Practical - 1

Name - Sarthak Abhaykumar Nahar

Batch - B4

Roll no. - 55

In [2]:

```
import pandas as pd

df = pd.read_csv('titanic.csv')
```

In [3]:

```
# Get the first five data
print(df.head())
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3   
1            2         1       1   
2            3         1       3   
3            4         1       1   
4            5         0       3   

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1   
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1   
2                             Heikkinen, Miss. Laina  female  26.0      0   
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1   
4                           Allen, Mr. William Henry    male  35.0      0   

   Parch            Ticket     Fare Cabin Embarked  
0      0         A/5 21171   7.2500   NaN        S  
1      0          PC 17599  71.2833   C85        C  
2      0  STON/O2. 3101282   7.9250   NaN        S  
3      0            113803  53.1000  C123        S  
4      0            373450   8.0500   NaN        S  
```

In [4]:

```
# Get bottom 5 record
print(df.tail())
```

```
     PassengerId  Survived  Pclass                                      Name  \
886          887         0       2                     Montvila, Rev. Juozas   
887          888         1       1              Graham, Miss. Margaret Edith   
888          889         0       3  Johnston, Miss. Catherine Helen "Carrie"   
889          890         1       1                     Behr, Mr. Karl Howell   
890          891         0       3                       Dooley, Mr. Patrick   

        Sex   Age  SibSp  Parch      Ticket   Fare Cabin Embarked  
886    male  27.0      0      0      211536  13.00   NaN        S  
887  female  19.0      0      0      112053  30.00   B42        S  
888  female   NaN      1      2  W./C. 6607  23.45   NaN        S  
889    male  26.0      0      0      111369  30.00  C148        C  
890    male  32.0      0      0      370376   7.75   NaN        Q  
```

In [5]:

```
#Get the Statistical Information abot the dataset
print(df.describe())
```

```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000   
mean    446.000000    0.383838    2.308642   29.699118    0.523008   
std     257.353842    0.486592    0.836071   14.526497    1.102743   
min       1.000000    0.000000    1.000000    0.420000    0.000000   
25%     223.500000    0.000000    2.000000   20.125000    0.000000   
50%     446.000000    0.000000    3.000000   28.000000    0.000000   
75%     668.500000    1.000000    3.000000   38.000000    1.000000   
max     891.000000    1.000000    3.000000   80.000000    8.000000   

            Parch        Fare  
count  891.000000  891.000000  
mean     0.381594   32.204208  
std      0.806057   49.693429  
min      0.000000    0.000000  
25%      0.000000    7.910400  
50%      0.000000   14.454200  
75%      0.000000   31.000000  
max      6.000000  512.329200  
```

In [7]:

```
#Get the number of rows and columns
```

```
print(df.shape)
```

```
(891, 12)
```

In [10]:

```
#Check for missing values

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

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [13]:

```
df['Age'].fillna(df['Age'].median(), inplace=True)
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
df.drop(columns=['Cabin'],inplace=True)
```

In [19]:

```
duplicates = df.duplicated().sum()
print(f"Number of duplicates rows: {duplicates}")
df.drop_duplicates(inplace=True)
```

```
Number of duplicates rows: 0
```

In [20]:

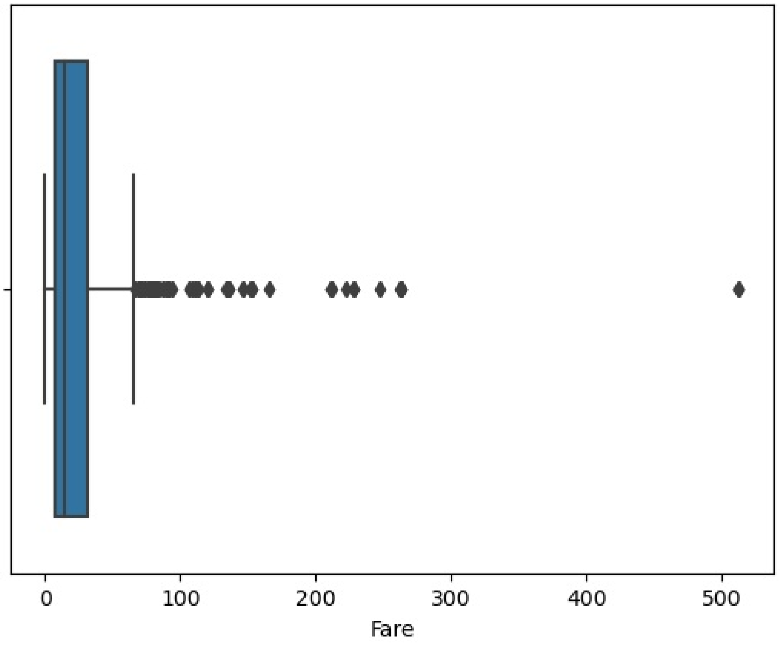
```
import matplotlib.pyplot as plt
```

In [21]:

```
import seaborn as sns
```

In [22]:

```
sns.boxplot(x=df['Fare'])
plt.show()
```



In [27]:

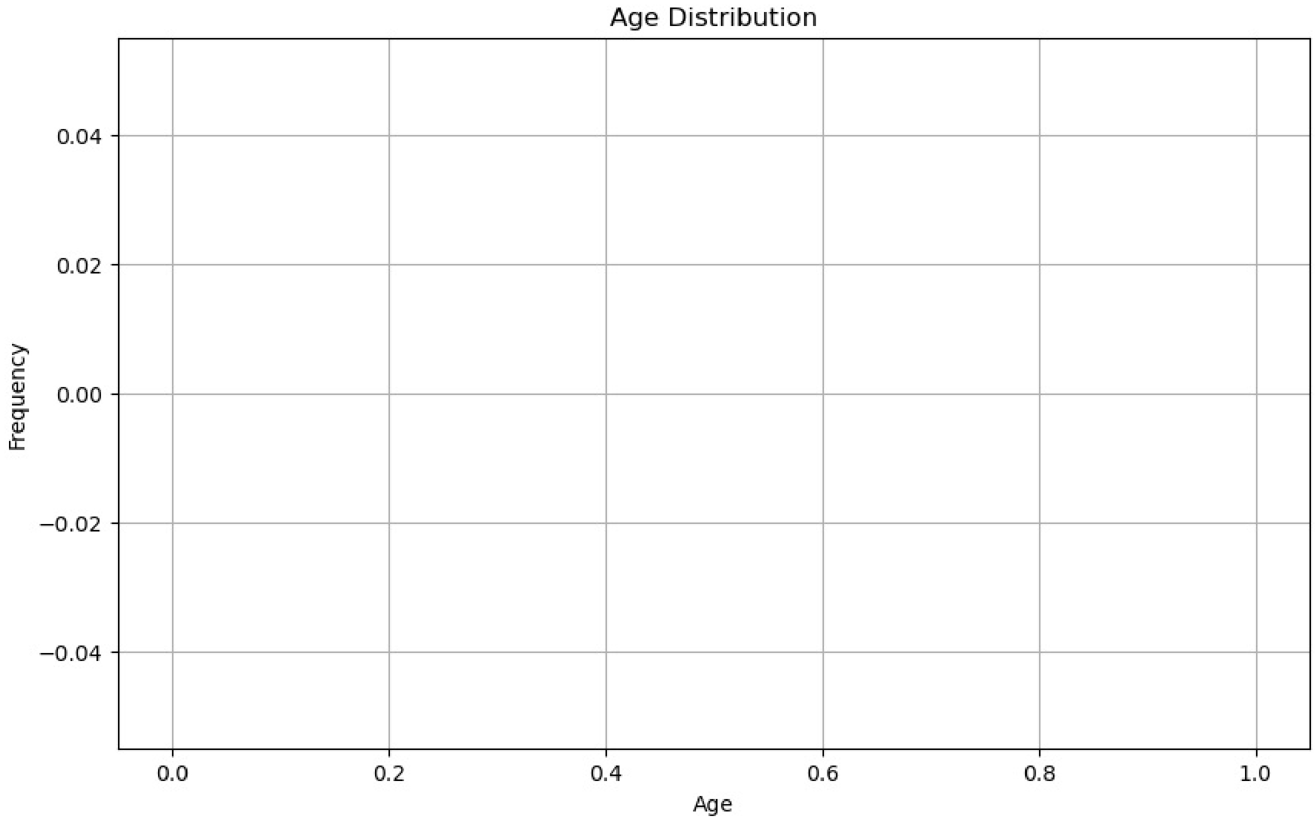
```
Q1 = df['Fare'].quantile(0.25)
Q3 = df['Fare'].quantile(0.75)
IQR = Q3-Q1
df = df[~((df['Fare'] < (Q1 - 1.5*IQR)) | (df['Fare'] > (Q3 - 1.5*IQR)))]
```

In [28]:

```
# Apply one-hot encoding to the 'Sex' column
df = pd.get_dummies(df, columns=['Sex', 'Embarked'], drop_first=True)
```

In [36]:

```
import matplotlib.pyplot as plt

# Univariate analysis for Age
plt.figure(figsize=(10, 6))
df['Age'].hist(bins=30)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



In [38]:

```
print(df.head())
print(df[['Pclass', 'Survived']].isnull().sum())
```

```
Empty DataFrame
Columns: [PassengerId, Survived, Pclass, Name, Age, SibSp, Parch, Ticket, Fare]
Index: []
Pclass      0.0
Survived    0.0
dtype: float64
```

In [40]:

```
import pandas as pd

# Load the dataset
df = pd.read_csv('titanic.csv')

# Check the initial data
print(df.head())
print(df.columns)
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3   
1            2         1       1   
2            3         1       3   
3            4         1       1   
4            5         0       3   

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1   
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1   
2                             Heikkinen, Miss. Laina  female  26.0      0   
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1   
4                           Allen, Mr. William Henry    male  35.0      0   

   Parch            Ticket     Fare Cabin Embarked  
0      0         A/5 21171   7.2500   NaN        S  
1      0          PC 17599  71.2833   C85        C  
2      0  STON/O2. 3101282   7.9250   NaN        S  
3      0            113803  53.1000  C123        S  
4      0            373450   8.0500   NaN        S  
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [41]:

```
# tells the no of columns
print(df.columns)
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [42]:

```
# Example: Check for dropped columns or filtering conditions
print(df.isnull().sum())

# Check any transformations applied to 'Pclass' or 'Survived'
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [43]:

```
# Reload the dataset
df = pd.read_csv('titanic.csv')

# Reapply preprocessing, ensuring not to drop or alter essential columns
df['Age'].fillna(df['Age'].median(), inplace=True)
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
df.drop(columns=['Cabin'], inplace=True)

# Confirm the presence of data
print(df['Pclass'].unique())
print(df['Survived'].unique())
```

```
[3 1 2]
[0 1]
```

In [44]:

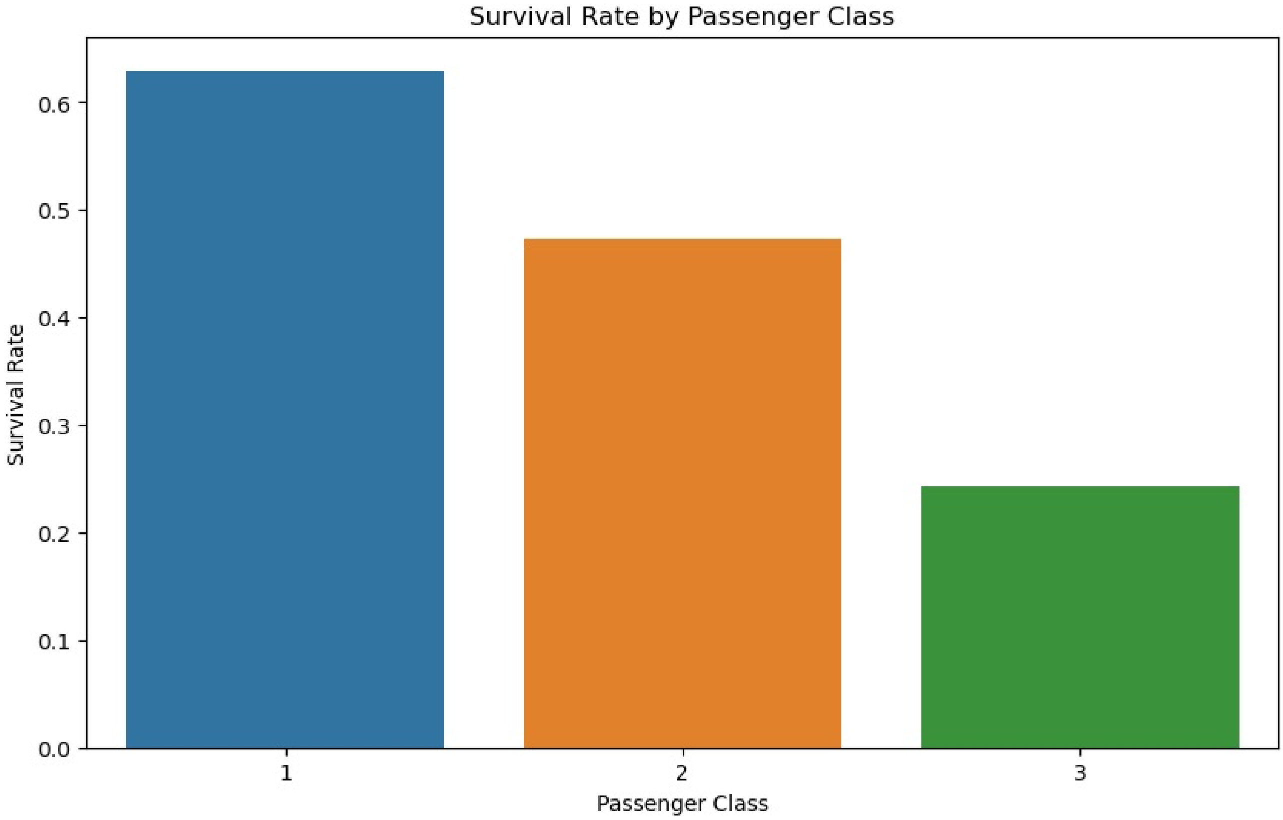
```
# to check the numeric value
print(df.dtypes)
```

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Embarked        object
dtype: object
```

