In [1]:

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

In [2]:

```
%matplotlib inline
```

In [3]:

```
train = pd.read_csv('C:/Users/LENOVO/Downloads/train.csv')
test = pd.read_csv('C:/Users/LENOVO/Downloads/test.csv')
gender_submission = pd.read_csv('C:/Users/LENOVO/Downloads/gender_submission.csv')
```

In [4]:

```
print("Train Data Info:")
print(train.info())
```

```
Train Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
None
```

In [5]:

```
print("\nTest Data Info:")
print(test.info())
```

```
Test Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Name         418 non-null    object 
 3   Sex          418 non-null    object 
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64  
 6   Parch        418 non-null    int64  
 7   Ticket       418 non-null    object 
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object 
 10  Embarked     418 non-null    object 
dtypes: float64(2), int64(4), object(5)
memory usage: 36.1+ KB
None
```

In [6]:

```
print("\nGender Submission Info:")
print(gender_submission.info())
```

```
Gender Submission Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 2 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Survived     418 non-null    int64
dtypes: int64(2)
memory usage: 6.7 KB
None
```

In [7]:

```
print("\nTrain Summary Statistics:")
print(train.describe(include='all'))
```

```
Train Summary Statistics:
        PassengerId    Survived      Pclass                     Name   Sex  \
count    891.000000  891.000000  891.000000                      891   891   
unique          NaN         NaN         NaN                      891     2   
top             NaN         NaN         NaN  Braund, Mr. Owen Harris  male   
freq            NaN         NaN         NaN                        1   577   
mean     446.000000    0.383838    2.308642                      NaN   NaN   
std      257.353842    0.486592    0.836071                      NaN   NaN   
min        1.000000    0.000000    1.000000                      NaN   NaN   
25%      223.500000    0.000000    2.000000                      NaN   NaN   
50%      446.000000    0.000000    3.000000                      NaN   NaN   
75%      668.500000    1.000000    3.000000                      NaN   NaN   
max      891.000000    1.000000    3.000000                      NaN   NaN   

               Age       SibSp       Parch  Ticket        Fare    Cabin  \
count   714.000000  891.000000  891.000000     891  891.000000      204   
unique         NaN         NaN         NaN     681         NaN      147   
top            NaN         NaN         NaN  347082         NaN  B96 B98   
freq           NaN         NaN         NaN       7         NaN        4   
mean     29.699118    0.523008    0.381594     NaN   32.204208      NaN   
std      14.526497    1.102743    0.806057     NaN   49.693429      NaN   
min       0.420000    0.000000    0.000000     NaN    0.000000      NaN   
25%      20.125000    0.000000    0.000000     NaN    7.910400      NaN   
50%      28.000000    0.000000    0.000000     NaN   14.454200      NaN   
75%      38.000000    1.000000    0.000000     NaN   31.000000      NaN   
max      80.000000    8.000000    6.000000     NaN  512.329200      NaN   

       Embarked  
count       889  
unique        3  
top           S  
freq        644  
mean        NaN  
std         NaN  
min         NaN  
25%         NaN  
50%         NaN  
75%         NaN  
max         NaN  
```

In [8]:

```
print("\nMissing Values in Train Data:")
print(train.isnull().sum())
```

```
Missing Values in Train Data:
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
```

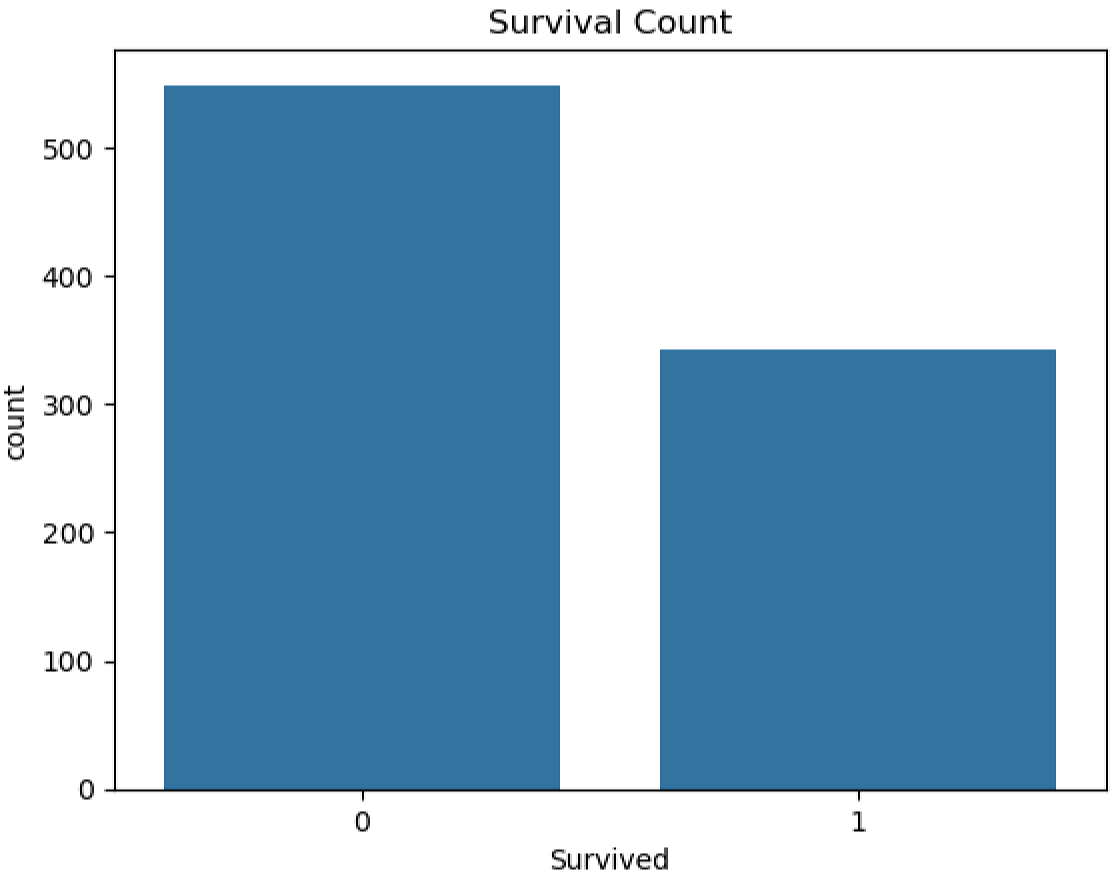
In [9]:

```
print("\nMissing Values in Test Data:")
print(test.isnull().sum())
```

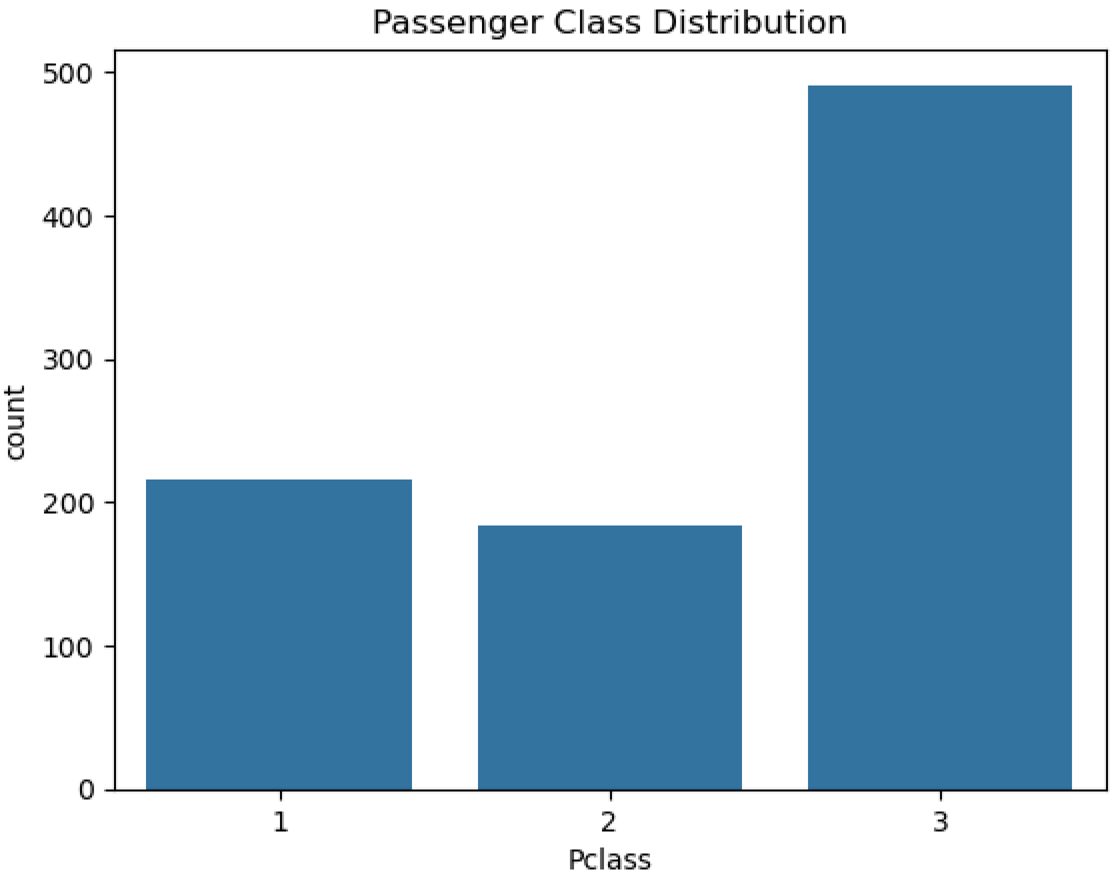
```
Missing Values in Test Data:
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [10]:

```
sns.countplot(x='Survived', data=train)
plt.title('Survival Count')
plt.show()
```

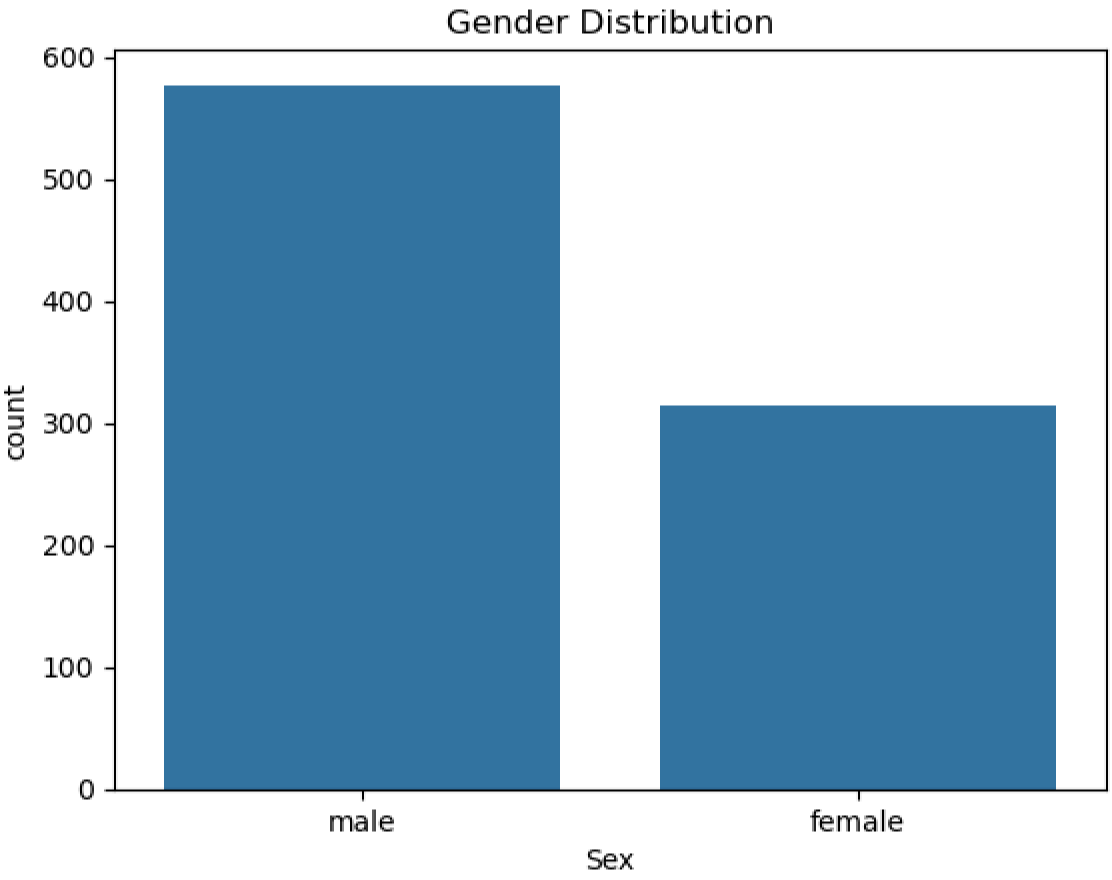


In [11]:
```
sns.countplot(x='Pclass', data=train)
plt.title('Passenger Class Distribution')
plt.show()
```



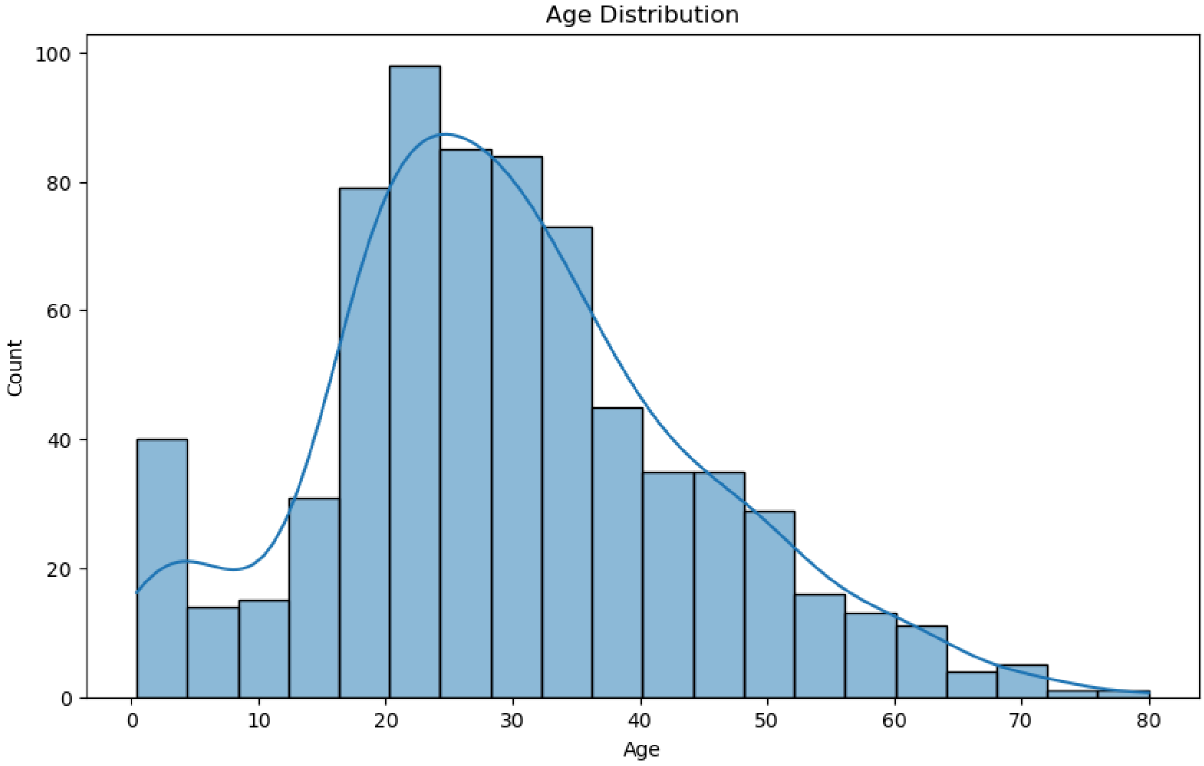
In [12]:

```
sns.countplot(x='Sex', data=train)
plt.title('Gender Distribution')
plt.show()
```



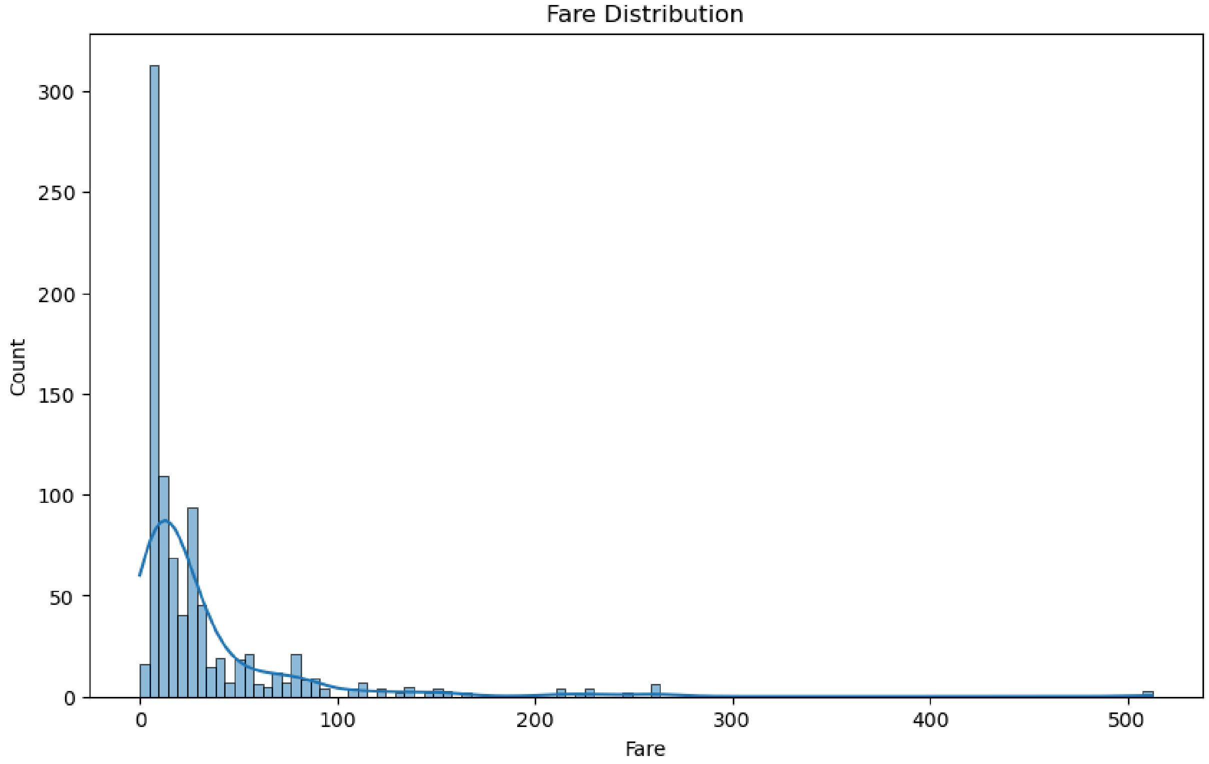
In [13]:

```
plt.figure(figsize=(10,6))
sns.histplot(train['Age'].dropna(), kde=True)
plt.title('Age Distribution')
plt.show()
```



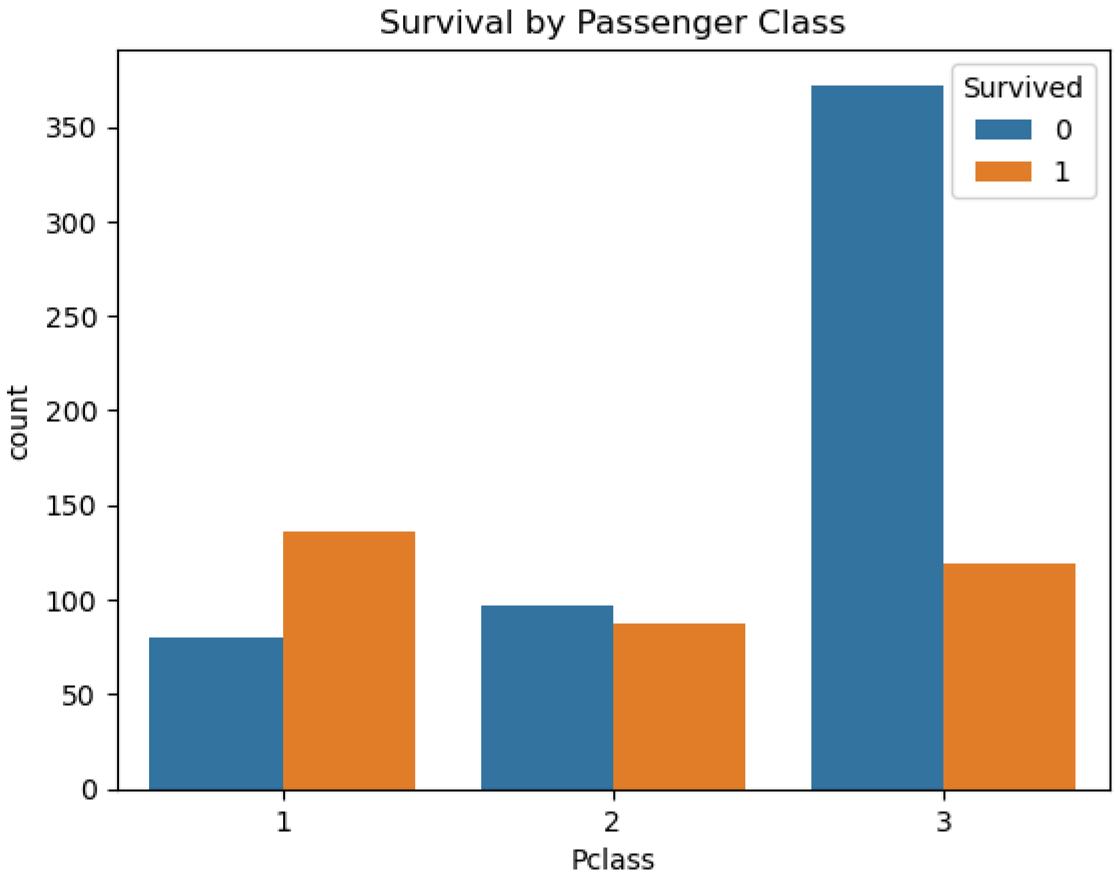
In [14]:

```
plt.figure(figsize=(10,6))
sns.histplot(train['Fare'], kde=True)
plt.title('Fare Distribution')
plt.show()
```



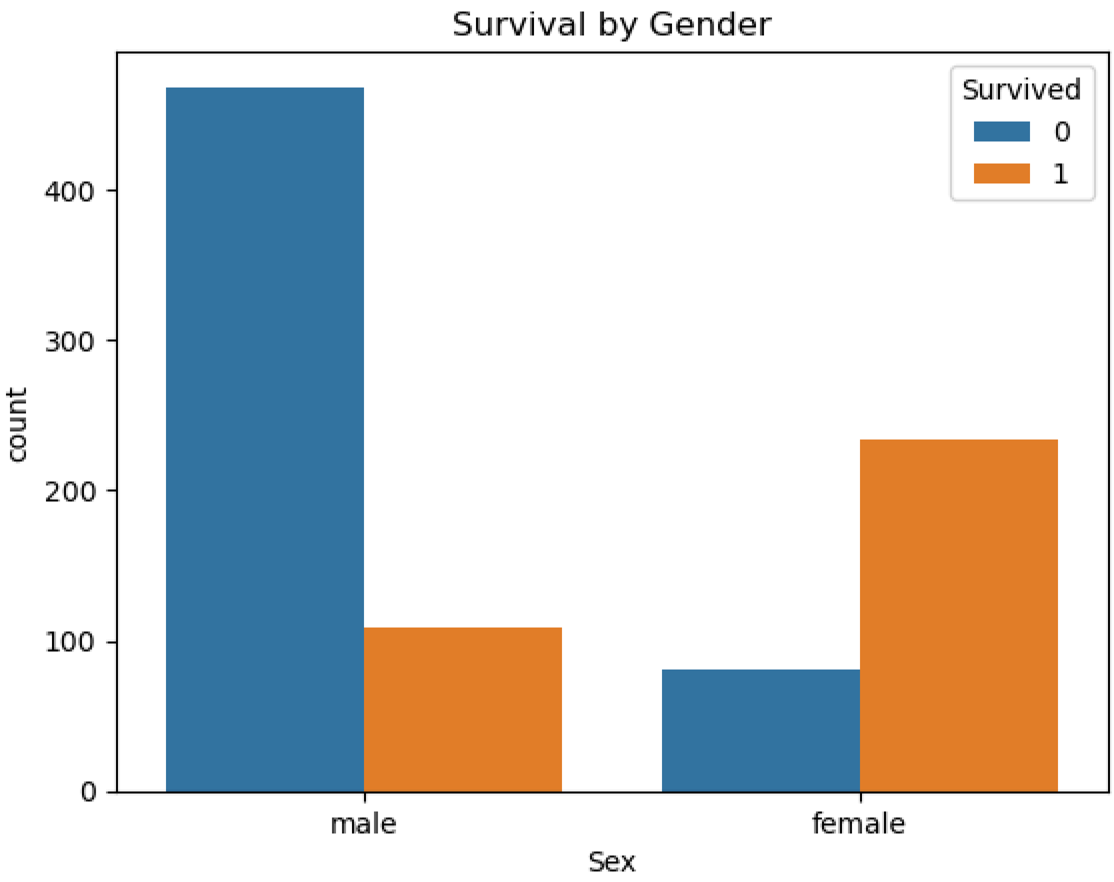
In [15]:

```
sns.countplot(x='Pclass', hue='Survived', data=train)
plt.title('Survival by Passenger Class')
plt.show()
```

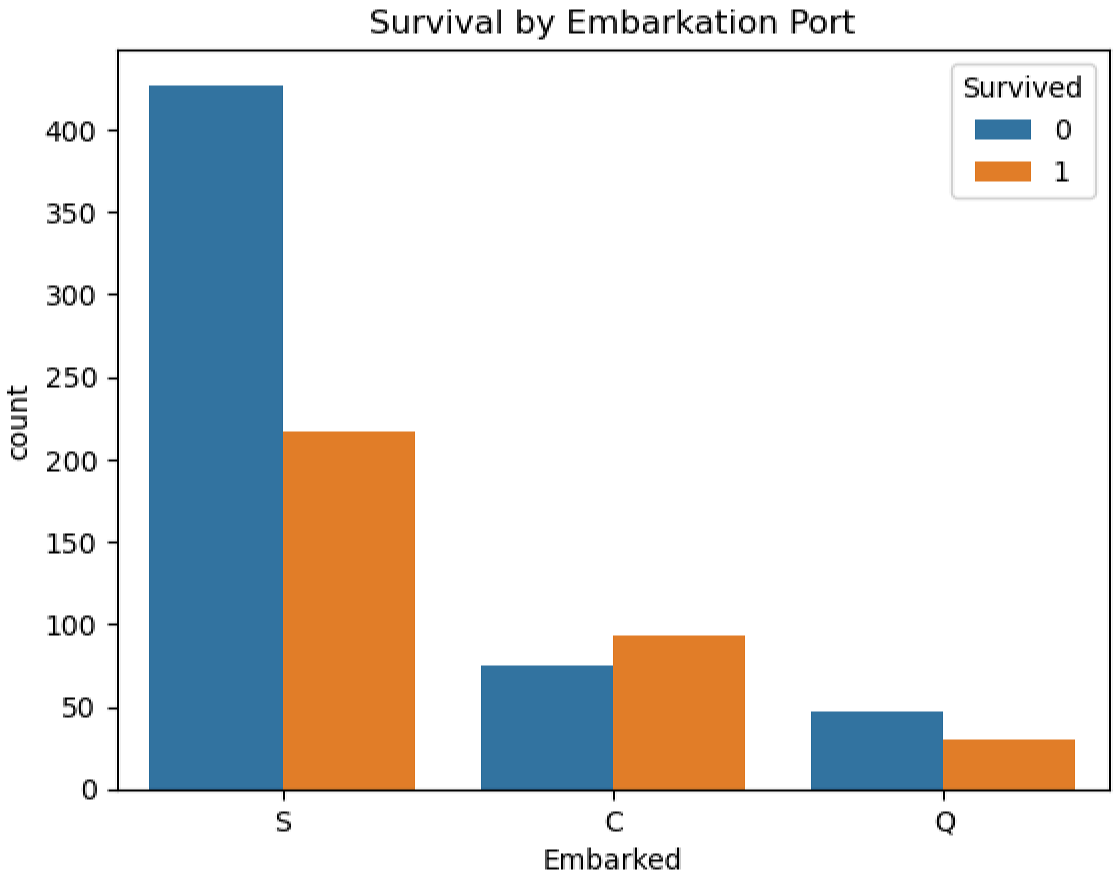


In [16]:

```
sns.countplot(x='Sex', hue='Survived', data=train)
plt.title('Survival by Gender')
plt.show()
```

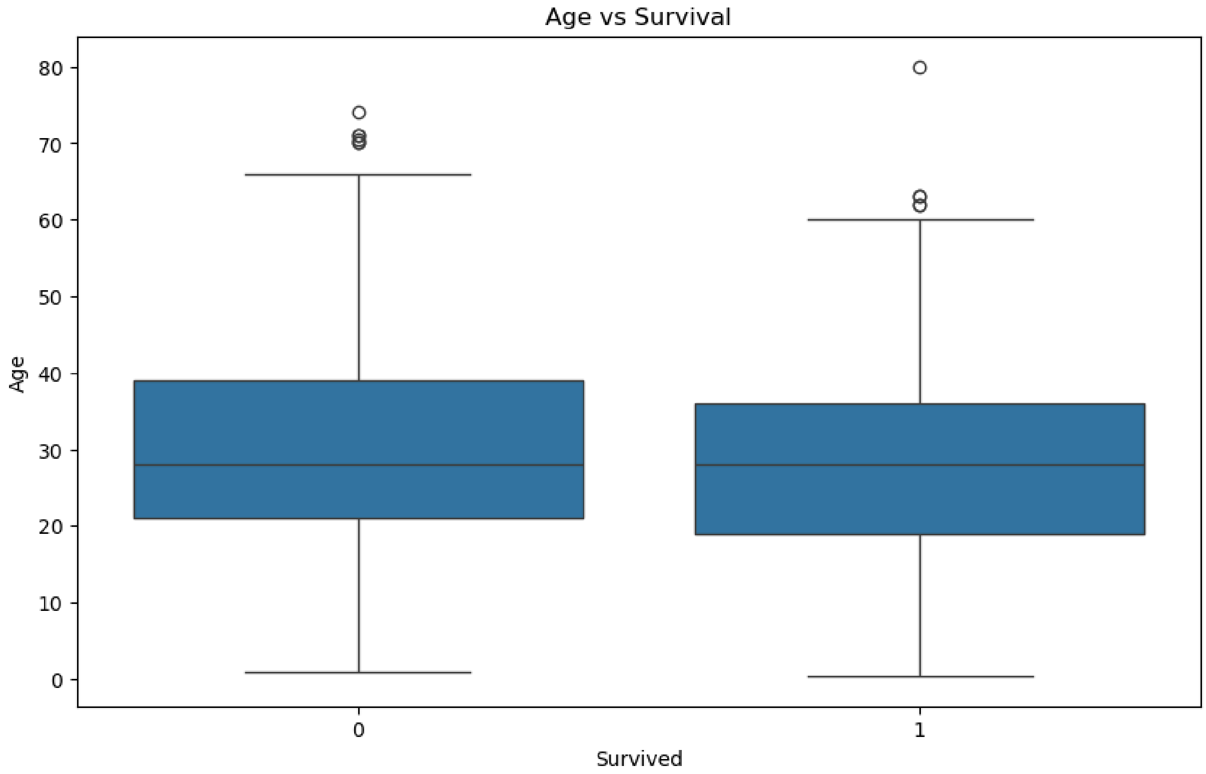


```
In [17]: sns.countplot(x='Embarked', hue='Survived', data=train)
         plt.title('Survival by Embarkation Port')
         plt.show()
```



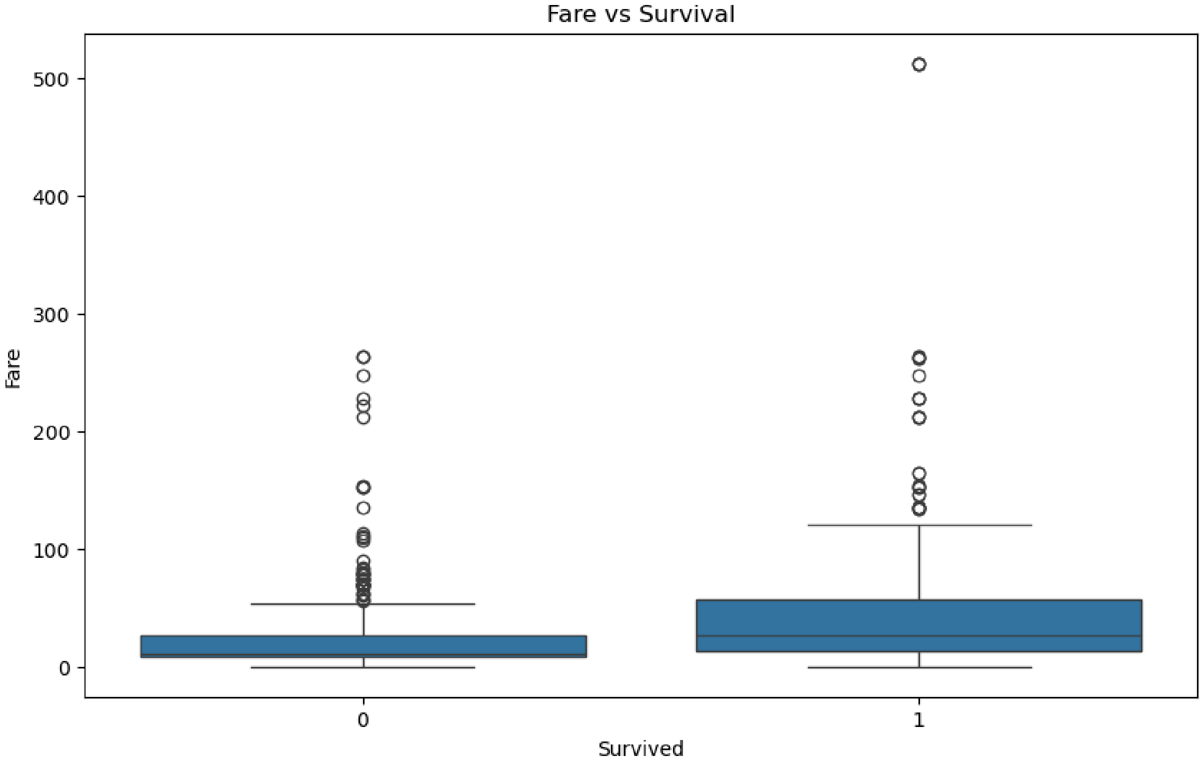
In [18]:

```python
plt.figure(figsize=(10,6))
sns.boxplot(x='Survived', y='Age', data=train)
plt.title('Age vs Survival')
plt.show()
```



In [19]:

```
plt.figure(figsize=(10,6))
sns.boxplot(x='Survived', y='Fare', data=train)
plt.title('Fare vs Survival')
plt.show()
```



In [20]:

```
train['Age'].fillna(train['Age'].median(), inplace=True)
test['Age'].fillna(test['Age'].median(), inplace=True)
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_15408\3278935906.py:1: FutureWarning: A
value is trying to be set on a copy of a DataFrame or Series through chained assignm
ent using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because
the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method
({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform
the operation inplace on the original object.


  train['Age'].fillna(train['Age'].median(), inplace=True)
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_15408\3278935906.py:2: FutureWarning: A
value is trying to be set on a copy of a DataFrame or Series through chained assignm
ent using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because
the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method
({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform
the operation inplace on the original object.


  test['Age'].fillna(test['Age'].median(), inplace=True)
```

In [21]:
```
train['Embarked'].fillna(train['Embarked'].mode()[0], inplace=True)
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_15408\1031565505.py:1: FutureWarning: A
value is trying to be set on a copy of a DataFrame or Series through chained assignm
ent using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because
the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method
({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform
the operation inplace on the original object.


  train['Embarked'].fillna(train['Embarked'].mode()[0], inplace=True)
```

In [22]:
```
train.drop('Cabin', axis=1, inplace=True)
test.drop('Cabin', axis=1, inplace=True)
```

In [23]:
```
test_with_predictions = pd.merge(test, gender_submission, on='PassengerId')
print("\nTest Data with Predictions:")
print(test_with_predictions.head())
```

```
Test Data with Predictions:
   PassengerId  Pclass                                          Name     Sex  \
0          892       3                              Kelly, Mr. James    male   
1          893       3              Wilkes, Mrs. James (Ellen Needs)  female   
2          894       2                     Myles, Mr. Thomas Francis    male   
3          895       3                              Wirz, Mr. Albert    male   
4          896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female   

    Age  SibSp  Parch   Ticket     Fare Embarked  Survived  
0  34.5      0      0   330911   7.8292        Q         0  
1  47.0      1      0   363272   7.0000        S         1  
2  62.0      0      0   240276   9.6875        Q         0  
3  27.0      0      0   315154   8.6625        S         0  
4  22.0      1      1  3101298  12.2875        S         1  
```

In [24]:

```
summary = """
Summary of Titanic Dataset EDA:

- Total Passengers in Training Set: {}
- Survival Rate: {:.2f}%
- Higher survival rates observed in:
    - Females compared to males.
    - Passengers in 1st class.
    - Passengers who embarked from Cherbourg.
- Age distribution is right-skewed, most passengers are between 20-40 years.
- Higher fare seems to correlate with better survival chances.
""".format(len(train), train['Survived'].mean() * 100)

print(summary)
```

```
Summary of Titanic Dataset EDA:

- Total Passengers in Training Set: 891
- Survival Rate: 38.38%
- Higher survival rates observed in:
    - Females compared to males.
    - Passengers in 1st class.
    - Passengers who embarked from Cherbourg.
- Age distribution is right-skewed, most passengers are between 20-40 years.
- Higher fare seems to correlate with better survival chances.
```

In [25]:

```
with open('Titanic_EDA_Summary.txt', 'w') as f:
    f.write(summary)
```