In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
titanic_data = pd.read_csv('train.csv')
```

In [2]:
```
# Display the first few rows of the dataset
print(titanic_data.head())

# Check the dimensions of the dataset
print(titanic_data.shape)

# Summary statistics
print(titanic_data.describe())

# Check data types and missing values
print(titanic_data.info())
```

```
   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
(891, 12)
       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
<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 [3]:
```
# Count of missing values
print(titanic_data.isnull().sum())

# Fill missing values for 'Age' with the median age
median_age = titanic_data['Age'].median()
titanic_data['Age'].fillna(median_age, inplace=True)

# Drop 'Cabin' because it has too many missing values
titanic_data.drop('Cabin', axis=1, inplace=True)

# Fill missing values for 'Embarked' with the mode
mode_embarked = titanic_data['Embarked'].mode()[0]
titanic_data['Embarked'].fillna(mode_embarked, inplace=True)

# Verify no more missing values
print(titanic_data.isnull().sum())
```

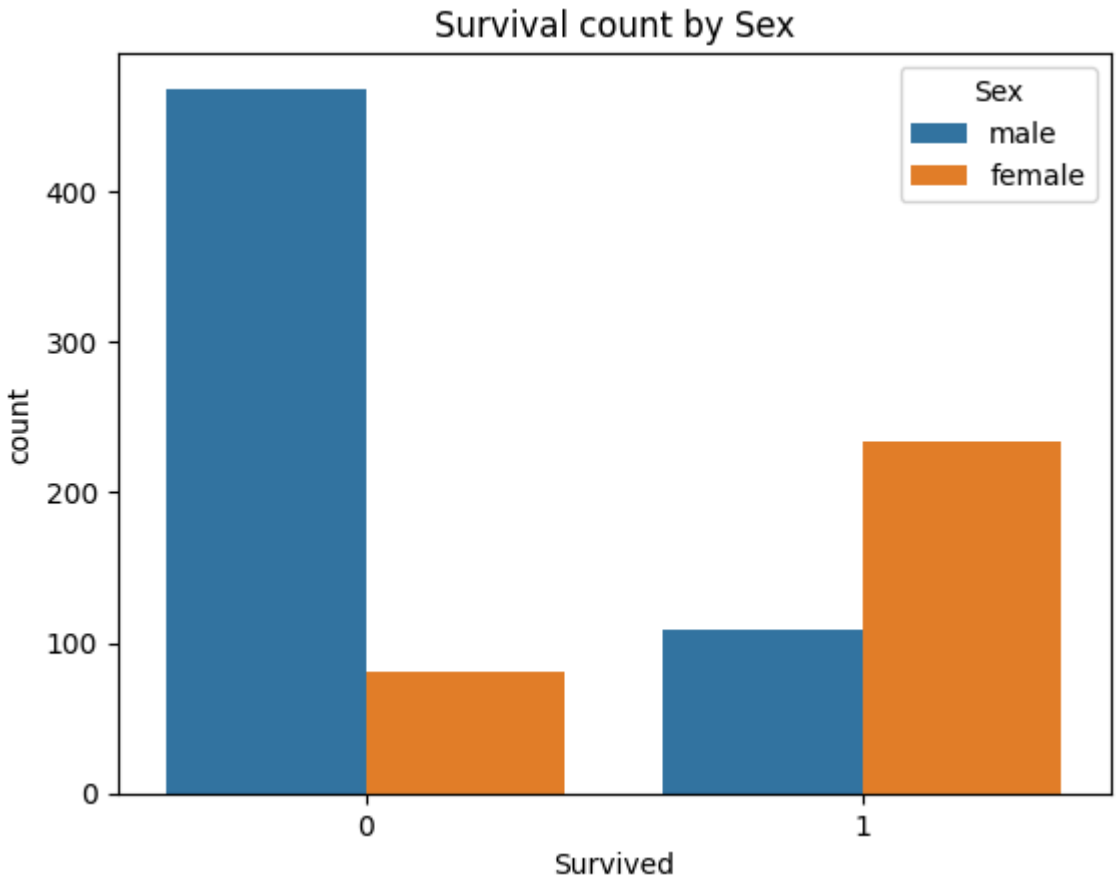
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
PassengerId    0
Survived       0
Pclass         0
Name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

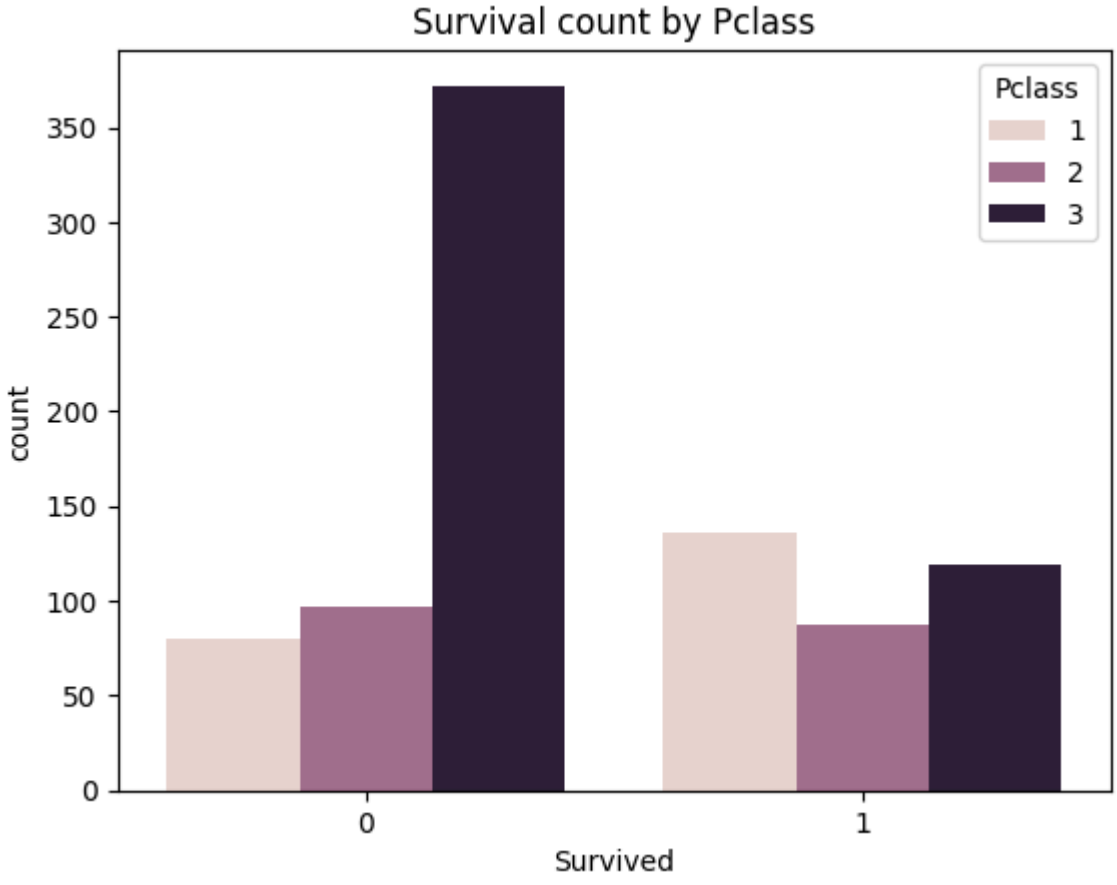
```
C:\Users\yogit\AppData\Local\Temp\ipykernel_13672\1848886882.py:6: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained as
signment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work becau
se the intermediate object on which we are setting values always behaves as a cop
y.

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


  titanic_data['Age'].fillna(median_age, inplace=True)
C:\Users\yogit\AppData\Local\Temp\ipykernel_13672\1848886882.py:13: FutureWarnin
g: A value is trying to be set on a copy of a DataFrame or Series through chained
assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work becau
se the intermediate object on which we are setting values always behaves as a cop
y.

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


  titanic_data['Embarked'].fillna(mode_embarked, inplace=True)
```

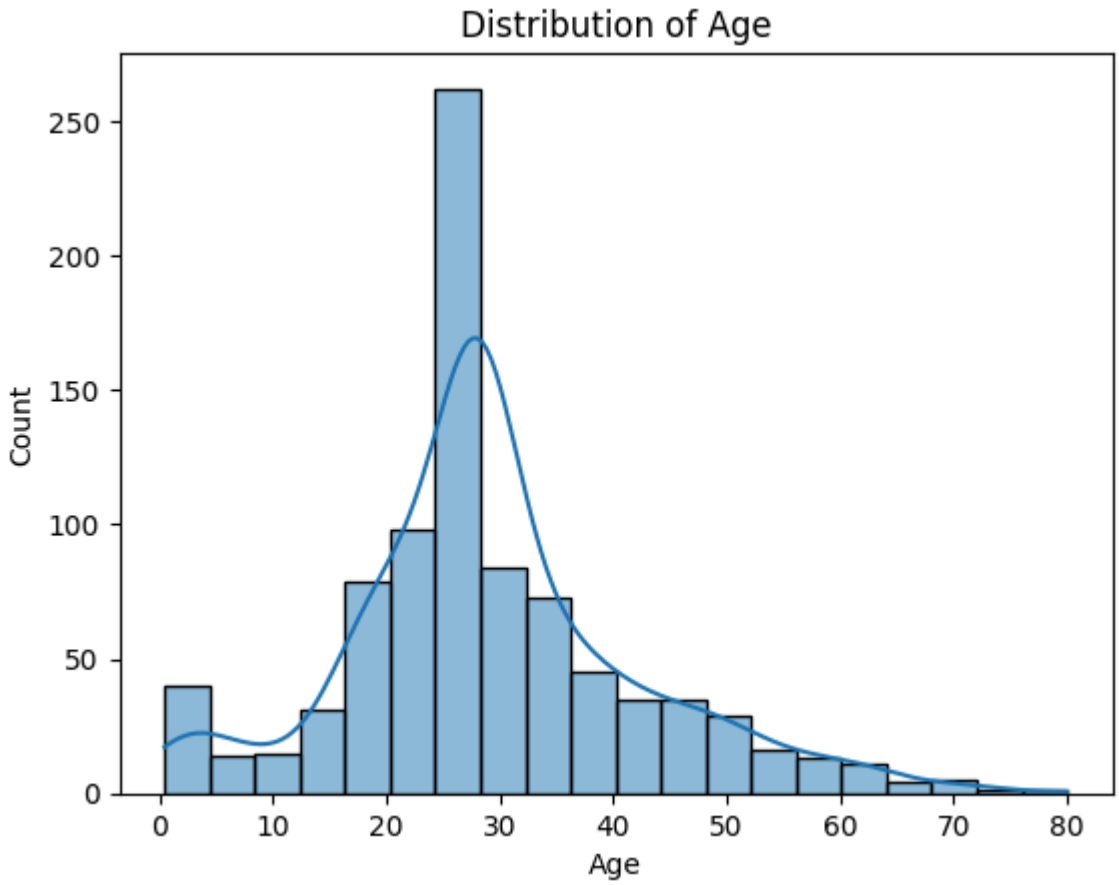
In [4]:

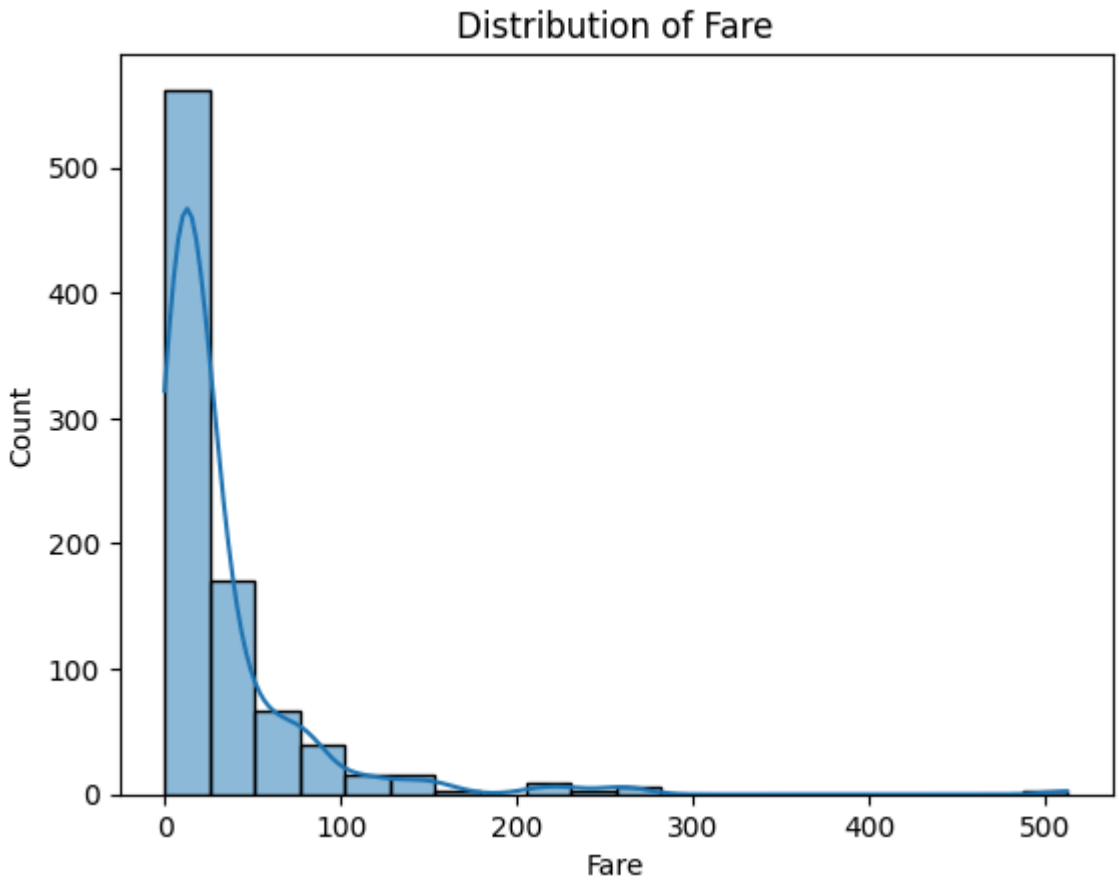
```python
# Explore survival rate by sex
sns.countplot(x='Survived', hue='Sex', data=titanic_data)
plt.title('Survival count by Sex')
plt.show()

# Explore survival rate by class
sns.countplot(x='Survived', hue='Pclass', data=titanic_data)
plt.title('Survival count by Pclass')
plt.show()

# Distribution of age
sns.histplot(titanic_data['Age'], bins=20, kde=True)
plt.title('Distribution of Age')
plt.show()

# Fare distribution
sns.histplot(titanic_data['Fare'], bins=20, kde=True)
plt.title('Distribution of Fare')
plt.show()

# Survival rate by age
sns.boxplot(x='Survived', y='Age', data=titanic_data)
plt.title('Survival by Age')
plt.show()

# Survival rate by fare
sns.boxplot(x='Survived', y='Fare', data=titanic_data)
plt.title('Survival by Fare')
plt.show()

# Correlation matrix
correlation_matrix = titanic_data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', square=True)
```

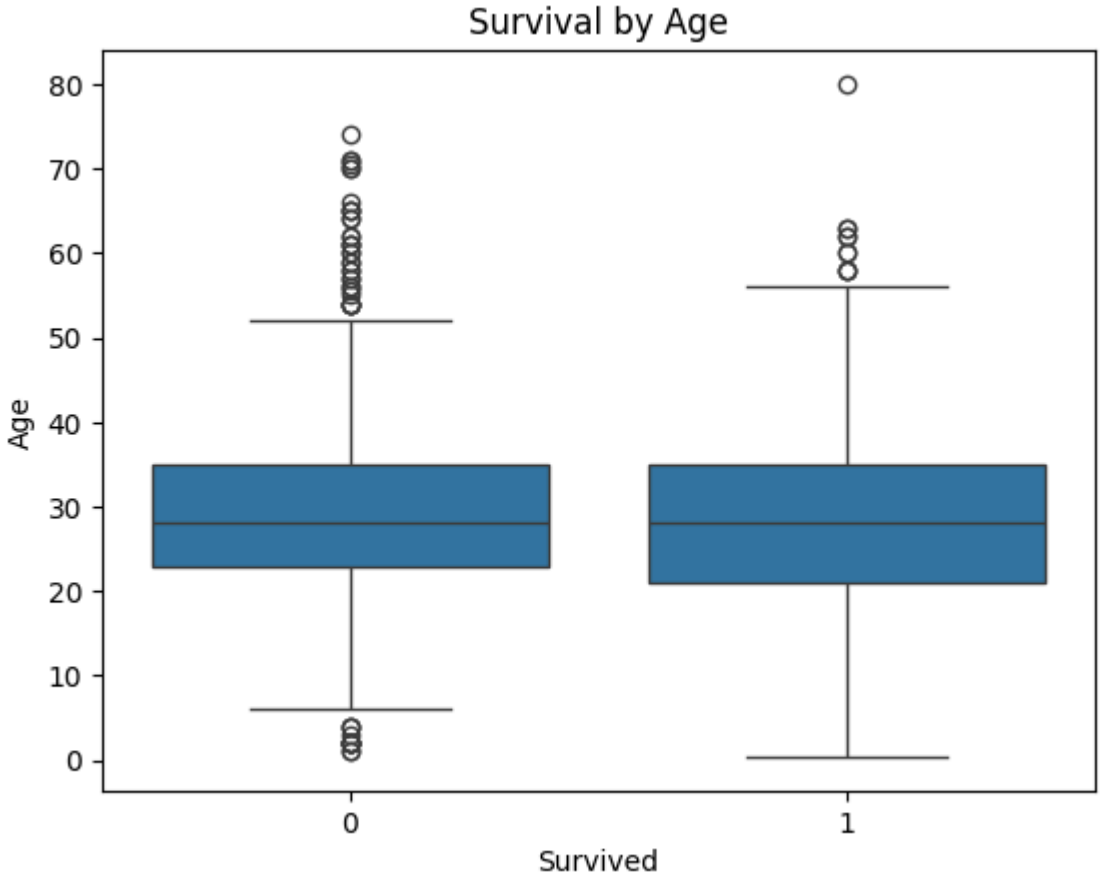
```
plt.title('Correlation Matrix')
plt.show()
```

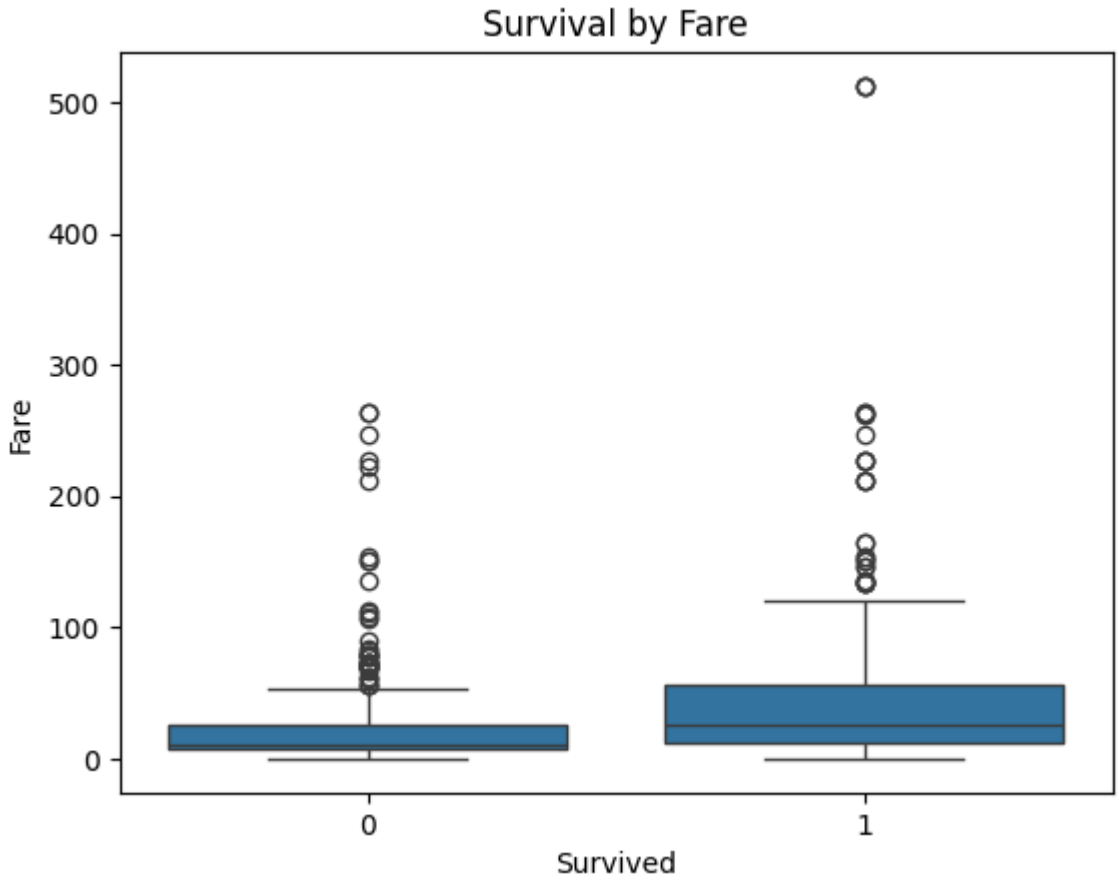





Distribution of Age
Count
Age
0
50
100
150
200
250
0
10
20
30
40
50
60
70
80


Distribution of Fare
Count
Fare
0
100
200
300
400
500
0
100
200
300
400
500


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


Survival by Fare
Fare
Survived
0
100
200
300
400
500
0
1

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[4], line 32
     29 plt.show()
     31 # Correlation matrix
---> 32 correlation_matrix = titanic_data.corr()
     33 sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', square=True)
     34 plt.title('Correlation Matrix')

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\frame.py:11049, in DataFrame.corr(self, method, min_periods, numeric_only)
  11047 cols = data.columns
  11048 idx = cols.copy()
> 11049 mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
  11051 if method == "pearson":
  11052     correl = libalgos.nancorr(mat, minp=min_periods)

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\frame.py:1993, in DataFrame.to_numpy(self, dtype, copy, na_value)
   1991 if dtype is not None:
   1992     dtype = np.dtype(dtype)
-> 1993 result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
   1994 if result.dtype is not dtype:
   1995     result = np.asarray(result, dtype=dtype)

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\internals\managers.py:1694, in BlockManager.as_array(self, dtype, copy, na_value)
   1692         arr.flags.writeable = False
   1693 else:
-> 1694     arr = self._interleave(dtype=dtype, na_value=na_value)
   1695     # The underlying data was copied within _interleave, so no need
   1696     # to further copy if copy=True or setting na_value
   1698 if na_value is lib.no_default:

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\pandas\core\internals\managers.py:1753, in BlockManager._interleave(self, dtype, na_value)
   1751     else:
   1752         arr = blk.get_values(dtype)
-> 1753     result[rl.indexer] = arr
   1754     itemmask[rl.indexer] = 1
   1756 if not itemmask.all():

ValueError: could not convert string to float: 'Braund, Mr. Owen Harris'
```

In [ ]: