

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

```
b = pd.read_csv("C:\\Users\\manth\\Downloads\\Telecom chain
analysis.csv")
b
```

	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]
```

```
b.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
```

It give information about table

```

b["TotalCharges"] = b["TotalCharges"].replace(" ", "0")
b["TotalCharges"] = b["TotalCharges"].astype("float")

```

Replacing blanks with 0 as tenure is 0 and no total charges are recorded

```
b.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
```

Below we see that how many null value are there (isnull) is used to give boolean value true or false After that sum() is used to see the total null value in that one column

```
b.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

It give statistics (count,min,max,std,mean)

```
b["customerID"].duplicated()
0      False
1      False
2      False
3      False
4      False
```

```

...
7038    False
7039    False
7040    False
7041    False
7042    False
Name: customerID, Length: 7043, dtype: bool

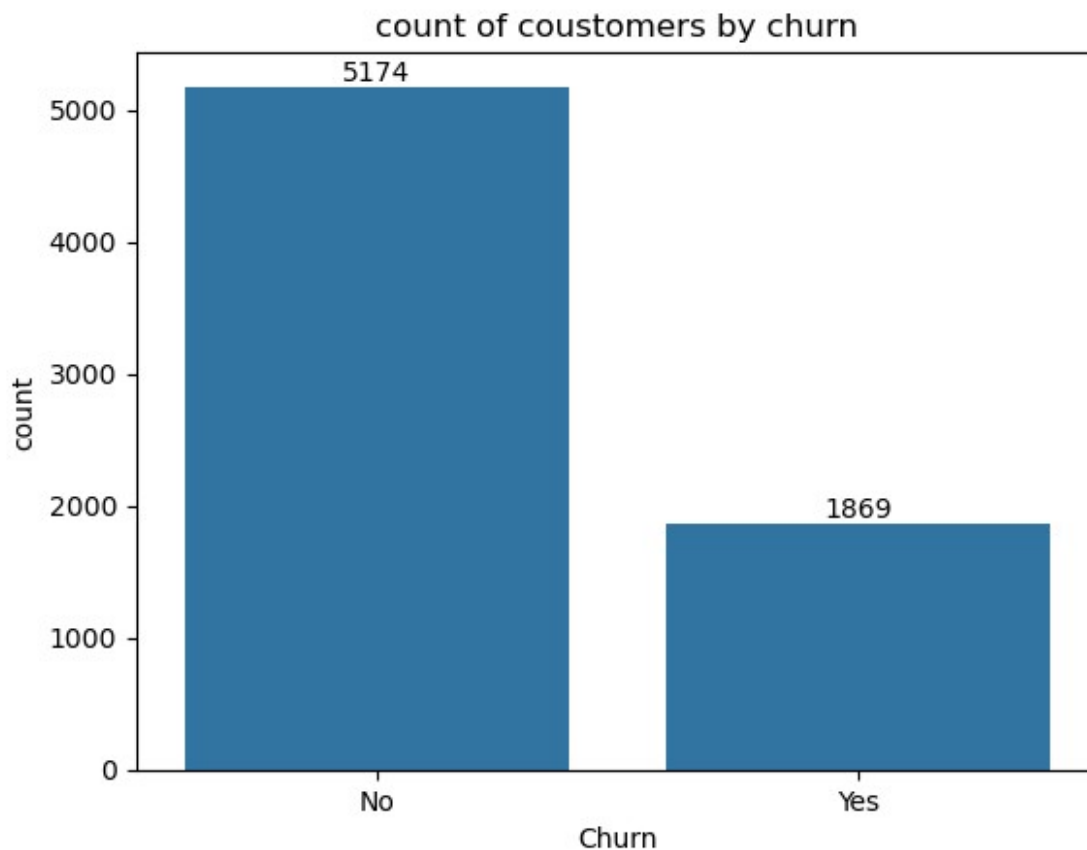
```

It gives the duplicate value df.duplicated() We can check a specific column (df["customerID"].duplicated())

```

ax = sns.countplot(x= 'Churn', data = b)
ax.bar_label(ax.containers[0])
plt.title("count of coustomers by churn")
plt.show()

