In [1]:
```
import pandas as pd
# Load the Titanic dataset
df = pd.read_csv(r"C:\Users\DANIEL\Downloads\train.csv")
df.head()
```

Out[1]:

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

In [13]:
```
# Check for missing values
print(df.isnull().sum())

# Fill missing values or drop rows/columns with missing values
df['Age'].fillna(df['Age'].median(), inplace=True)
df.dropna(subset=['Embarked'], inplace=True)

# Correct data types if necessary
df['Survived'] = df['Survived'].astype('category')
```

```
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
```

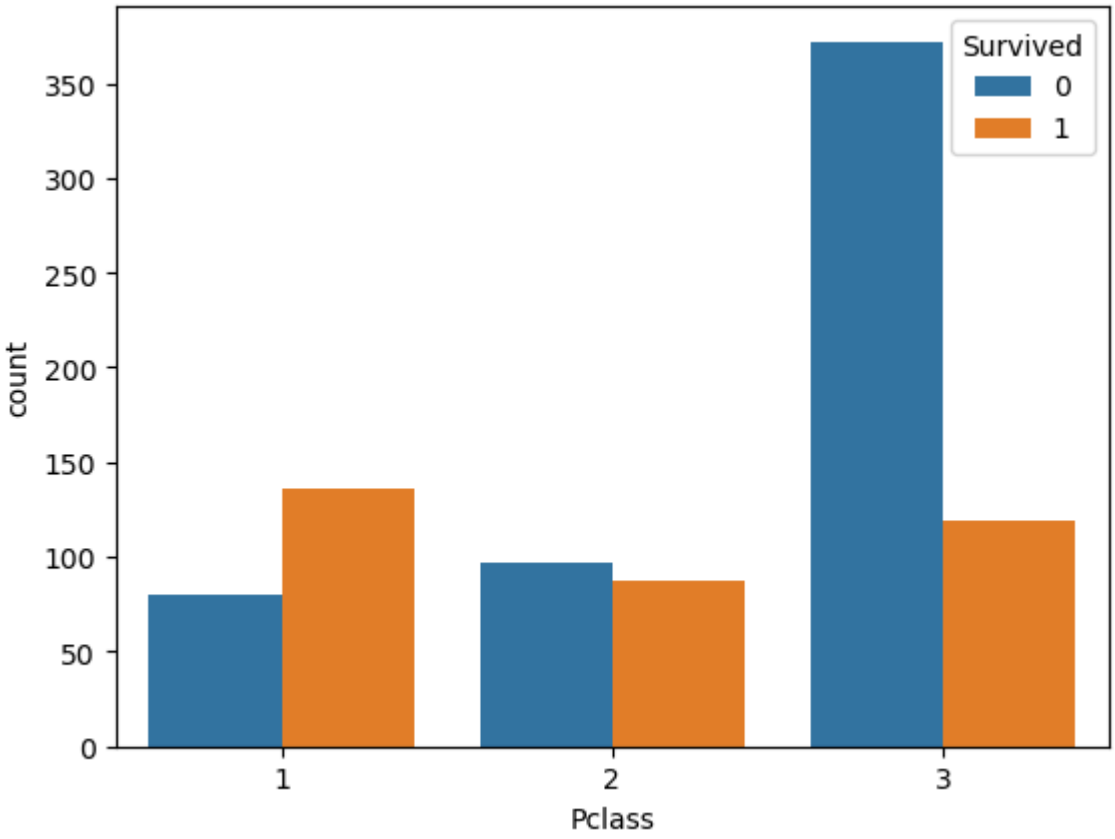
In [2]:
```
df.describe(include='all')
```

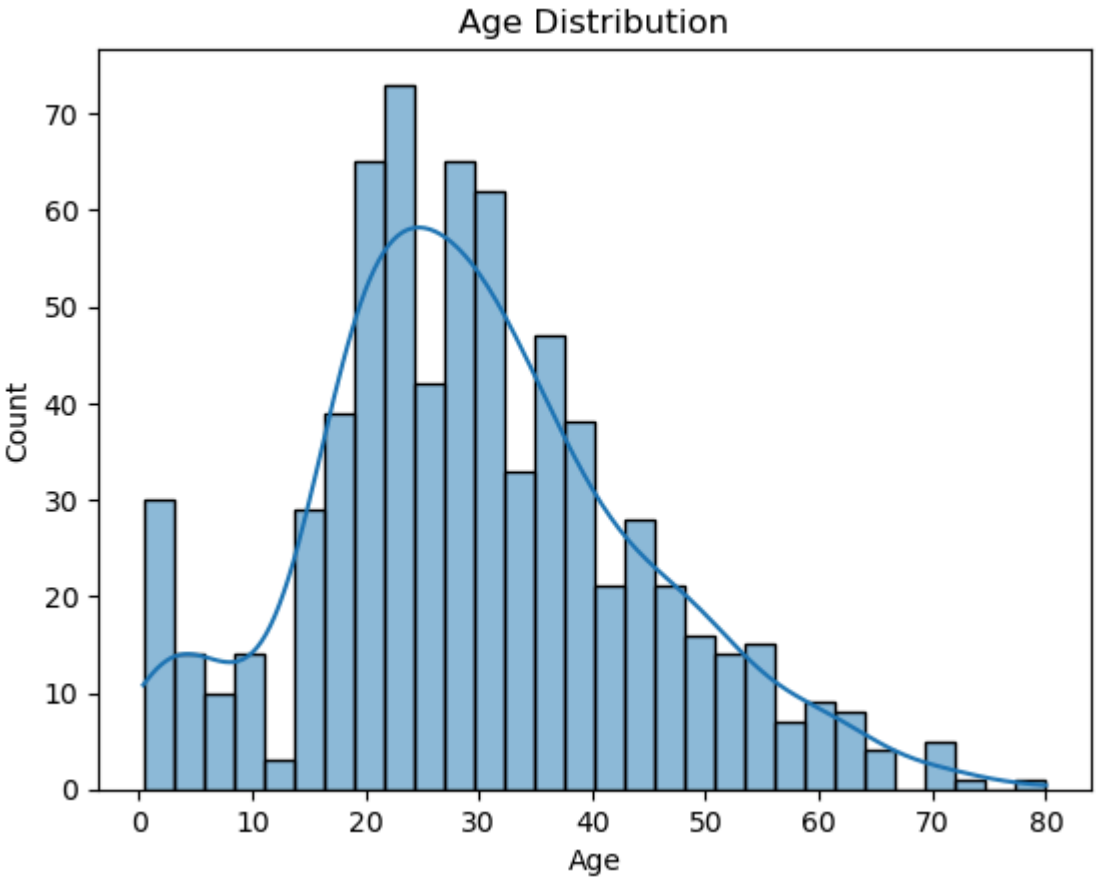
Out[2]:

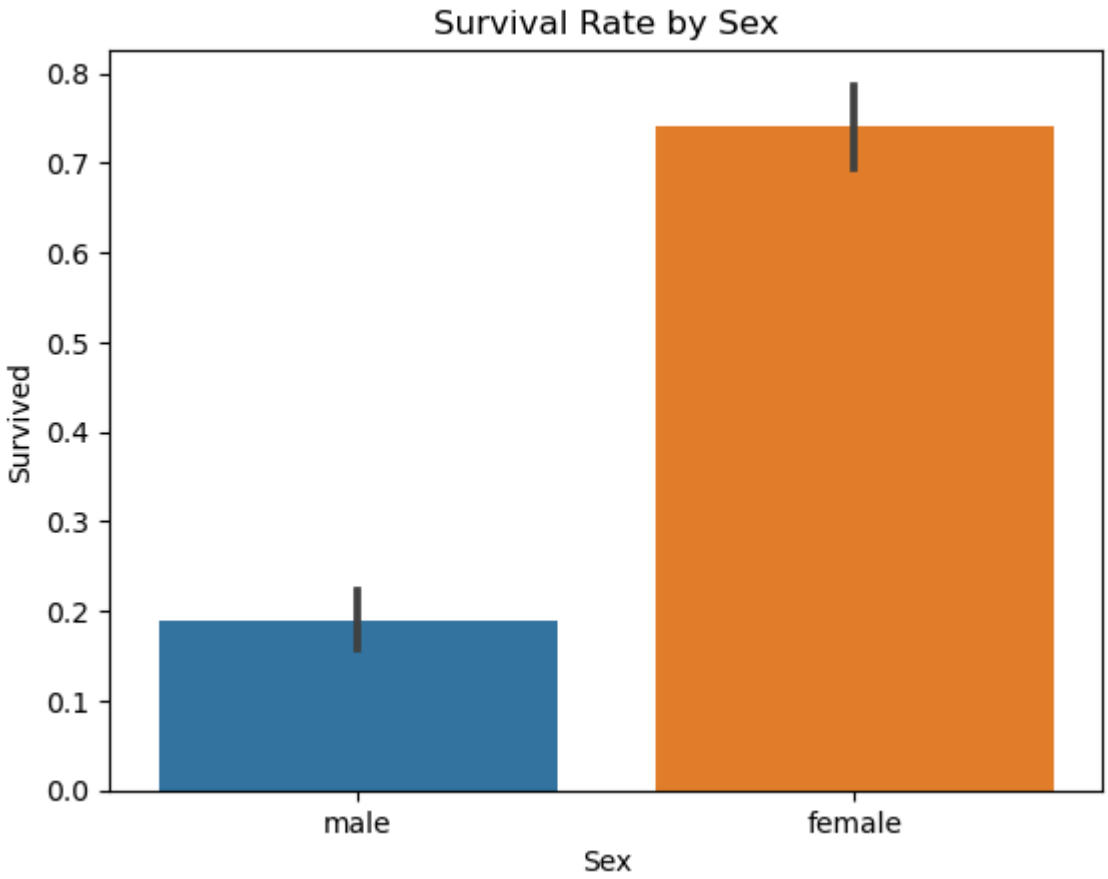
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Pa
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000(
unique	NaN	NaN	NaN	891	2	NaN	NaN	N
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	NaN	NaN	N
freq	NaN	NaN	NaN	1	577	NaN	NaN	N
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.3815
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806(
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000(
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000(
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000(
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000(
max	891.000000	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.000(

In [5]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Statistical summary
print(df.describe())

# Relationship between survival and passenger class
sns.countplot(x='Pclass', hue='Survived', data=df)
plt.show()

# Age distribution of passengers
sns.histplot(df['Age'], bins=30, kde=True)
plt.title('Age Distribution')
plt.show()

# Survival rate by gender
sns.barplot(x='Sex', y='Survived', data=df)
plt.title('Survival Rate by Sex')
plt.show()
```

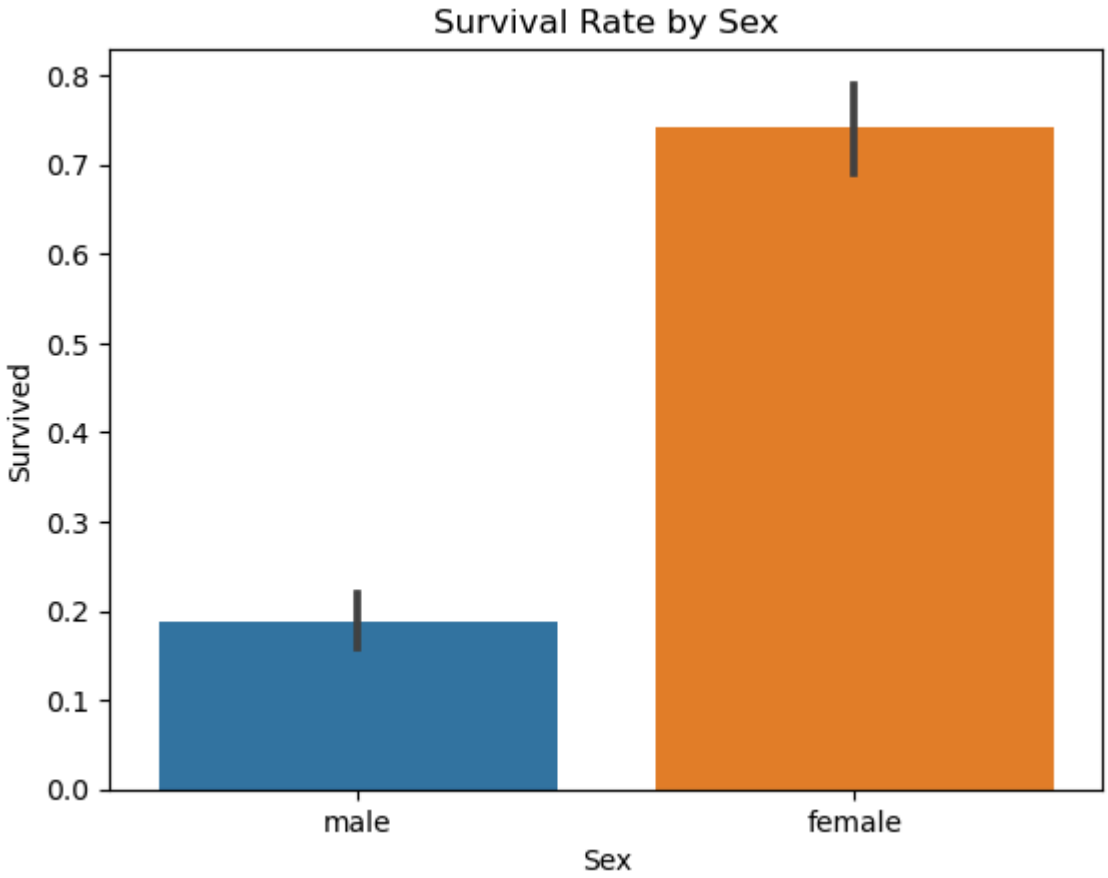
```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    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
```

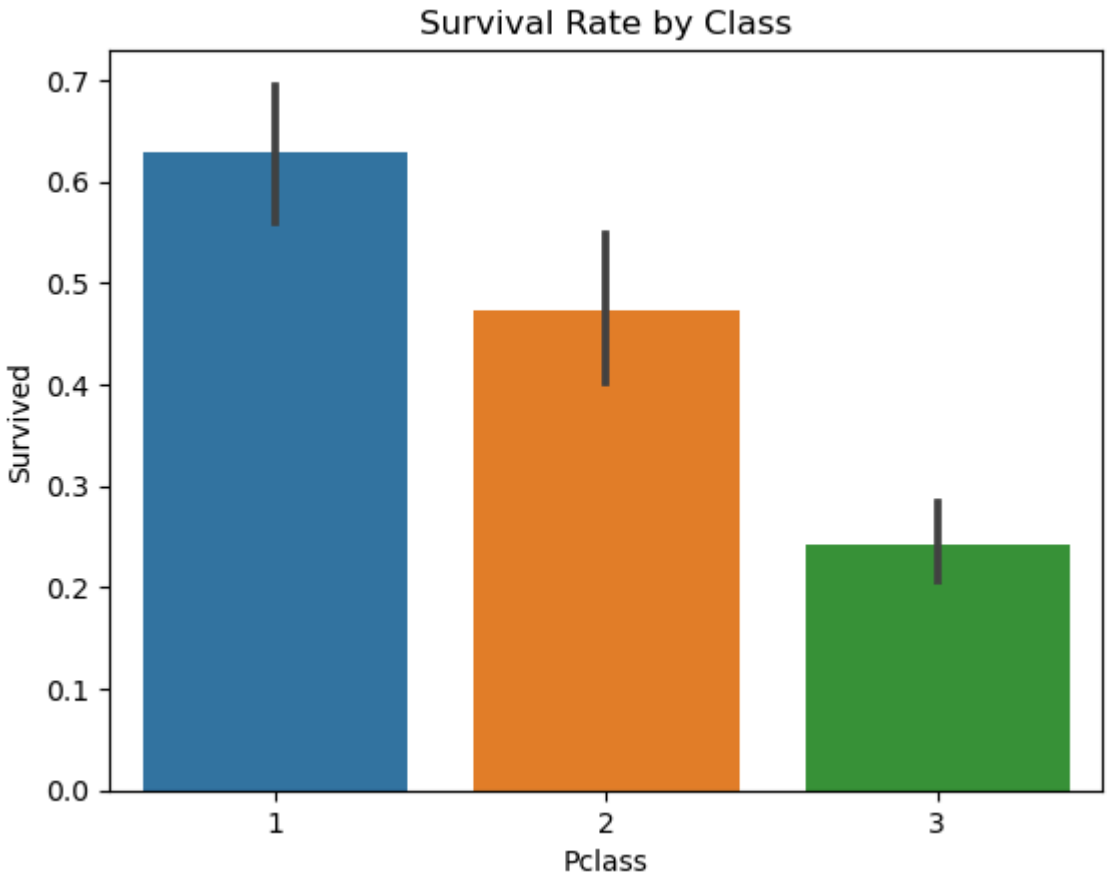

Survived
0
1
350
300
250
200
150
100
50
0
count
1
2
3
Pclass

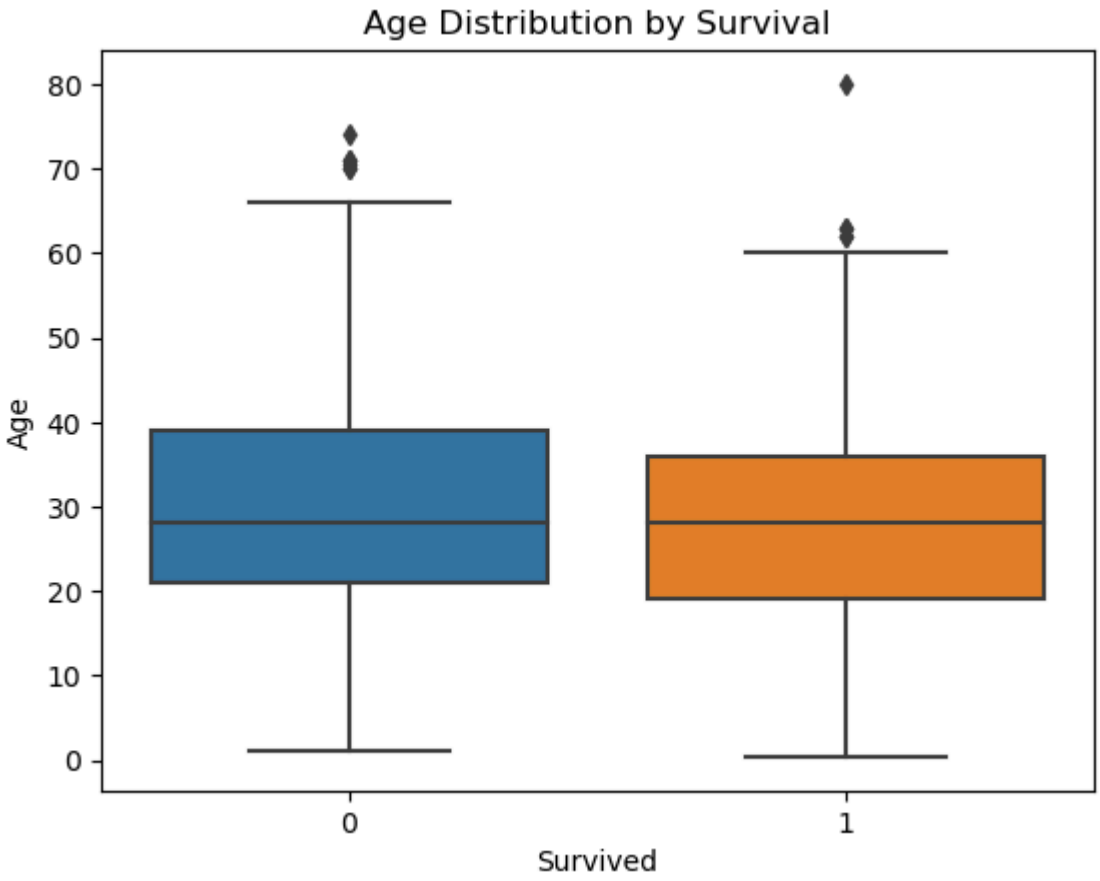

Age Distribution
70
60
50
40
30
20
10
0
Count
0
10
20
30
40
50
60
70
80
Age


Survival Rate by Sex
Survived
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
male
female
Sex

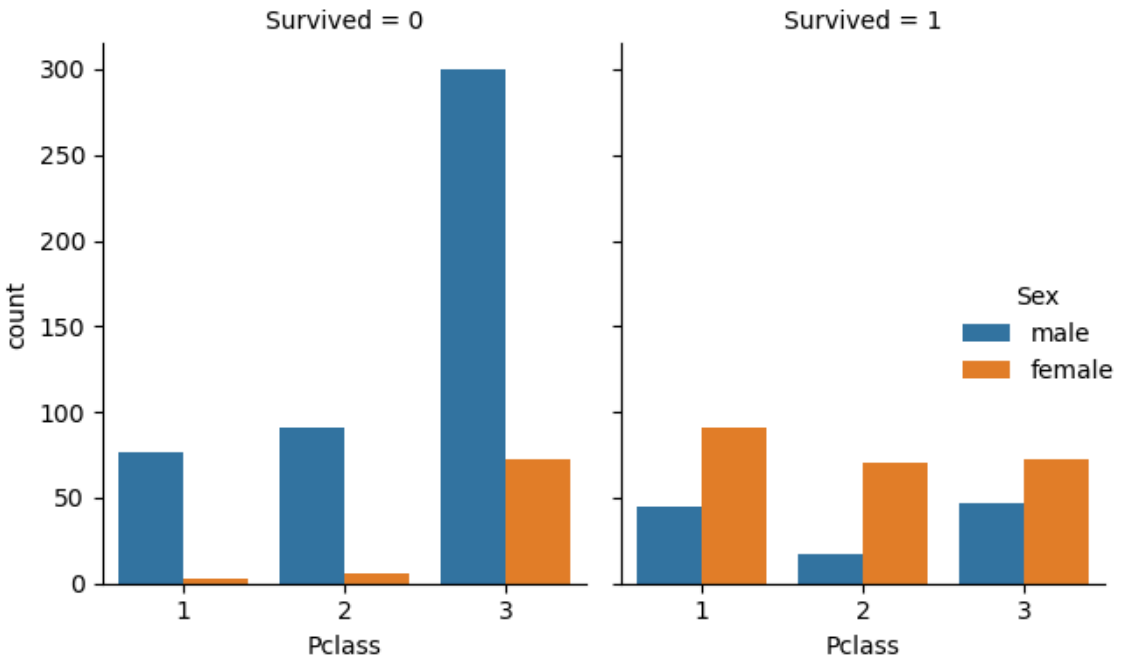
In [12]:
```
# Survival rate by class
sns.barplot(x='Pclass', y='Survived', data=df)
plt.title('Survival Rate by Class')
plt.show()
# Age distribution by survival
sns.boxplot(x='Survived', y='Age', data=df)
plt.title('Age Distribution by Survival')
plt.show()
```



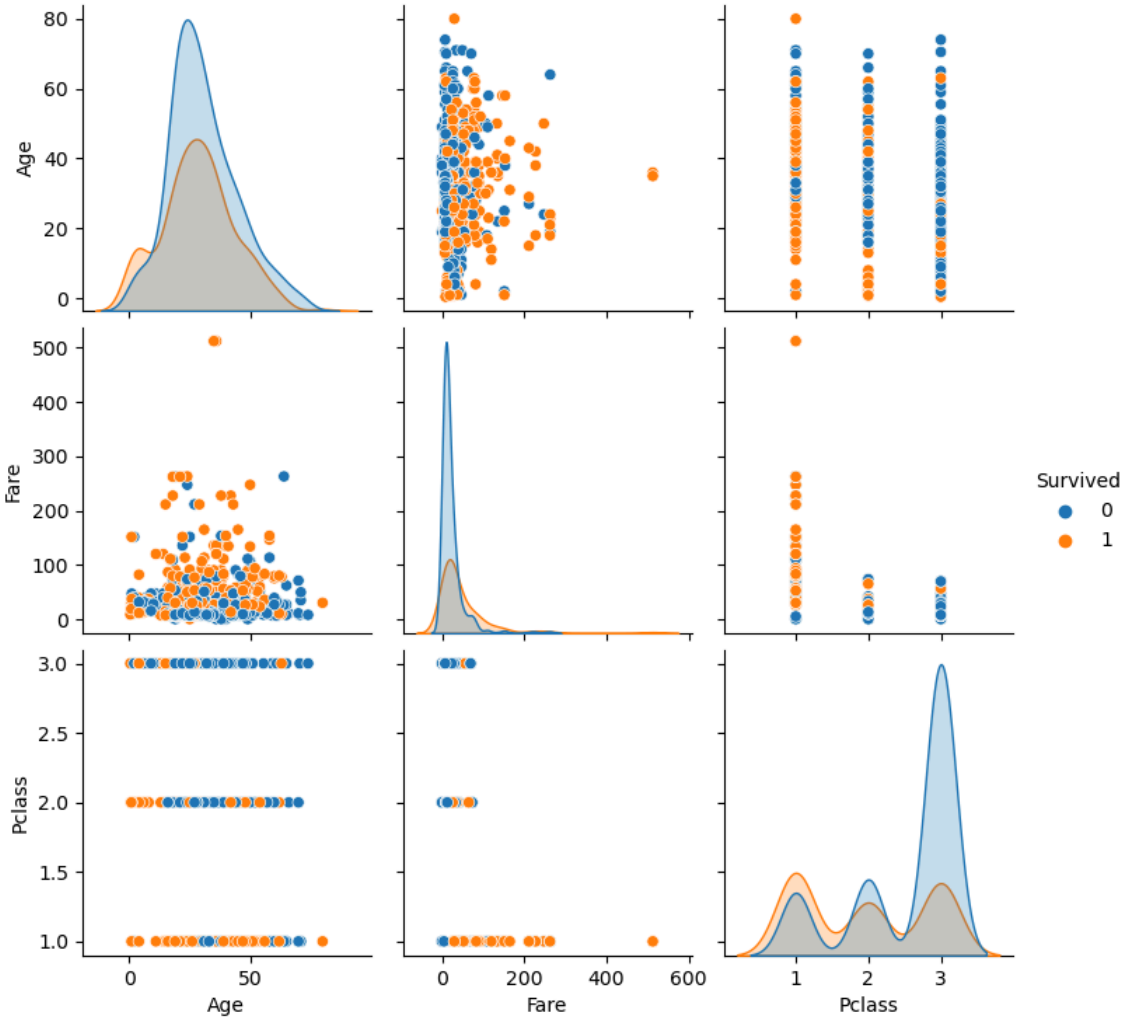

Survival Rate by Class
Survived
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
1
2
3
Pclass


Age Distribution by Survival
Age
0
10
20
30
40
50
60
70
80
0
1
Survived

In [9]:
```
sns.catplot(x='Pclass', hue='Sex', col='Survived', data=df, kind='count', he
plt.tight_layout()
plt.show()
```



In [8]:
```
# Pairplot of numerical features colored by survival
sns.pairplot(df, hue='Survived', vars=['Age', 'Fare', 'Pclass'])
plt.show()
```



In [ ]: