Exploratory Analysis of Telecom Customer Churn Factors

Importing important libraries

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

Reading the data set

```
df=pd.read_csv('/content/Telco-Customer-Churn.csv')
df.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...

5 rows × 23 columns

Data Cleaning

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7032 entries, 0 to 7031
Data columns (total 23 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7032 non-null   object
 1   gender            7032 non-null   object
 2   SeniorCitizen     7032 non-null   int64
 3   Partner           7032 non-null   object
 4   Dependents        7032 non-null   object
 5   tenure            7032 non-null   int64
 6   PhoneService      7032 non-null   object
 7   MultipleLines     7032 non-null   object
 8   InternetService   7032 non-null   object
 9   OnlineSecurity    7032 non-null   object
 10  OnlineBackup      7032 non-null   object
 11  DeviceProtection  7032 non-null   object
 12  TechSupport       7032 non-null   object
 13  StreamingTV       7032 non-null   object
 14  StreamingMovies   7032 non-null   object
 15  Contract          7032 non-null   object
 16  PaperlessBilling  7032 non-null   object
 17  PaymentMethod     7032 non-null   object
 18  MonthlyCharges    7032 non-null   float64
 19  TotalCharges      7032 non-null   float64
 20  Churn             7032 non-null   object
 21  Unnamed: 21       0 non-null      float64
 22  Tenure Group      1 non-null      object
dtypes: float64(3), int64(2), object(18)
memory usage: 1.2+ MB
```

```
df.columns
```

```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
```

```
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn',
       'Unnamed: 21', 'Tenure Group'],
      dtype='object')
```

```
#dropping unnecessary columns
df.drop(['customerID', 'Unnamed: 21', 'Tenure Group'], axis=1, inplace=True)
```

```
#Converting total charges into numeric
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df.dropna(inplace=True)
```

```
#Converted senior citizen value from 0 and 1 to yes and No.
def conv(value):
    if value == 1:
        return 'Yes'
    else:
        return 'No'

# Apply the function to the 'SeniorCitizen' column
df['SeniorCitizen'] = df['SeniorCitizen'].apply(conv)
```

```
df.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	D
0	Female	No	Yes	No	1	No	No phone service	DSL	No	Yes	
1	Male	No	No	No	34	Yes	No	DSL	Yes	No	
2	Male	No	No	No	2	Yes	No	DSL	Yes	Yes	
3	Male	No	No	No	45	No	No phone service	DSL	Yes	No	
4	Female	No	No	No	2	Yes	No	Fiber optic	No	No	

Next steps: Generate code with df View recommended plots New interactive sheet

Summary of numerical columns (statistical info for numerical columns)

```
df.describe()
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges	Unnamed: 21
count	7032.000000	7032.000000	7032.000000	7032.000000	0.0
mean	0.162400	32.421786	64.798208	2283.300441	NaN
std	0.368844	24.545260	30.085974	2266.771362	NaN
min	0.000000	1.000000	18.250000	18.800000	NaN
25%	0.000000	9.000000	35.587500	401.450000	NaN
50%	0.000000	29.000000	70.350000	1397.475000	NaN
75%	0.000000	55.000000	89.862500	3794.737500	NaN
max	1.000000	72.000000	118.750000	8684.800000	NaN

```
# Value counts of the target column
df['Churn'].value_counts()

# Percentage distribution
df['Churn'].value_counts(normalize=True) * 100
```

	proportion
Churn	
No	73.421502
Yes	26.578498

dtype: float64

Univariate Analysis

```
print(df.groupby('gender')['Churn'].value_counts(normalize=True))
```

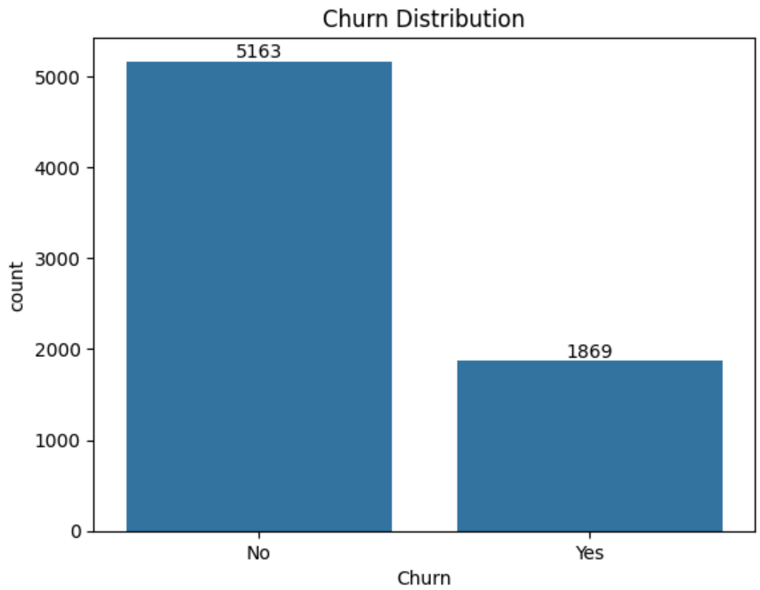
```
gender  Churn
Female  No       0.730405
        Yes      0.269595
Male    No       0.737954
        Yes      0.262046
Name: proportion, dtype: float64
```

```
print(df.groupby('Contract')['Churn'].value_counts(normalize=True))
```

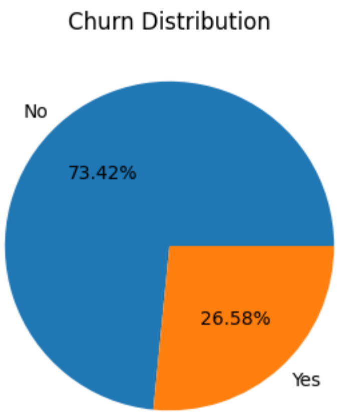
```
Contract        Churn
Month-to-month  No       0.572903
                Yes      0.427097
One year        No       0.887228
                Yes      0.112772
Two year        No       0.971513
                Yes      0.028487
Name: proportion, dtype: float64
```

Visualize Churn Distribution:

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



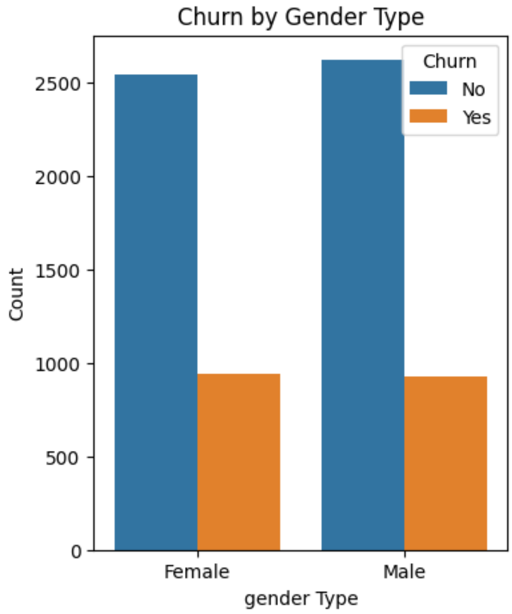

```
#This is a pie chart showing overall churn in percentage
plt.figure(figsize=(4, 4))
gb = df.groupby('Churn').agg({'Churn': 'count'})
plt.pie(gb['Churn'], labels = gb.index, autopct='%1.2f%%')
plt.title('Churn Distribution')
plt.show()
```



From the given pie chart, most customers stay, but about 27% leave. This shows there is a potential area for improvement in customer retention.

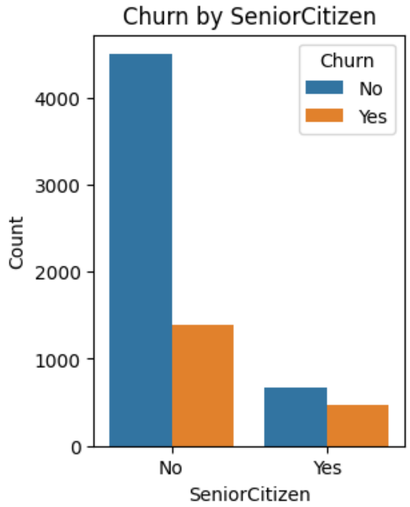
Now, exploring the factors behind churn

```
plt.figure(figsize=(4,5))
sns.countplot(x='gender', hue='Churn', data=df)
plt.title('Churn by Gender Type')
plt.xlabel('gender Type')
plt.ylabel('Count')
plt.show()
```

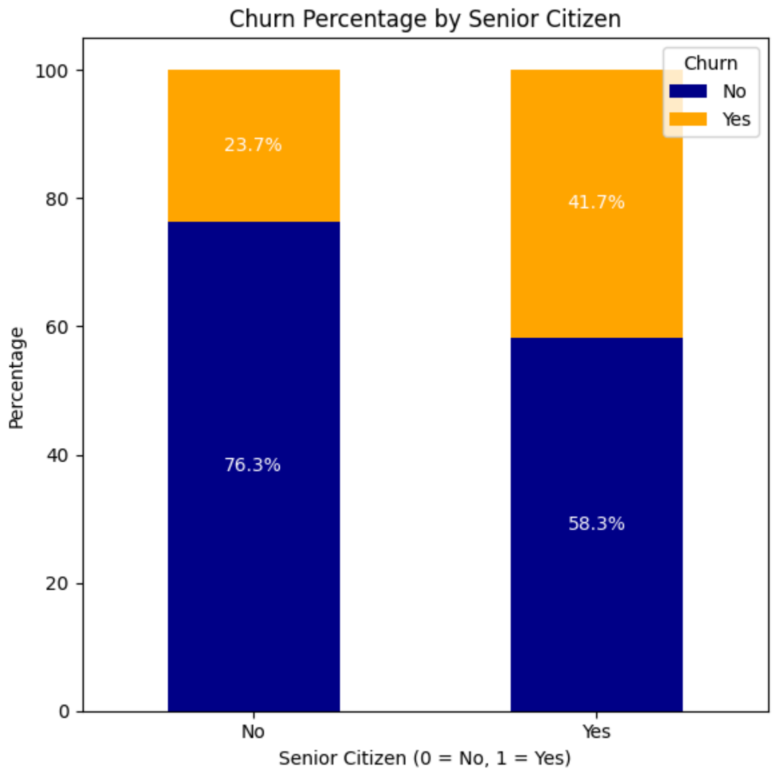


From the given plot, we can say that the churn is not gender specific.

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

plt.title('Churn by SeniorCitizen')
plt.xlabel('SeniorCitizen')
plt.ylabel('Count')
plt.show()
```



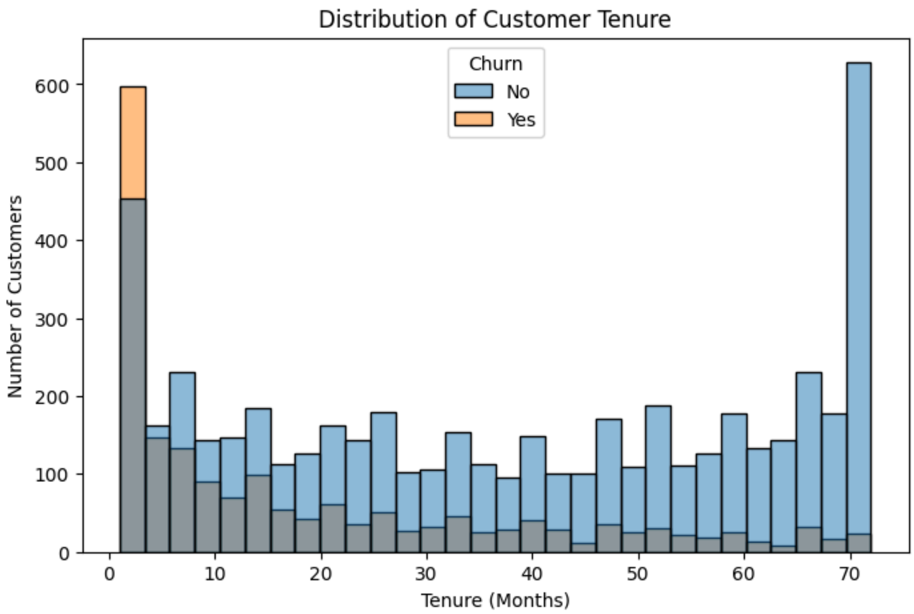

```
# Step 1: Count Churn for each SeniorCitizen group
count_data = df.groupby(['SeniorCitizen', 'Churn']).size().unstack()

# Step 2: Convert counts to percentage
percent_data = count_data.div(count_data.sum(axis=1), axis=0) * 100

# Step 3: Plot the stacked bar chart
ax = percent_data.plot(kind='bar', stacked=True, figsize=(6, 6), color=['darkblue', 'orange'])

plt.title('Churn Percentage by Senior Citizen')
plt.xlabel('Senior Citizen (0 = No, 1 = Yes)')
plt.ylabel('Percentage')
plt.xticks(rotation=0)

# Step 4: Add percentage labels
for i in range(len(percent_data)):
    bottom = 0
    for j in range(len(percent_data.columns)):
        value = percent_data.iloc[i, j]
        if value > 0:
            ax.text(i, bottom + value / 2, f'{value:.1f}%', ha='center', va='center', fontsize=10, color='white')
            bottom += value

plt.legend(title='Churn', loc='upper right')
plt.tight_layout()
plt.show()

```



Senior Citizens have comapritavily more chur than non-senior citizens

Tenure Distribution (How long customers have stayed)

```
plt.figure(figsize=(8, 5))
sns.histplot(data = df, x = 'tenure',bins=30, color='skyblue', hue = 'Churn')
plt.title('Distribution of Customer Tenure')
plt.xlabel('Tenure (Months)')
plt.ylabel('Number of Customers')
plt.show()
```

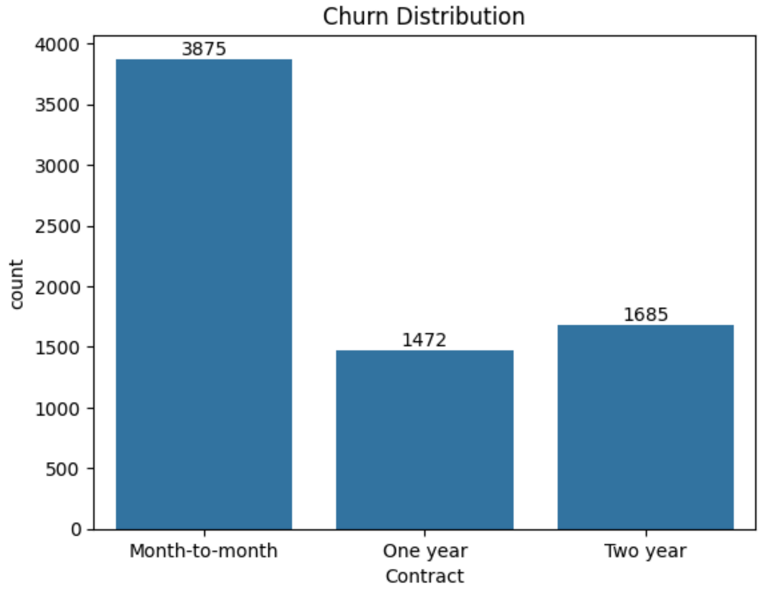


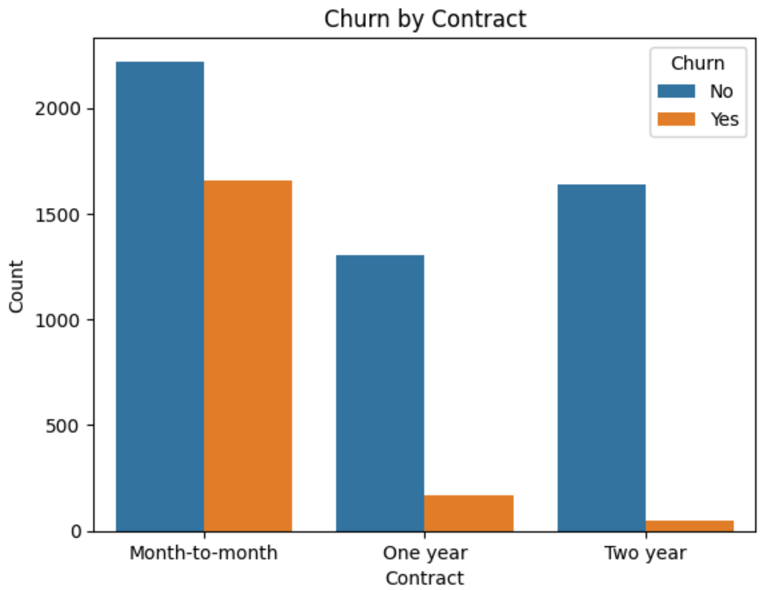
Observations from the Tenure Histogram:

1. High Churn at the Beginning (0–1 months): This might reflect poor onboarding, unmet expectations, or uncompetitive offerings for new users.
2. Steady Decline and Flat Midsection (10–60 months): Customers who stay beyond the first few months tend to continue for a relatively steady period, showing moderate retention.
3. Another Peak at 70–72 Months: These could be loyal customers who've been with the company for the full duration. This segment may be highly satisfied or have long-term contracts.

Churn by contract

```
# Count plot of Churn
ax = sns.countplot(x='Contract', data=df)
plt.title('Churn Distribution')
ax.bar_label(ax.containers[0])
plt.show()

sns.countplot(x='Contract', hue='Churn', data=df)
plt.title('Churn by Contract')
plt.xlabel('Contract')
plt.ylabel('Count')
plt.show()
```





From the given chart, customer retention improves with longer contract durations, and month-to-month plans exhibit the highest churn rates than those who have 1 or 2 years plans. This trend suggests that incentivizing longer contracts could reduce churn rates.

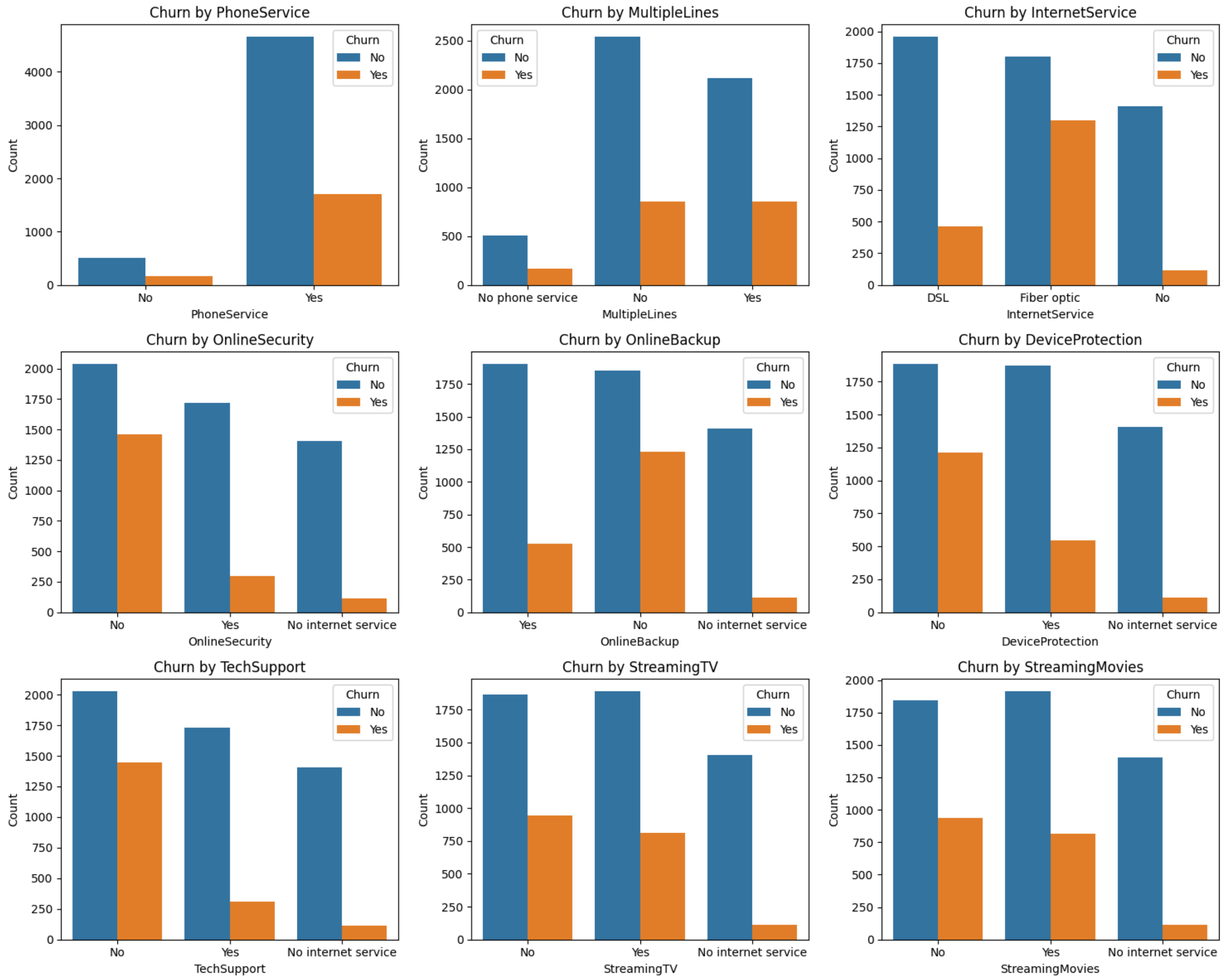
```
df.columns.values
```

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

Service-Wise Churn Comparison in Telecom Dataset