In [45]:

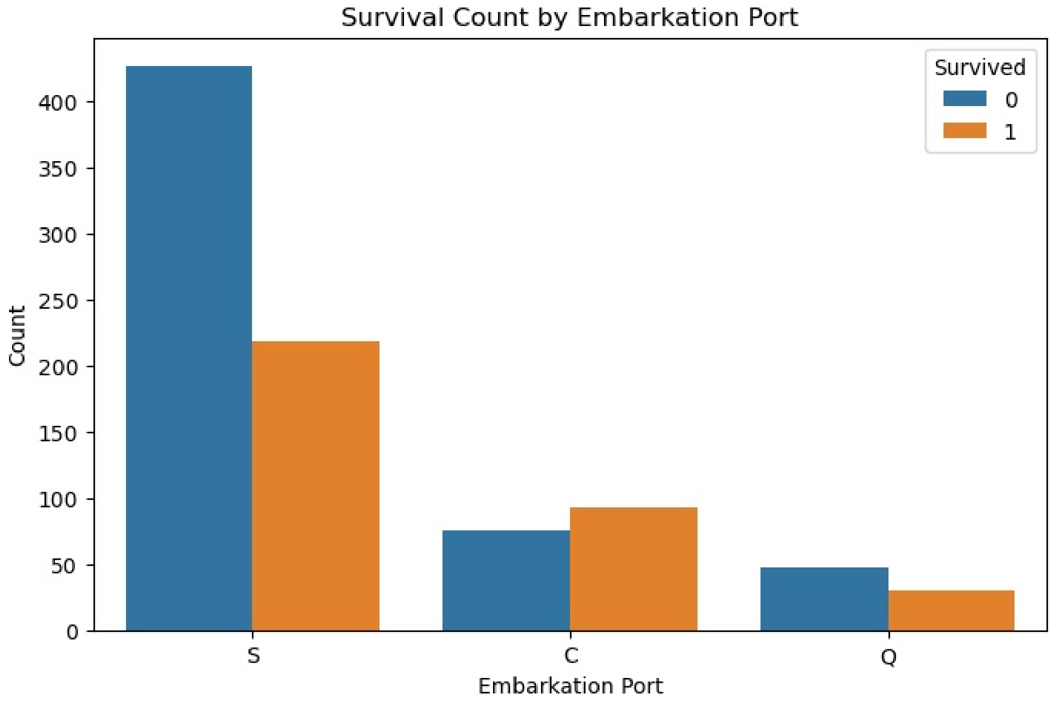
```
import seaborn as sns
import matplotlib.pyplot as plt

# Ensure the DataFrame is not empty and the columns exist
if not df.empty and 'Pclass' in df.columns and 'Survived' in df.columns:
    plt.figure(figsize=(10, 6))
    sns.barplot(x='Pclass', y='Survived', data=df, ci=None)
    plt.title('Survival Rate by Passenger Class')
    plt.xlabel('Passenger Class')
    plt.ylabel('Survival Rate')
    plt.show()
else:
    print("Data is not available for plotting.")
```

```
C:\Users\cse\AppData\Local\Temp\ipykernel_13196\257418103.py:7: FutureWarning: 

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(x='Pclass', y='Survived', data=df, ci=None)
```



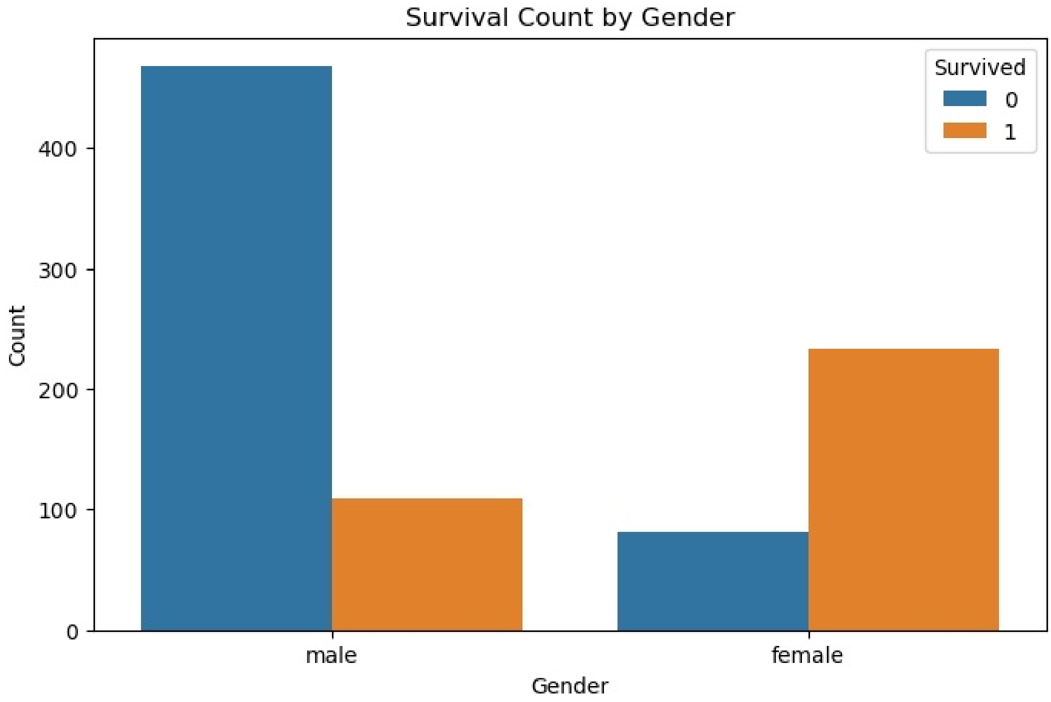
In [47]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Bivariate analysis: Survived vs. Embarked
plt.figure(figsize=(8, 5))
sns.countplot(x='Embarked', hue='Survived', data=df)
plt.title('Survival Count by Embarkation Port')
plt.xlabel('Embarkation Port')
plt.ylabel('Count')
plt.legend(title='Survived', loc='upper right')
plt.show()
```



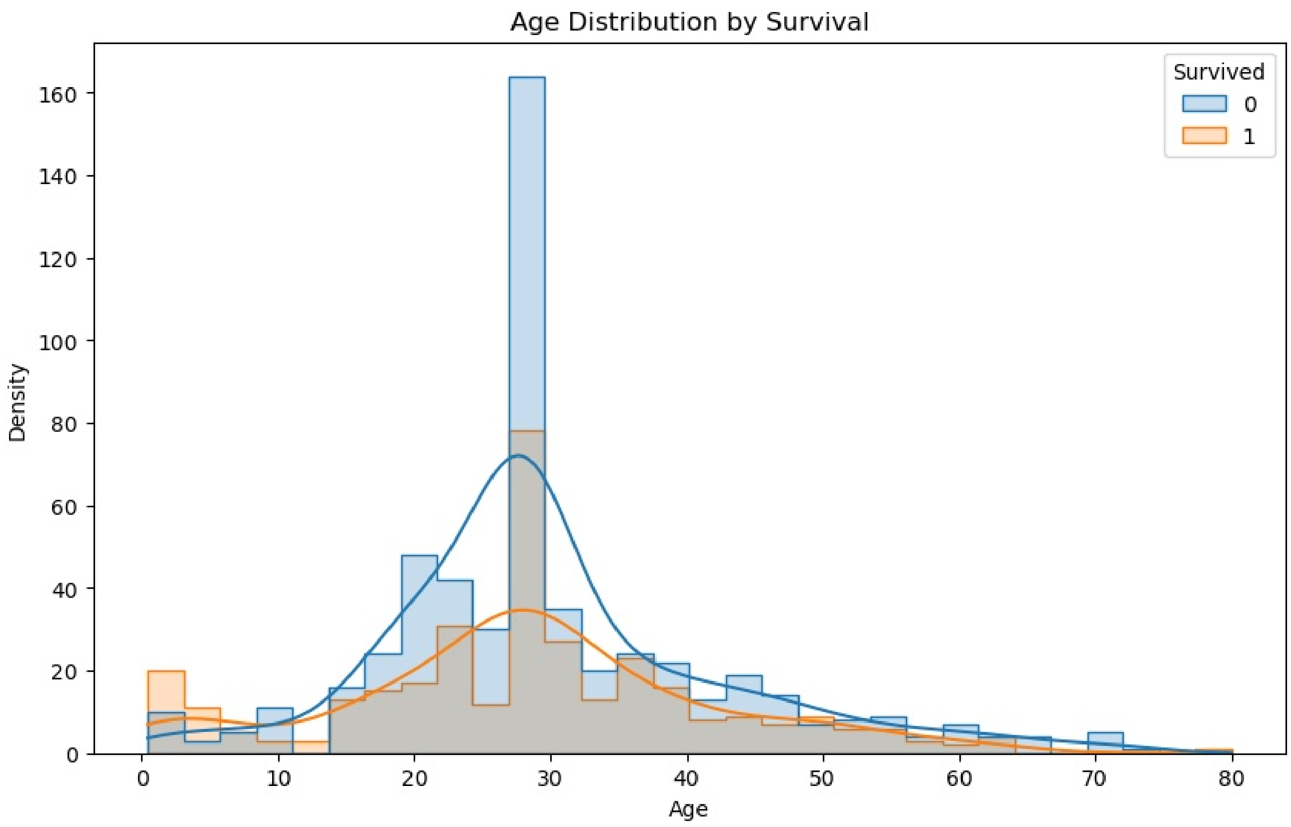
In [48]:

```
# Bivariate analysis: Survived vs. Sex
plt.figure(figsize=(8, 5))
sns.countplot(x='Sex', hue='Survived', data=df)
plt.title('Survival Count by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.legend(title='Survived', loc='upper right')
plt.show()
```

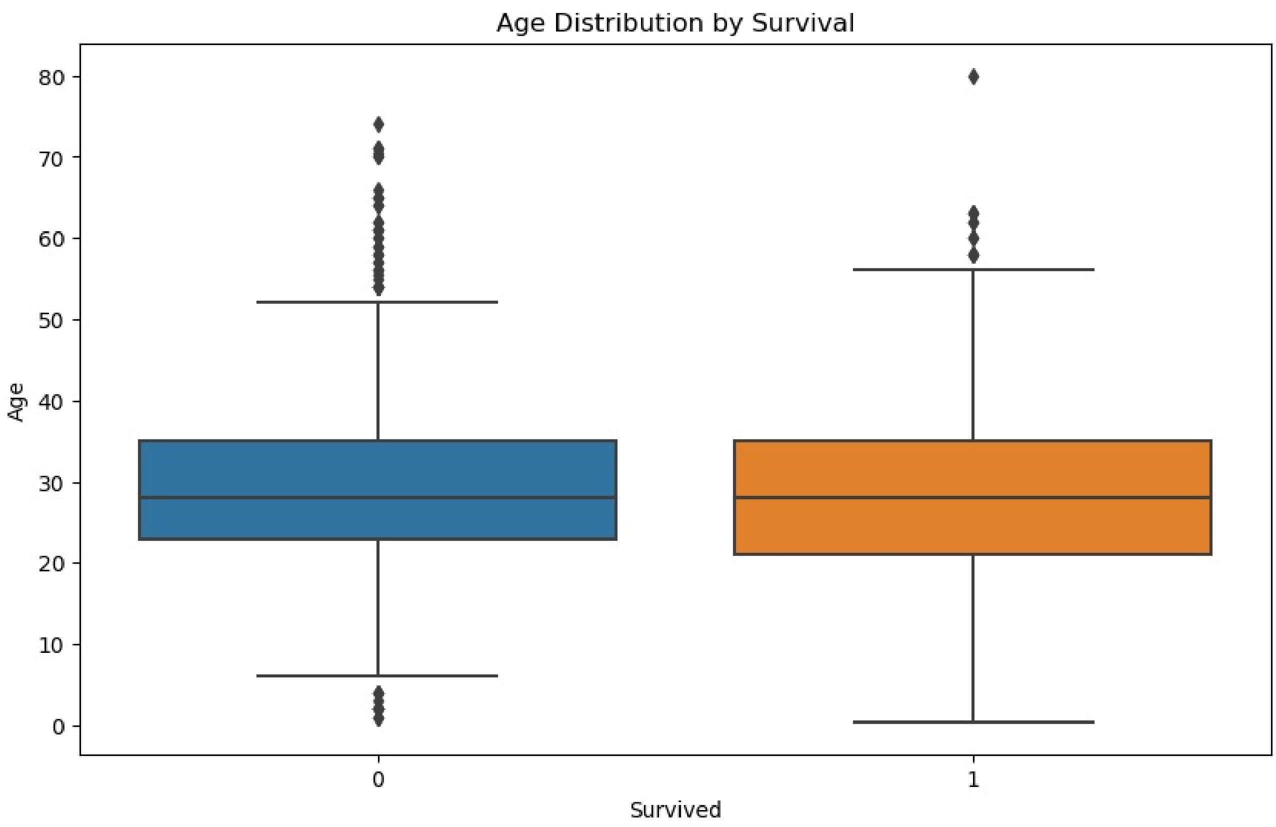


In [49]:

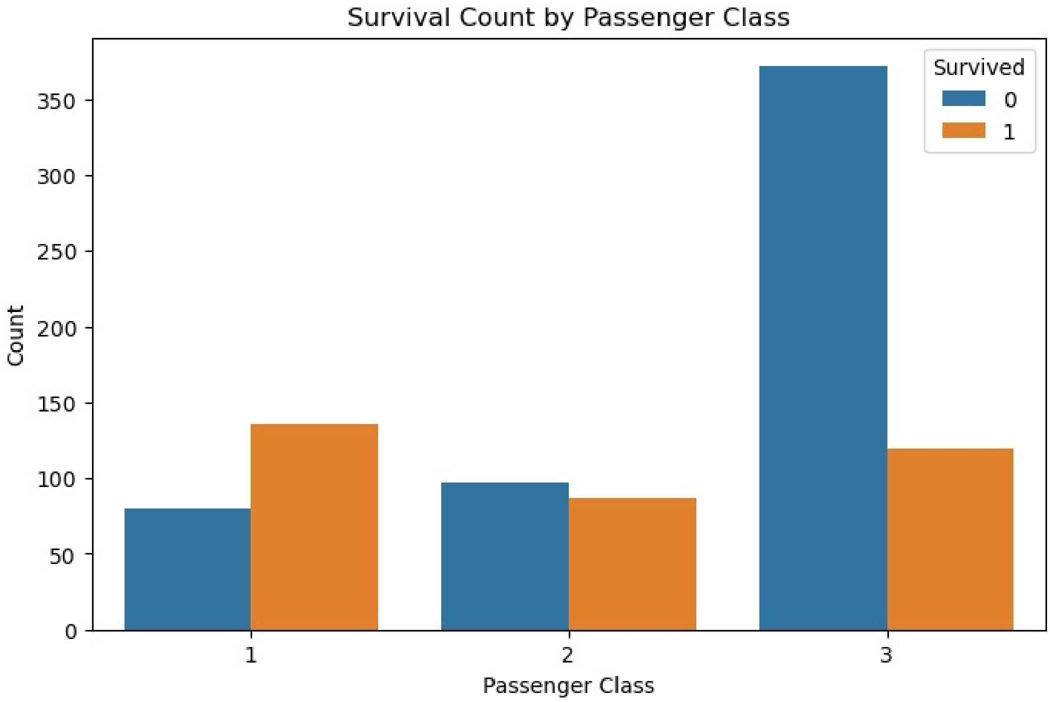
```
# Bivariate analysis: Survived vs. Age
plt.figure(figsize=(10, 6))
sns.histplot(data=df, x='Age', hue='Survived', kde=True, bins=30, element='step')
plt.title('Age Distribution by Survival')
plt.xlabel('Age')
plt.ylabel('Density')
plt.show()
```



In [50]:
```
# Boxplot: Age distribution by Survival
plt.figure(figsize=(10, 6))
sns.boxplot(x='Survived', y='Age', data=df)
plt.title('Age Distribution by Survival')
plt.xlabel('Survived')
plt.ylabel('Age')
plt.show()
```



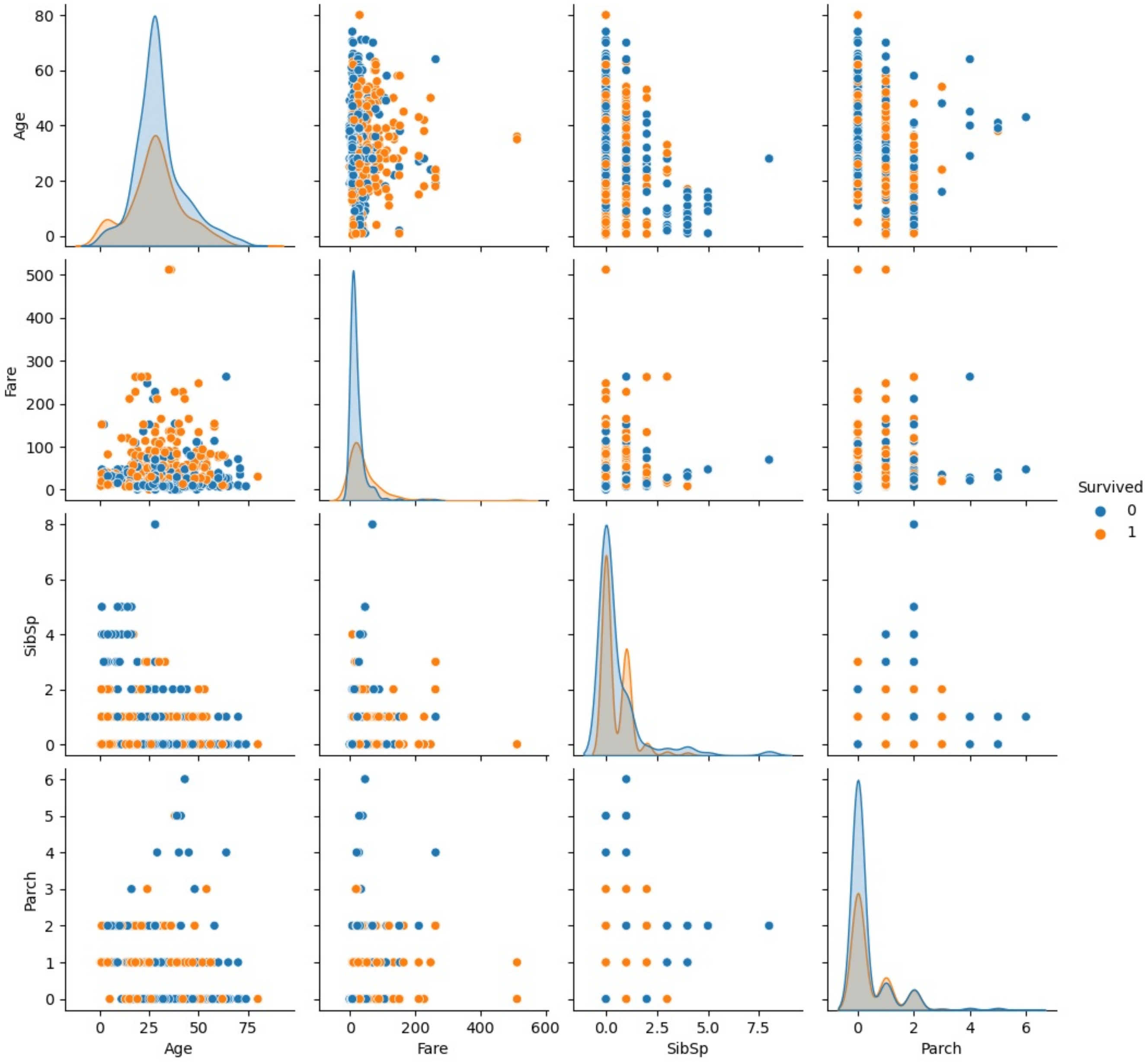
In [51]:
```
# Bivariate analysis: Survived vs. Pclass
plt.figure(figsize=(8, 5))
sns.countplot(x='Pclass', hue='Survived', data=df)
plt.title('Survival Count by Passenger Class')
plt.xlabel('Passenger Class')
plt.ylabel('Count')
plt.legend(title='Survived', loc='upper right')
plt.show()
```



In [52]:

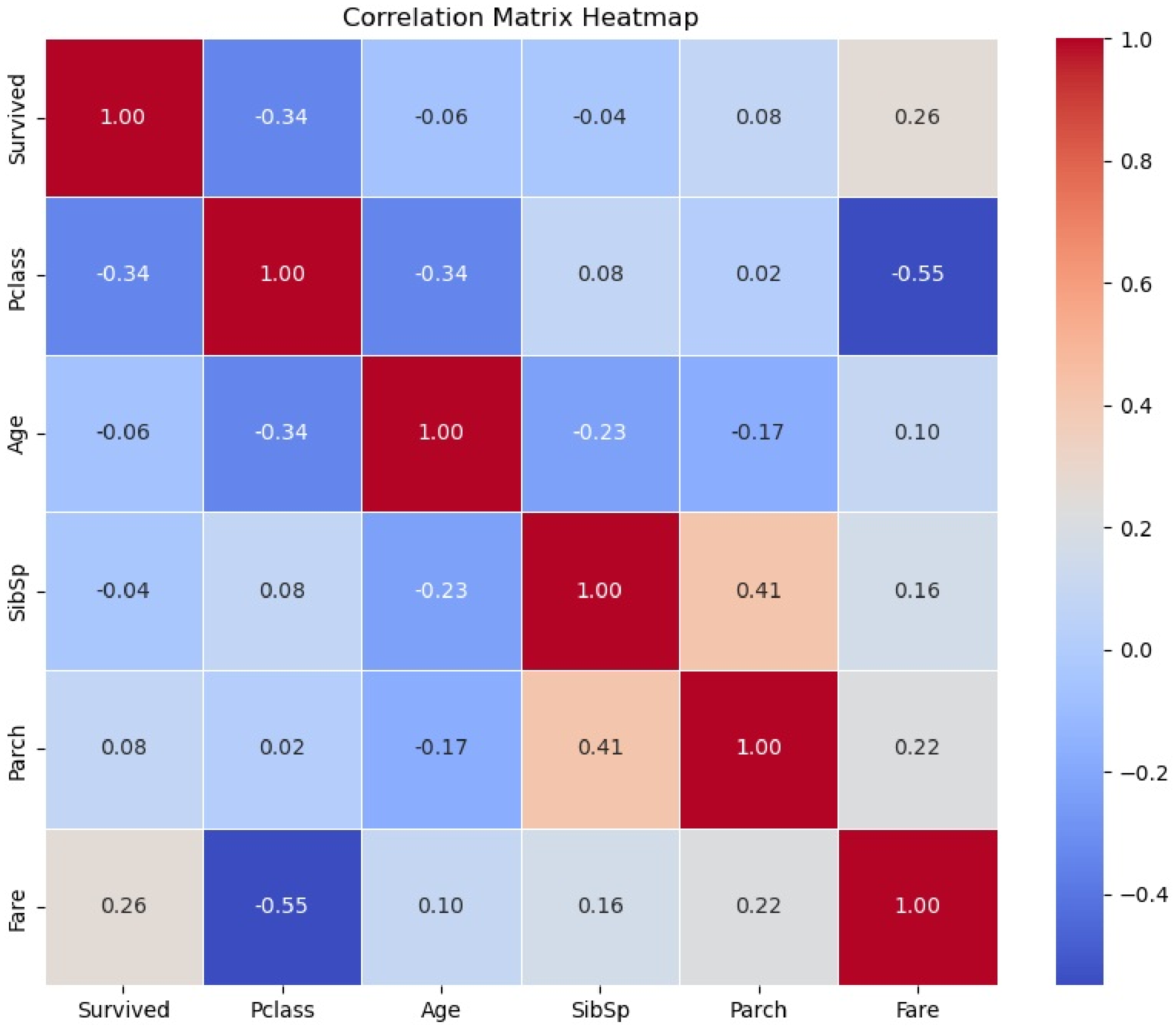
```
import seaborn as sns
import matplotlib.pyplot as plt

# Pair plot for selected features
#Pair Plot: This is useful for seeing all pairwise relationships and distributions in one plot.
sns.pairplot(df, hue='Survived', vars=['Age', 'Fare', 'SibSp', 'Parch'], diag_kind='kde')
plt.suptitle('Pair Plot of Features', y=1.02)
plt.show()
```

```
C:\Users\cse\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to
tight
  self._figure.tight_layout(*args, **kwargs)
```


Pair Plot of Features

In [53]:

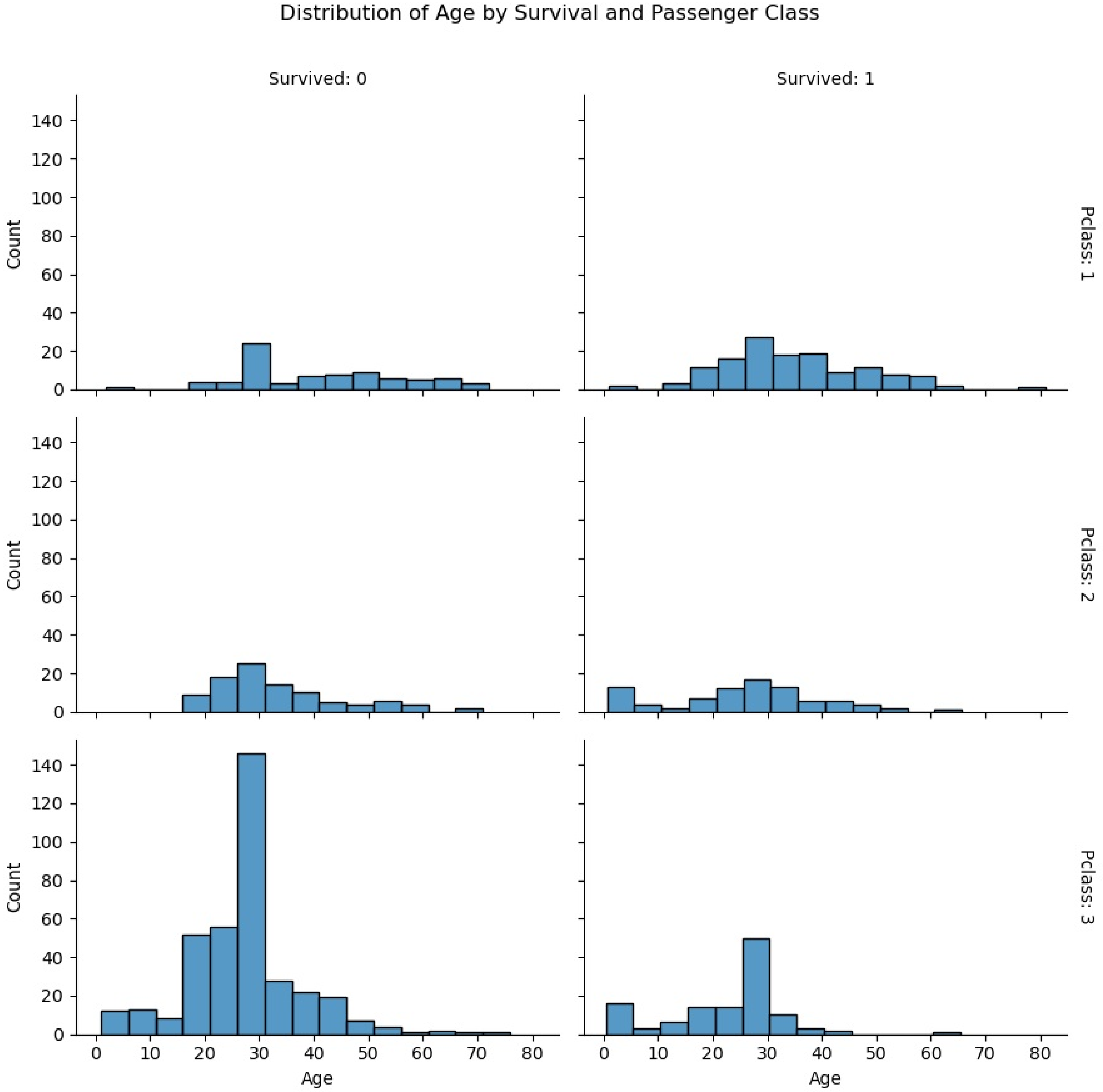
```
# Compute correlation matrix
corr = df[['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']].corr()

# Plot heatmap
#Correlation Heatmap: Provides a visual summary of relationships between numerical variables.
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Correlation Matrix Heatmap')
plt.show()
```



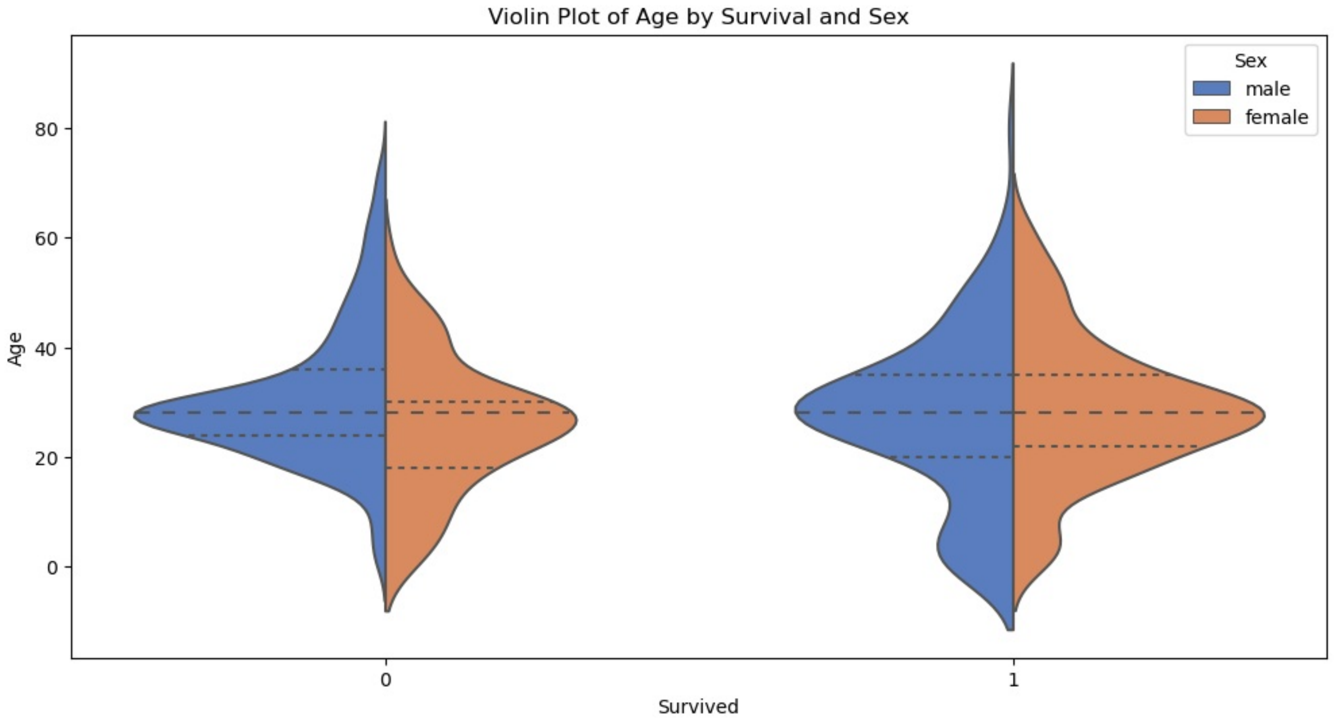
In [54]:

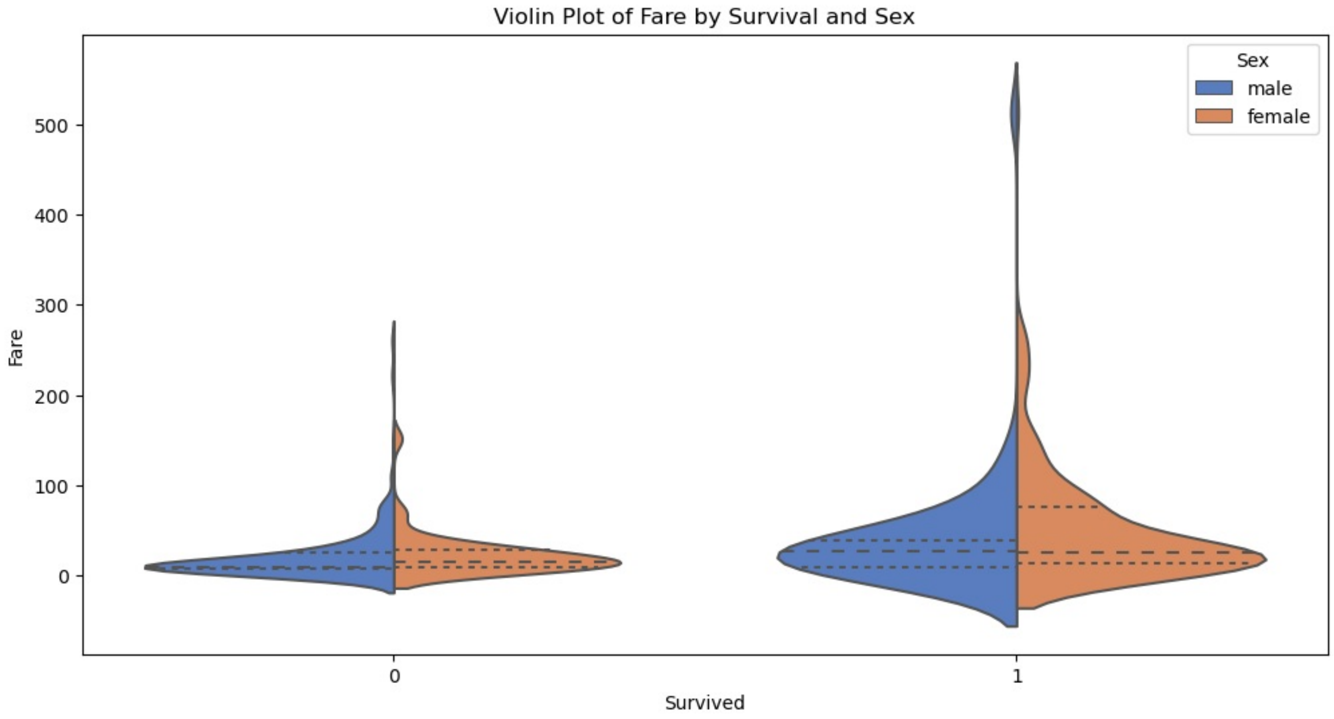
```
# Facet Grid for Survived vs Age and Pclass
# Facet Grid: Allows you to compare distributions across subsets of data, which is great for exploring interact
g = sns.FacetGrid(df, col='Survived', row='Pclass', margin_titles=True, height=3, aspect=1.5)
g.map_dataframe(sns.histplot, x='Age', binwidth=5)
g.set_axis_labels('Age', 'Count')
g.set_titles(col_template='Survived: {col_name}', row_template='Pclass: {row_name}')
g.fig.subplots_adjust(top=0.9)
g.fig.suptitle('Distribution of Age by Survival and Passenger Class')
plt.show()
```

```
C:\Users\cse\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed to
tight
  self._figure.tight_layout(*args, **kwargs)
```



In [55]:
```
# Violin plot for Age and Fare by Survived and Sex
# Violin Plot: Displays the distribution of data points, showing the probability density of the data at differe
plt.figure(figsize=(12, 6))
sns.violinplot(x='Survived', y='Age', hue='Sex', data=df, split=True, inner='quart', palette='muted')
plt.title('Violin Plot of Age by Survival and Sex')
plt.xlabel('Survived')
plt.ylabel('Age')
plt.show()

plt.figure(figsize=(12, 6))
sns.violinplot(x='Survived', y='Fare', hue='Sex', data=df, split=True, inner='quart', palette='muted')
plt.title('Violin Plot of Fare by Survival and Sex')
plt.xlabel('Survived')
plt.ylabel('Fare')
plt.show()
```





In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js