```python
import pandas as pd
titanic_df = pd.read_csv('Titanic-Dataset.csv')

print(titanic_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
```

```python
titanic_df.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

import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10,6))
sns.heatmap(titanic_df.isnull(), cbar=False, cmap='viridis')
plt.title('Missing Values in the Titanic Dataset')
plt.show()

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

titanic_df.drop(columns=['Cabin'], inplace=True)

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

print(titanic_df.isnull().sum())
```
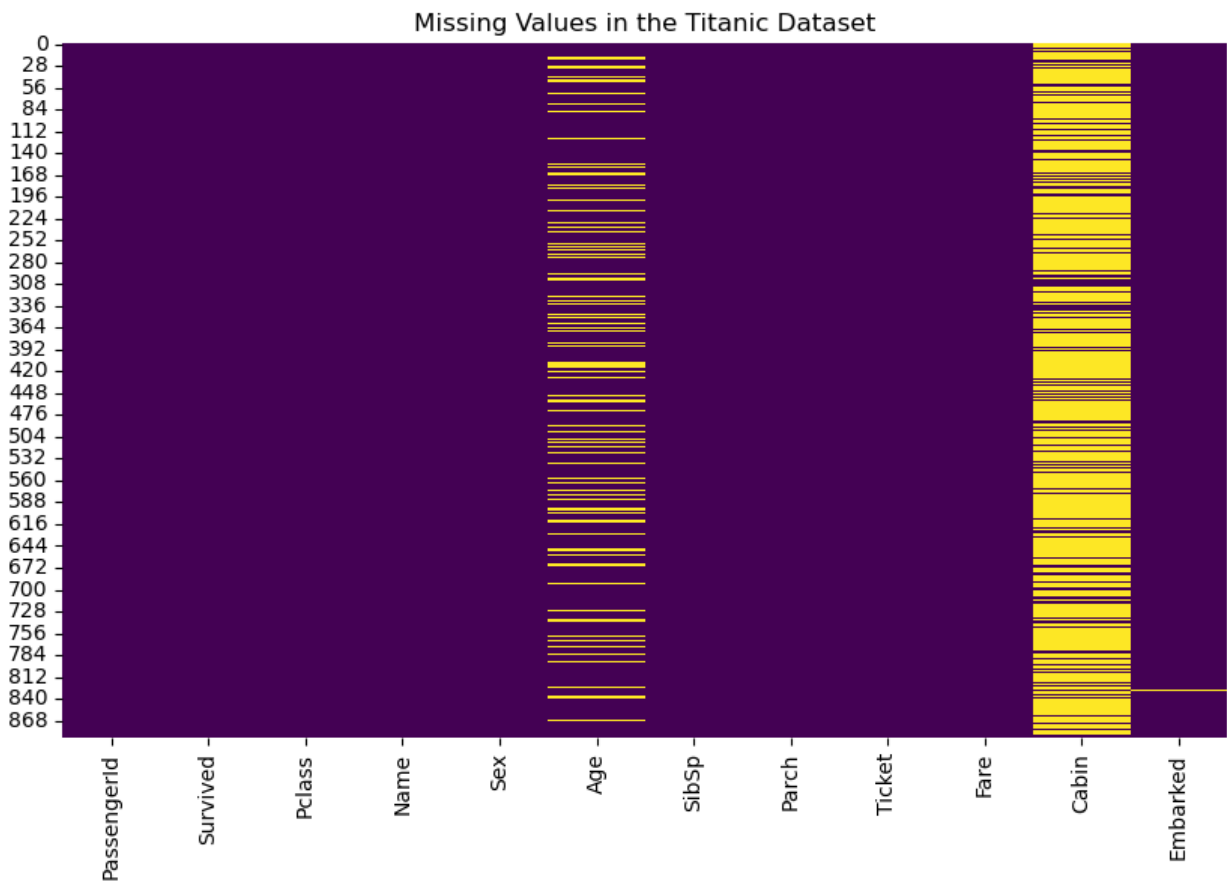

Missing Values in the Titanic Dataset

```
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\Dibyam Jyoti Pradhan\AppData\Local\Temp\
ipykernel_17240\1712176759.py:9: FutureWarning: 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 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.


  titanic_df['Age'].fillna(titanic_df['Age'].median(), inplace=True)
C:\Users\Dibyam Jyoti Pradhan\AppData\Local\Temp\
ipykernel_17240\1712176759.py:13: FutureWarning: 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 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.


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

print(titanic_df.describe(include='all'))

       PassengerId    Survived      Pclass                           Name
Sex  \
count   891.000000  891.000000  891.000000                            891
891
```

	PassengerId	Survived	Pclass	Name	Sex
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	Embarked
count	891.000000	891.000000	891.000000	891	891.000000	891
unique	NaN	NaN	NaN	681	NaN	3
top	NaN	NaN	NaN	347082	NaN	S
freq	NaN	NaN	NaN	7	NaN	646
mean	29.361582	0.523008	0.381594	NaN	32.204208	NaN
std	13.019697	1.102743	0.806057	NaN	49.693429	NaN
min	0.420000	0.000000	0.000000	NaN	0.000000	NaN
25%	22.000000	0.000000	0.000000	NaN	7.910400	NaN
50%	28.000000	0.000000	0.000000	NaN	14.454200	NaN
75%	35.000000	1.000000	0.000000	NaN	31.000000	NaN
max	80.000000	8.000000	6.000000	NaN	512.329200	NaN

```python
plt.figure(figsize=(10, 6))


plt.subplot(2, 3, 1)
```

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


plt.subplot(2, 3, 2)
sns.countplot(x='Pclass', data=titanic_df)
plt.title('Passenger Class Count')


plt.subplot(2, 3, 3)
sns.histplot(titanic_df['Age'], kde=True)
plt.title('Age Distribution')


plt.subplot(2, 3, 4)
sns.histplot(titanic_df['Fare'], kde=True)
plt.title('Fare Distribution')


plt.subplot(2, 3, 5)
sns.countplot(x='Sex', data=titanic_df)
plt.title('Gender Count')

plt.tight_layout()
plt.show()
```
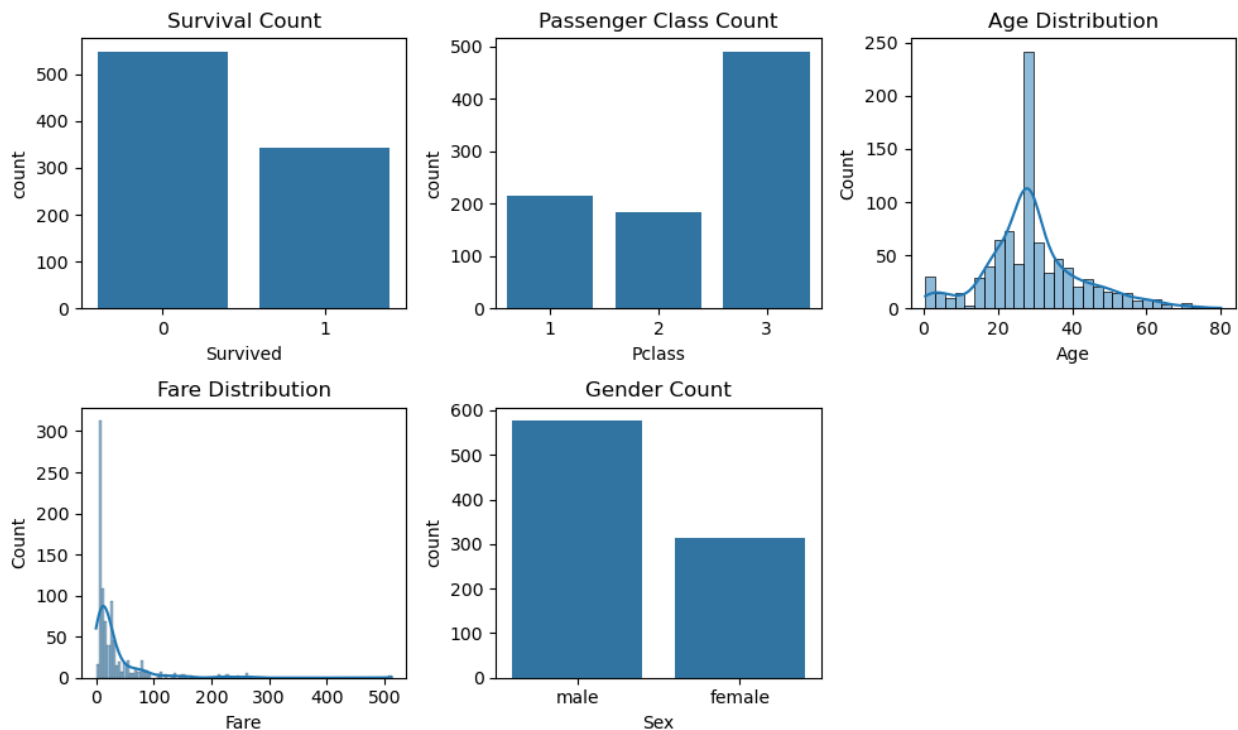
```python
plt.figure(figsize=(12, 8))


plt.subplot(2, 2, 1)
sns.countplot(x='Pclass', hue='Survived', data=titanic_df)
plt.title('Survival Count by Passenger Class')


plt.subplot(2, 2, 2)
sns.countplot(x='Sex', hue='Survived', data=titanic_df)
plt.title('Survival Count by Gender')


plt.subplot(2, 2, 3)
sns.kdeplot(data=titanic_df[titanic_df['Survived']==1]['Age'],
label='Survived', shade=True)
sns.kdeplot(data=titanic_df[titanic_df['Survived']==0]['Age'],
label='Not Survived', shade=True)
plt.title('Age Distribution by Survival')
plt.legend()


plt.subplot(2, 2, 4)
sns.kdeplot(data=titanic_df[titanic_df['Survived']==1]['Fare'],
label='Survived', shade=True)
sns.kdeplot(data=titanic_df[titanic_df['Survived']==0]['Fare'],
label='Not Survived', shade=True)
plt.title('Fare Distribution by Survival')
plt.legend()

plt.tight_layout()
plt.show()
```

```
C:\Users\Dibyam Jyoti Pradhan\AppData\Local\Temp\
ipykernel_17240\1288160612.py:15: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data=titanic_df[titanic_df['Survived']==1]['Age'],
label='Survived', shade=True)
C:\Users\Dibyam Jyoti Pradhan\AppData\Local\Temp\
ipykernel_17240\1288160612.py:16: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data=titanic_df[titanic_df['Survived']==0]['Age'],
label='Not Survived', shade=True)
C:\Users\Dibyam Jyoti Pradhan\AppData\Local\Temp\
ipykernel_17240\1288160612.py:22: FutureWarning:
```
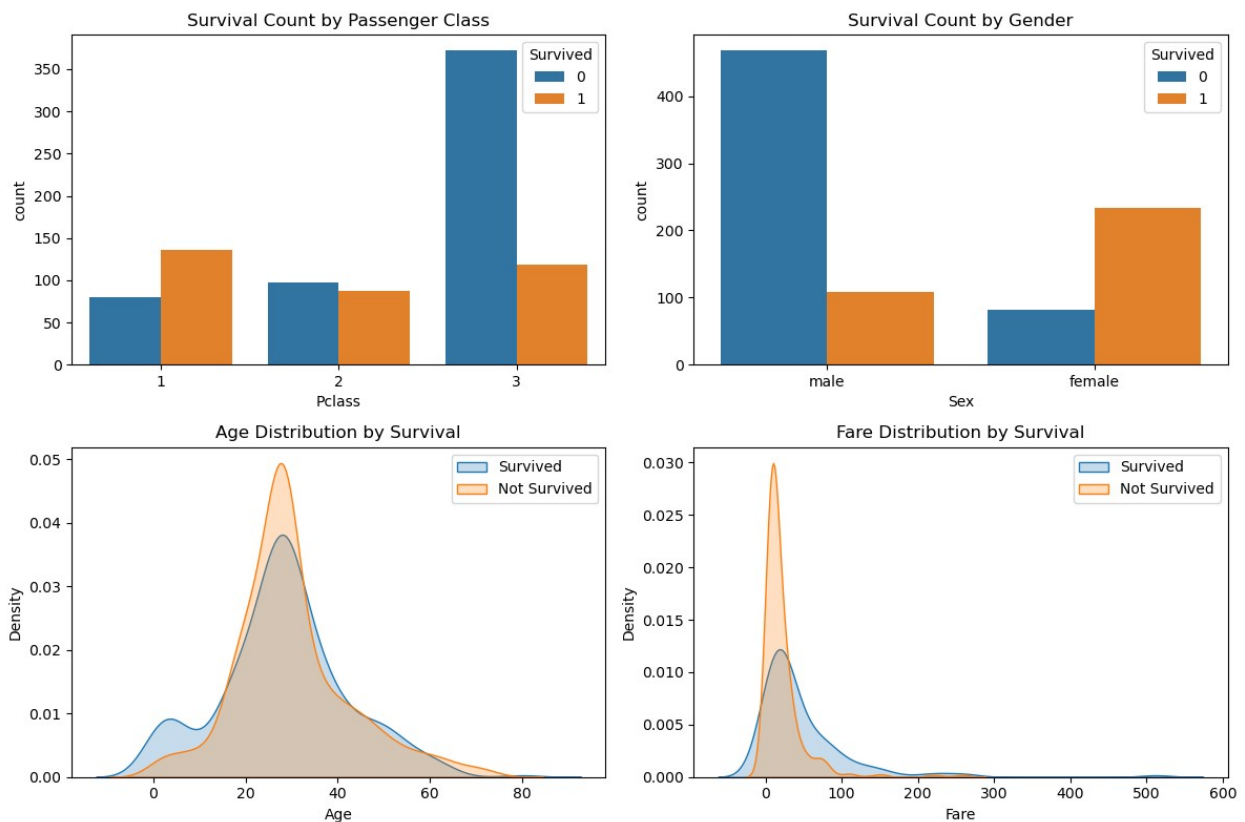
```
plt.figure(figsize=(10, 6))
sns.pairplot(titanic_df[['Survived', 'Pclass', 'Age', 'Fare', 'Sex']],
hue='Survived', palette='Set1')

plt.show()

<Figure size 1000x600 with 0 Axes>
```