In [1]:

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

df = pd.read_csv("train (1).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 [2]:

```
df.shape
df.info()
df.describe()
df.isnull().sum()
df.dtypes
```

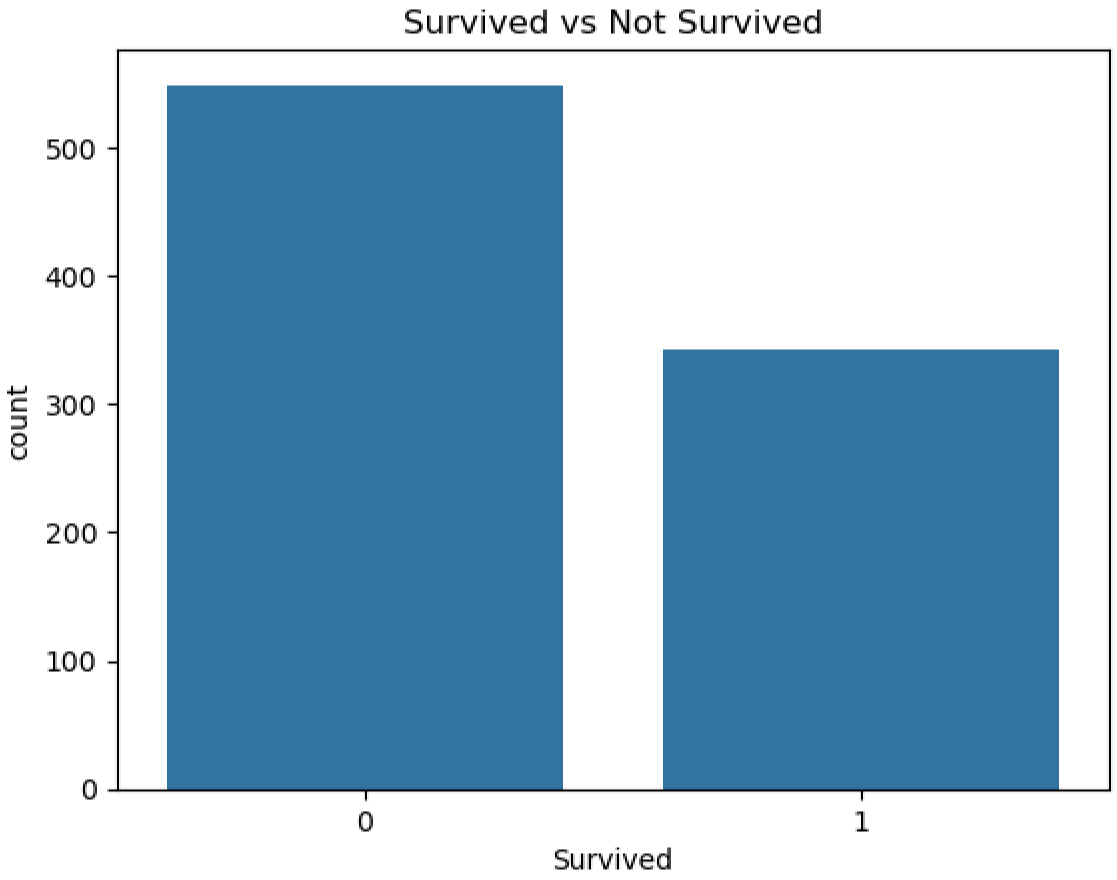
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    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
```

Out[2]:

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

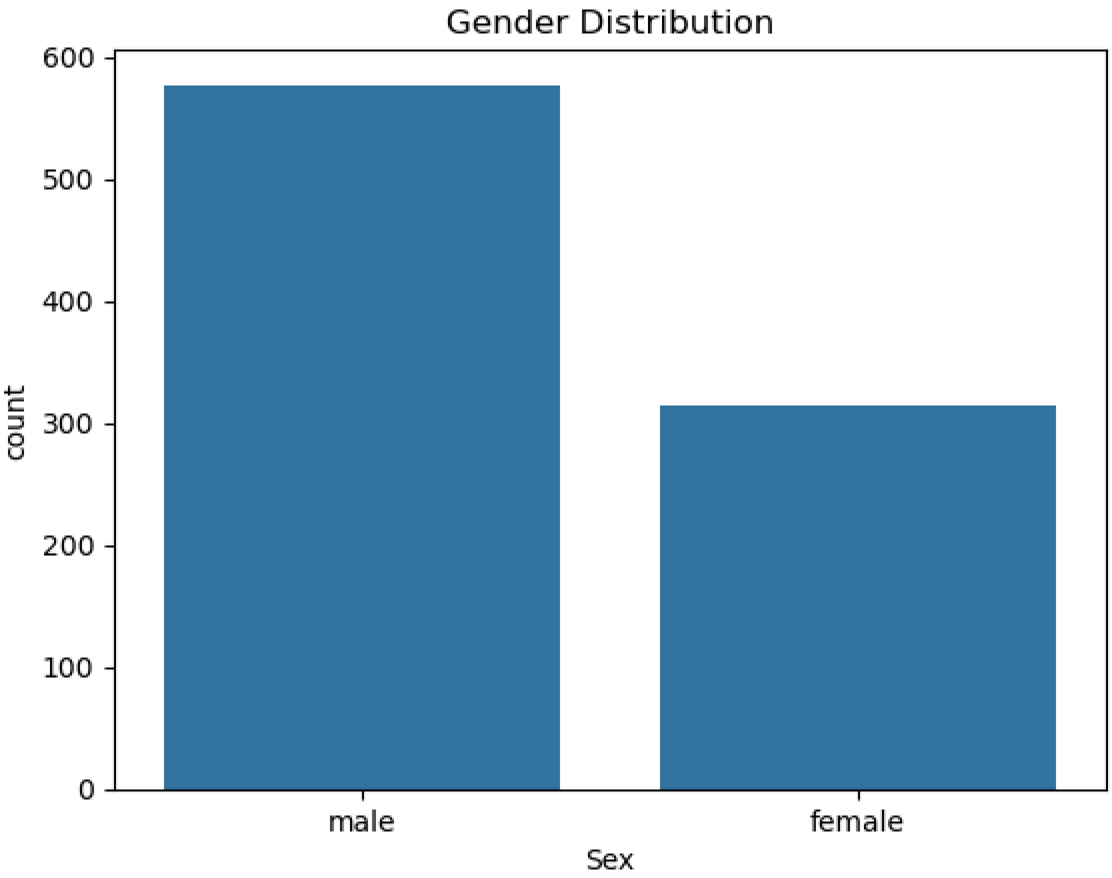
In [3]:

```
sns.countplot(x='Survived', data=df)
plt.title("Survived vs Not Survived")
plt.show()
```

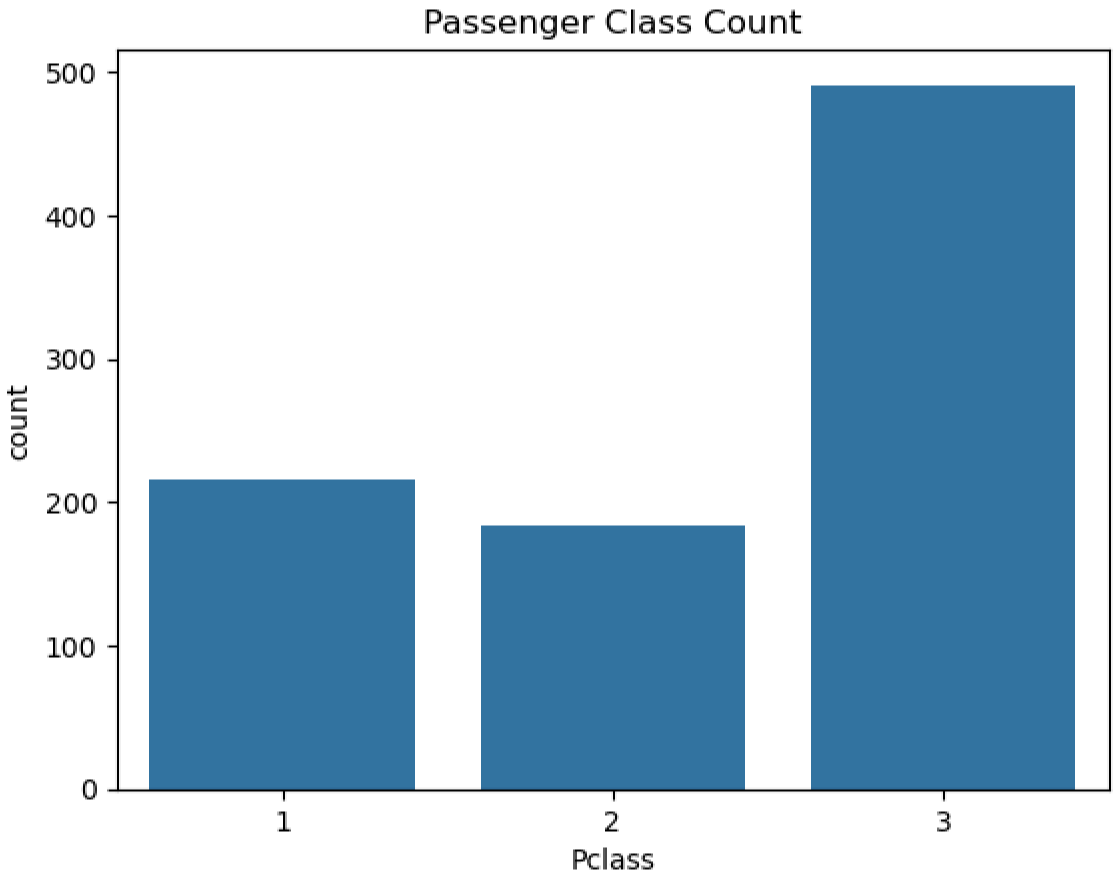


In [4]:

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

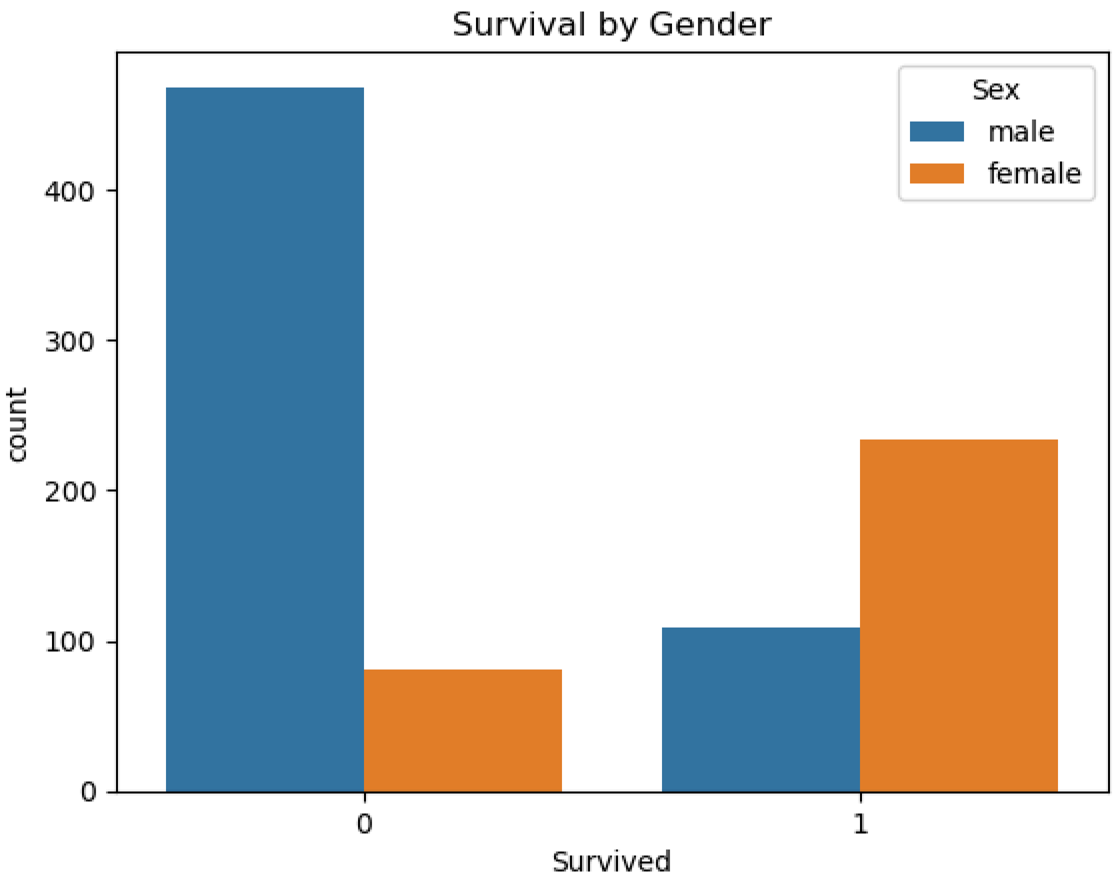


In [5]:
```
sns.countplot(x='Pclass', data=df)
plt.title("Passenger Class Count")
plt.show()
```



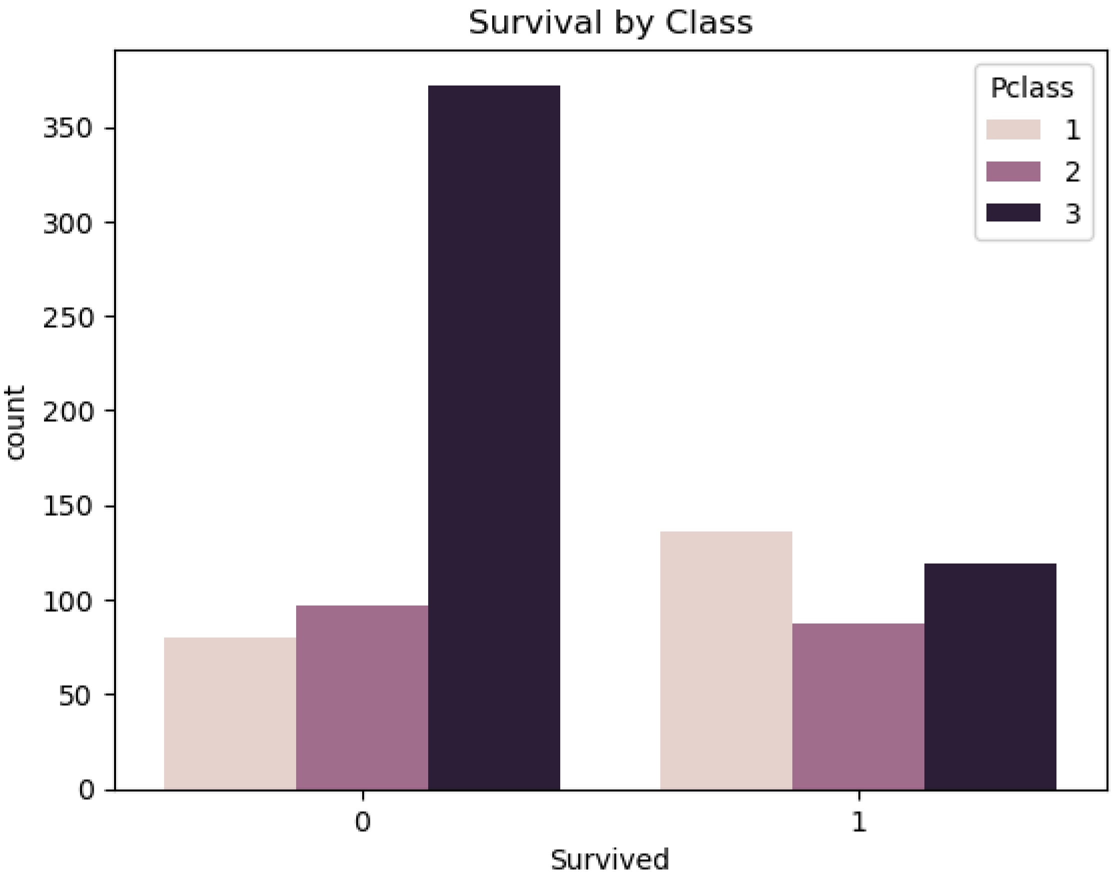
In [6]:

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



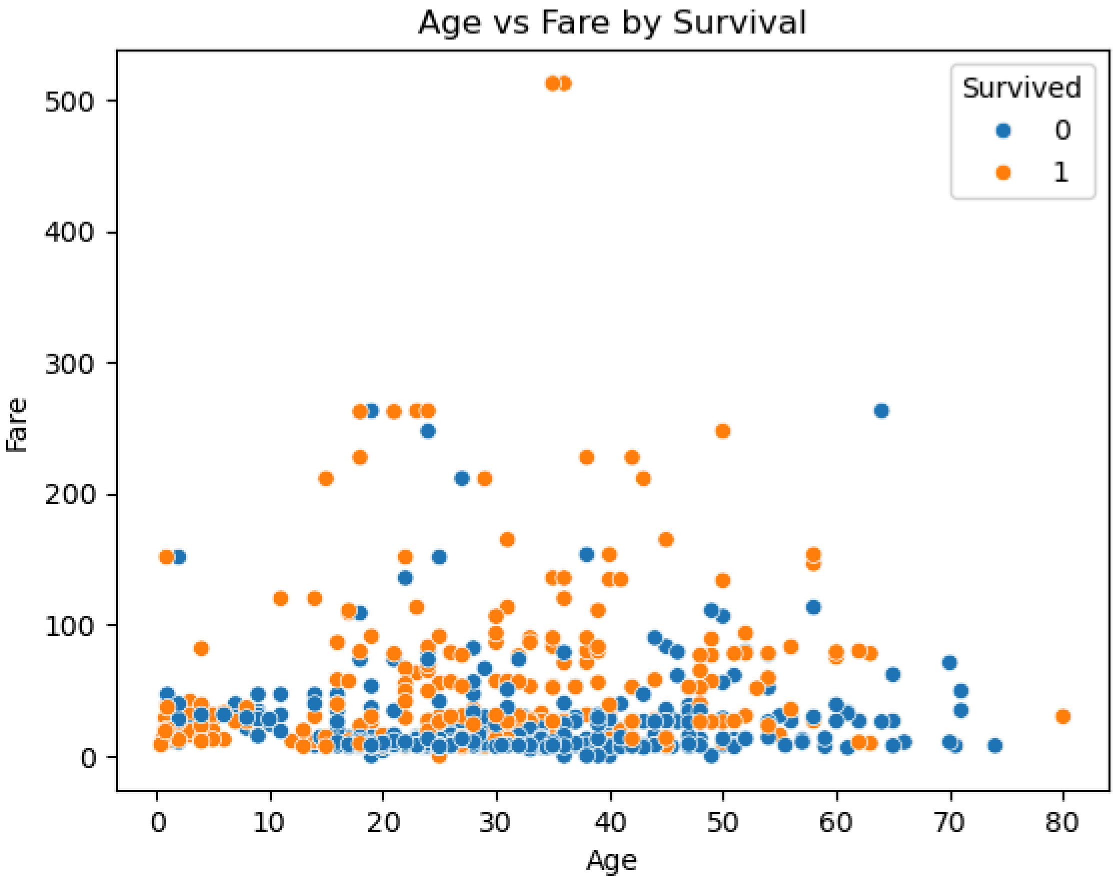
```
In [ ]:
```

```
In [7]: sns.countplot(x='Survived', hue='Pclass', data=df)
        plt.title("Survival by Class")
        plt.show()
```



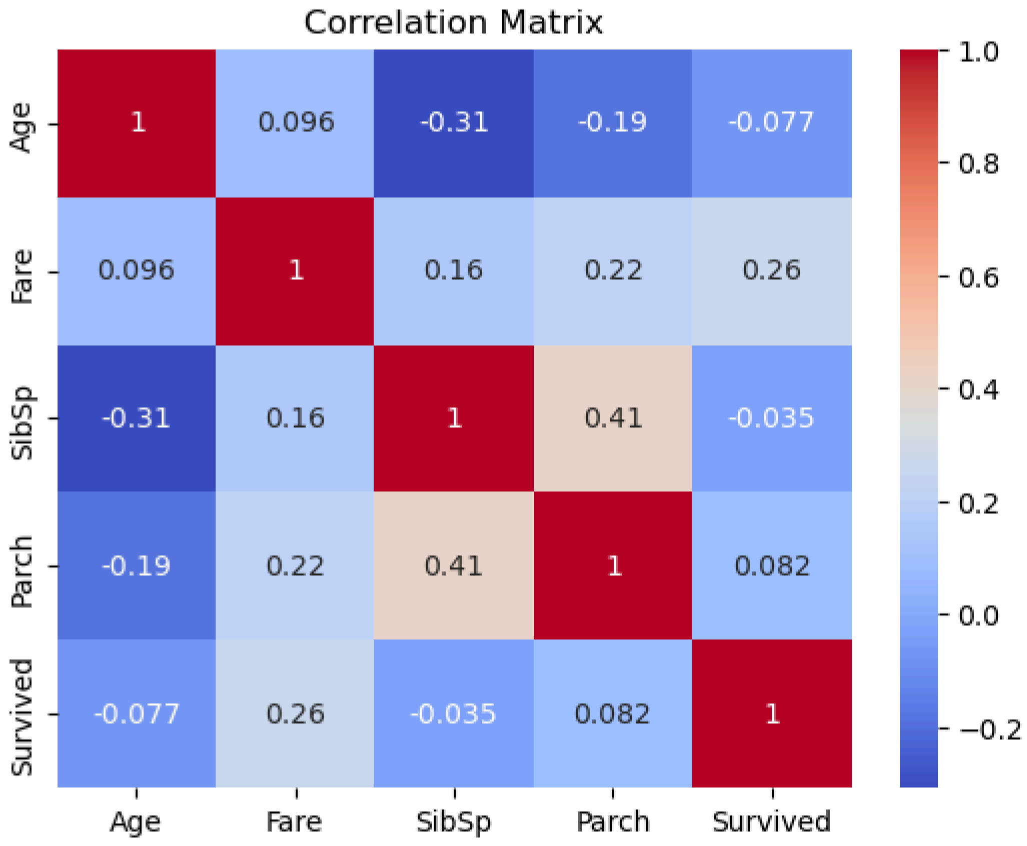
In [8]:

```
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=df)
plt.title("Age vs Fare by Survival")
plt.show()
```



In [ ]:

In [9]:
```
numeric_df = df[['Age', 'Fare', 'SibSp', 'Parch', 'Survived']]
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```



In [ ]:

```

```

In [10]:

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

```
C:\Users\adithya\AppData\Local\Temp\ipykernel_23572\1413761131.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained assig
nment 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.


  df['Age'].fillna(df['Age'].median(), inplace=True)
C:\Users\adithya\AppData\Local\Temp\ipykernel_23572\1413761131.py:2: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained assig
nment 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.


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

In [11]:

```
df['Cabin_known'] = df['Cabin'].notnull().astype(int)
```

In [12]:

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

```
C:\Users\adithya\AppData\Local\Temp\ipykernel_23572\1413761131.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained assig
nment 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.


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

In [15]:

```
print(
Final Insights:
- Females had a much higher survival rate.
- Passengers in 1st class survived more often than those in 3rd class.
- Younger passengers (children) survived more often.
- High fares were slightly associated with higher survival.
- Most missing data was in 'Age', 'Cabin', and 'Embarked'.
)
```

```
  Cell In[15], line 4
    - Passengers in 1st class survived more often than those in 3rd class.
                    ^
SyntaxError: invalid decimal literal
```

In [16]:

```
print(
"""Final Insights:
- Females had a much higher survival rate.
- Passengers in 1st class survived more often than those in 3rd class.
- Younger passengers (children) survived more often.
- High fares were slightly associated with higher survival.
- Most missing data was in 'Age', 'Cabin', and 'Embarked'."""
)
```

```
Final Insights:
- Females had a much higher survival rate.
- Passengers in 1st class survived more often than those in 3rd class.
- Younger passengers (children) survived more often.
- High fares were slightly associated with higher survival.
- Most missing data was in 'Age', 'Cabin', and 'Embarked'.
```

In [ ]: