AIM :- Perform data cleaning and exploratory data analysis (EDA) on a dataset of your choice, such as the Titanic dataset from Kaggle. Explore the relationships between variables and identify patterns and trends in the data.

Dataset Link :- https://www.kaggle.com/c/titanic/data

Importing Libraries.

In [2]:

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

Reading CSV File.

In [3]:

```
df=pd.read_csv("data1-csv.csv")
```

Accessing the top 5 rows of the Dataset.

In [63]:

```
df.head()
```

Out[63]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Accessing the bottom 5 rows of the Dataset.

In [64]:

```
df.tail()
```

Out[64]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [45]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 888 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  888 non-null    int64  
 1   Survived     888 non-null    int64  
 2   Pclass       888 non-null    int64  
 3   Name         888 non-null    object 
 4   Sex          888 non-null    object 
 5   Age          711 non-null    float64
 6   SibSp        888 non-null    int64  
 7   Parch        888 non-null    int64  
 8   Ticket       888 non-null    object 
 9   Fare         888 non-null    float64
 10  Cabin        202 non-null    object 
 11  Embarked     886 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 90.2+ KB
```

In [53]:

```
df.shape
```

Out[53]:

```
(888, 12)
```

Checking for the null values.

In [65]:

```
df.isnull()
```

Out[65]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False	True	False
887	False	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	False	False	True	False	False	False	False	True	False
889	False	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	False	True	False

891 rows × 12 columns

Calculating Mathematical Stastical Terms.

In [66]:

```
df.describe()
```

Out[66]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [ ]:

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

Data Cleaning.

Drop Unuseful Columns.

In [67]:

```
df.drop(columns='Ticket' , inplace= True)
df.drop(columns='PassengerId' , inplace=True)
df.drop(columns='Cabin' , inplace=True)
```

In [68]:

```
df.head()
```

Out[68]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S

In [7]:

```
df.dropna(how ='all')
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

Filling Age Column With Mean Age.

In [80]:
```
mean=df.Age.mean()
df.Age.fillna(np.random.randint(mean) , inplace=True)
```

In [79]:
```
df.isnull().sum()
```

Out[79]:
```
Survived    0
Pclass      0
Name        0
Sex         0
Age         0
SibSp       0
Parch       0
Fare        0
Embarked    2
dtype: int64
```

Filling Embarked column.

In [82]:
```
df.Embarked.fillna(df.Embarked.mode()[0] , inplace=True)
```

In [83]:
```
df.head()
```

Out[83]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S

Calculating the sum of null values.

In [84]:
```
df.isnull().sum()
```

Out[84]:

```
Survived    0
Pclass      0
Name        0
Sex         0
Age         0
SibSp       0
Parch       0
Fare        0
Embarked    0
dtype: int64
```

Analysis of Age Column.

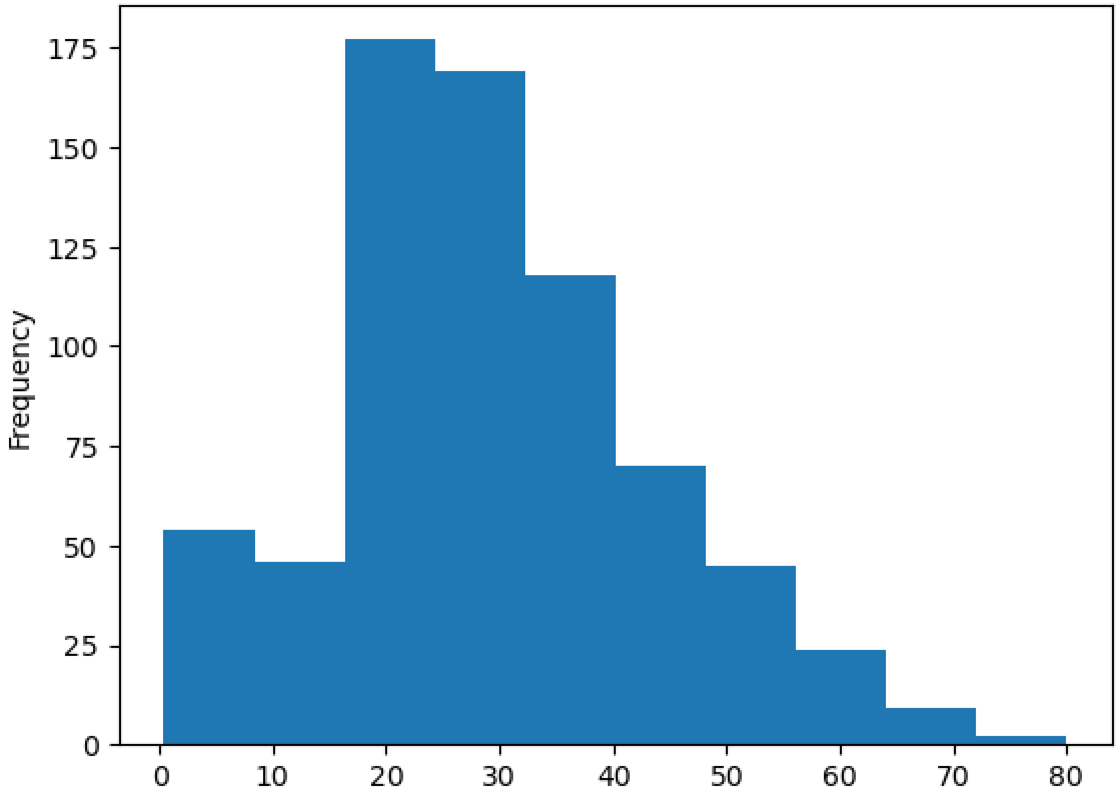
Plotting Histogram of Age Column.

In [13]:

```
df['Age'].plot.hist()
```

Out[13]:

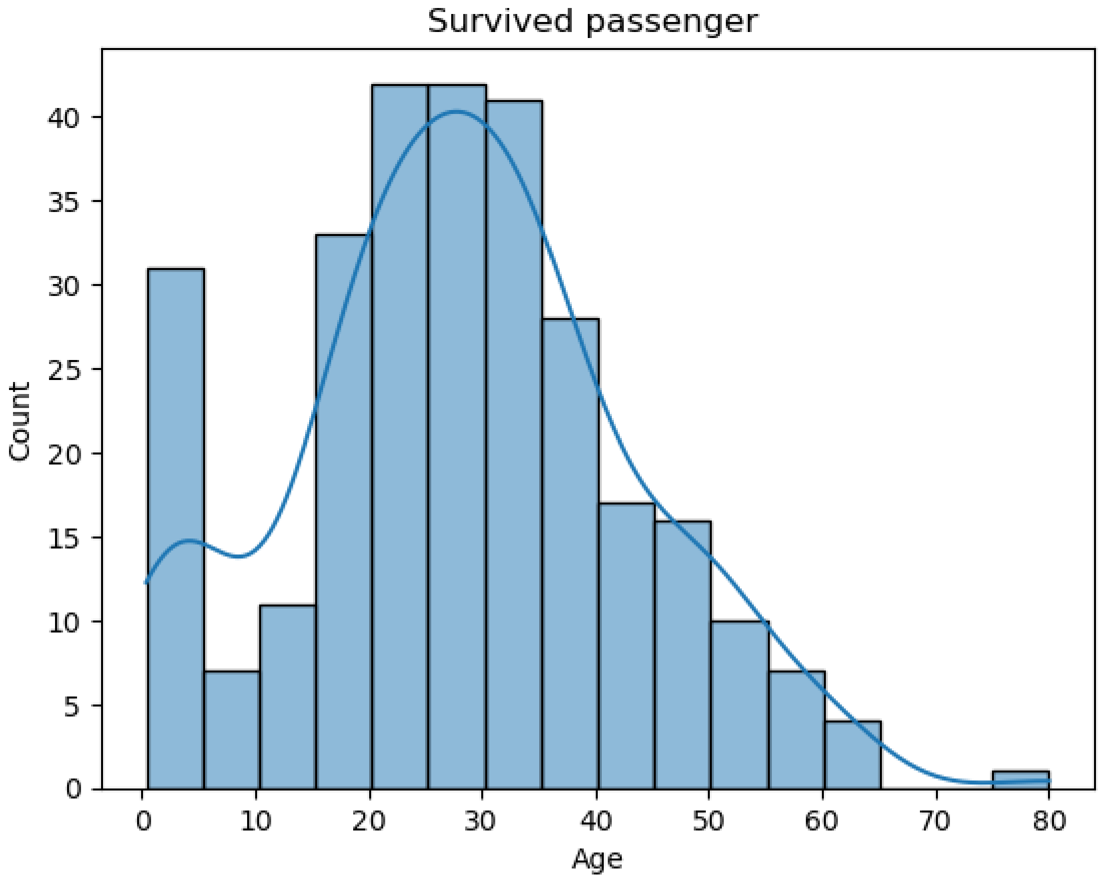
```
<Axes: ylabel='Frequency'>
```



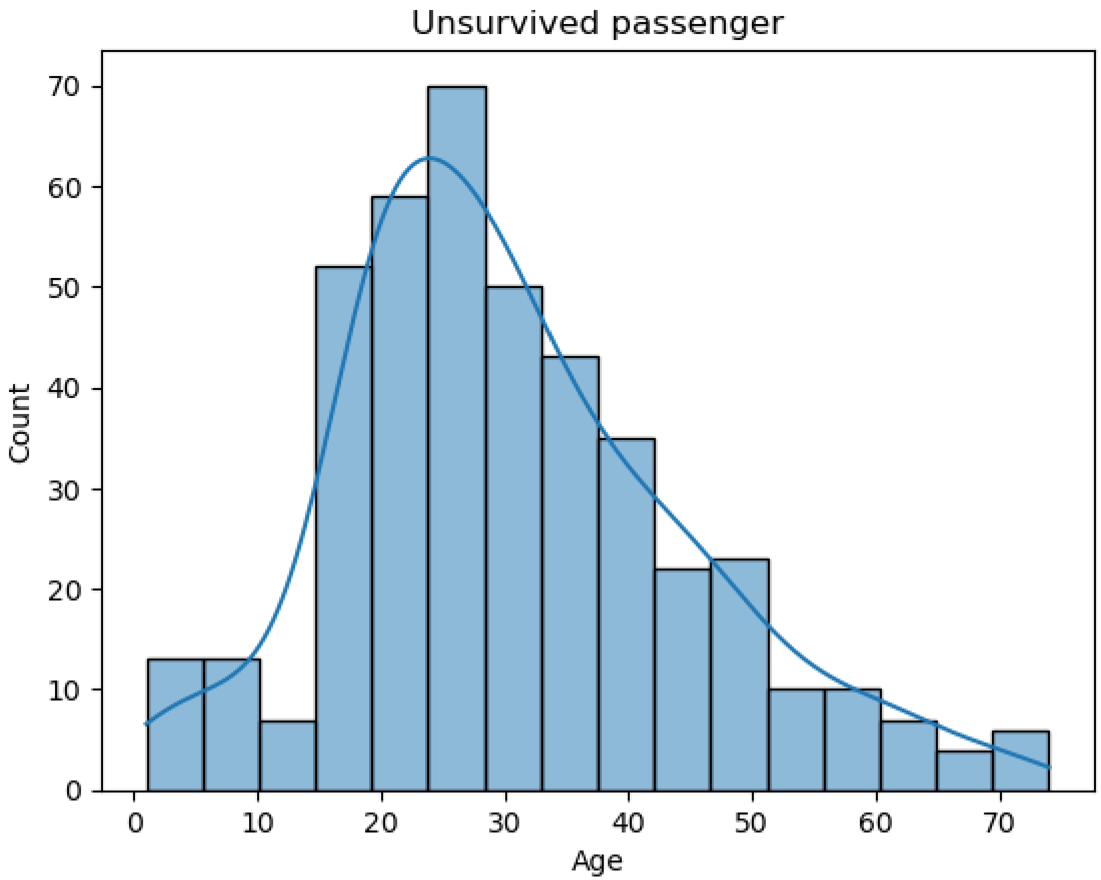
In [8]:
```
survived=df[df['Survived']==1]

unsurvived=df[df['Survived']==0]
```

In [9]:
```
plt.title('Survived passenger ')
plot=sns.histplot(data=survived , x='Age' , kde=True)
```



In [10]:
```
plt.title('Unsurvived passenger')
plot=sns.histplot(data=unsurvived , x='Age' , kde=True )
```



```
In [75]: df.head()
```

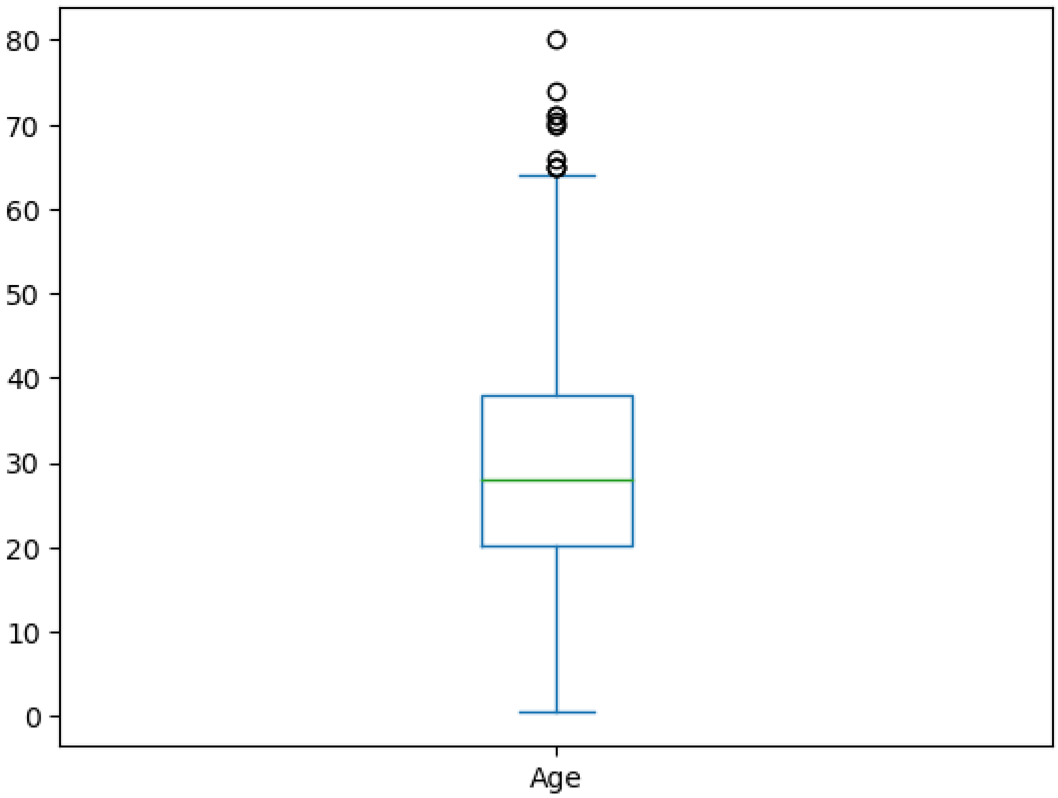
Out[75]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S

Plotting Scatter box of Age Column.

```
In [14]: df['Age'].plot.box()
```

```
Out[14]: <Axes: >
```



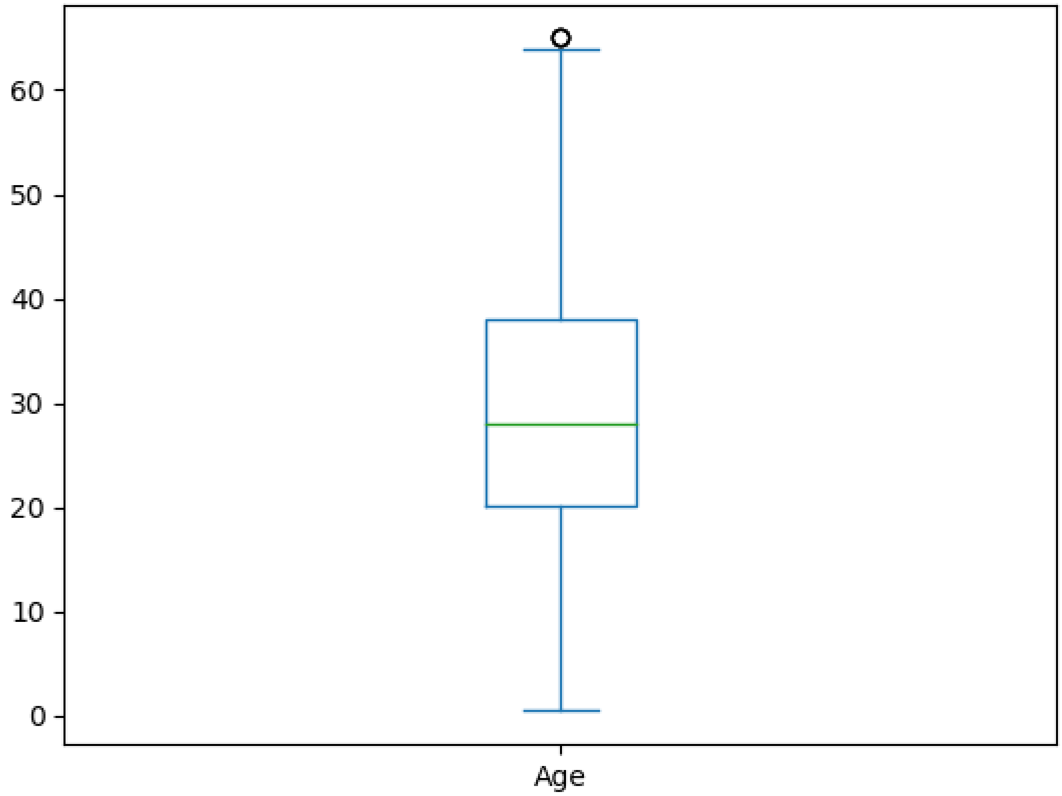
Outliers Treatment in Age Column.

```
In [15]: df.loc[df['Age']>65,'Age']=np.mean(df['Age'])
```

Plotting Scatter plot after treating outliers in Age Column.

```
In [16]: df['Age'].plot.box()
```

```
Out[16]: <Axes: >
```



Bivariate Analysis.

Correlation between 'Survived' and 'Pclass'.

```
In [20]: df['Survived'].corr(df['Pclass'])
```

```
Out[20]: -0.33848103596101503
```

```
In [21]: df[['Survived' , 'Pclass']].corr()
```

Out[21]:

	Survived	Pclass
Survived	1.000000	-0.338481
Pclass	-0.338481	1.000000

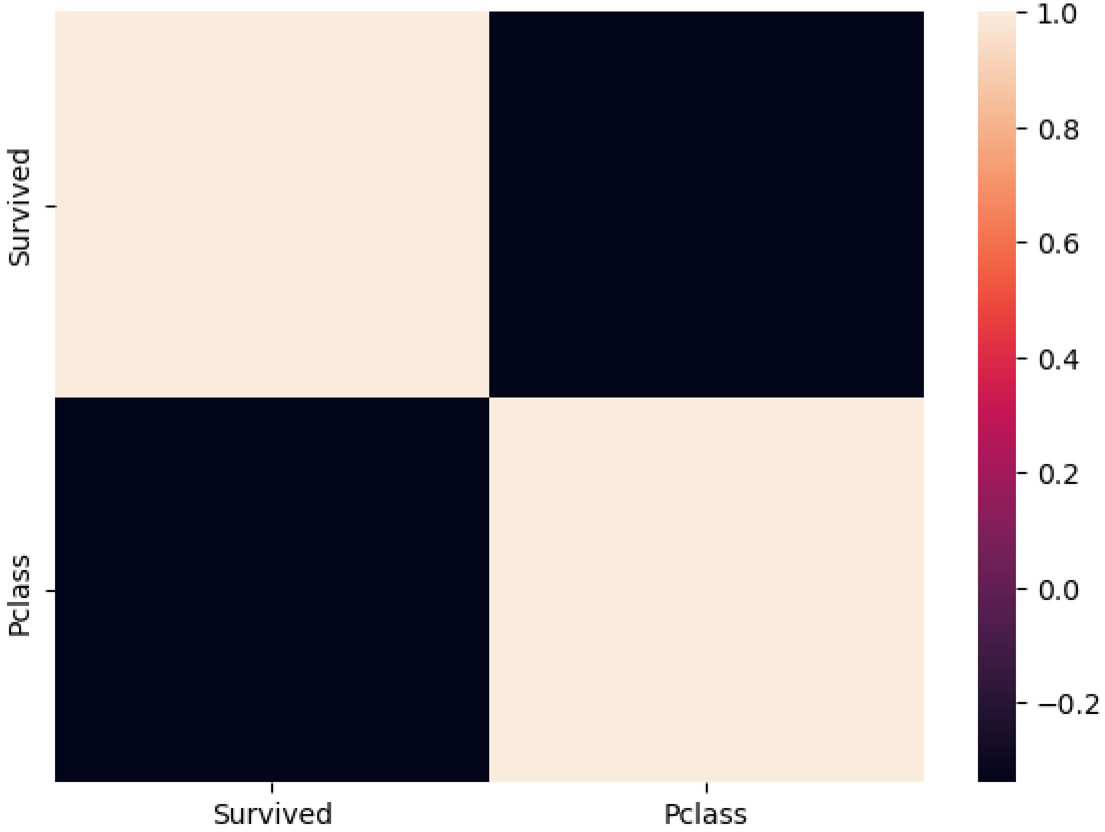
Plotting heatmap of correlation.

In [22]:

```
plt.figure(figsize=(7,5))
sns.heatmap(df[['Survived' , 'Pclass']].corr())
```

Out[22]:

```
<Axes: >
```



Correlation between 'Parch' and 'Pclass'.

In [23]:

```
df['Parch'].corr(df['Pclass'])
```

Out[23]:

```
0.018442671310748497
```

In [24]:

```
df[['Pclass' , 'Parch']].corr()
```

Out[24]:

	Pclass	Parch
Pclass	1.000000	0.018443
Parch	0.018443	1.000000

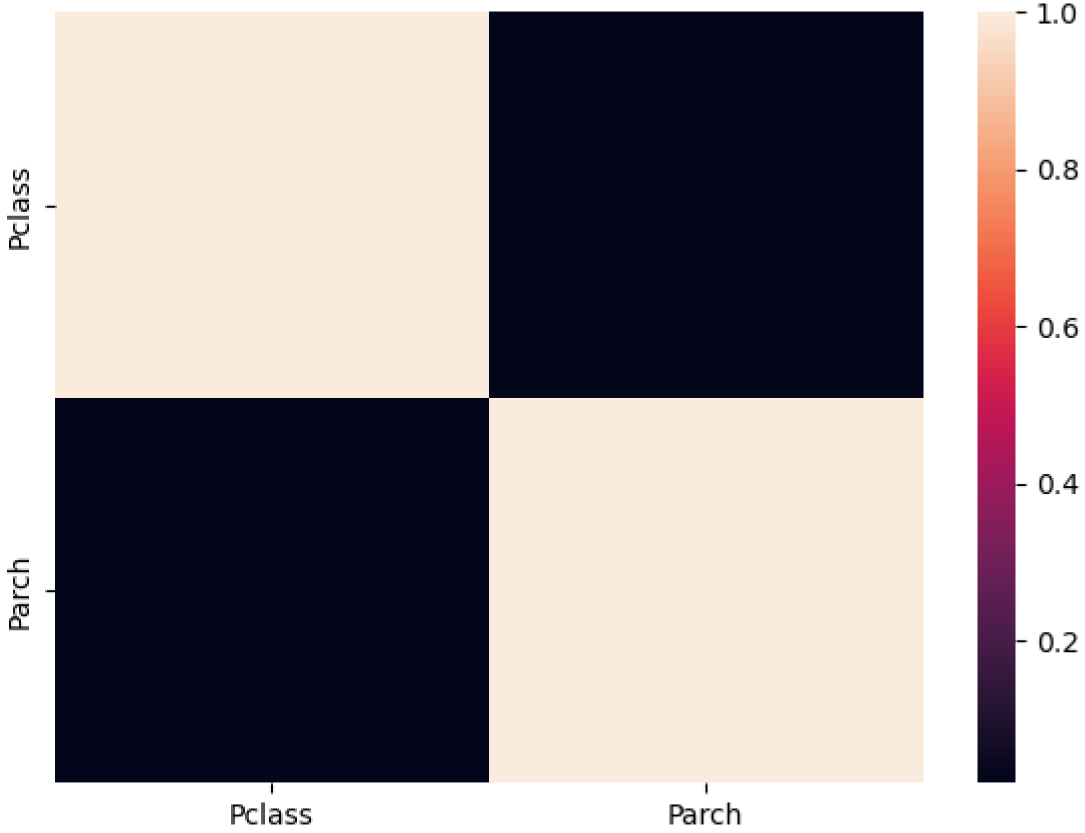
Plotting heatmap of correlation.

In [25]:

```
plt.figure(figsize=(7,5))
sns.heatmap(df[['Pclass' , 'Parch']].corr())
```

Out[25]:

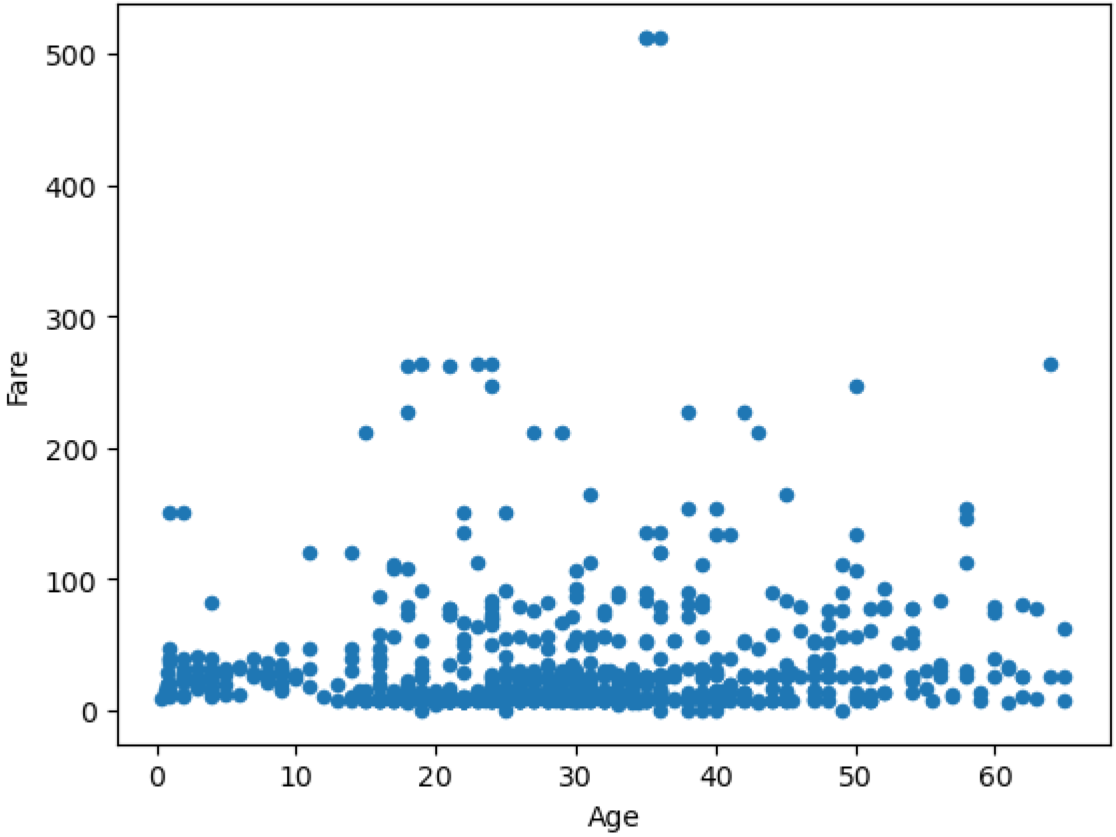
```
<Axes: >
```



Scatter Plot b/w 'Fare'and 'Age'.

In [26]:
```
df.plot.scatter('Age','Fare')
```

Out[26]:
```
<Axes: xlabel='Age', ylabel='Fare'>
```



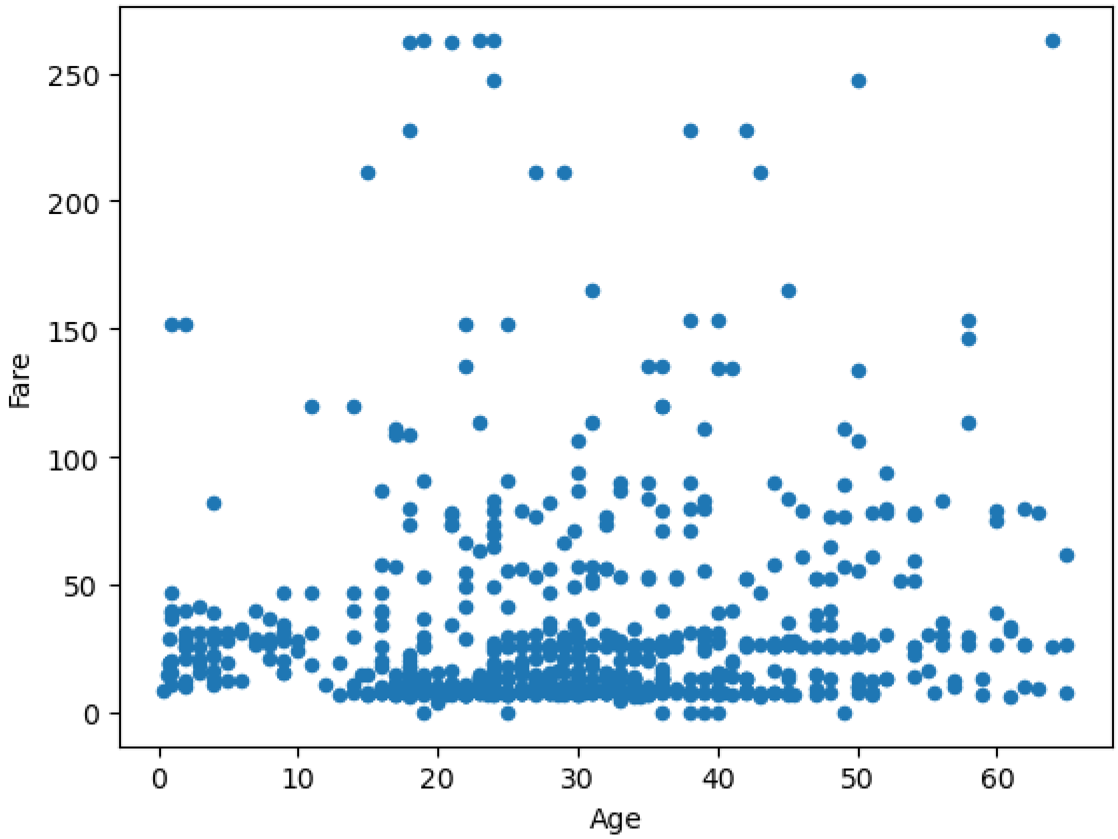
Treating Outliers in Fare Column.

```
In [27]: df=df[df['Fare']<300]
```

Plotting Scatter plot after treating outliers.

```
In [28]: df.plot.scatter('Age','Fare')
```

```
Out[28]: <Axes: xlabel='Age', ylabel='Fare'>
```



Univariate Analysis.

Counting the Values of Male and Female.

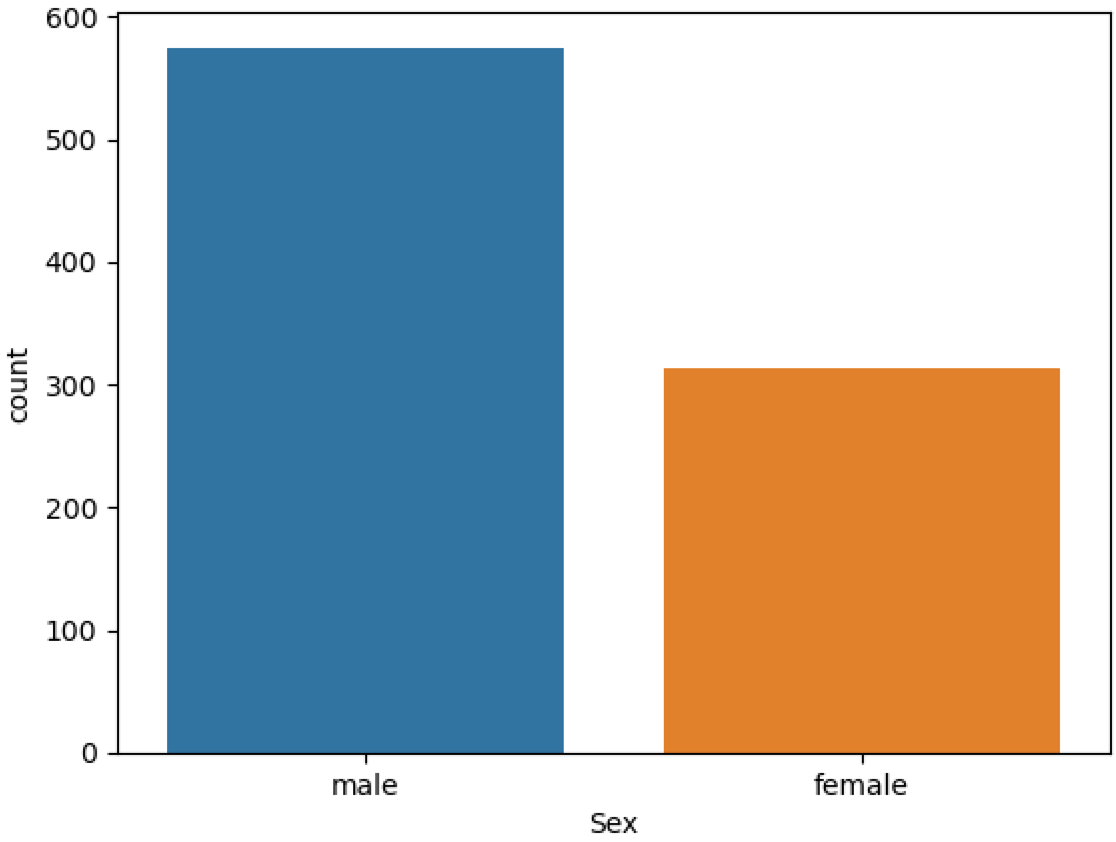
```
In [29]: df['Sex'].value_counts()
```

```
Out[29]: male      575
         female    313
         Name: Sex, dtype: int64
```

Plotting bar graph of Sex Column.

```
In [30]: sns.countplot(x='Sex',data=df)
```

```
Out[30]: <Axes: xlabel='Sex', ylabel='count'>
```



Counting values of 'Pclass'.

```
In [5]: df['Pclass'].value_counts()
```

```
Out[5]: 3    491
        1    216
        2    184
        Name: Pclass, dtype: int64
```

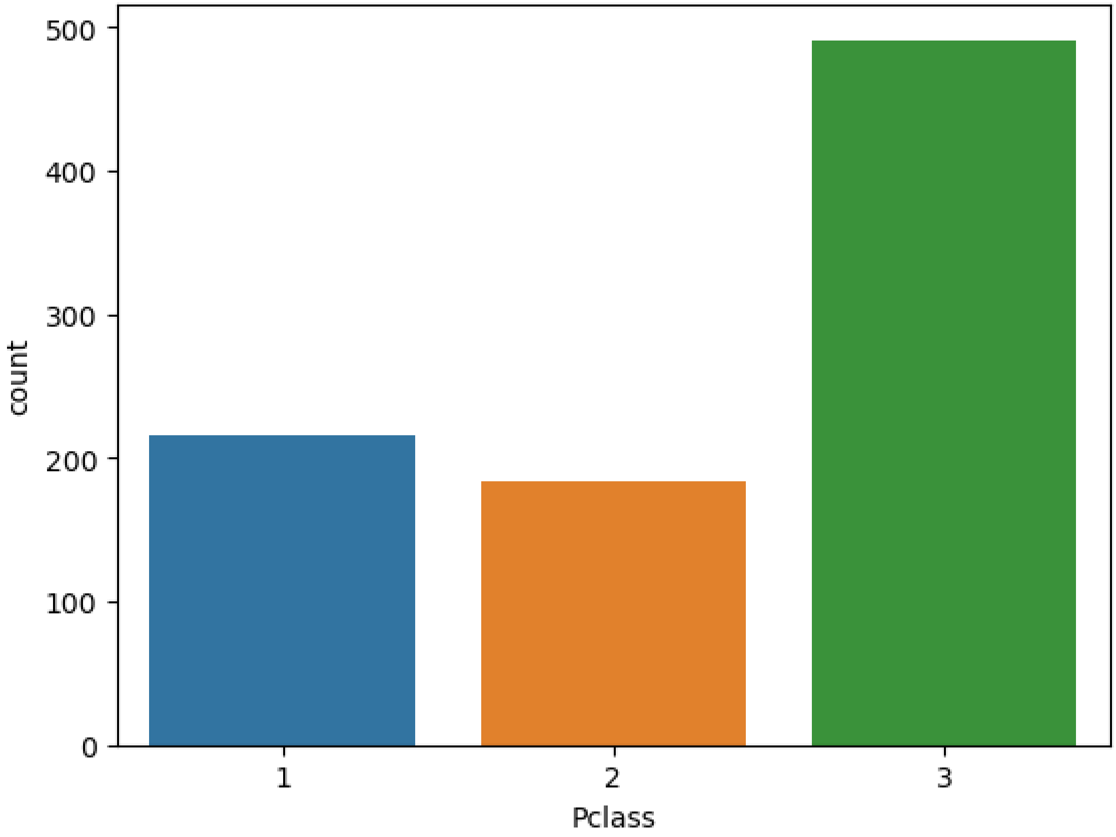
Plotting bar graph of Pclass Column.

In [4]:

```
sns.countplot(x='Pclass',data=df)
```

Out[4]:

```
<Axes: xlabel='Pclass', ylabel='count'>
```



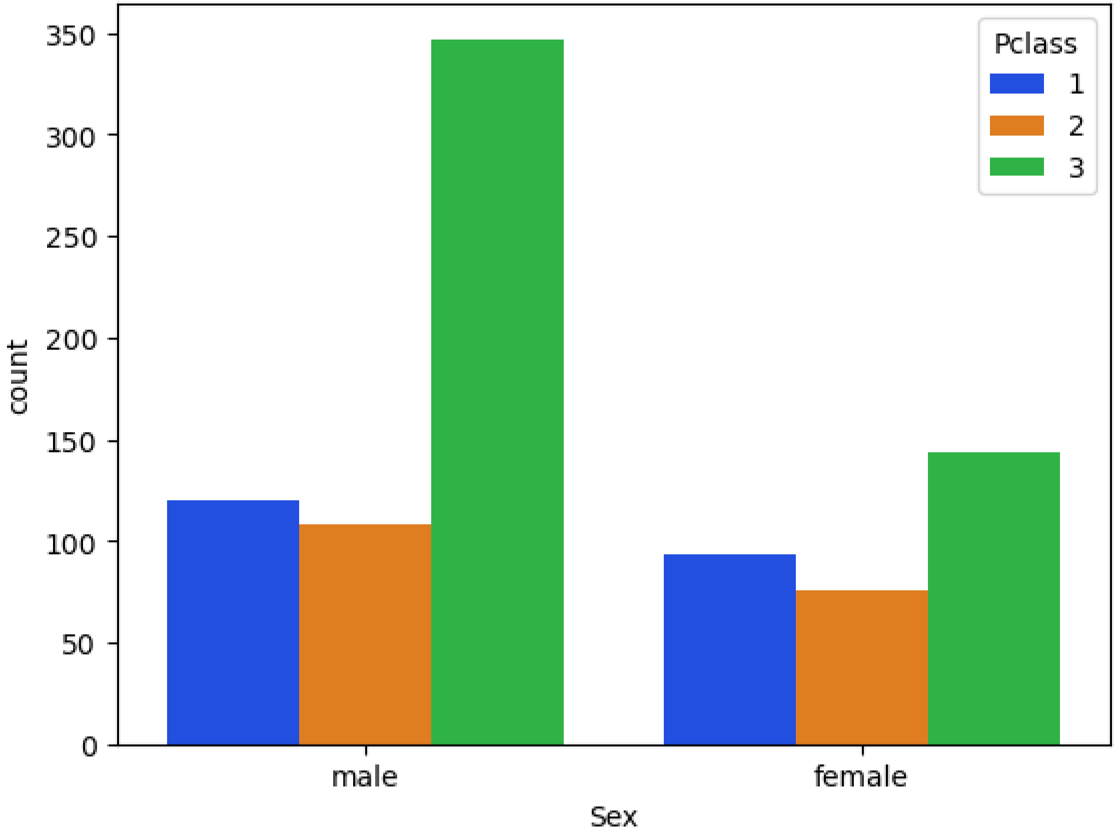
Plotting bargraph b/w 'Sex' and 'Pclass' Column.

In [32]:

```
sns.countplot(x='Sex', hue='Pclass',data=df, palette='bright')
```

Out[32]:

```
<Axes: xlabel='Sex', ylabel='count'>
```



Counting the Values of 'Survived' Column.

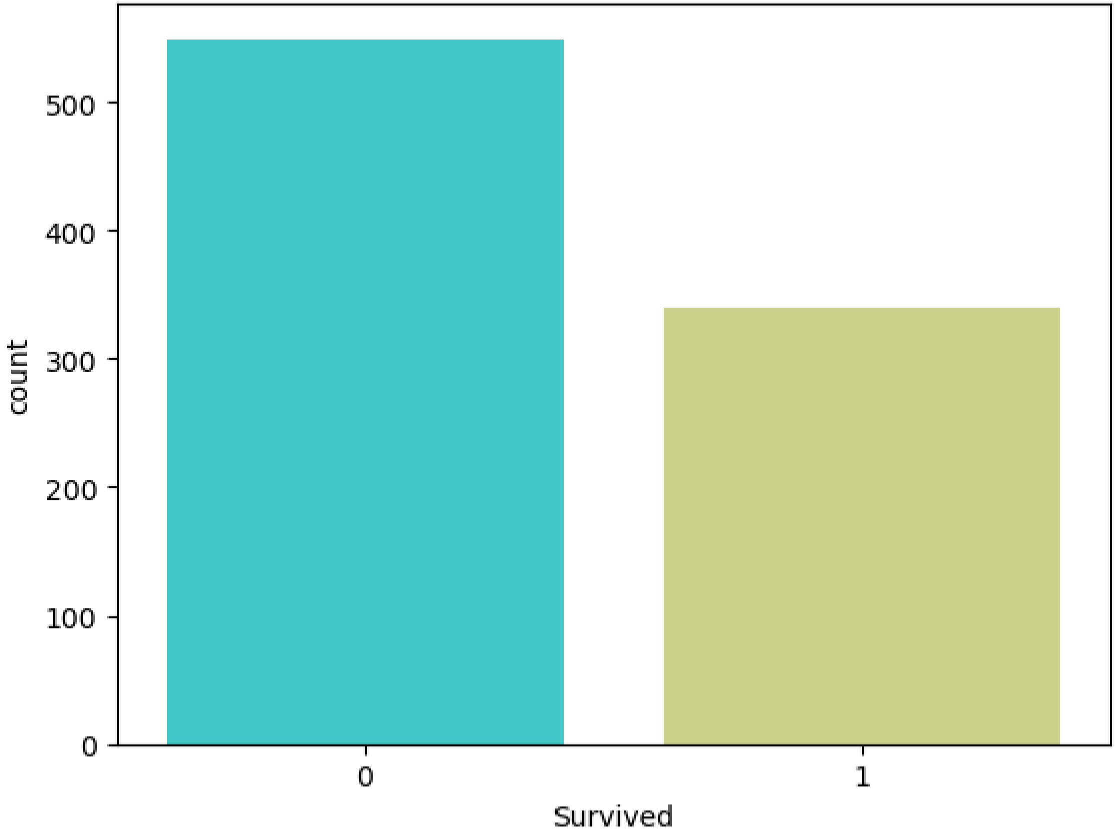
```
In [33]: df['Survived'].value_counts()
```

```
Out[33]: 0    549
         1    339
         Name: Survived, dtype: int64
```

Plotting bar graph of 'Survived' Column.

```
In [34]: sns.countplot(x='Survived',data=df, palette='rainbow')
```

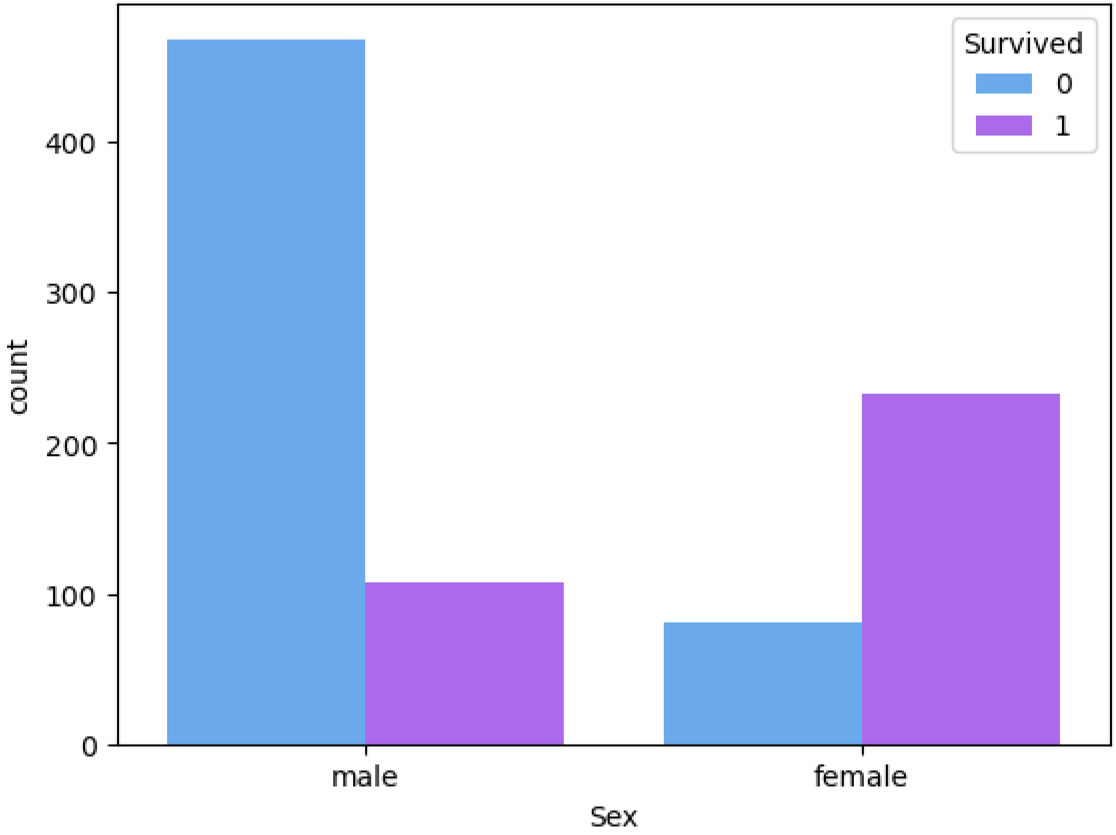
```
Out[34]: <Axes: xlabel='Survived', ylabel='count'>
```



Plotting bargraph b/w 'Sex' and 'Survived' Column.

```
In [35]: sns.countplot(x='Sex', hue='Survived',data=df, palette='cool')
```

```
Out[35]: <Axes: xlabel='Sex', ylabel='count'>
```



Counting 'SibSp' column.

In [36]:

```
df['SibSp'].value_counts()
```

Out[36]:

```
0    605
1    209
2     28
4     18
3     16
8      7
5      5
Name: SibSp, dtype: int64
```

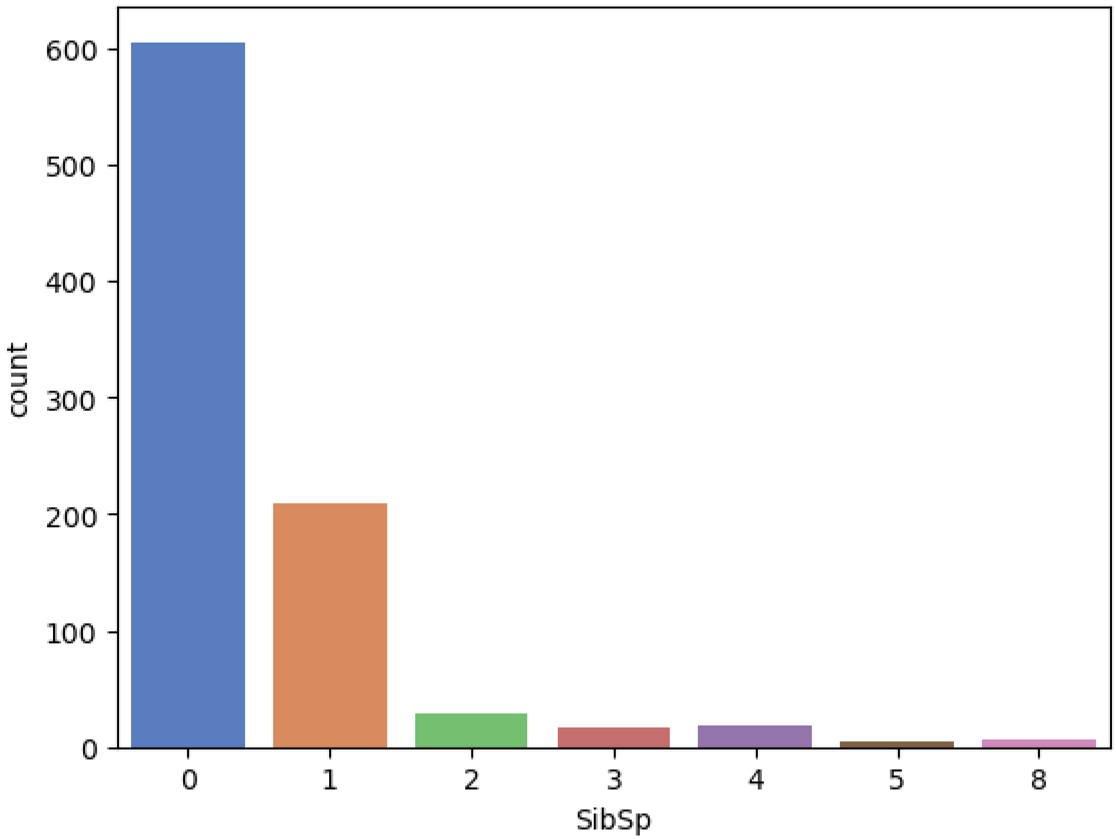
Plotting bar graph of 'SibSp' Column.

In [37]:

```
sns.countplot(x='SibSp',data=df, palette='muted')
```

Out[37]:

```
<Axes: xlabel='SibSp', ylabel='count'>
```



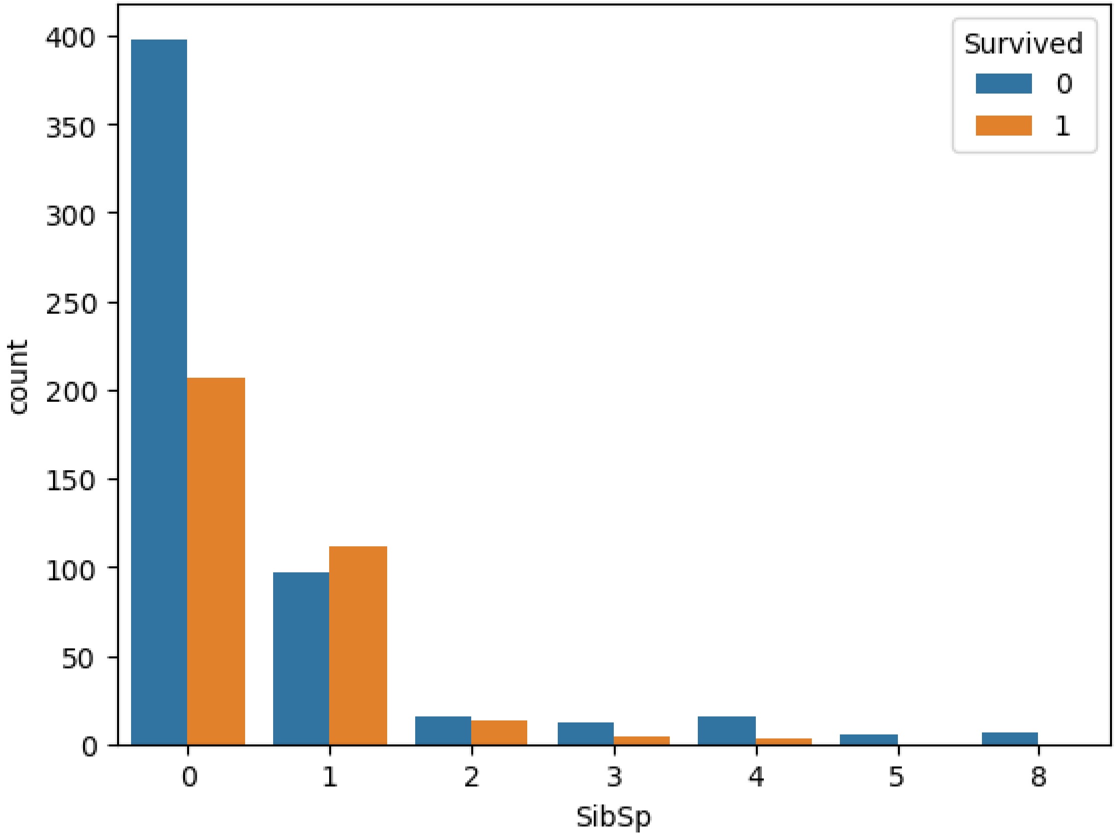
Plotting bargraph b/w 'SibSp' and 'Survived' Column.

In [38]:

```
sns.countplot(x='SibSp',hue='Survived',data=df)
```

Out[38]:

```
<Axes: xlabel='SibSp', ylabel='count'>
```



Counting Values of 'Embarked' Column.

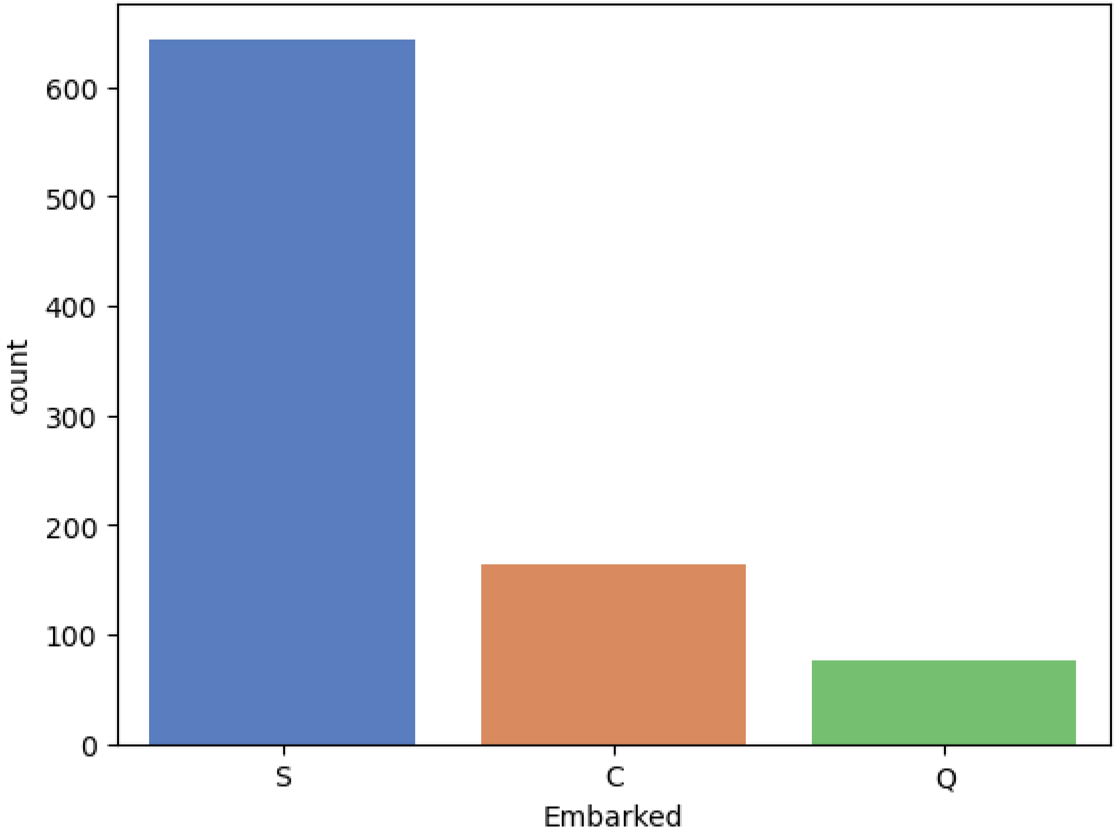
```
In [69]: df['Embarked'].value_counts()
```

```
Out[69]: S    644
         C    168
         Q     77
         Name: Embarked, dtype: int64
```

Plotting bar graph of 'Embarked' Column.

```
In [39]: sns.countplot(x='Embarked',data=df, palette='muted')
```

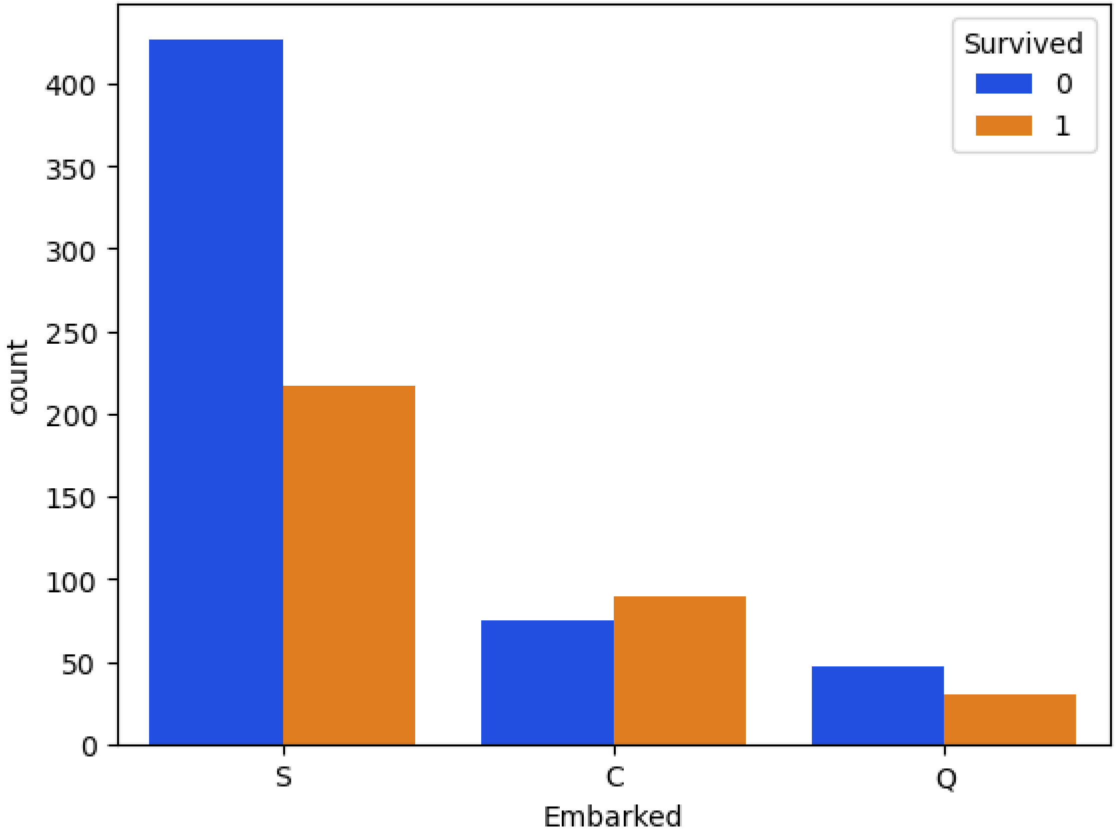
```
Out[39]: <Axes: xlabel='Embarked', ylabel='count'>
```



Plotting bargraph b/w 'Emabrked' and 'Survived' Column.

```
In [40]: sns.countplot(x='Embarked',hue='Survived',data=df, palette='bright')
```

```
Out[40]: <Axes: xlabel='Embarked', ylabel='count'>
```



Counting the Values of 'Parch' Column.

```
In [41]: df['Parch'].value_counts()
```

```
Out[41]: 0    676
         1    117
         2     80
         5      5
         3      5
         4      4
         6      1
         Name: Parch, dtype: int64
```

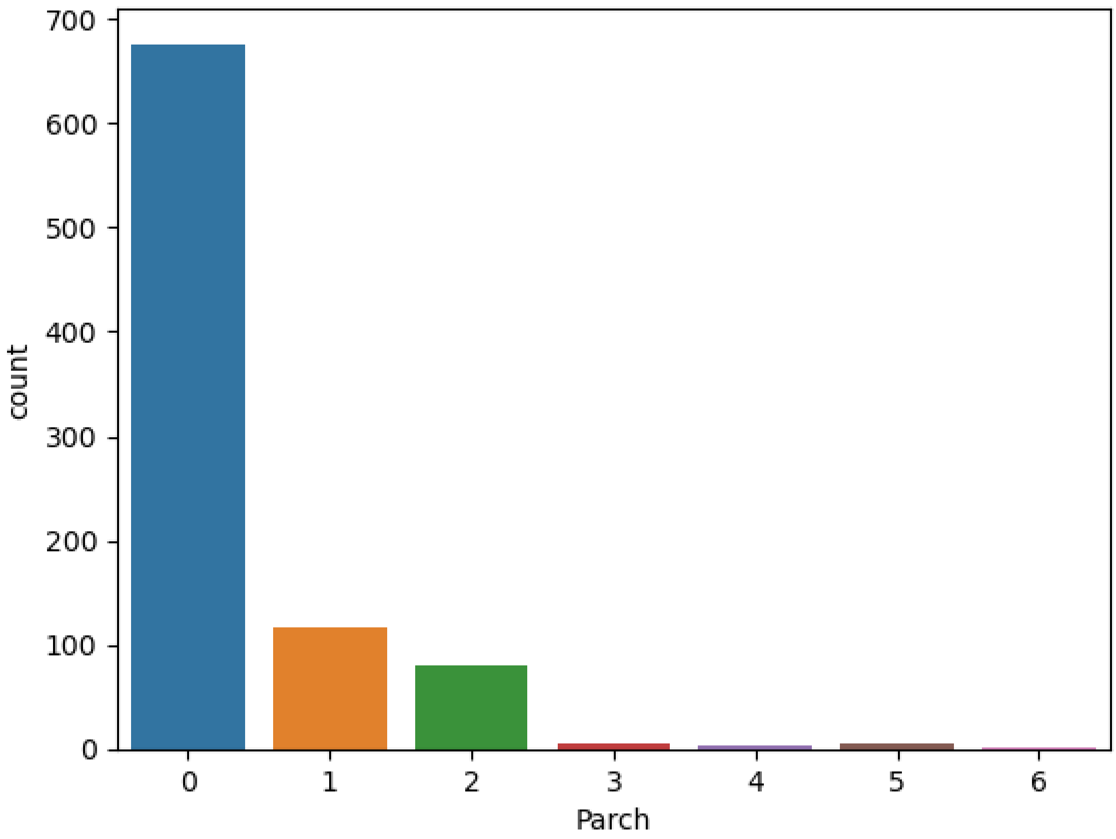
Plotting bar graph of 'Parch' Column.

In [ ]:

```
sns.countplot(x='Parch',data=df)
```

Out[ ]:

```
<Axes: xlabel='Parch', ylabel='count'>
```



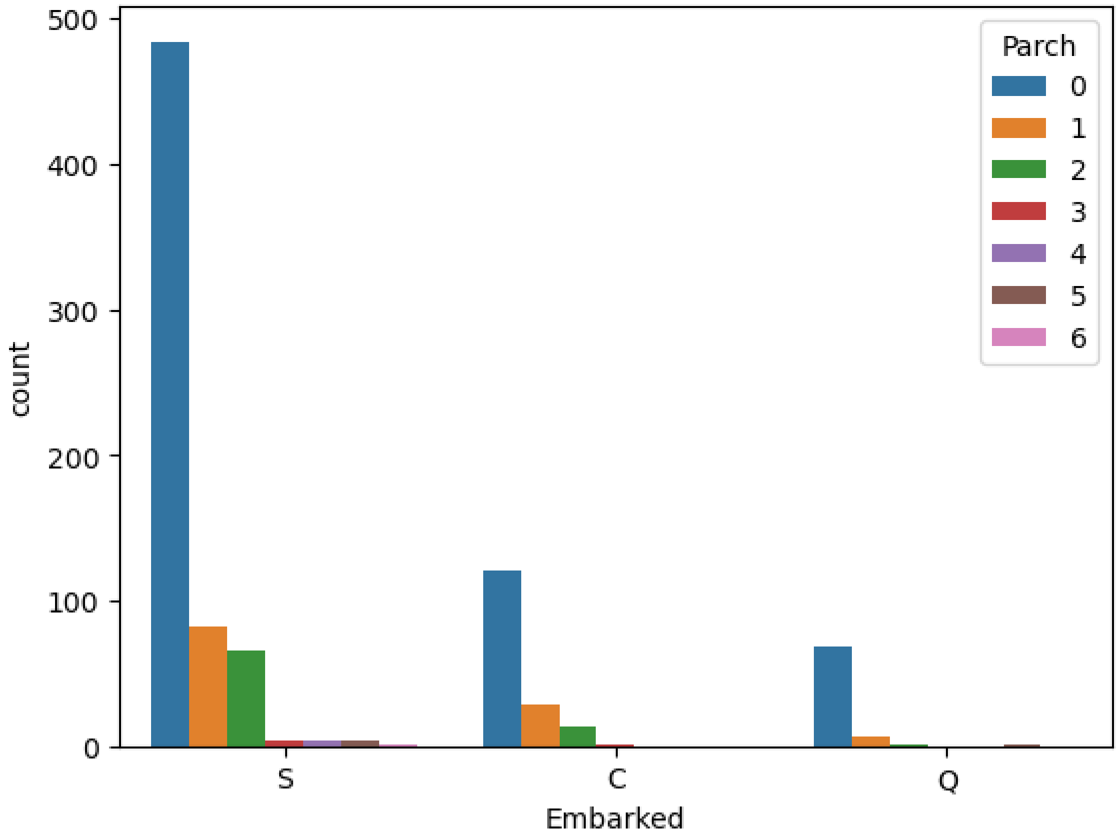
Plotting bargraph b/w 'Emabrked' and 'Parch' Column.

In [43]:

```
sns.countplot(x='Embarked',hue='Parch',data=df)
```

Out[43]:

```
<Axes: xlabel='Embarked', ylabel='count'>
```



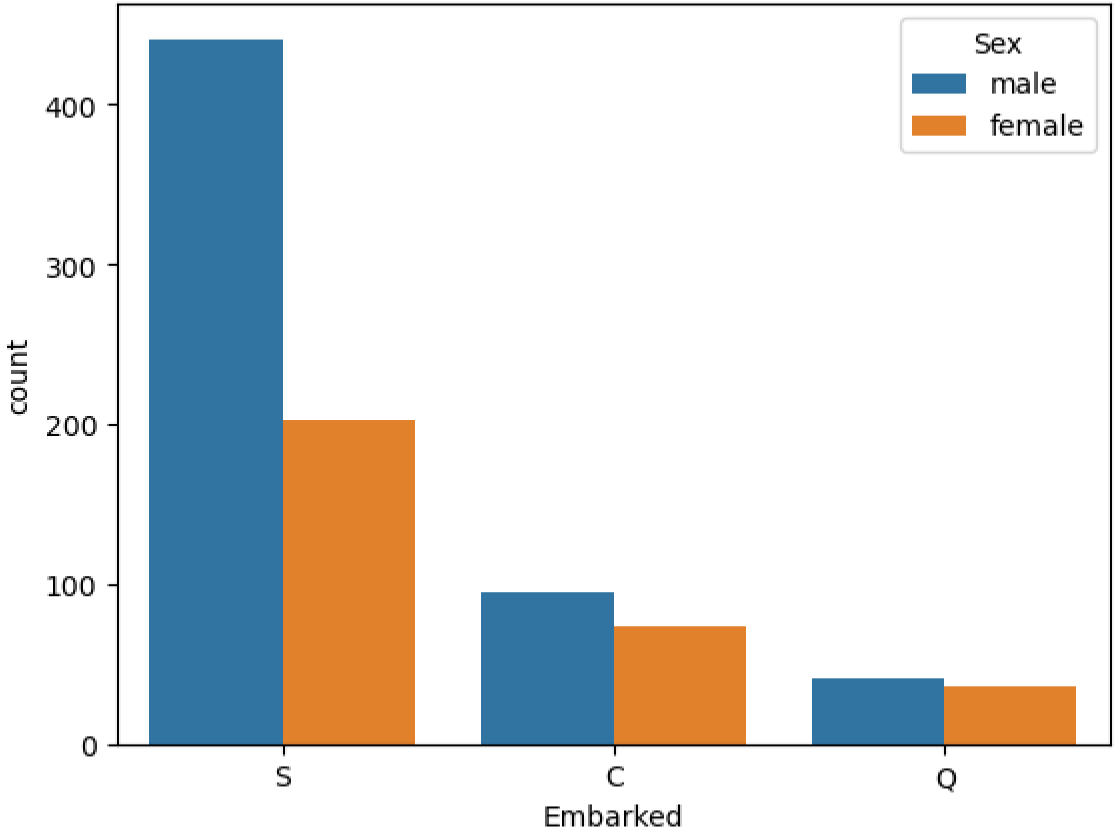
Plotting bargraph b/w 'Emabrked' and 'Sex' Column.

In [70]:

```
sns.countplot(x='Embarked',hue='Sex',data=df)
```

Out[70]:

```
<Axes: xlabel='Embarked', ylabel='count'>
```



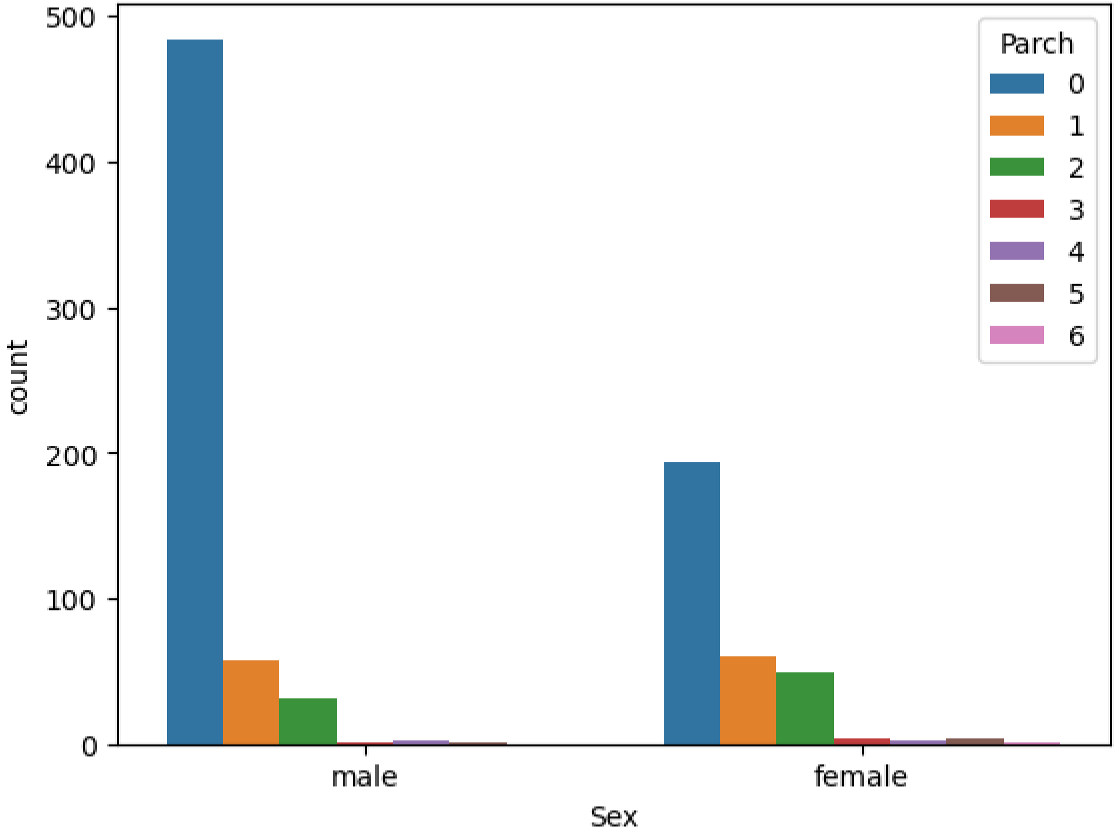
Plotting bargraph b/w 'Sex' and 'Parch' Column.

In [71]:

```
sns.countplot(x='Sex',hue='Parch',data=df)
```

Out[71]:

```
<Axes: xlabel='Sex', ylabel='count'>
```



In [ ]:

In [ ]: