In [1]:
```
pip install pandas
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandas in c:\users\pc\appdata\roaming\python\python313\site-packages (2.2.3)
Requirement already satisfied: numpy>=1.26.0 in c:\users\pc\appdata\roaming\python\python313\site-packages (from
pandas) (2.2.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\pc\appdata\roaming\python\python313\site-packa
ges (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\pc\appdata\roaming\python\python313\site-packages (from
pandas) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\pc\appdata\roaming\python\python313\site-packages (fro
m pandas) (2025.1)
Requirement already satisfied: six>=1.5 in c:\users\pc\appdata\roaming\python\python313\site-packages (from pyth
on-dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [2]:
```
pip install matplotlib
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: matplotlib in c:\users\pc\appdata\roaming\python\python313\site-packages (3.10.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\pc\appdata\roaming\python\python313\site-packages (f
rom matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\pc\appdata\roaming\python\python313\site-packages (from
matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\pc\appdata\roaming\python\python313\site-packages (
from matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\pc\appdata\roaming\python\python313\site-packages (
from matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in c:\users\pc\appdata\roaming\python\python313\site-packages (from m
atplotlib) (2.2.3)
Requirement already satisfied: packaging>=20.0 in c:\users\pc\appdata\roaming\python\python313\site-packages (fr
om matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in c:\users\pc\appdata\roaming\python\python313\site-packages (from mat
plotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\pc\appdata\roaming\python\python313\site-packages (f
rom matplotlib) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\pc\appdata\roaming\python\python313\site-package
s (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\pc\appdata\roaming\python\python313\site-packages (from pyth
on-dateutil>=2.7->matplotlib) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [3]:
```
pip install seaborn
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: seaborn in c:\users\pc\appdata\roaming\python\python313\site-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\pc\appdata\roaming\python\python313\site-package
s (from seaborn) (2.2.3)
Requirement already satisfied: pandas>=1.2 in c:\users\pc\appdata\roaming\python\python313\site-packages (from s
eaborn) (2.2.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\pc\appdata\roaming\python\python313\site-pack
ages (from seaborn) (3.10.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\pc\appdata\roaming\python\python313\site-packages (f
rom matplotlib!=3.6.1,>=3.4->seaborn) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\pc\appdata\roaming\python\python313\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\pc\appdata\roaming\python\python313\site-packages (
from matplotlib!=3.6.1,>=3.4->seaborn) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\pc\appdata\roaming\python\python313\site-packages (
from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.8)
Requirement already satisfied: packaging>=20.0 in c:\users\pc\appdata\roaming\python\python313\site-packages (fr
om matplotlib!=3.6.1,>=3.4->seaborn) (24.2)
Requirement already satisfied: pillow>=8 in c:\users\pc\appdata\roaming\python\python313\site-packages (from mat
plotlib!=3.6.1,>=3.4->seaborn) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\pc\appdata\roaming\python\python313\site-packages (f
rom matplotlib!=3.6.1,>=3.4->seaborn) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\pc\appdata\roaming\python\python313\site-package
s (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\pc\appdata\roaming\python\python313\site-packages (from
pandas>=1.2->seaborn) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\pc\appdata\roaming\python\python313\site-packages (fro
m pandas>=1.2->seaborn) (2025.1)
Requirement already satisfied: six>=1.5 in c:\users\pc\appdata\roaming\python\python313\site-packages (from pyth
on-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [4]:

```
pip install numpy
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in c:\users\pc\appdata\roaming\python\python313\site-packages (2.2.3)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [28]:

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

In [6]:

```
df= pd.read_csv('C:/Users/PC/Desktop/AIML and DA/customer churn.csv')
df
```

Out[6]:

	customerID	tenure	PhoneService	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	7590-VHVEG	1	No	Month-to-month	Yes	Electronic check	29.85	29.85	No
1	5575-GNVDE	34	Yes	One year	No	Mailed check	56.95	1889.5	No
2	3668-QPYBK	2	Yes	Month-to-month	Yes	Mailed check	53.85	108.15	Yes
3	7795-CFOCW	45	No	One year	No	Bank transfer (automatic)	42.30	1840.75	No
4	9237-HQITU	2	Yes	Month-to-month	Yes	Electronic check	70.70	151.65	Yes
...	...	...	...	...	...	...	...	...	...
7037	2569-WGERO	72	Yes	Two year	Yes	Bank transfer (automatic)	21.15	1419.4	No
7038	6840-RESVB	24	Yes	One year	Yes	Mailed check	84.80	1990.5	No
7039	2234-XADUH	72	Yes	One year	Yes	Credit card (automatic)	103.20	7362.9	No
7040	4801-JZAZL	11	No	Month-to-month	Yes	Electronic check	29.60	346.45	No
7041	8361-LTMKD	4	Yes	Month-to-month	Yes	Mailed check	74.40	306.6	Yes

7042 rows × 9 columns

In [10]:

```
df.head()
```

Out[10]:

	customerID	tenure	PhoneService	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	7590-VHVEG	1	No	Month-to-month	Yes	Electronic check	29.85	29.85	No
1	5575-GNVDE	34	Yes	One year	No	Mailed check	56.95	1889.5	No
2	3668-QPYBK	2	Yes	Month-to-month	Yes	Mailed check	53.85	108.15	Yes
3	7795-CFOCW	45	No	One year	No	Bank transfer (automatic)	42.30	1840.75	No
4	9237-HQITU	2	Yes	Month-to-month	Yes	Electronic check	70.70	151.65	Yes

inspection of data

In [11]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7042 entries, 0 to 7041
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7042 non-null   object 
 1   tenure            7042 non-null   int64  
 2   PhoneService      7042 non-null   object 
 3   Contract          7042 non-null   object 
 4   PaperlessBilling  7042 non-null   object 
 5   PaymentMethod     7042 non-null   object 
 6   MonthlyCharges    7042 non-null   float64
 7   TotalCharges      7042 non-null   object 
 8   Churn             7042 non-null   object 
dtypes: float64(1), int64(1), object(7)
memory usage: 495.3+ KB
```

replacing blanks with 0 as tenure is 0 and no total charges are replaced

also the data type of total charges converted to object to float

In [17]:

```
df['TotalCharges'] = df['TotalCharges'].replace(' ','0')
df['TotalCharges'] = df['TotalCharges'].astype('float')
```

In [15]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7042 entries, 0 to 7041
Data columns (total 9 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7042 non-null   object 
 1   tenure            7042 non-null   int64  
 2   PhoneService      7042 non-null   object 
 3   Contract          7042 non-null   object 
 4   PaperlessBilling  7042 non-null   object 
 5   PaymentMethod     7042 non-null   object 
 6   MonthlyCharges    7042 non-null   float64
 7   TotalCharges      7042 non-null   float64
 8   Churn             7042 non-null   object 
dtypes: float64(2), int64(1), object(6)
memory usage: 495.3+ KB
```

In [20]:

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

Out[20]:

```
customerID          0
tenure              0
PhoneService        0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

In [21]:

```
df.describe()
```

Out[21]:

	tenure	MonthlyCharges	TotalCharges
count	7042.000000	7042.000000	7042.000000
mean	32.366373	64.755886	2279.086083
std	24.557955	30.088238	2266.302524
min	0.000000	18.250000	0.000000
25%	9.000000	35.500000	398.550000
50%	29.000000	70.350000	1394.075000
75%	55.000000	89.850000	3783.600000
max	72.000000	118.750000	8684.800000

In [23]:

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

Out[23]:

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

In [27]:

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

Out[27]:

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

df conv(value):

```
#if value == 1:
   #return 'yes'
#else:
    #return 'no'
```

df['Columnname'] = df['Columnname'].apply(conv)

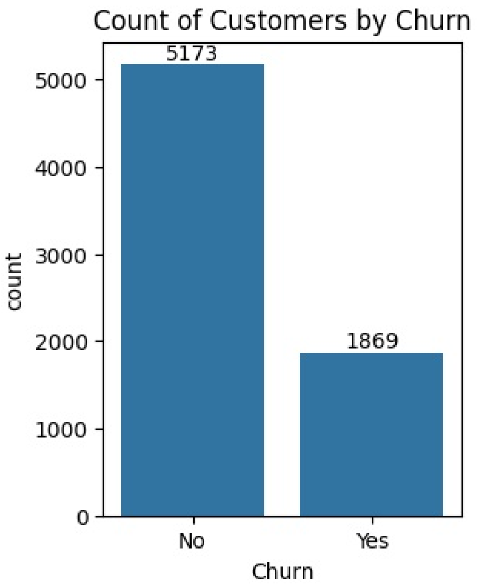
converted 0 and 1 value coulnm to yes/no to make it easier to understand

understanding the churn

In [13]:

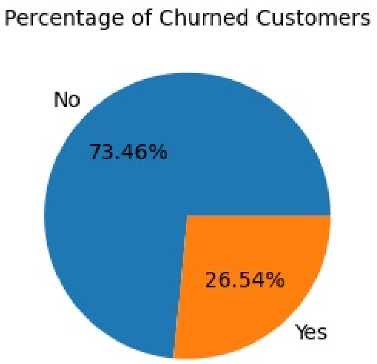
```python
plt.figure(figsize = (3,4))
ax = sns.countplot(x = 'Churn', data = df)

ax.bar_label(ax.containers[0])
plt.title('Count of Customers by Churn')
plt.show()
```



In [24]:

```python
plt.figure(figsize = (3,4))
gb = df.groupby("Churn").agg({"Churn":'count'})
plt.title('Percentage of Churned Customers', fontsize = 10)
plt.pie(gb['Churn'], labels = gb.index, autopct = '%1.2f%%')
plt.show()
```



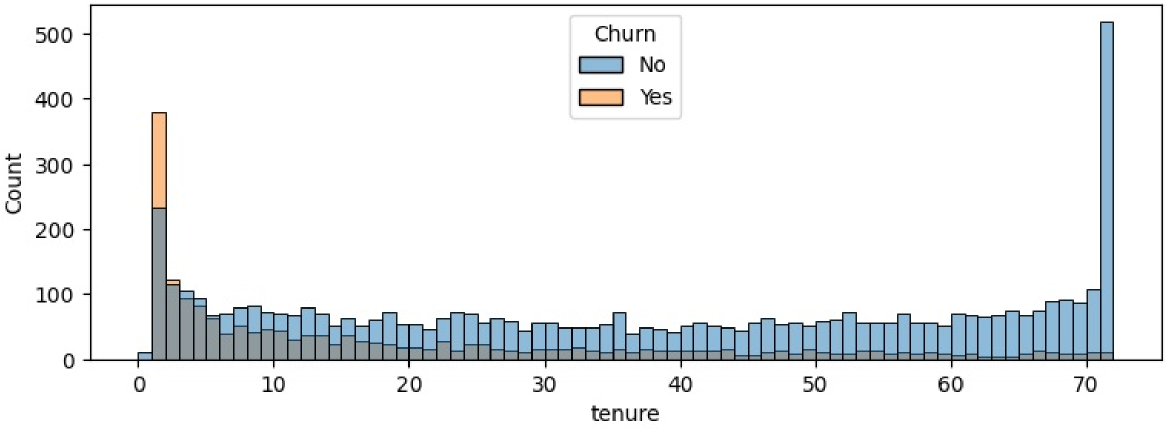
from the given pie chart we can conclude that 26.54% of our

customers are churned out

let's explore reason behind it.

In [17]:

```
plt.figure(figsize = (9,3))
sns.histplot(x = 'tenure', data= df , bins = 72, hue = 'Churn')
plt.show()
```

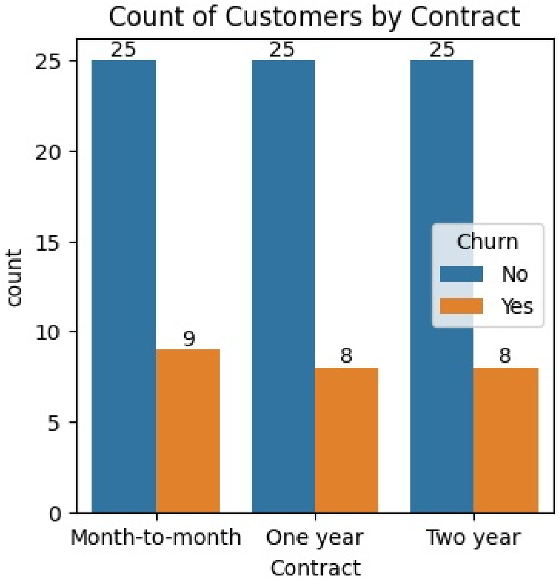


people who have used our services for a long time have stayed and people who have used our services #1 or 2 months have churned

In [86]:

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

ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title('Count of Customers by Contract')
plt.show()
```



people who have contract of month to month are likely to churn as compared to one year or two year plan

In [31]:

```
df.columns.values
```

Out[31]:

```
array(['customerID', 'tenure', 'PhoneService', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

In [64]:

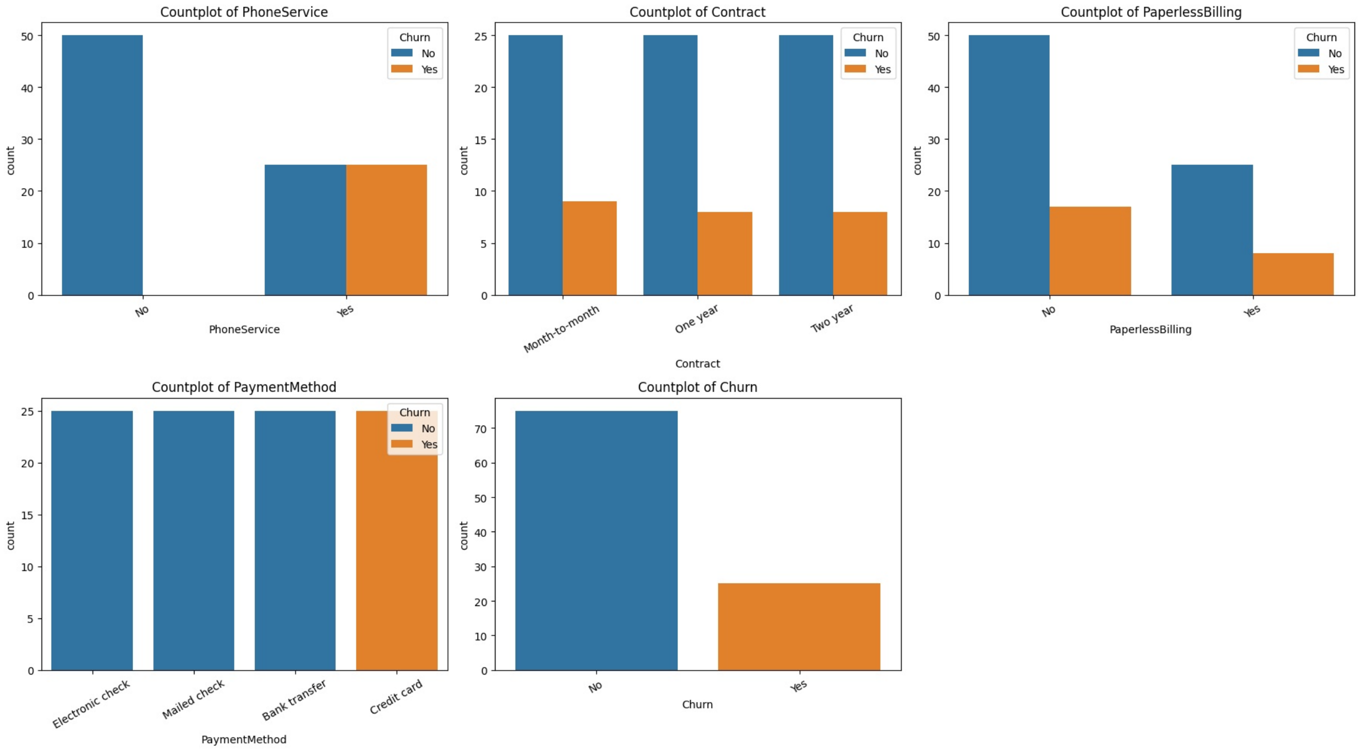
```
# Sample Data (Replace this with your actual dataset)
data = {
    'customerID': range(1, 101),
    'tenure': [i % 12 for i in range(1, 101)],
    'PhoneService': ['Yes' if i % 2 == 0 else 'No' for i in range(1, 101)],
    'Contract': ['Month-to-month', 'One year', 'Two year'] * 33 + ['Month-to-month'],
    'PaperlessBilling': ['Yes' if i % 3 == 0 else 'No' for i in range(1, 101)],
    'PaymentMethod': ['Electronic check', 'Mailed check', 'Bank transfer', 'Credit card'] * 25,
    'MonthlyCharges': [round(50 + (i % 10) * 5, 2) for i in range(1, 101)],
    'TotalCharges': [round(500 + (i * 10), 2) for i in range(1, 101)],
    'Churn': ['Yes' if i % 4 == 0 else 'No' for i in range(1, 101)]
}
df = pd.DataFrame(data)

# List of categorical columns for count plots
categorical_columns = ['PhoneService', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'Churn']

# Create subplots
fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(18, 10))
axes = axes.flatten()

# Generate count plots
for i, col in enumerate(categorical_columns):
    sns.countplot(data=df, x=col, hue='Churn', ax=axes[i])
    axes[i].set_title(f'Countplot of {col}')
    axes[i].tick_params(axis='x', rotation=30)

# Remove empty subplot if needed
fig.delaxes(axes[-1])
plt.tight_layout()
plt.show()
```


Countplot of PhoneService

Countplot of Contract

Countplot of PaperlessBilling

Countplot of PaymentMethod

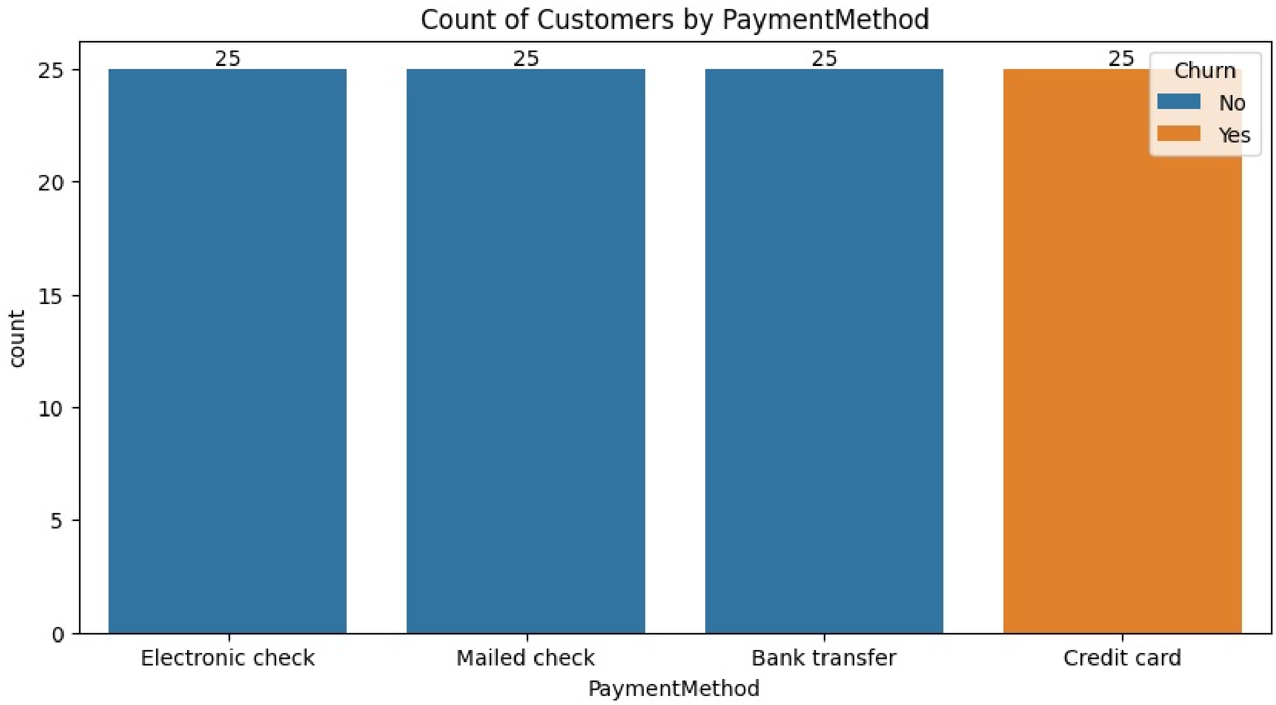
Countplot of Churn

The data visualization shows churn distribution based on different factors. Churn is higher among customers with month-to-month contracts and those using paperless billing. Phone service does not significantly impact churn, while payment methods have a relatively even distribution. Overall, most customers did not churn, but a smaller proportion did.

In [87]:

```
plt.figure(figsize = (10,5))
ax = sns.countplot(x = 'PaymentMethod', data = df, hue = 'Churn')

ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.title('Count of Customers by PaymentMethod')
plt.show()
```


Count of Customers by PaymentMethod
25
25
25
25
Churn
No
Yes
count
0
5
10
15
20
25
Electronic check
Mailed check
Bank transfer
Credit card
PaymentMethod

```
In [ ]:
```