Customer Churn Analysis

Import all Libraries

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Read the Data

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

In [6]:
```
df.head()
```

Out[6]:

	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 × 21 columns

In [5]:
```
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
```

Replacing blanks with O as tenure is 0 and no of total charge are recorded

In [8]:
```
df['TotalCharges'] = df['TotalCharges'].replace(" ",0)
df['TotalCharges'] = df['TotalCharges'].astype("float")
```

In [10]:
```
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 [12]:
```
df.isnull().sum().sum()
```

Out[12]: 0

display statistcal summary

In [15]:
```
df.describe()
```

Out[15]:

	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

In [50]:
```
df.describe(include='all')
```

Out[50]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	Online
count	7043	7043	7043.000000	7043	7043	7043.000000	7043	7043	7043	
unique	7043	2	NaN	2	2	NaN	2	3	3	
top	7590-VHVEG	Male	NaN	No	No	NaN	Yes	No	Fiber optic	
freq	1	3555	NaN	3641	4933	NaN	6361	3390	3096	
mean	NaN	NaN	0.162147	NaN	NaN	32.371149	NaN	NaN	NaN	
std	NaN	NaN	0.368612	NaN	NaN	24.559481	NaN	NaN	NaN	
min	NaN	NaN	0.000000	NaN	NaN	0.000000	NaN	NaN	NaN	
25%	NaN	NaN	0.000000	NaN	NaN	9.000000	NaN	NaN	NaN	
50%	NaN	NaN	0.000000	NaN	NaN	29.000000	NaN	NaN	NaN	
75%	NaN	NaN	0.000000	NaN	NaN	55.000000	NaN	NaN	NaN	
max	NaN	NaN	1.000000	NaN	NaN	72.000000	NaN	NaN	NaN	

11 rows × 21 columns

In [17]:
```
df.duplicated().sum()
```

Out[17]:
```
0
```

In [19]:
```
df_cleaned = df.drop_duplicates()

print("Original rows:",len(df))
print("Rows after removing duplicates:", len(df_cleaned))
print("No. of Duplicates remove:", len(df) - len(df_cleaned))
```

```
Original rows: 7043
Rows after removing duplicates: 7043
No. of Duplicates remove: 0
```

In [21]:
```
df['PaymentMethod'].value_counts()
```

Out[21]:
```
PaymentMethod
Electronic check             2365
Mailed check                 1612
Bank transfer (automatic)    1544
Credit card (automatic)      1522
Name: count, dtype: int64
```

checking for missing value

In [24]:
```
missing_values = df.isnull().sum()
print(missing_values)
```

```
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
```

Churn Value counts

In [29]:
```
df["Churn"].value_counts()
```

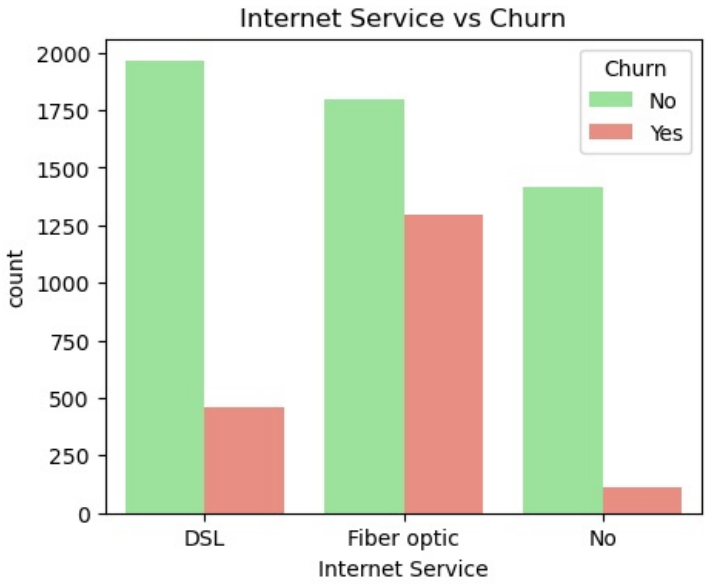
Out[29]:
```
Churn
No     5174
Yes    1869
Name: count, dtype: int64
```

Relationships and trends

Count Plot – Internet Service vs Churn

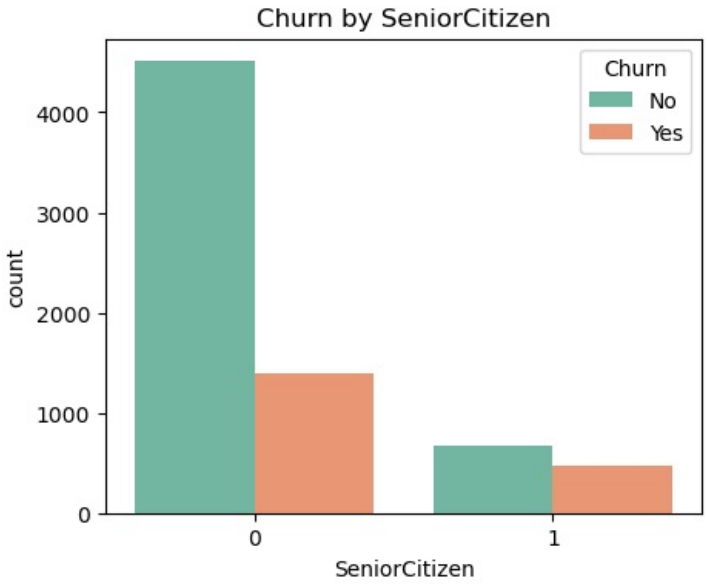
In [75]:
```
# : Count Plot

plt.figure(figsize =(5,4))
sns.countplot(x="InternetService", hue="Churn", data=df,
              palette={"No": "lightgreen", "Yes": "salmon"})
plt.title("Internet Service vs Churn")
plt.xlabel("Internet Service")
```

Out[75]:
```
Text(0.5, 0, 'Internet Service')
```



Gender and Churn Rate

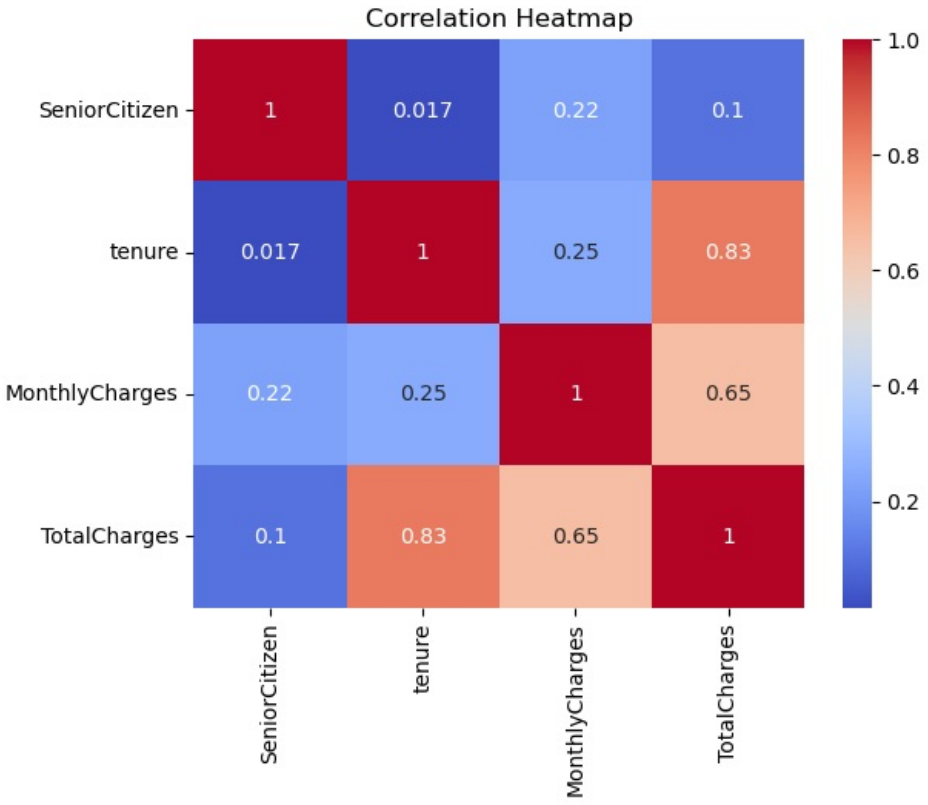
In [48]:
```
# : count plot

plt.figure(figsize =(5,4))
sns.countplot(x="SeniorCitizen", data= df, hue="Churn", palette="Set2")
plt.title("Churn by SeniorCitizen")
plt.show()
```



Visualization :-

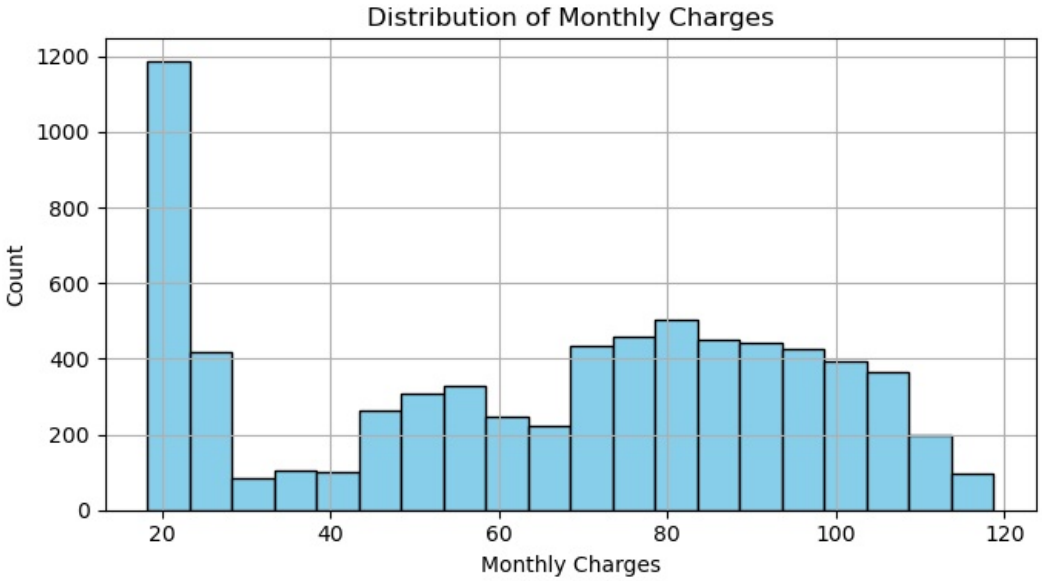
What is the correlation between numeric features ?

In [46]:
```
# Heatmap

sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```



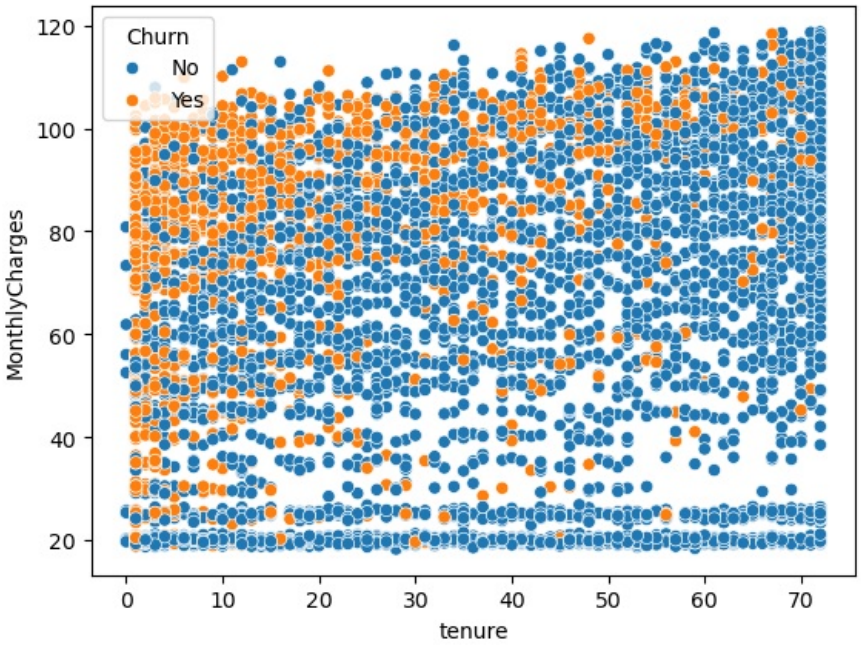
What is the distribution of Monthly Charges among customers, and are there any noticeable patterns or concentrations in the data?

In [57]:
```
# Histogram:

ax = df['MonthlyCharges'].plot(
    kind='hist',
    bins=20,
    figsize=(7,4),
    color='skyblue',
    edgecolor='black',
    grid=True
)

plt.title("Distribution of Monthly Charges")
plt.xlabel("Monthly Charges")
plt.ylabel("Count")
plt.tight_layout()
plt.show()
```



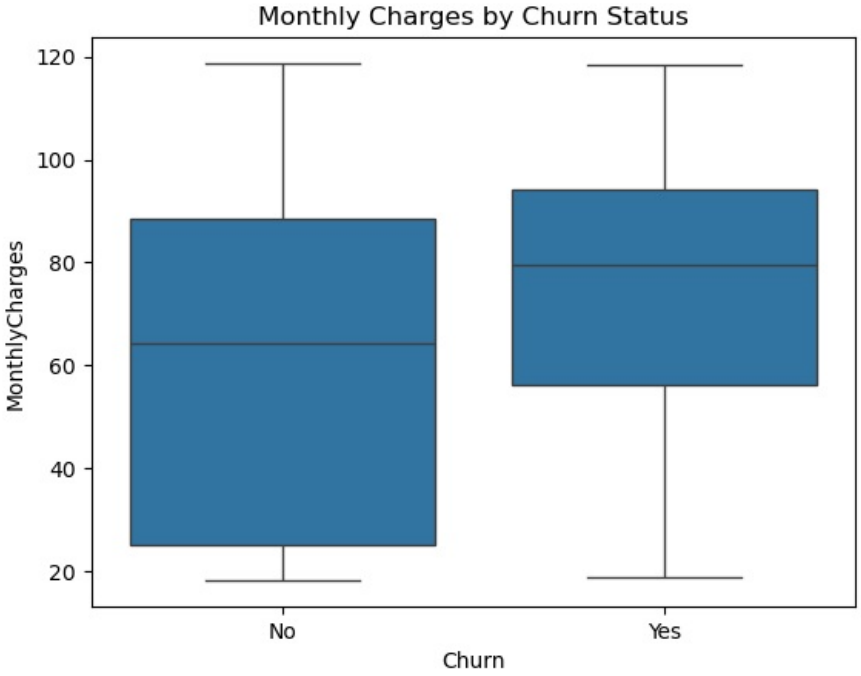
What do scatterplots show between tenure and monthly charges?

In [10]:
```
# Scatter plot

ax = sns.scatterplot(x='tenure', y='MonthlyCharges', hue='Churn', data=df)
plt.show()
```



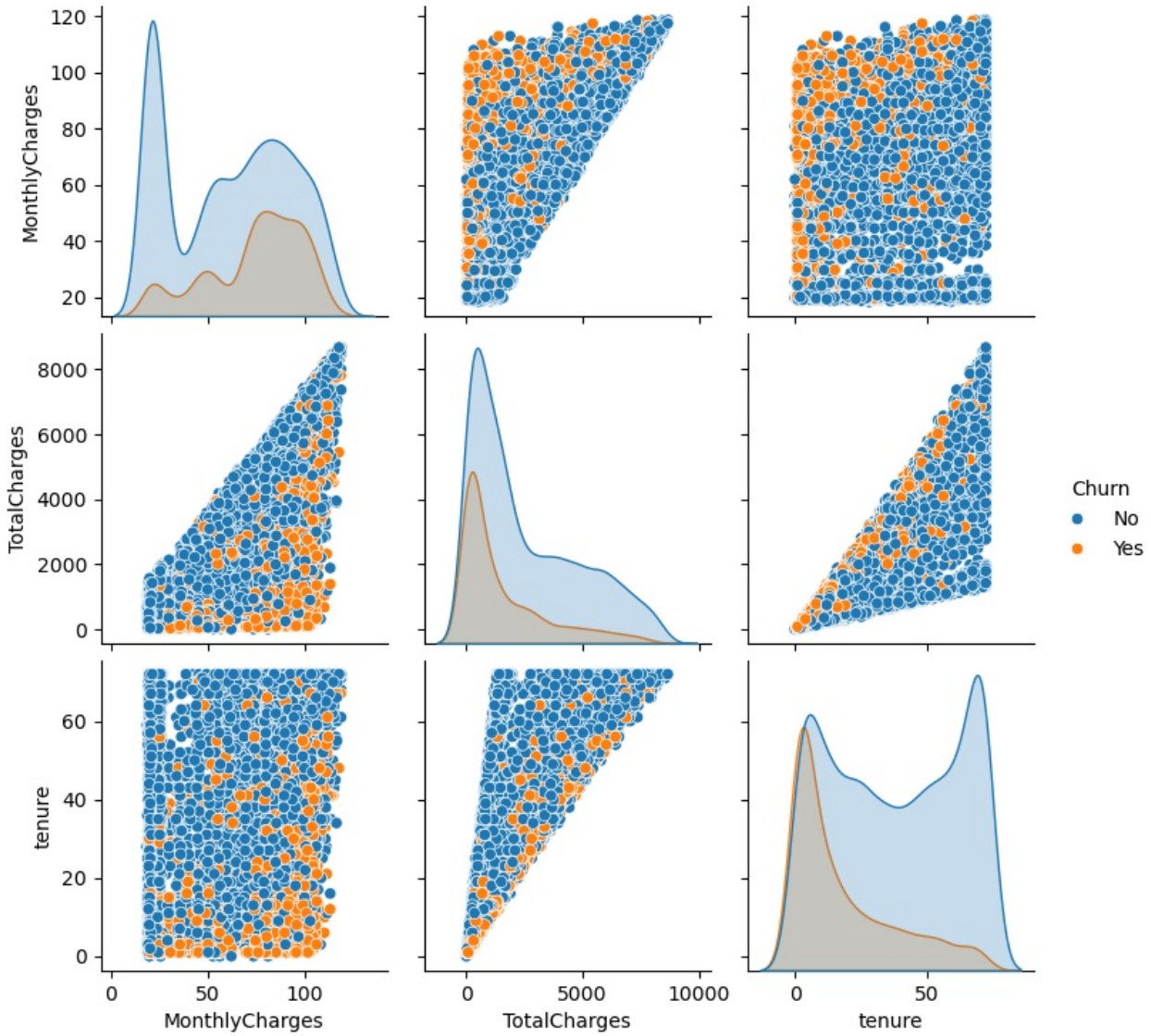
Churned customers have higher Monthly Charges?

In [68]:
```
# Box plot

sns.boxplot(x="Churn", y="MonthlyCharges", data=df)
plt.title("Monthly Charges by Churn Status")
plt.show()
```



Can we identify any visible relationships or patterns between Monthly Charges, Total Charges, and Tenure based on customer churn?

In [71]:
```
# Pair plot

sns.pairplot(df[['MonthlyCharges', 'TotalCharges', 'tenure', 'Churn']], hue='Churn')
plt.show()
```



In [ ]: