In [1]:
```
#Importing Libraries:

#Import Pandas Library and make it as pd:
import pandas as pd

#Import NumPy Library and make it as np:
import numpy as np

#Import PyPlot from Matplotlib Library and make it as plt:
import matplotlib.pyplot as plt

#Import Seaborn Library and make it as sns:
import seaborn as sns

#Inorder to Supress Warnings import Filterwarnings:
from warnings import filterwarnings
filterwarnings('ignore')
```

In [7]:
```
df = pd.read_csv('titanic.csv')
df
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

418 rows × 12 columns

In [9]:
```
# Display head of the Dataset:
# head() displays first five rows:

df.head()
```

Out[9]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [11]:
```
#Display tail of Dataset:
# tail() displays last five rows:
df.tail()
```

Out[11]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	0	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	0	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	0	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

In [13]:
```
# Finding size of the data:
df.shape
```

Out[13]:
```
(418, 12)
```

In [15]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Survived     418 non-null    int64
 2   Pclass       418 non-null    int64
 3   Name         418 non-null    object
 4   Sex          418 non-null    object
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64
 7   Parch        418 non-null    int64
 8   Ticket       418 non-null    object
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object
 11  Embarked     418 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

In [17]:
```
df.describe()
```

Out[17]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	0.363636	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.481622	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	0.000000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	0.000000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	1.000000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	1.000000	3.000000	76.000000	8.000000	9.000000	512.329200

In [19]:
```
# Unique values in the dataset:
df.nunique()
```

Out[19]:
```
PassengerId    418
Survived         2
Pclass           3
Name           418
Sex              2
Age             79
SibSp            7
Parch            8
Ticket         363
Fare           169
Cabin           76
Embarked         3
dtype: int64
```

In [21]:
```
# Check the variable types:

df.dtypes
```

Out[21]:
```
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 [23]:
```
# Check for the Duplicate Values from the Dataset:
df.duplicated().sum()
```

Out[23]:
```
0
```

In [25]:
```
# Cheek for the Null values from the Dataset:
df.isnull().sum()
```

Out[25]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [27]:
```
# Skewness of the Dataset:
df.skew(numeric_only = True)
```

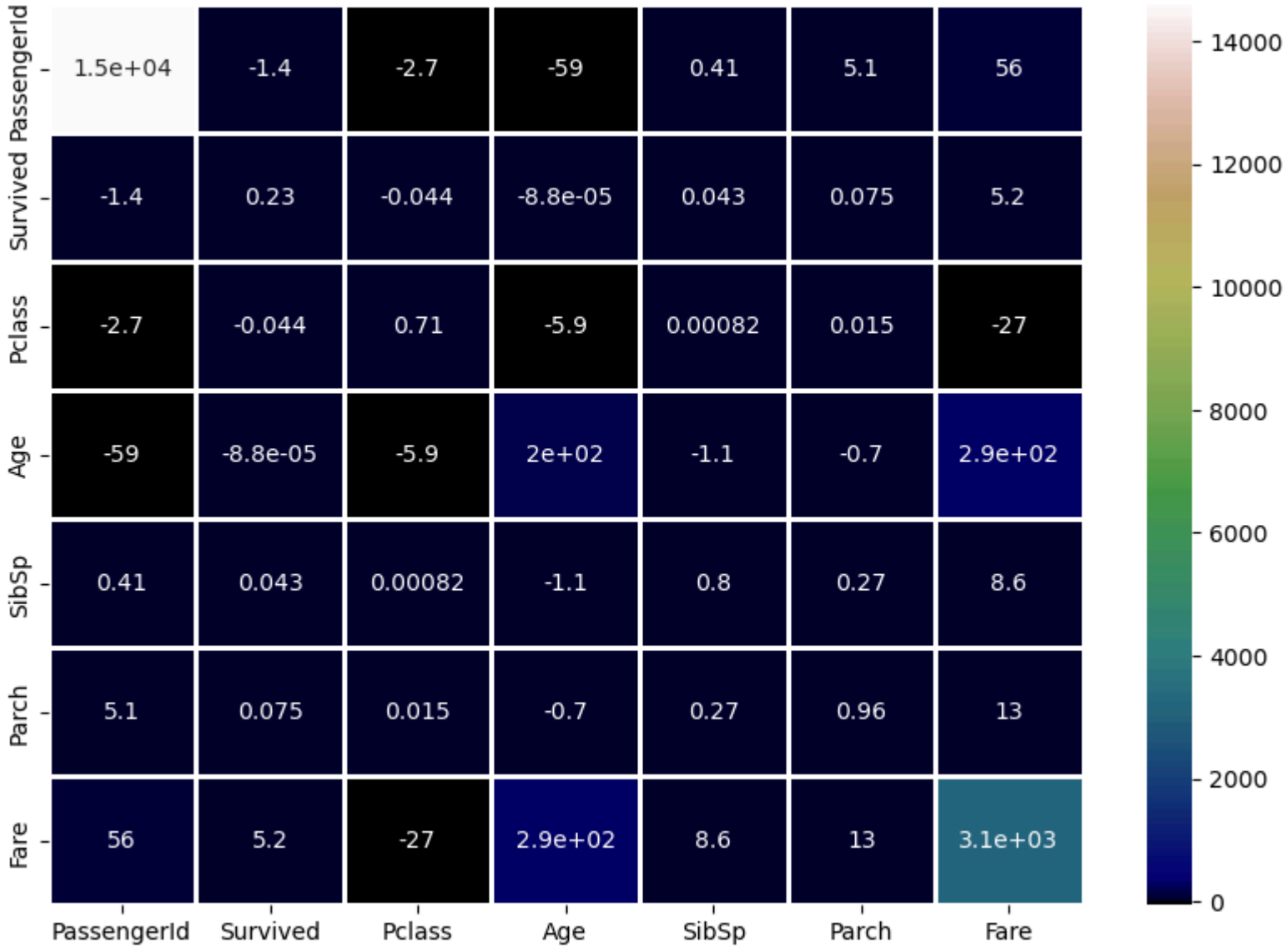
Out[27]:
```
PassengerId    0.000000
Survived       0.568991
Pclass        -0.534170
Age            0.457361
SibSp          4.168337
Parch          4.654462
Fare           3.687213
dtype: float64
```

In [29]:
```
# Kurtosis of the Dataset;
df.kurt(numeric_only = True)
```

Out[29]:
```
PassengerId    -1.200000
Survived       -1.684332
Pclass         -1.382666
Age             0.083783
SibSp          26.498712
Parch          31.412513
Fare           17.921595
dtype: float64
```

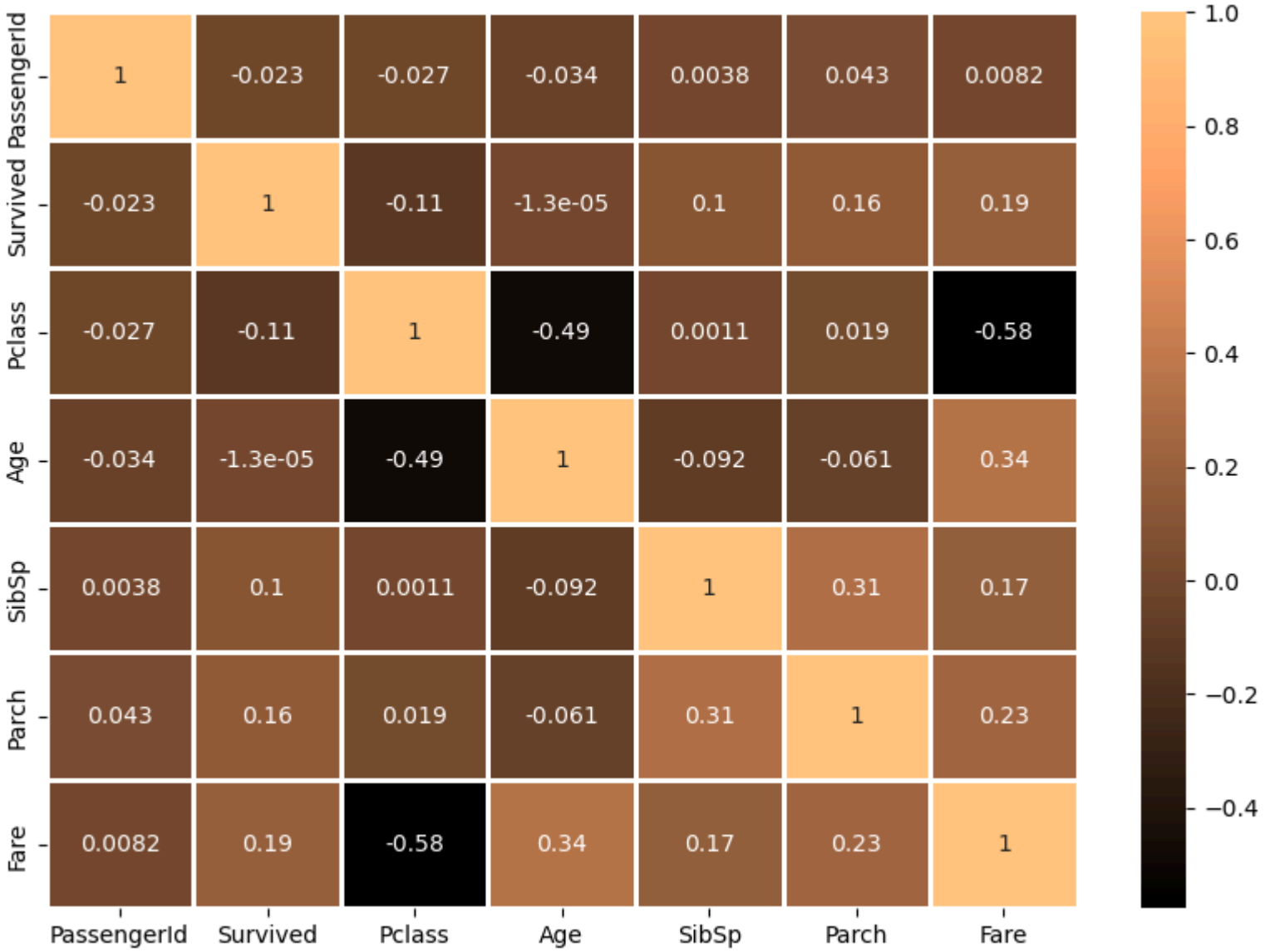
In [33]:

```
# Covariance
cov = df.cov(numeric_only = True)
cov

fig,ax = plt.subplots(figsize = (10,7))

sns.heatmap(cov,annot = True,linewidth = 0.95,
            cmap = 'gist_earth',fmt = '.2g')

plt.show()
```



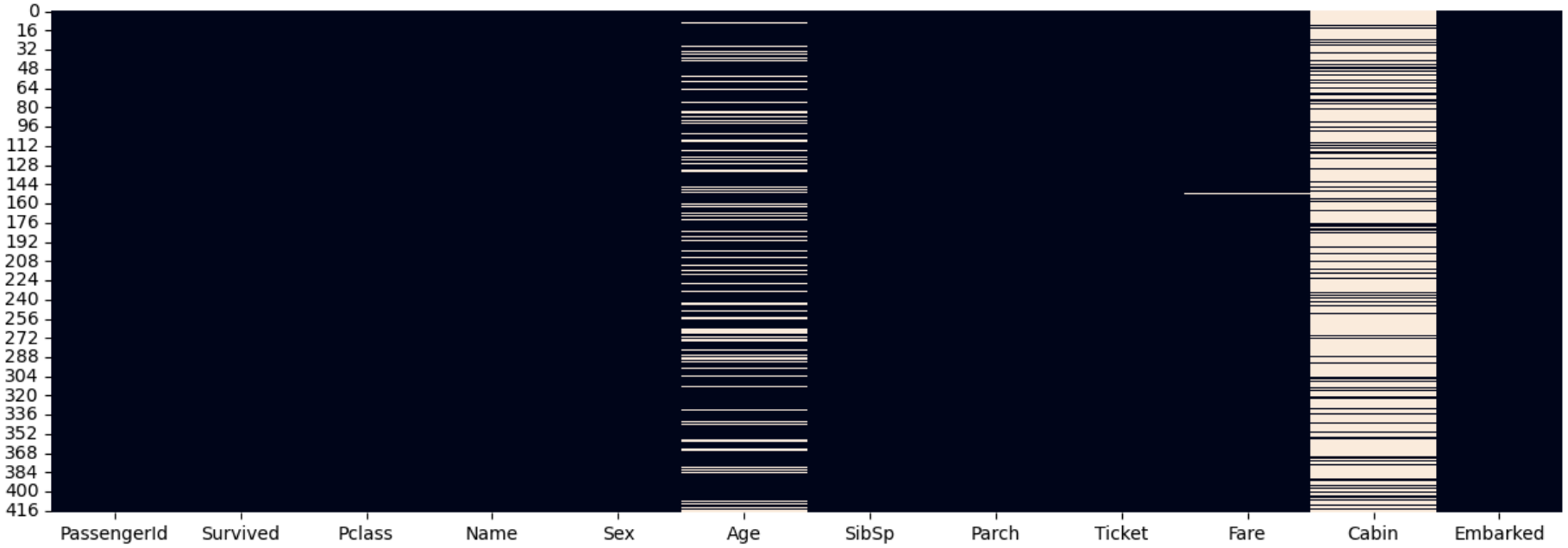
In [35]:
```
# Correlation Matrix

corr = df.corr(numeric_only = True)
corr
```

```
fig,ax = plt.subplots(figsize = (10,7))

sns.heatmap(corr,annot = True,linewidths = 0.95,
            cmap = 'copper',fmt = '.2g')

plt.show()
```



In [31]:
```
# Visualize missing values using heat map:

plt.rcParams['figure.figsize'] = [15,5]

sns.heatmap(df.isnull(), cbar = False)
```

```
plt.show()
```



In [37]:
```
# Visualizing Missing Values

missing_values = df.isnull().sum()

total = df.isnull().sum().sort_values(ascending = False)

percent = ((df.isnull().sum()/df.shape[0])*100)

percent = percent.sort_values(ascending = False)

missing_data = pd.concat([total,percent],axis = 1,
                         keys = ['Total','Percentage'])

missing_data['Type'] = df[missing_data.index].dtypes

missing_data
```

Out[37]:

	Total	Percentage	Type
Cabin	327	78.229665	object
Age	86	20.574163	float64
Fare	1	0.239234	float64
PassengerId	0	0.000000	int64
Survived	0	0.000000	int64
Pclass	0	0.000000	int64
Name	0	0.000000	object
Sex	0	0.000000	object
SibSp	0	0.000000	int64
Parch	0	0.000000	int64
Ticket	0	0.000000	object
Embarked	0	0.000000	object

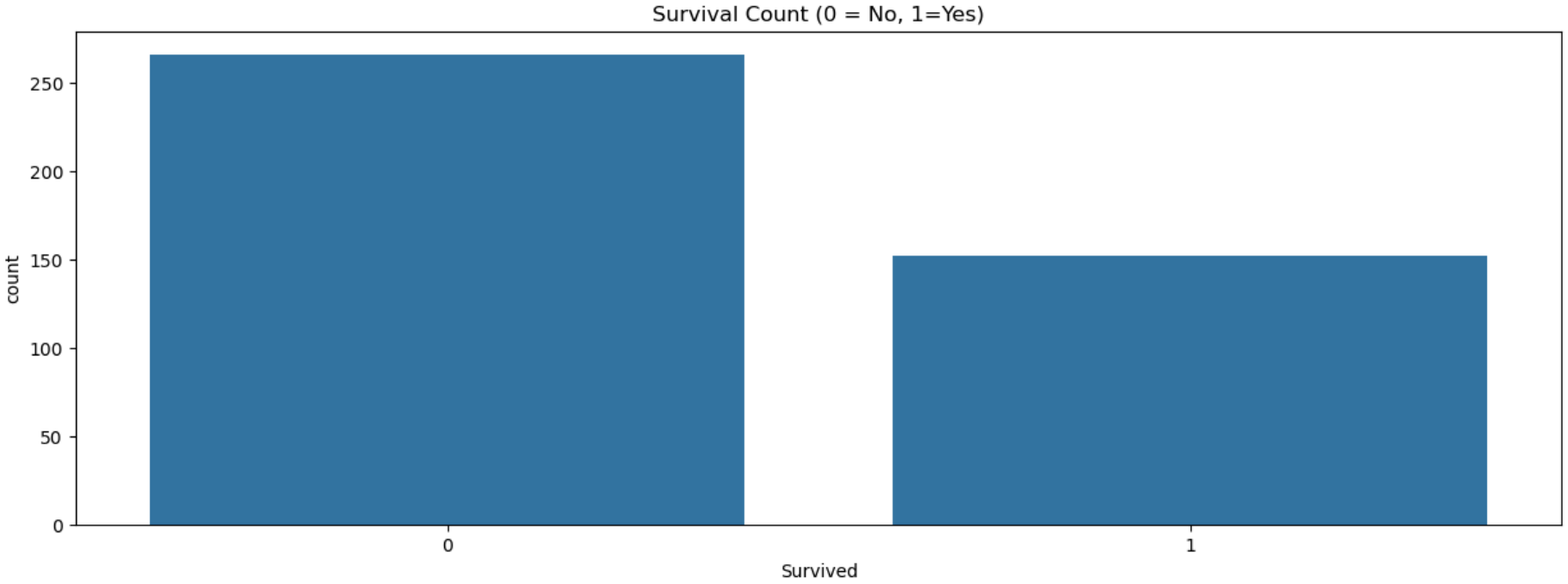
In [39]:
```
# Fill missing values

# Fill 'Age' with median
df['Age'].fillna(df['Age'].median(),inplace=True)

# Fill 'Embarked' with mode
df['Embarked'].fillna(df['Embarked'].mode()[0],inplace=True)

# Drop 'Cabin' due to too many missing values
df.drop(columns=['Cabin'],inplace=True)
```

In [41]:
```
# Sanitary check for missing values

df.isnull().sum()
```

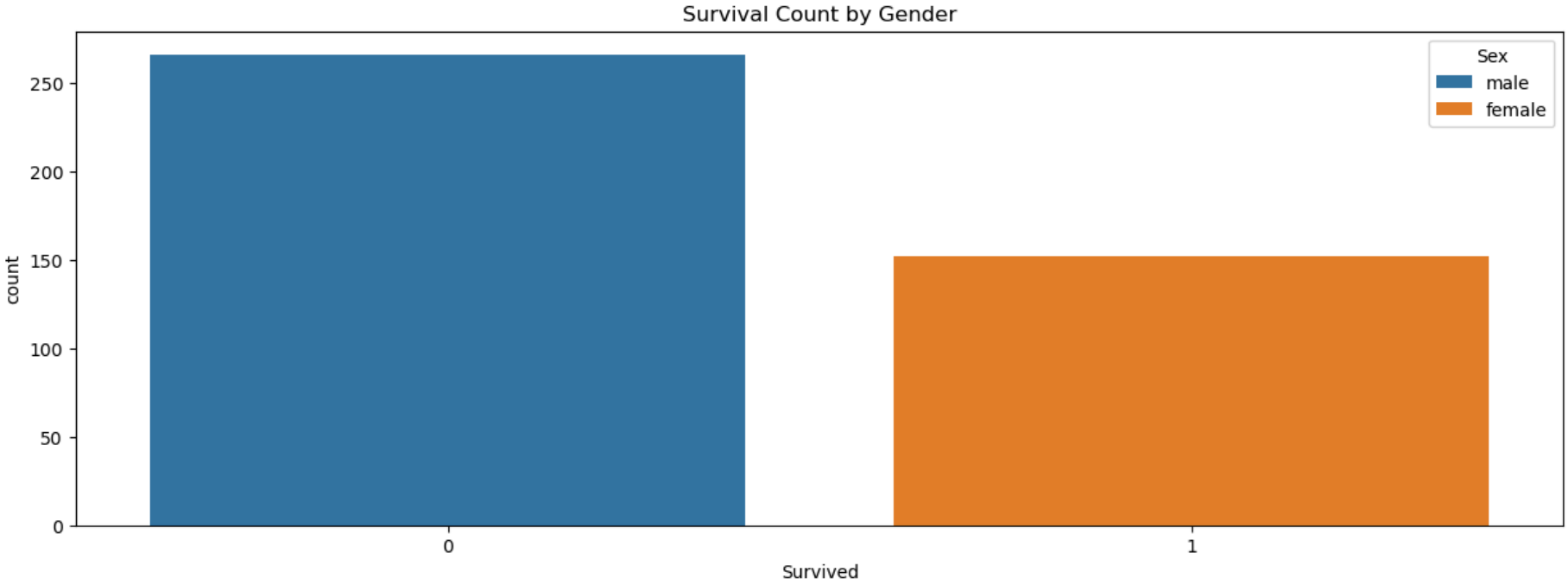
Out[41]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           1
Embarked       0
dtype: int64
```

Visualization

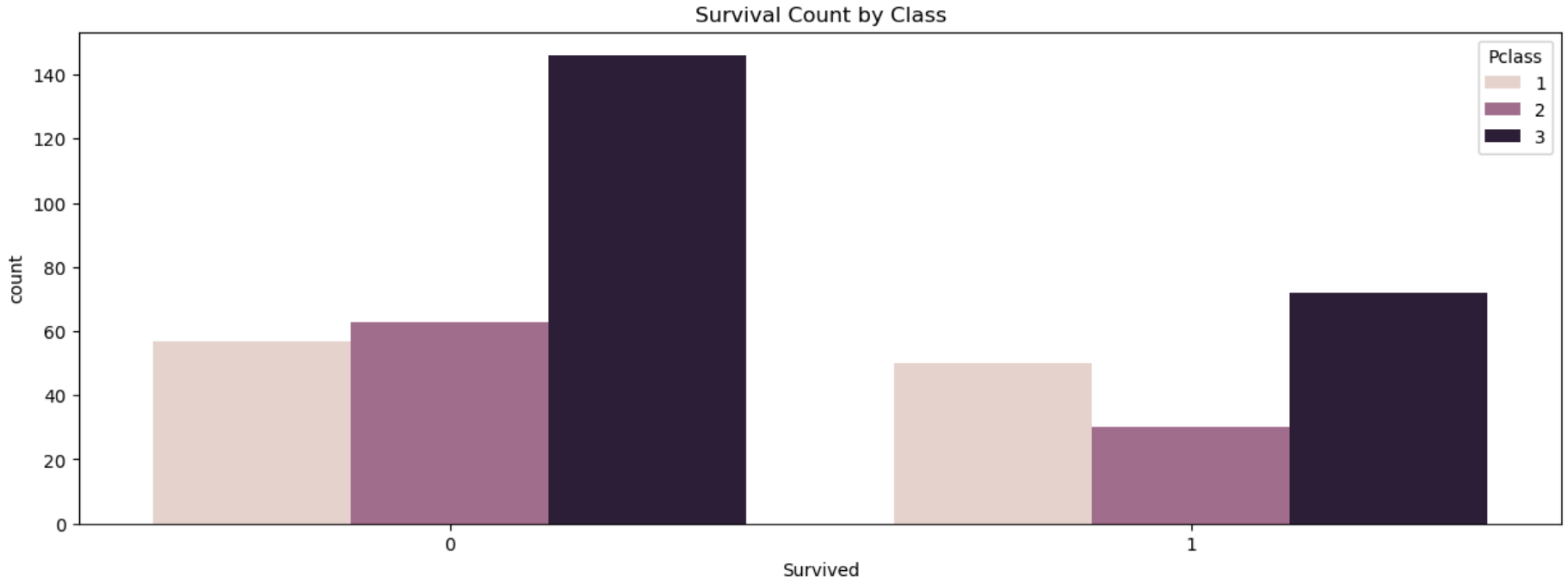
In [44]:
```
# Understand the distribution of target variable 'Survived'

sns.countplot(x='Survived',data=df)
plt.title('Survival Count (0 = No, 1=Yes)')

plt.show()
```



In [46]:
```
# Analyze Survival rate by Gender

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

plt.show()
```



In [48]:
```
# Analyze Survival rate by passenger class

sns.countplot(x='Survived', hue='Pclass',data=df)
plt.title('Survival Count by Class')

plt.show()
```



In [50]:
```
# Encode categorical variables

df_encoded = df.copy()
df_encoded['Sex'] = df_encoded['Sex'].map({'male':0,'female':1})
df_encoded['Embarked'] = df_encoded['Embarked'].map({'S':0,'C':1,'Q':2})
```

In [52]:
```
# Final check of dataset
df_encoded.head()
```

Out[52]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	892	0	3	Kelly, Mr. James	0	34.5	0	0	330911	7.8292	2
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	1	47.0	1	0	363272	7.0000	0
2	894	0	2	Myles, Mr. Thomas Francis	0	62.0	0	0	240276	9.6875	2
3	895	0	3	Wirz, Mr. Albert	0	27.0	0	0	315154	8.6625	0
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	1	22.0	1	1	3101298	12.2875	0

Summary

The Exploratory Data Analysis (EDA) on the Titanic dataset revealed valuable insights into the survival patterns of passengers aboard the ill-fated ship. Here's a consolidated summary of our findings:

Missing Values Handling:

- The dataset had missing values in the Age, Cabin, and Embarked columns.
- We filled Age with the median, Embarked with the mode, and dropped Cabin due to excessive missing data.

Survival Distribution:

- About 62% of passengers did not survive, while 38% survived.

Gender Impact:

- Females had a much higher survival rate than males.
- Survival rate among females was significantly higher, likely due to the "women and children first" policy.

Passenger Class:

- First-class passengers had the highest survival rate, while third-class had the lowest.
- Socioeconomic status played a crucial role in survival.

Age Factor:

- Most passengers were aged 20 to 40.

-Some children had higher survival rates, aligning with emergency rescue priorities.

Embarkation Port:

- Passengers who embarked from Cherbourg ('C') showed a higher survival rate, likely because many first-class passengers boarded there.

Fare & Correlation:

- Higher ticket fare showed a positive correlation with survival, indirectly highlighting wealth and class.

Feature Engineering & Preparation:

Categorical variables were encoded for future predictive modeling.

THANK YOU