```
# List of service-related columns
cols = [
    'PhoneService', 'MultipleLines', 'InternetService',
    'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
    'TechSupport', 'StreamingTV', 'StreamingMovies'
]

n_cols = 3
n_rows = (len(cols) + n_cols - 1) // n_cols #calculate numbers of rows needed

#create subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, n_rows * 4)) #adjust fig as needed

# Flatten the axes array for easier iteration
axes = axes.flatten()
```

```

for i, col in enumerate(cols):
  # Use the single Axes object from the flattened array
  sns.countplot(x=col, hue='Churn', data=df, ax=axes[i])
  axes[i].set_title(f'Churn by {col}')
  axes[i].set_xlabel(col)
  axes[i].set_ylabel('Count')

#Remove empty subplots
# Start the loop from the number of columns to remove the remaining axes
for i in range(len(cols), n_rows * n_cols):
  fig.delaxes(axes[i])

plt.tight_layout()
plt.show()
```



Here are the key insights from these subplots.

1. PhoneService: Customers with Phone Service are more likely to churn compared to those without.

However, a majority still do not churn, indicating that this service alone isn't a strong churn driver.

2. MultipleLines: Churn is higher among customers who have multiple lines than those who do not.

No phone service group has the lowest churn, but it's also a small segment.

3. InternetService: Fiber optic users show a much higher churn rate than DSL or those without internet.

This may suggest dissatisfaction with fiber service or pricing.

4. OnlineSecurity: Churn is significantly higher among those without online security.

Customers who have OnlineSecurity tend to stay longer.

5. OnlineBackup: Similar to OnlineSecurity, customers without online backup churn more.

Offering backup services may reduce churn.

6. DeviceProtection: Customers without device protection show a higher churn rate.

Those who opt for this add-on appear more committed to the service.

7. TechSupport: One of the strongest patterns: customers without tech support churn the most.

Tech support availability correlates with retention.

8. StreamingTV: Customers with StreamingTV churn slightly more than those without it.

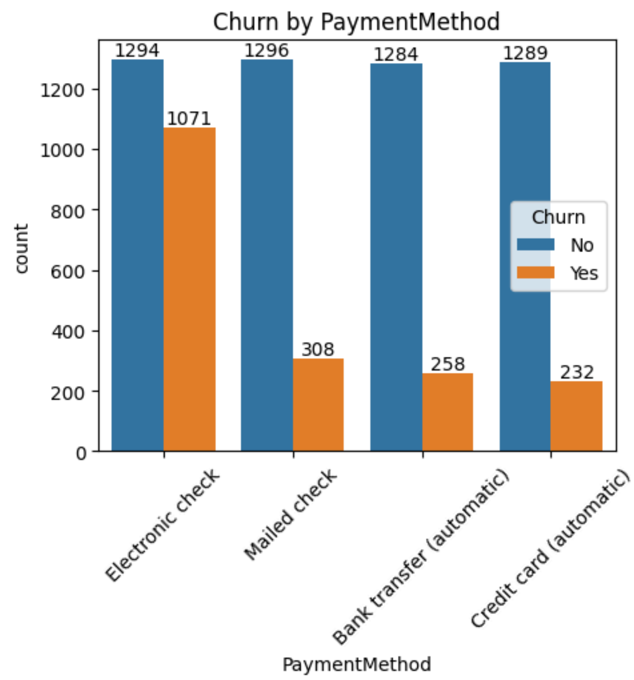
Still, not as strong an indicator as security/tech support.

9. StreamingMovies: Churn is higher among those who use StreamingMovies compared to those who don't.

The difference is moderate, similar to StreamingTV.

Churn by Payment Method

```
plt.figure(figsize=(5, 4))
ax = sns.countplot(x='PaymentMethod', hue='Churn', data=df)

# plt.xlabel('PaymentMethod')
# plt.ylabel('Count')
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.xticks(rotation=45)
plt.title('Churn by PaymentMethod')
plt.show()
```

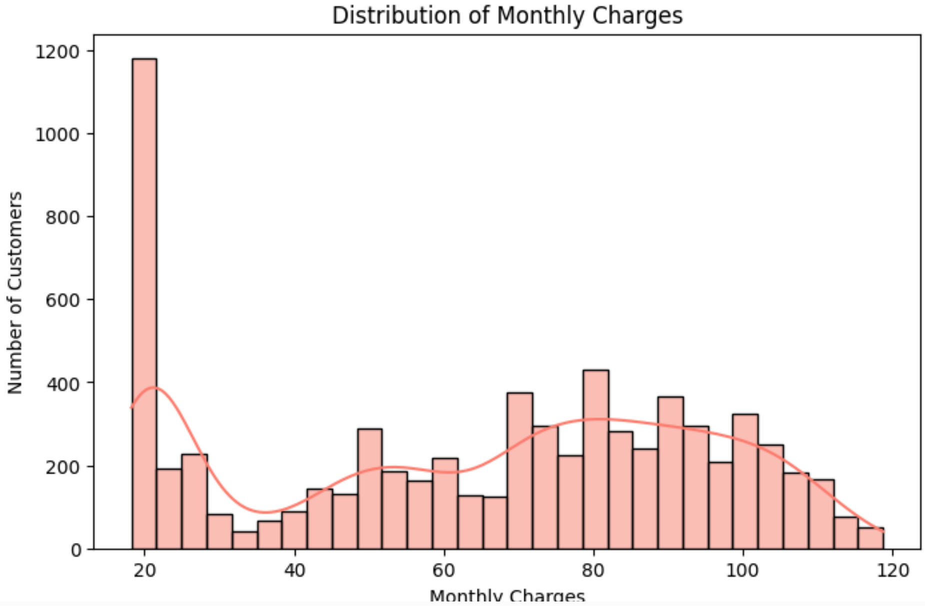


Insights from Churn by PaymentMethod Plot:

- Electronic Check users show the highest churn rate.
- 1071 customers churned vs 1294 who stayed.
- This suggests electronic check users might be less loyal or more price-sensitive.
- Mailed Check, Bank Transfer (automatic), and Credit Card (automatic) users have significantly lower churn rates.
- Each of these methods shows much higher "No" (non-churn) counts compared to "Yes".
- Indicates that automatic payments are correlated with higher customer retention.

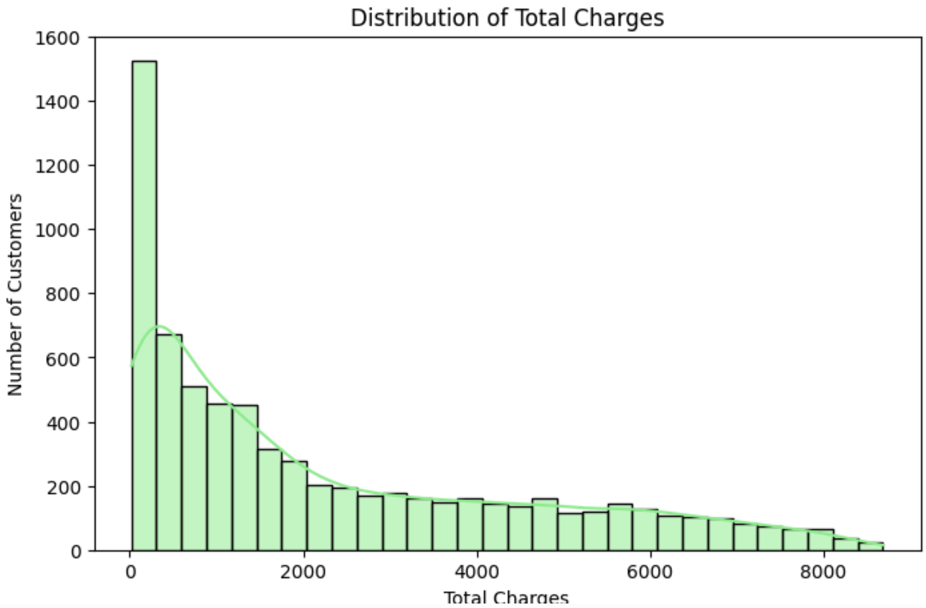
MonthlyCharges Distribution

```
plt.figure(figsize=(8, 5))
sns.histplot(df['MonthlyCharges'], kde=True, bins=30, color='salmon')
plt.title('Distribution of Monthly Charges')
plt.xlabel('Monthly Charges')
plt.ylabel('Number of Customers')
plt.show()
```



TotalCharges Distribution

```
plt.figure(figsize=(8, 5))
sns.histplot(df['TotalCharges'], kde=True, bins=30, color='lightgreen')
plt.title('Distribution of Total Charges')
plt.xlabel('Total Charges')
plt.ylabel('Number of Customers')
plt.show()
```

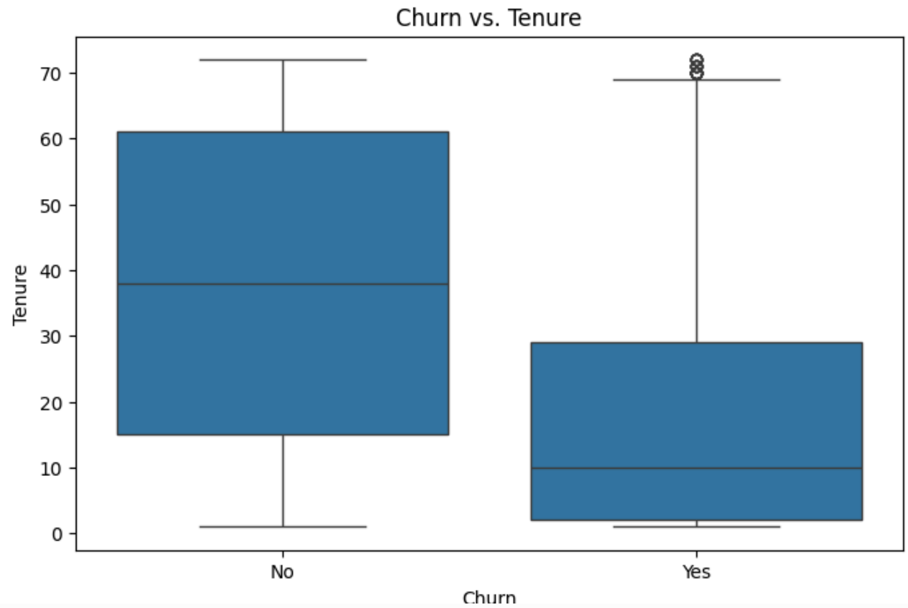


Bivariate Analysis:

This helps us understand how each feature affects customer churn.

1. Churn vs. Tenure

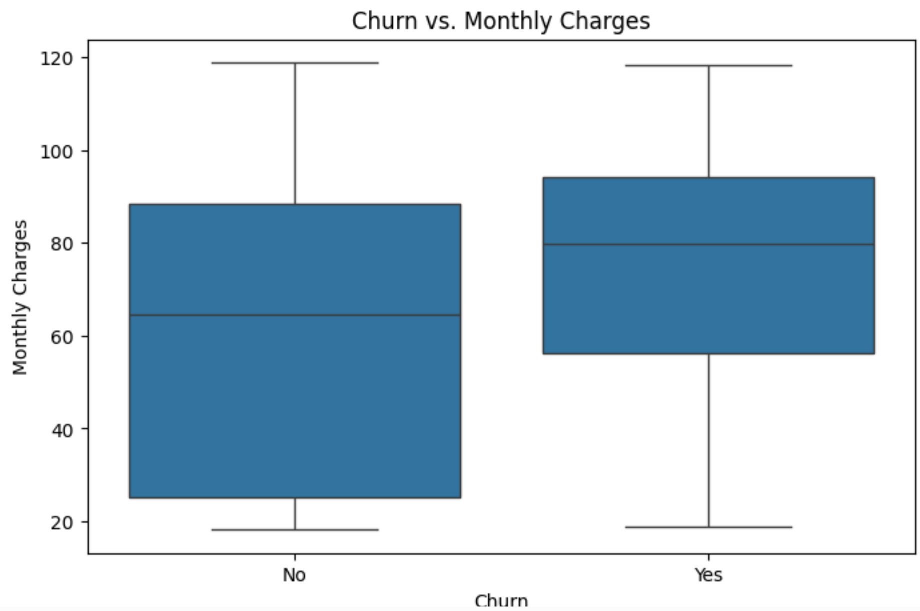
```
plt.figure(figsize=(8, 5))
sns.boxplot(x='Churn', y='tenure', data=df)
plt.title('Churn vs. Tenure')
plt.xlabel('Churn')
plt.ylabel('Tenure')
plt.show()
```



Insight: Churned customers often have lower tenure (shorter stay with the company).

2. Churn vs. Monthly Charges

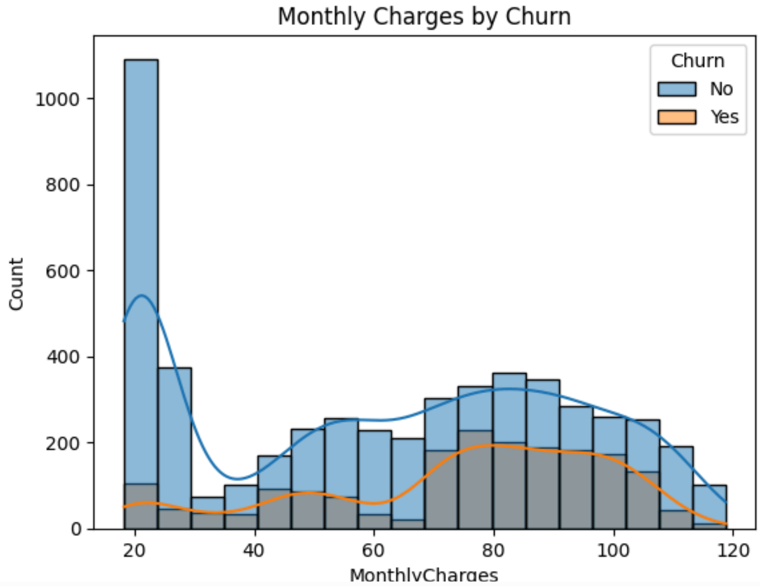
```
plt.figure(figsize=(8, 5))
sns.boxplot(x='Churn', y='MonthlyCharges', data=df)
plt.title('Churn vs. Monthly Charges')
plt.xlabel('Churn')
plt.ylabel('Monthly Charges')
plt.show()
```



- Insight: Customers paying higher monthly charges are more likely to churn.

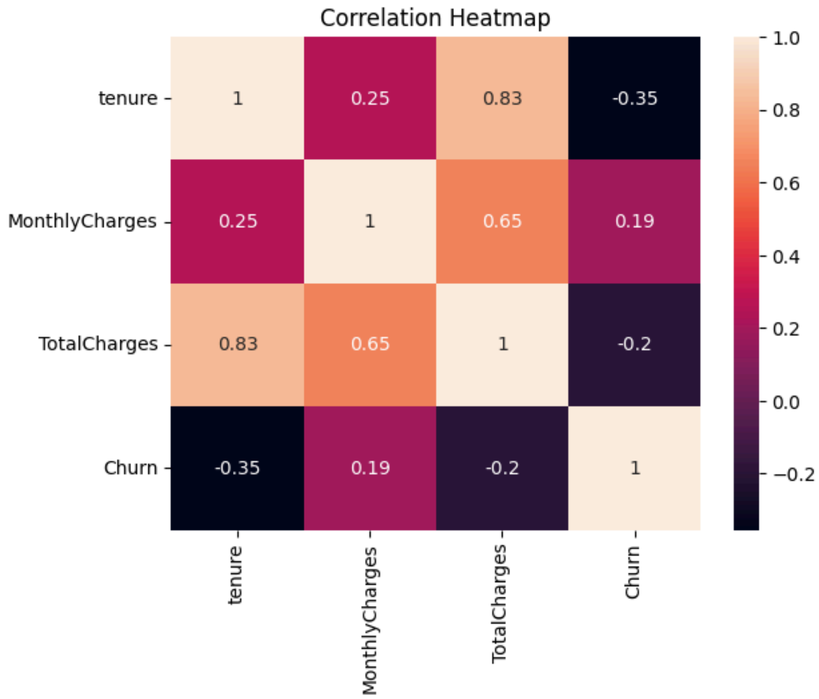
3. Churn vs. Categorical Features (e.g., Contract Type)

