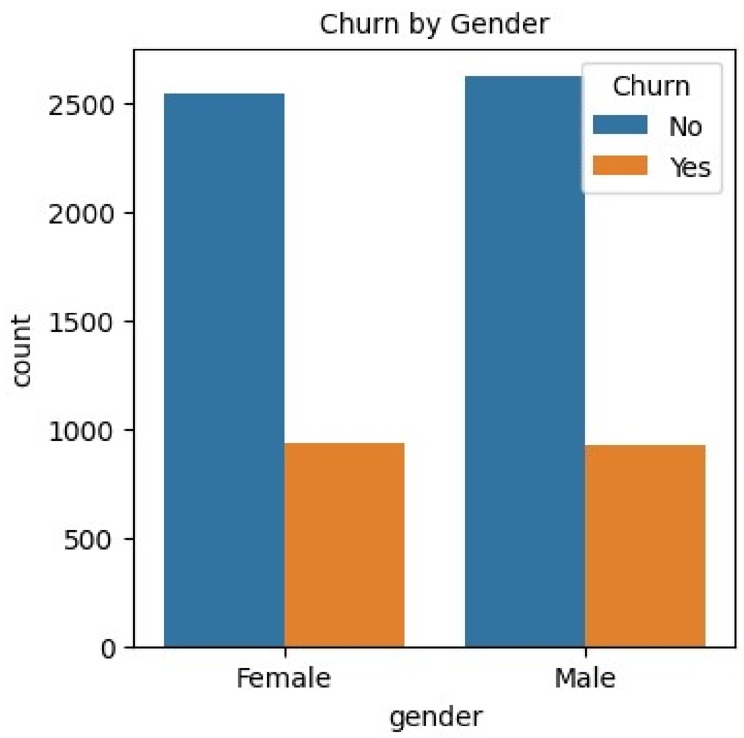
```
plt.figure(figsize=(4,4))
gb= df.groupby("Churn").agg({'Churn': "count"})
plt.pie(gb['Churn'], labels=gb.index, autopct="%1.2f%%")
plt.title("Percentage of Churned Customers", fontsize=10)
plt.show()
```



#From the above pie chart, we can conclude that 26.54% of our customers has churned out

#Churn by Gender

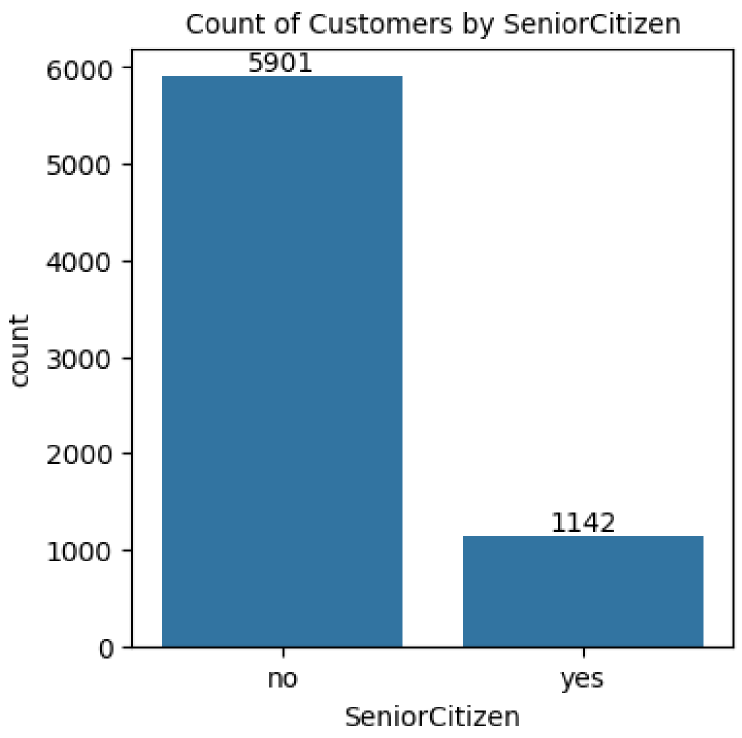
```
plt.figure(figsize=(4,4))
sns.countplot(x='gender', data=df, hue="Churn")
plt.title("Churn by Gender", fontsize=10)
plt.show()
```



#From above column chart we see that equal amount of people are churning out not based on gender specific

#Count of Customers by SeniorCitizen

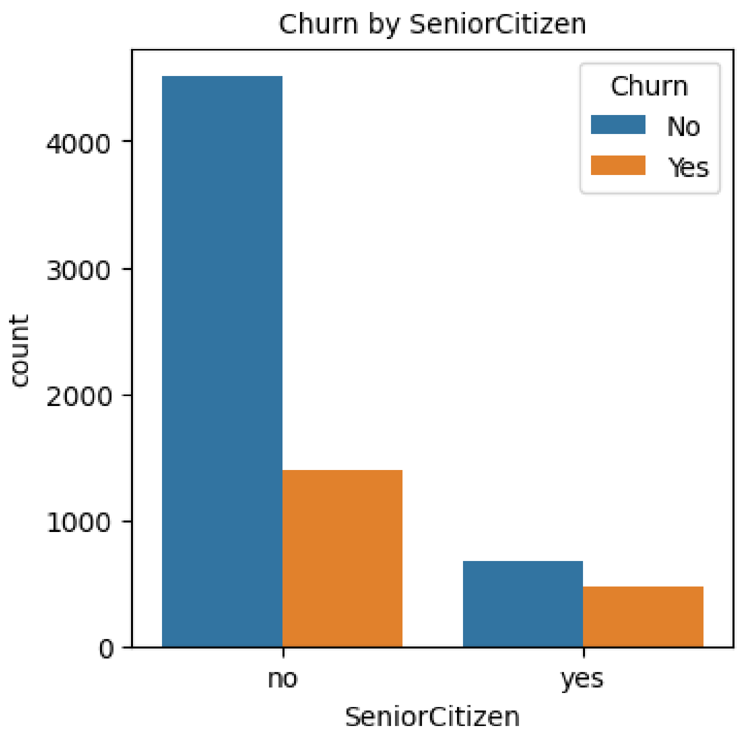
```
plt.figure(figsize=(4,4))
ax=sns.countplot(x='SeniorCitizen', data=df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by SeniorCitizen", fontsize=10)
plt.show()
```



#From above graph we notice that around 1142 customers are senior citizen

#Churn by SeniorCitizen

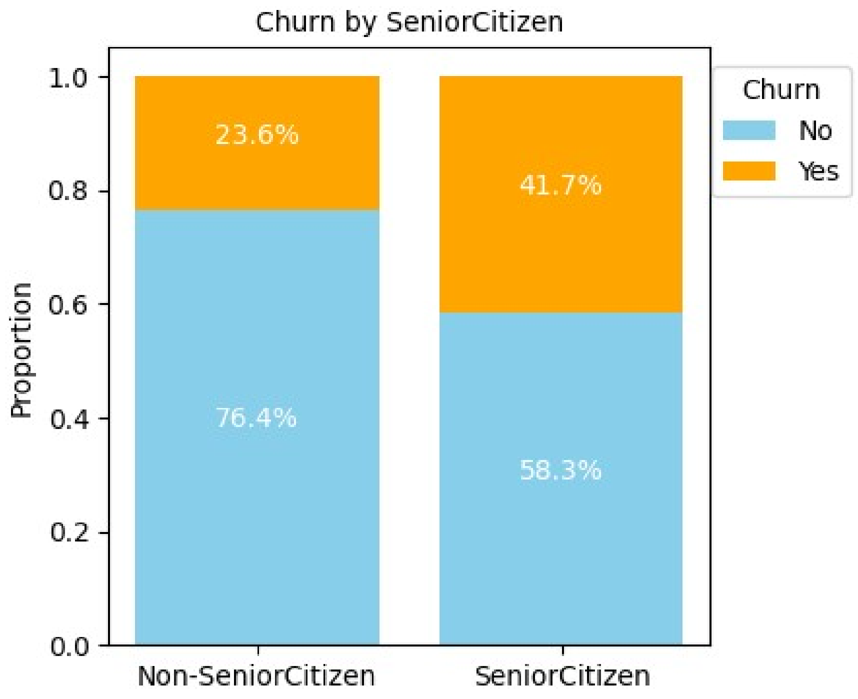
```
plt.figure(figsize=(4,4))
sns.countplot(x='SeniorCitizen', data=df, hue="Churn")
plt.title("Churn by SeniorCitizen", fontsize=10)
plt.show()
```



#We are trying to plot this same graph in stack column chart

```
# Step 1: Calculate counts and percentages
counts = df.groupby(['SeniorCitizen',
'Churn']).size().unstack(fill_value=0)
percentages = counts.div(counts.sum(axis=1), axis=0)

# Use the actual column names instead of 0 and 1
churn_categories = percentages.columns

# Step 2: Plot stacked bar chart
plt.figure(figsize=(4, 4))

# Plot bars for each category (No Churn and Churn)
plt.bar(counts.index, percentages.iloc[:, 0],
label=churn_categories[0], color='skyblue')
plt.bar(counts.index, percentages.iloc[:, 1],
bottom=percentages.iloc[:, 0], label=churn_categories[1],
color='orange')

# Step 3: Add percentage labels
for i in range(len(counts)):
    plt.text(i, percentages.iloc[i, 0] / 2, f'{percentages.iloc[i, 0]
* 100:.1f}%', ha='center', color='white')
    plt.text(i, percentages.iloc[i, 0] + percentages.iloc[i, 1] / 2,
f'{percentages.iloc[i, 1] * 100:.1f}%', ha='center', color='white')
```

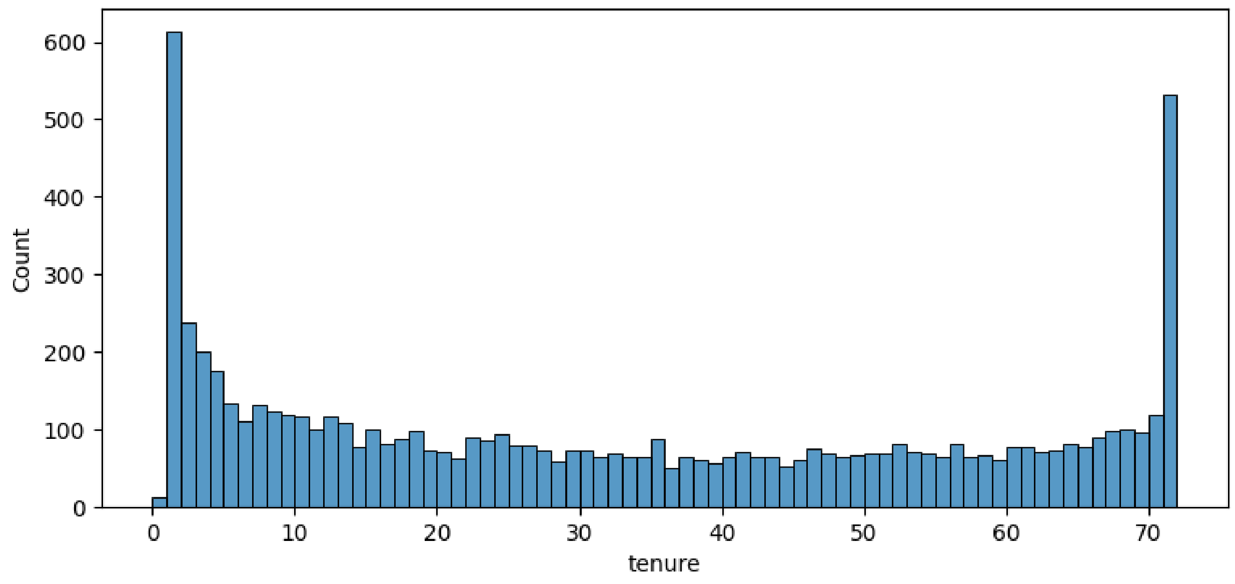
```
# Step 4: Title, labels, and legend
plt.title("Churn by SeniorCitizen", fontsize=10)
plt.xticks(ticks=[0, 1], labels=['Non-SeniorCitizen',
'SeniorCitizen'])
plt.ylabel('Proportion')
plt.legend(title='Churn', bbox_to_anchor=(.98,.99))

# Show plot
plt.show()
```



#Comparitive a greater percentage of people in SeniorCitizen category have churned out

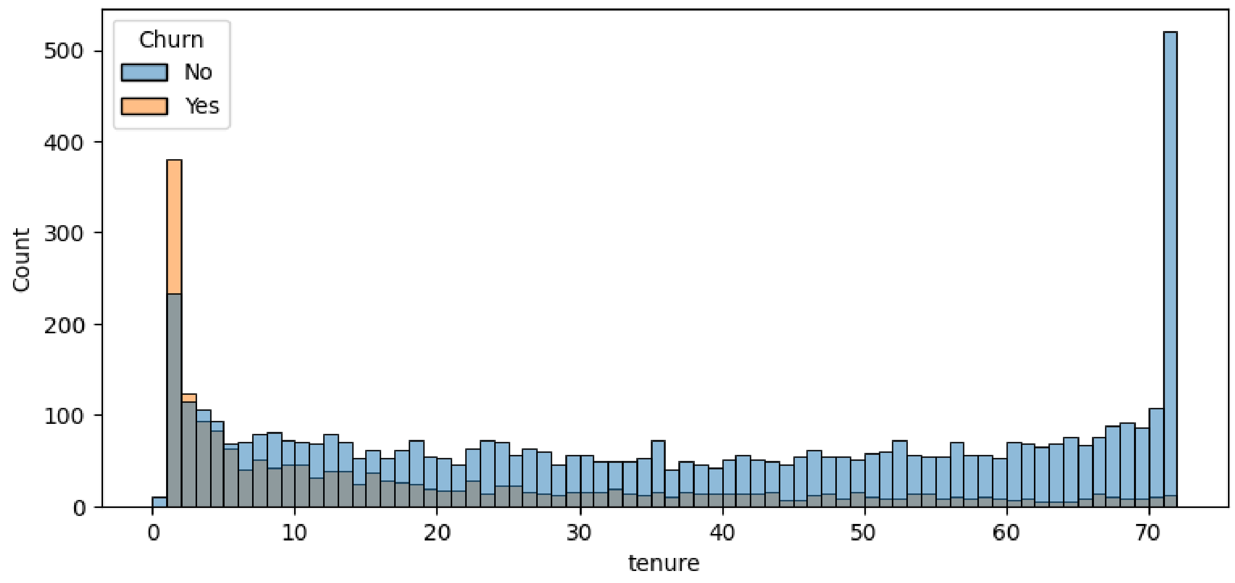
#Ploting histogram - count of customers by tenure--we have used bin size= 72(max)

```
plt.figure(figsize=(9,4))
sns.histplot(x='tenure', data=df, bins=72)
plt.show()
```



#Modify the above code by adding hue='Churn'

```
plt.figure(figsize=(9,4))
sns.histplot(x='tenure', data=df, bins=72, hue='Churn' )
plt.show()
```

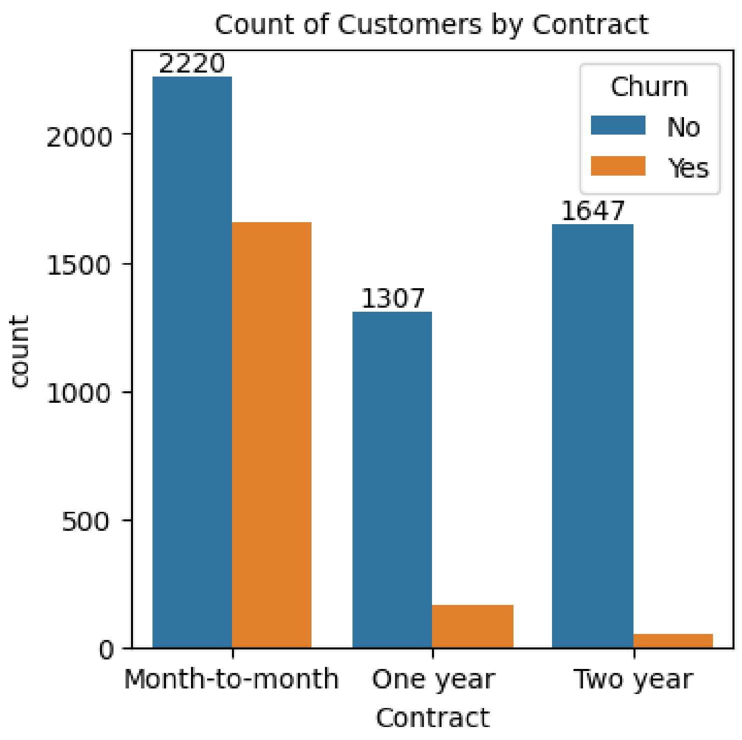


#From the graph we get to know that people who have used our services for a long time have stayed and people who have used our services for 1 or 2 months have churned out

#Count of Customers by Contract

```
plt.figure(figsize=(4,4))
ax=sns.countplot(x='Contract', data=df,hue= 'Churn')
```

```
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Contract", fontsize=10)
plt.show()
```



#From above graph, we can conclude that most customers with monthly contracts are likely to churned out as compared to others with 1 or 2 years contract

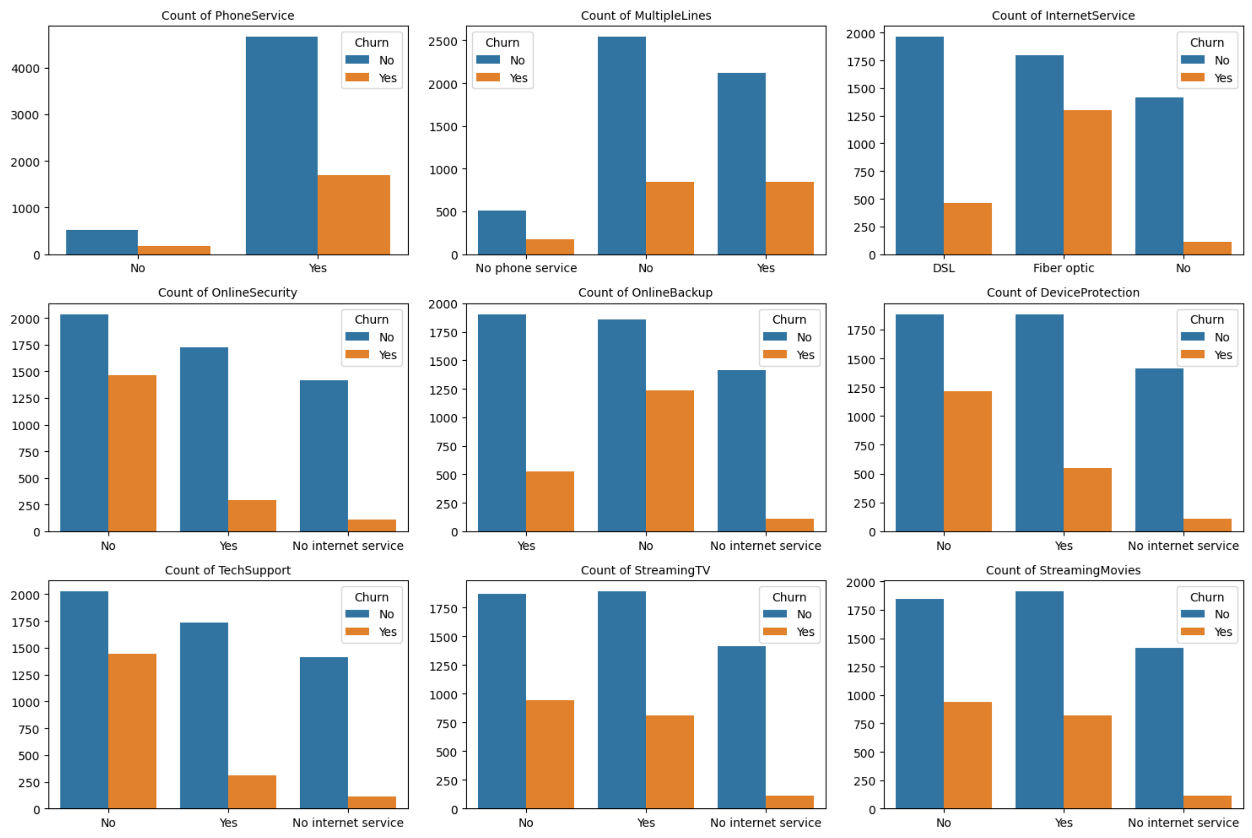
#We are trying to get all columns

```
df.columns.values

array(['customerID', 'gender', 'SeniorCitizen', 'Partner',
'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

#We want to see the characteristics of each services provided by company

```
# Columns to plot
columns = ['PhoneService', 'MultipleLines', 'InternetService',
           'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
           'TechSupport', 'StreamingTV', 'StreamingMovies']
```

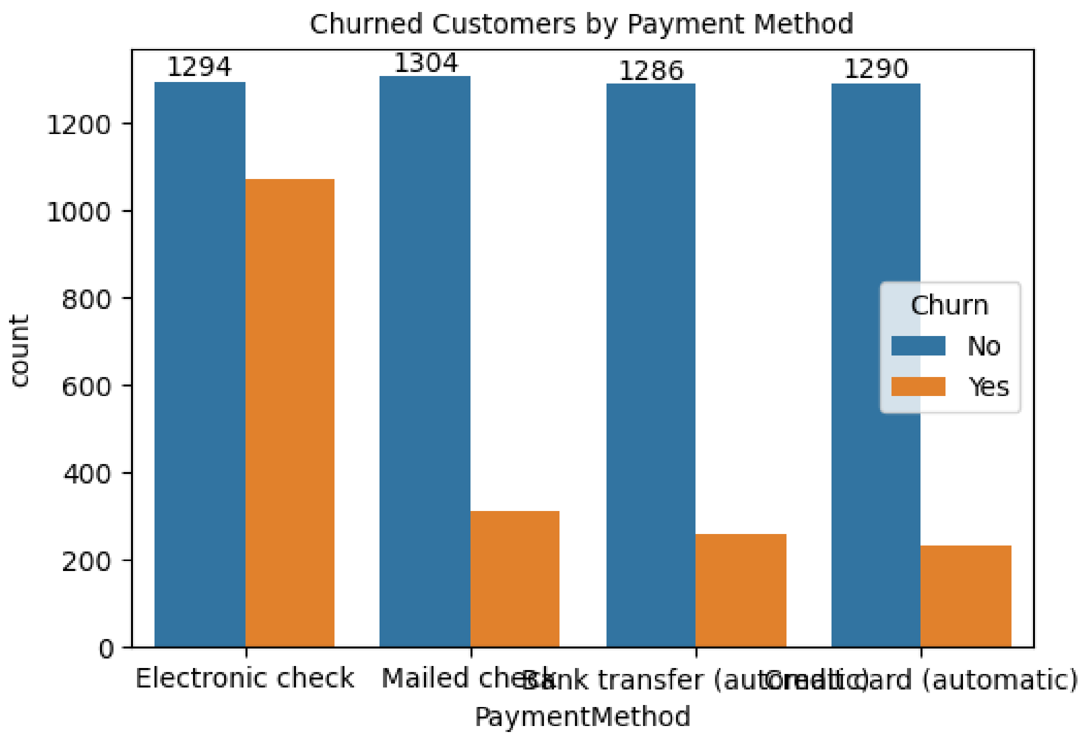
```python
# Step 1: Set up the subplot grid
n_cols = 3  # Number of columns in the subplot grid
n_rows = len(columns) // n_cols + (len(columns) % n_cols > 0)  # Rows
based on the number of columns
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, 10))  # Adjust
figsize as needed
axes = axes.flatten()  # Flatten axes array to easily iterate over it

# Step 2: Loop through each column and create a countplot for each
for i, col in enumerate(columns):
    sns.countplot(x=col, data=df, ax=axes[i], hue='Churn')
    axes[i].set_title(f'Count of {col}', fontsize=10)
    axes[i].set_xlabel('')  # Optional: remove x-axis label for
cleaner look
    axes[i].set_ylabel('')  # Optional: remove y-axis label for
cleaner look

# Step 3: Remove any empty subplots (if the number of plots is not a
perfect grid)
for i in range(len(columns), len(axes)):
    fig.delaxes(axes[i])

# Step 4: Adjust the layout and display
plt.tight_layout()
plt.show()
```



#From above plots, we can conclude the following-- InternetService: Customers with fiber optic service show a higher churn rate compared to those using DSL or no internet service. OnlineSecurity, OnlineBackup, TechSupport: Customers without these services are more likely to churn, while those with these services show lower churn rates. PhoneService and MultipleLines: While most customers have phone services, churn rates are similar between customers with and without multiple lines. Overall, lack of internet-related services like security, backup, and tech support appears to be associated with higher churn rates.

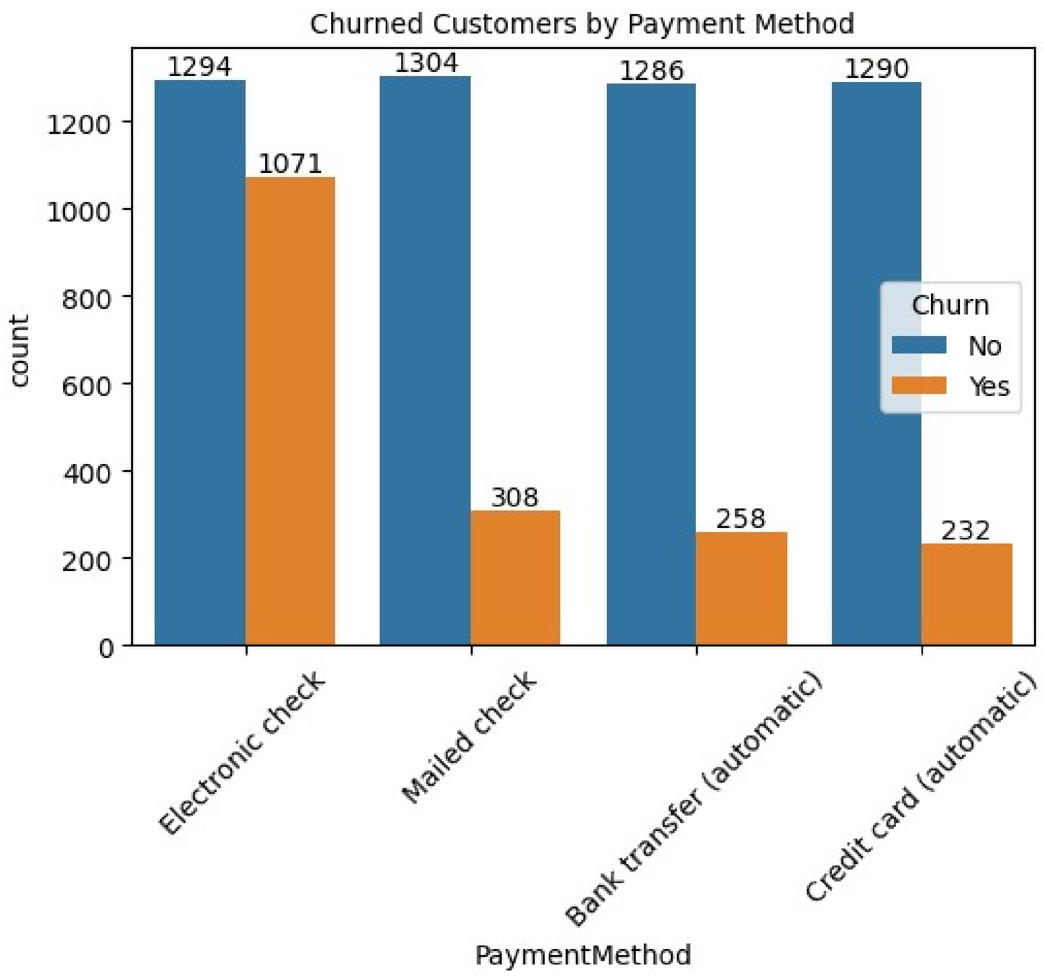
#Churned customers by PaymentMethod

```
plt.figure(figsize=(6,4))
ax=sns.countplot(x='PaymentMethod', data=df,hue= 'Churn')
ax.bar_label(ax.containers[0])
plt.title("Churned Customers by Payment Method", fontsize=10)
plt.show()
```



#We can see that x labels are over written_ so to correct it modify the above code as below

```
plt.figure(figsize=(6,4))
ax=sns.countplot(x='PaymentMethod', data=df,hue= 'Churn')
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.xticks(rotation=45)
plt.title("Churned Customers by Payment Method", fontsize=10)
plt.show()
```



#From above graph we can conclude that customers are likely to churned out when they are using electronic check as a payment method