```



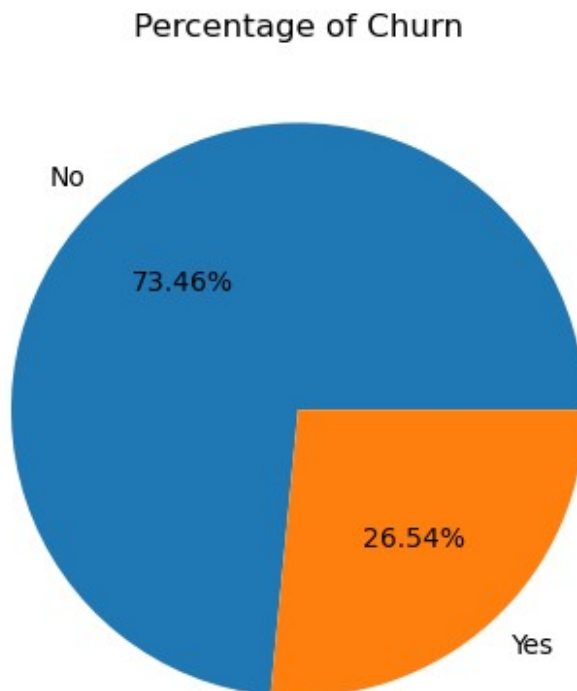
This countplot visualizes the distribution of customers based on the Churn variable, which indicates whether a customer has left the company (Yes) or stayed (No).

```

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

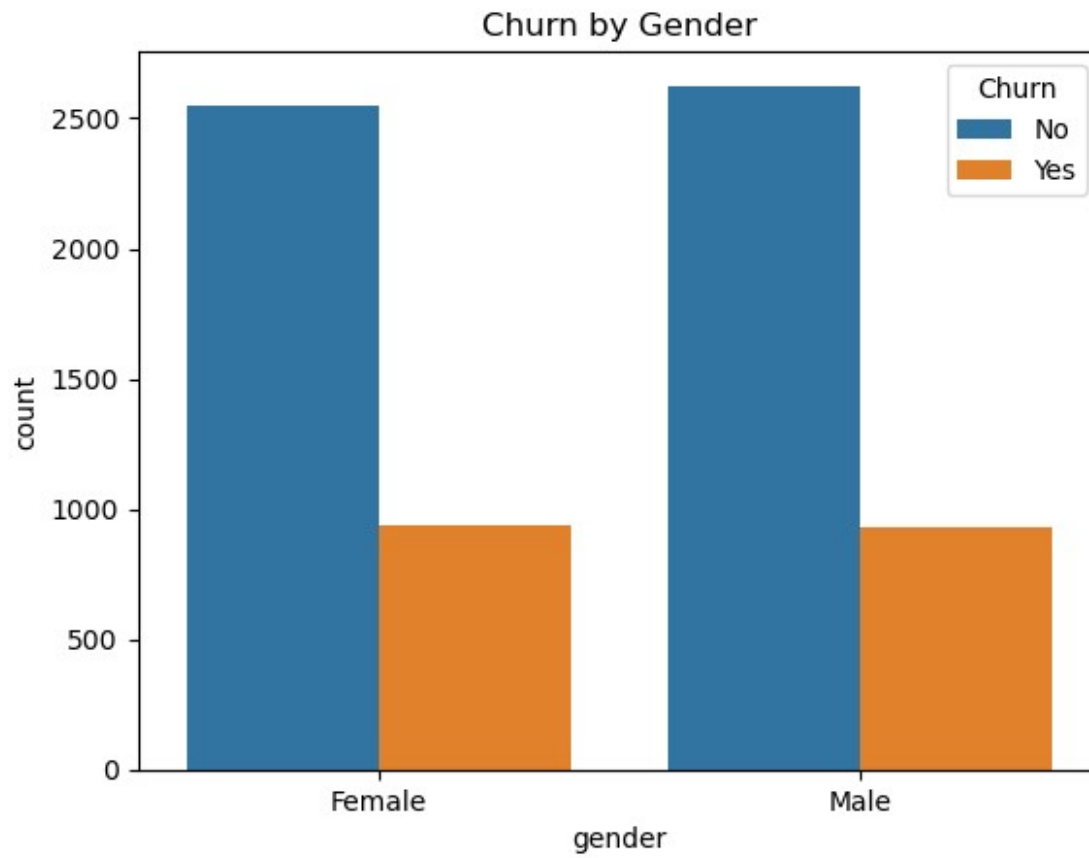
```

```
plt.title("Percentage of Churn")  
plt.show()
```



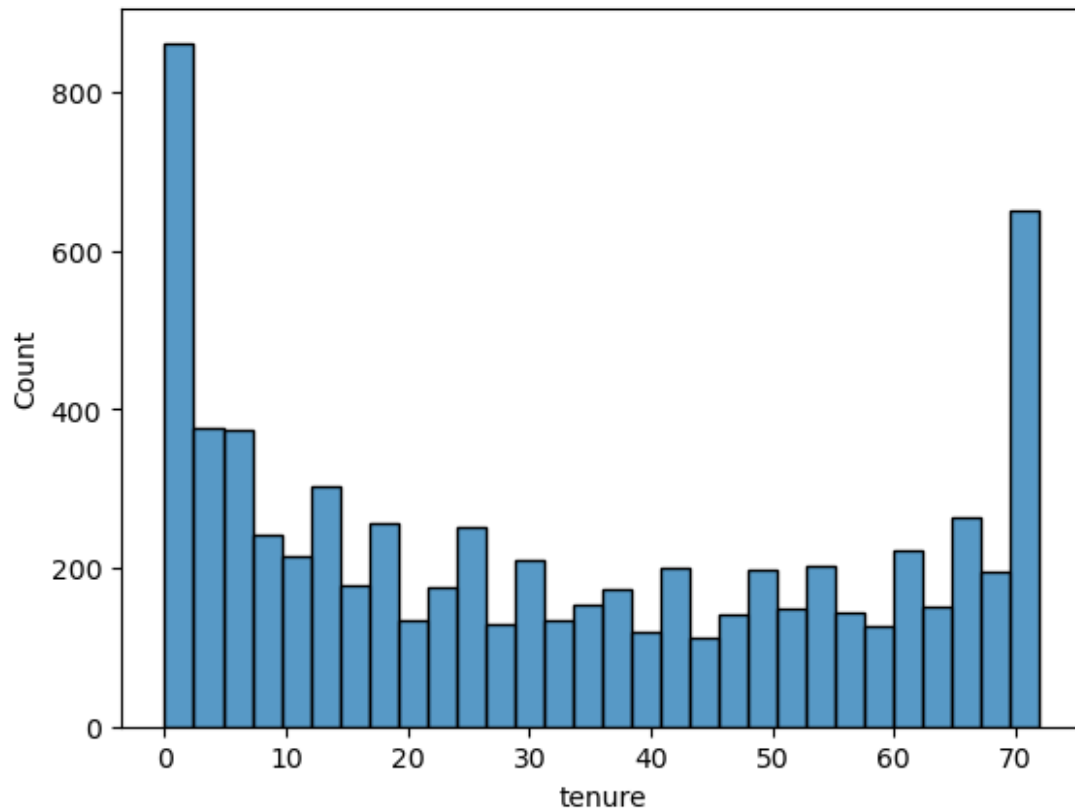
In this pie chart we can see that 26% our Customers are churn out

```
# plt.figure(figsize = (3,4))  
sns.countplot(x = "gender",data = b, hue= "Churn")  
plt.title("Churn by Gender")  
plt.show()
```



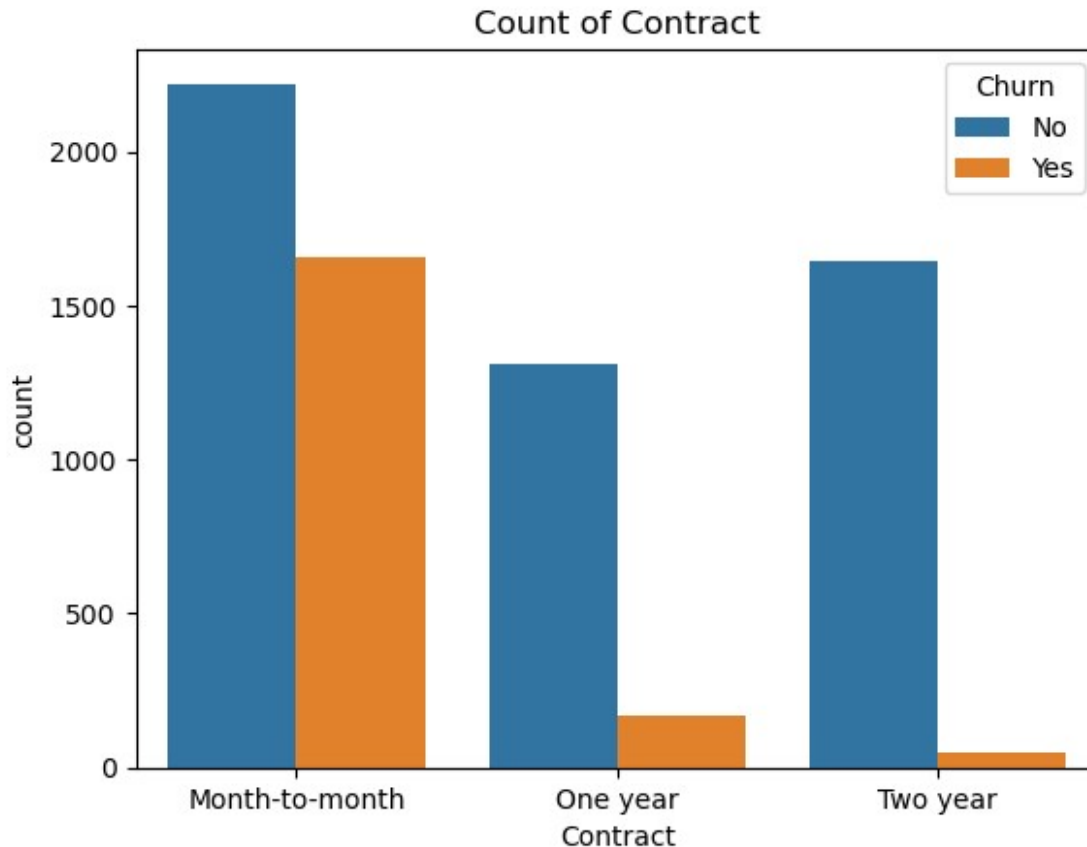
In this we can see churn by Gender that which gender in customer is churn out most

```
sns.histplot(x = "tenure", data = b, bins = 30)  
plt.show()
```



In this we can see that large no. of customers are new & many have been with the company for long time while fewer customers fall in between

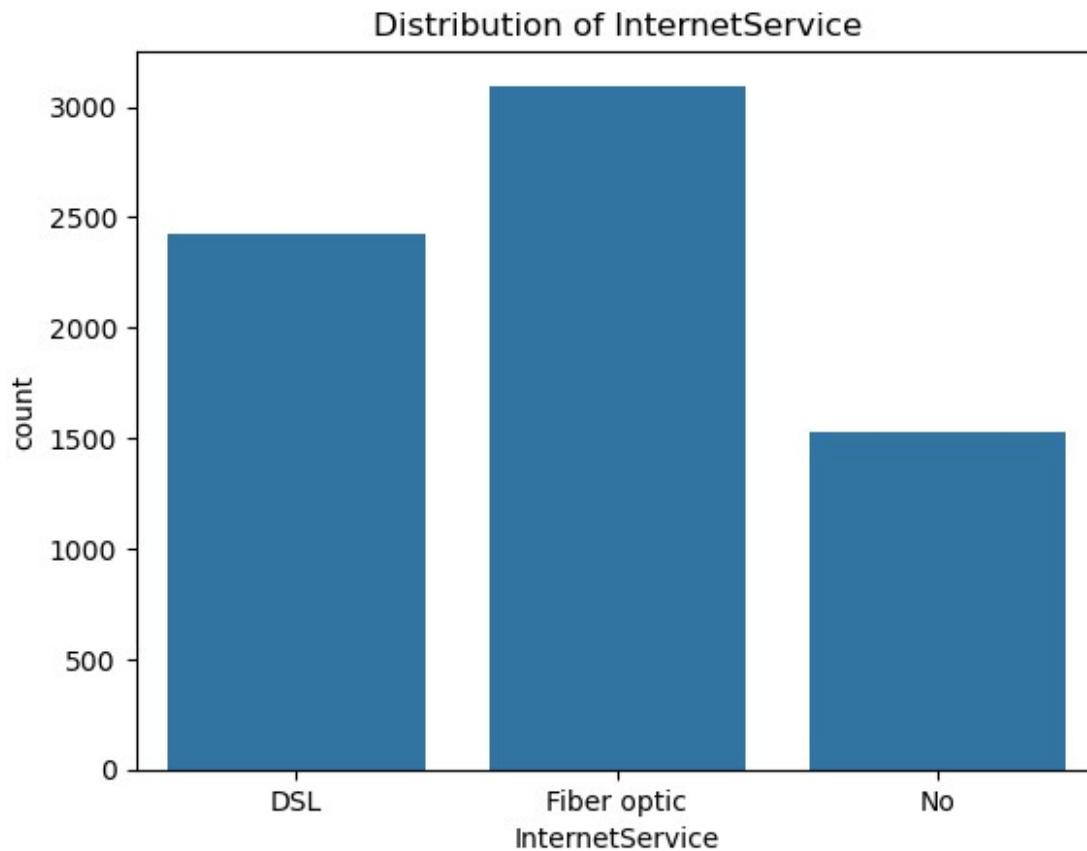
```
sns.countplot(x = "Contract",data = b, hue= "Churn")  
plt.title("Count of Contract")  
plt.show()
```

People who have month to month are likely to churn then from those who have 1 or 2 yrs contract

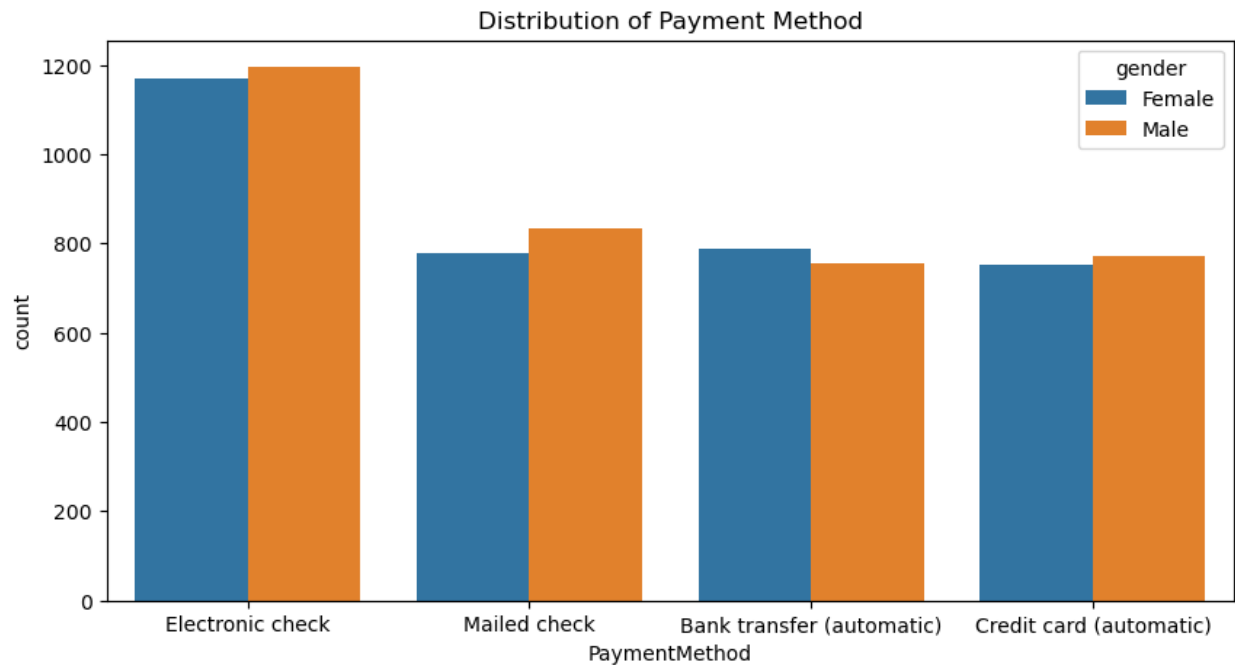
```
# b.columns.values
array(['customerID', 'gender', 'SeniorCitizen', 'Partner',
      'Dependents',
      'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
      'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
      'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
   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

sns.countplot(x = "InternetService", data = b)
plt.title("Distribution of InternetService")
plt.show()
```



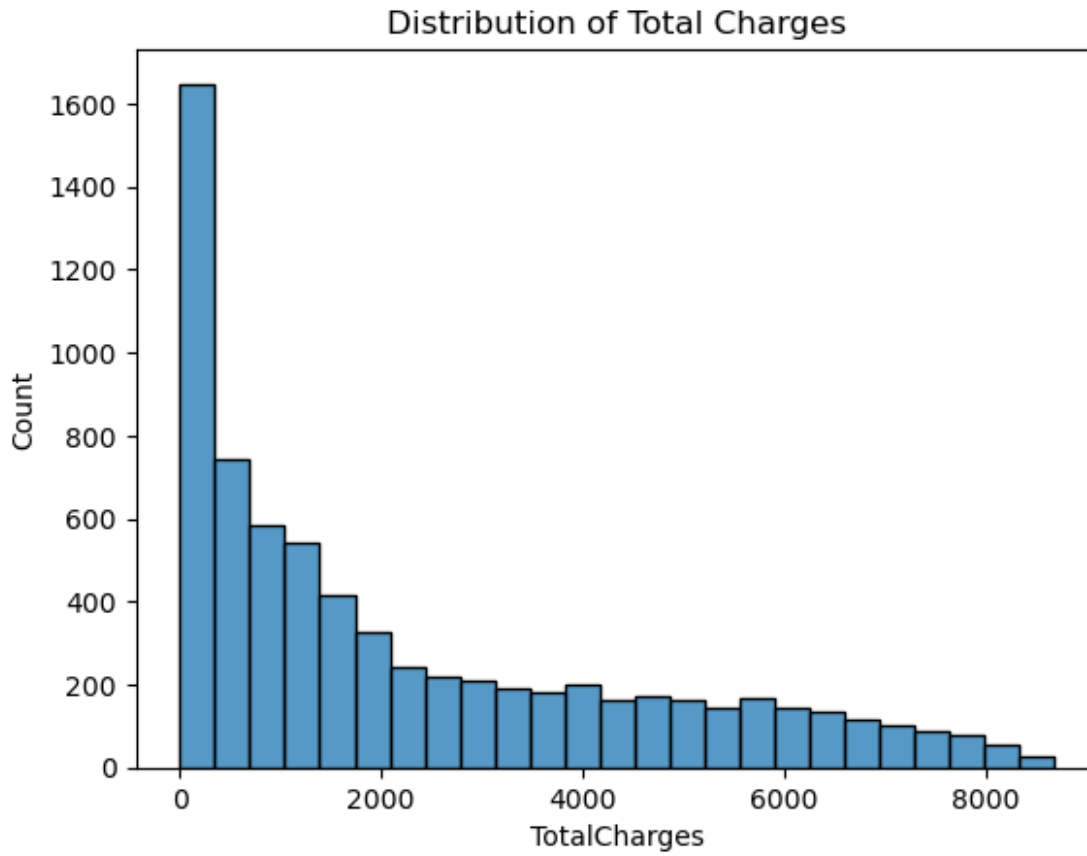
This countplot shows the distribution of different types of internet services subscribed to by customers in the dataset.

```
plt.figure(figsize = (10,5))
sns.countplot(x = "PaymentMethod", data = b, hue="gender")
plt.title("Distribution of Payment Method")
plt.show()
```



This countplot visualizes how different payment methods are distributed among customers, separated by gender (Male and Female).

```
sns.histplot(x = "TotalCharges",data = b)
plt.title("Distribution of Total Charges")
plt.show()
```



This histogram displays the distribution of `TotalCharges` across all customers in the dataset.

Conclusion -

- The analysis revealed that customer churn is significantly affected by factors such as contract type, tenure, payment method, internet service, and monthly charges.
- Customers with month-to-month contracts, lower tenure, and higher monthly charges showed a higher tendency to churn.
- Fiber optic internet users and those using electronic checks as a payment method were also more likely to leave the service.
- Demographic factors like gender had little impact on churn, while service-related factors such as tech support, online security and streaming services influenced customer decision.
- By addressing these issues, the company can improve customer satisfaction, reduce churn rates, and enhance long-term profitability.