```
# Histogram of MonthlyCharges
sns.histplot(data=df, x='MonthlyCharges', hue='Churn', kde=True)
plt.title('Monthly Charges by Churn')
plt.show()
```



- Customers with lower monthly charges are significantly more likely to churn compared to those with higher charges.

```
# Heatmap of correlations (after converting categorical to numeric if needed)
df_encoded = df.copy()
df_encoded['Churn'] = df_encoded['Churn'].map({'Yes': 1, 'No': 0})
corr = df_encoded.corr(numeric_only=True)
sns.heatmap(corr, annot=True)
plt.title('Correlation Heatmap')
plt.show()
```



The heatmap shows weak to moderate relationships among the variables. Key points are:

- Longer customer tenure is strongly linked to higher total charges.
- Monthly charges are positively related to total charges.
- Customers with shorter tenure are more likely to churn.
- Other relationships among variables are generally weak.

Overall, tenure and charges are closely connected, and shorter tenure is associated with increased churn risk.

Conclusion

The customer churn analysis effectively highlighted the key patterns and features that influence customer attrition in a telecom setting. Using exploratory data analysis, we identified meaningful trends that can support data-driven decision-making for improving customer retention.

Key Insights:

- **Tenure is Critical:** Customers with a shorter tenure (i.e., newer customers) are far more likely to churn compared to long-term customers.
- **Contract Type Matters:** Month-to-month contract users show the highest churn rates. In contrast, those with one- or two-year contracts are more loyal.
- **Monthly Charges Impact Churn:** Higher monthly charges are associated with an increased likelihood of churn, especially when combined with short tenure.
- **Total Charges Are Not Directly Indicative:** While total charges show distribution differences, their standalone impact on churn is less significant compared to tenure or contract.
- **Gender Has Minimal Effect:** Churn patterns between male and female customers are nearly identical, indicating gender is not a strong predictor.
- **Churn Rate:** Around 26.6% of the customers in the dataset have churned, while 73.4% have remained.