

```
In [1]: pip install pandas
!pip install matplotlib
!pip install numpy
!pip install seaborn

Requirement already satisfied: pandas in
c:\python\python312\lib\site-packages (2.2.
3)
Requirement already satisfied: numpy>=1.26.
0 in c:\python\python312\lib\site-packages
(from pandas) (2.2.2)
Requirement already satisfied: python-dateu
til>=2.8.2 in c:\python\python312\lib\site-pa
ckages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1
in c:\python\python312\lib\site-packages (f
rom pandas) (2025.1)
Requirement already satisfied: tzdata>=202
2.7 in c:\python\python312\lib\site-packag
es (from pandas) (2025.1)
Requirement already satisfied: six>=1.5 in
c:\python\python312\lib\site-packages (from
python-dateutil>=2.7->matplotlib) (1.17.0)
Requirement already satisfied: matplotlib
in c:\python\python312\lib\site-packages (0.1
3.10.0)
Requirement already satisfied: contourpy>=
1.0.1 in c:\python\python312\lib\site-packa
ges (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.1
0 in c:\python\python312\lib\site-packages (f
rom matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=
4.22.0 in c:\python\python312\lib\site-pack
ages (from matplotlib) (4.55.8)
Requirement already satisfied: kiwisolver>=
1.3.1 in c:\python\python312\lib\site-packa
ges (from matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23
in c:\python\python312\lib\site-packages (f
rom matplotlib) (2.2.2)
Requirement already satisfied: packaging>=2
0.0 in c:\python\python312\lib\site-packag
es (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in
c:\python\python312\lib\site-packages (from
matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=
2.3.1 in c:\python\python312\lib\site-packa
ges (from matplotlib) (3.2.1)
Requirement already satisfied: python-dateu
til>=2.7 in c:\python\python312\lib\site-pa
ckages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in
c:\python\python312\lib\site-packages (from
python-dateutil>=2.7->matplotlib) (1.17.0)
Requirement already satisfied: numpy in
c:\python\python312\lib\site-packages (2.2.
2)
Requirement already satisfied: seaborn in
c:\python\python312\lib\site-packages (0.1
3.2)
Requirement already satisfied: numpy!=1.24.
0,>=1.20 in c:\python\python312\lib\site-pa
ckages (from seaborn) (2.2.2)
Requirement already satisfied: pandas>=1.2
in c:\python\python312\lib\site-packages (f
rom seaborn) (2.2.3)
Requirement already satisfied: matplotlib!=
3.6.1,>=3.4 in c:\python\python312\lib\site
-packages (from seaborn) (3.10.0)
Requirement already satisfied: contourpy>=
1.0.1 in c:\python\python312\lib\site-packa
ges (from matplotlib!=3.6.1,>=3.4->seaborn)
(1.3.1)
Requirement already satisfied: cycler>=0.1
0 in c:\python\python312\lib\site-packages (f
rom matplotlib!=3.6.1,>=3.4->seaborn) (0.1
2.1)
Requirement already satisfied: fonttools>=
4.22.0 in c:\python\python312\lib\site-pack
ages (from matplotlib!=3.6.1,>=3.4->seabo
rn) (4.55.8)
Requirement already satisfied: kiwisolver>=
1.3.1 in c:\python\python312\lib\site-packa
ges (from matplotlib!=3.6.1,>=3.4->seabo
rn) (1.4.8)
Requirement already satisfied: packaging>=2
0.0 in c:\python\python312\lib\site-packag
es (from matplotlib!=3.6.1,>=3.4->seaborn)
(24.2)
Requirement already satisfied: pillow>=8 in
c:\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (11.1.0)
Requirement already satisfied: pyparsing>=
2.3.1 in c:\python\python312\lib\site-packa
ges (from matplotlib!=3.6.1,>=3.4->seabo
rn) (3.2.1)
Requirement already satisfied: python-dateu
til>=2.7 in c:\python\python312\lib\site-pa
ckages (from matplotlib!=3.6.1,>=3.4->seab
orn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1
in c:\python\python312\lib\site-packages (f
rom pandas) (2025.1)
Requirement already satisfied: tzdata>=202
2.7 in c:\python\python312\lib\site-packag
es (from pandas) (2025.1)
Requirement already satisfied: six>=1.5 in
c:\python\python312\lib\site-packages (from
python-dateutil>=2.7->matplotlib) (1.17.0)
Requirement already satisfied: numpy>=1.26
in c:\python\python312\lib\site-packages (f
rom pandas) (2.2.2)

In [20]:
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

In [4]:
df = pd.read_csv('Customer Churn.csv')

In [11]:
df

Out[11]:
   customerID  gender  SeniorCitizen  Partner  D
0          7590-  Female             0      Yes
   VHVVEG
1          5575-  Male             0      No
   GNVDE
2          3668-  Male             0      No
   QPYBK
3          7795-  Male             0      No
   CFOWC
4          9237-  Female            0      No
   HQITU
...         ...
7038        6840-  Male             0      Yes
   RESVB
7039        2234-  Female            0      Yes
   XADUH
7040        4801-  Female            0      Yes
   JZAZL
7041        8361-  Male             1      Yes
   LTMKD
7042        3186-  Male             0      No
   AJIEK
7043 rows x 21 columns

In [5]:
df.head()

Out[5]:
   customerID  gender  SeniorCitizen  Partner  Depen
0          7590-  Female             0      Yes
   VHVVEG
1          5575-  Male             0      No
   GNVDE
2          3668-  Male             0      No
   QPYBK
3          7795-  Male             0      No
   CFOWC
4          9237-  Female            0      No
   HQITU
5 rows x 21 columns

In [13]:
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(18)
memory usage: 1.1+ MB
```

replacing the blanks as 0 having tenure is 0

```
In [6]:
df["TotalCharges"] = df["TotalCharges"].rep
df["TotalCharges"] = df["TotalCharges"].ast

In [7]:
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

In [8]:
df.isnull().sum().sum()

Out[8]:
np.int64(0)

In [9]:
df.describe()

Out[9]:
   SeniorCitizen      tenure  MonthlyCharges
count  7043.000000  7043.000000  7043.000000
mean      0.162147    32.371149    64.761692
std       0.368612    24.559481    30.090047
min       0.000000    0.000000    18.250000
25%       0.000000    9.000000    35.500000
50%       0.000000    29.000000    70.350000
75%       0.000000    55.000000    89.850000
max       1.000000    72.000000   118.750000

In [10]:
df["customerID"].duplicated().sum()

Out[10]:
np.int64(0)

Covering 0 and 1 values of senior citizen to yes
or no to make it easier to understand

In [11]:
def conv(value):
    if value==1:
        return "yes"
    else:
        return "no"
df["SeniorCitizen"] = df["SeniorCitizen"].a

In [32]:
df.head(30)

Out[32]:
   customerID  gender  SeniorCitizen  Partner  Dep
0          7590-  Female             no      Yes
   VHVVEG
1          5575-  Male             no      No
   GNVDE
2          3668-  Male             no      No
   QPYBK
3          7795-  Male             no      No
   CFOWC
4          9237-  Female            no      No
   HQITU
5          9305-  Female             no      No
   CDSKC
6          1452-  Male             no      No
   KIOVK
7          6713-  Female             no      No
   OKOMC
8          7892-  Female             no      Yes
   POOKP
9          6388-  Male             no      No
   TABGU
10         9763-  Male             no      Yes
   GRSKD
11         7469-  Male             no      No
   LKBCI
12         8091-  Male             no      Yes
   TTVAX
13         0280-  Male             no      No
   XJGEX
14         5129-  Male             no      No
   JLPIS
15         3655-  Female             no      Yes
   SNQYZ
16         8191-  Female             no      No
   XWSZG
17         9959-  Male             no      No
   WOFKT
18         4190-  Female             no      Yes
   MFLUW
19         4183-  Female             no      No
   MYFRB
20         8779-  Male             yes      No
   QRDMV
21         1680-  Male             no      Yes
   VDCWW
22         1066-  Male             no      No
   JKSGK
23         3638-  Female             no      Yes
   WEABW
24         6322-  Male             no      Yes
   HRPFA
25         6865-  Female             no      No
   JZSKO
26         6467-  Male             no      Yes
   CHFZW
27         8665-  Male             no      Yes
   UTDHZ
28         5248-  Male             no      Yes
   YGJNJ
29         8773-  Female             no      No
   HHUOZ
30 rows x 21 columns

In [35]:
ax = sns.countplot(x = 'Churn', data = df)
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Churn")
plt.show()

Count of Customers by Churn

In [39]:
plt.figure(figsize = (3,4))
gb = df.groupby("Churn").agg(['Churn':"count
plt.pie(gb["Churn"], labels = gb.index, aut
plt.title("Percentage of Churned Customer
plt.show()

Percentage of Churned Customeres

from the given pie chart
we can conclude that
26.54% of our
customers have
churned out

Now let's explore the
reason behind it.

In [49]:
plt.figure(figsize = (3,3))
sns.countplot(x = df["gender"], data= df, h
plt.title("Churn by Gender")
plt.show()

Churn by Gender

In [50]:
plt.figure(figsize = (3,3))
sns.countplot(x = df["SeniorCitizen"], data
plt.title("Churn by Gender")
plt.show()

Churn by Gender

In [59]:
plt.figure(figsize = (4,4))
ax = sns.countplot(x = 'SeniorCitizen', dat
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Senior Cit
plt.show()

Count of Customers by Senior Citizen

In [60]:
grouped = df.groupby(["SeniorCitizen", "Chu
grouped = grouped.div(grouped.sum(axis=1),
# Plot
fig, ax = plt.subplots(figsize=(4, 4))
grouped.plot(kind="bar", stacked=True, colo
# Add percentage labels
for i, bars in enumerate(ax.containers):
    for bar in bars:
        height = bar.get_height()
        if height > 0: # Avoid showing lab
            ax.text(
                bar.get_x() + bar.get_width
                bar.get_y() + height / 2,
                f"{height:.1f}%", # Format
                ha="center", va="center", f

# Labels & Title
plt.xlabel("Senior Citizen")
plt.ylabel("Percentage")
plt.title("Churn by Senior Citizen (Stacked
plt.legend(title="Churn", bbox_to_anchor =
plt.show()

Churn by Senior Citizen (Stacked %)

comparative a greater
percentage of people in
senior citizen category
have churned

In [68]:
plt.figure(figsize = (9,4))
sns.histplot(x = "tenure", data = df, bins
plt.show()

People who have used
our services for a long
time have stayed and
people who have used
our sevices #1 or 2
months have churned

In [71]:
plt.figure(figsize = (4,4))
ax = sns.countplot(x = 'Contract', data = d
ax.bar_label(ax.containers[0])
plt.title("Count of Customers by Contract")
plt.show()

Count of Customers by Contract

People those who are
having only the monthly
Contract they are more
likly to Churn compare
to those who had taken
the one year or two
year contact

In [73]:
df.columns.values

Out[73]:
array(['customerID', 'gender', 'SeniorCitiz
en', 'Partner', 'Dependents',
'tenure', 'PhoneService', 'MultipleL
ines', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'D
eviceProtection', 'Contract', 'Stream
ingMovies', 'Contract',
'PaperlessBilling', 'PaymentMethod',
'TotalCharges', 'Churn'], dtype=object)

In [78]:
fig, axes = plt.subplots(3, 3, figsize=(15,
axes = axes.flatten()

# Generate countplots
for i, col in enumerate(columns):
    sns.countplot(data=df, x=col, hue=df["C
    axes[i].set_title(f'Countplot of {col}')
    axes[i].set_xlabel('')
    axes[i].tick_params(axis='x', rotation=

Count of Dependents
Count of MultipleLines
Count of OnlineSecurity
Count of OnlineBackup
Count of DeviceProtection
Count of TechSupport
Count of StreamingTV
Count of StreamingMovies
Count of Contract
Count of PaperlessBilling
Count of PaymentMethod
Count of MonthlyCharges
Count of TotalCharges
Count of Churn

Across various service categories like Phone
Service, Internet Service, Online Security, Tech
Support, and Streaming Services. In most
categories, a significant number of customers
have opted for "No" services, while those with
services are split between the two groups.
Internet-related services show a strong presence
of "No internet service" responses, suggesting a
portion of customers do not use internet-based
features.

In [83]:
plt.figure(figsize = (6,4))
ax = sns.countplot(x = 'PaymentMethod', dat
ax.bar_label(ax.containers[1])
plt.title("Count of Customers by PaymentMet
plt.xticks(rotation =40)
plt.show()

Count of Customers by PaymentMethod

Customers are likely to
churn when they are
using electroic check
as a payment method

In [ ]:

```