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


# Set visualization styles
sns.set(style="whitegrid")
```

Load the Dataset

```
train = pd.read_csv('train.csv')
test_data = pd.read_csv("test.csv")
```

1. Basic Overview

```
print("\n--- First 5 Rows ---\n")
print(train.head())
```

```
--- First 5 Rows ---

   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
```

```
print("\n--- First 5 Rows ---\n")
print(test_data.head())
```

```
--- First 5 Rows ---

   PassengerId  Pclass                                          Name     Sex  \
0          892       3                              Kelly, Mr. James    male
1          893       3              Wilkes, Mrs. James (Ellen Needs)  female
2          894       2                     Myles, Mr. Thomas Francis    male
3          895       3                              Wirz, Mr. Albert    male
4          896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female

    Age  SibSp  Parch   Ticket     Fare Cabin Embarked
0  34.5      0      0   330911   7.8292   NaN        Q
1  47.0      1      0   363272   7.0000   NaN        S
2  62.0      0      0   240276   9.6875   NaN        Q
3  27.0      0      0   315154   8.6625   NaN        S
4  22.0      1      1  3101298  12.2875   NaN        S
```

```
print("\n--- Dataset Info ---\n")
print(train.info())
```

```
--- Dataset 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
None
```

```
print("\n--- Statistical Summary ---\n")
print(train.describe())
```

```
--- Statistical Summary ---

       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
```

```
print("\n--- Checking Missing Values ---\n")
print(train.isnull().sum())
```

```
--- Checking Missing Values ---

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
```

🛠 Fill missing values after checking

```
# Fill missing Age values with median
train['Age'] = train['Age'].fillna(train['Age'].median())
```

```
# Fill missing Embarked values with mode
train['Embarked'] = train['Embarked'].fillna(train['Embarked'].mode()[0])
```

```
# Check again if missing values are gone
print("\n--- Missing Values After Filling ---\n")
print(train.isnull().sum())
```

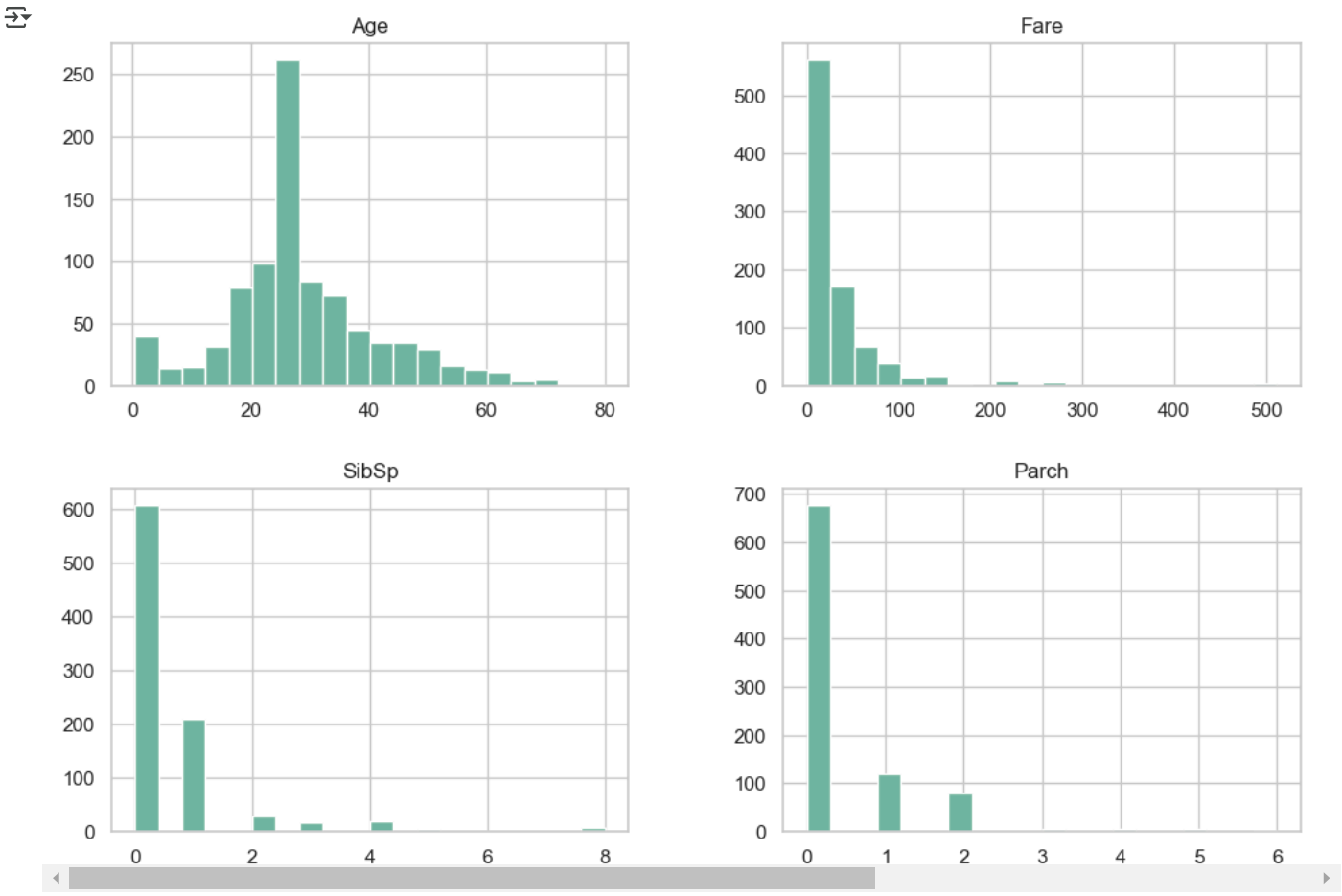
```
--- Missing Values After Filling ---

PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
```

```
Fare            0
Cabin         687
Embarked        0
dtype: int64
```

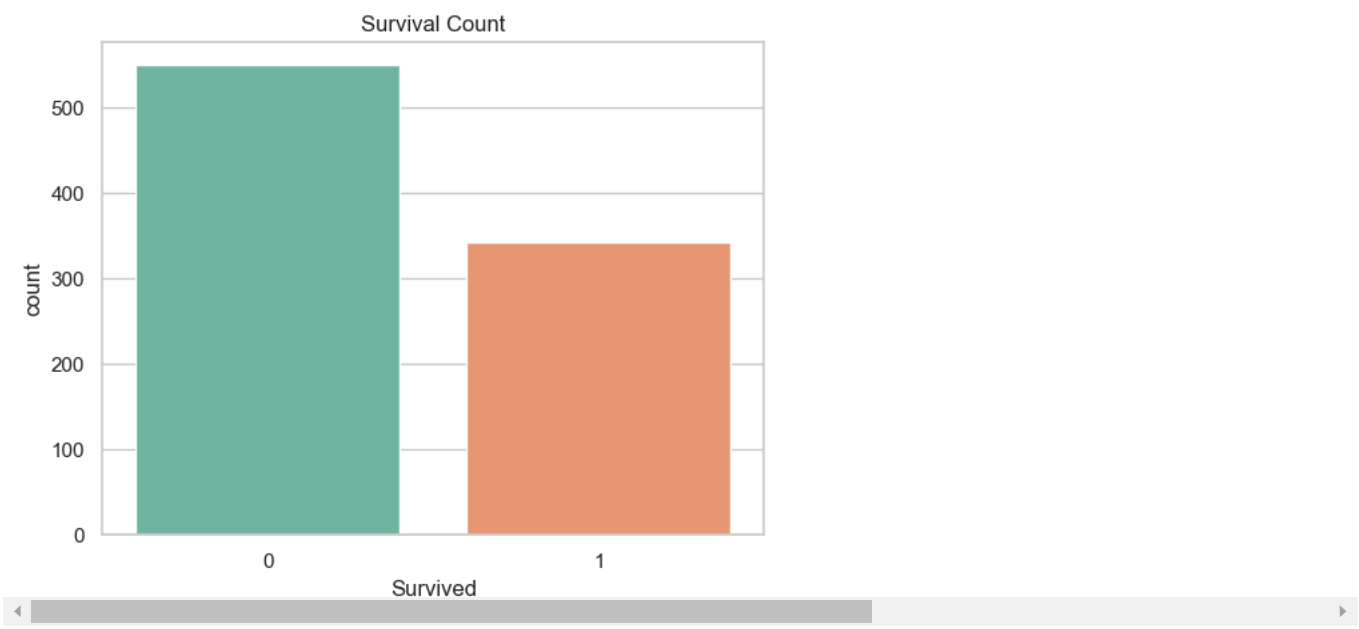
```
# Optional Drop 'Cabin' column
train = train.drop('Cabin', axis=1)
```

3. Univariate Analysis

```
# Plot histograms for numerical features
train[['Age','Fare','SibSp','Parch']].hist(bins=20, figsize=(12,8),color='#72b6a1')
plt.show()
```

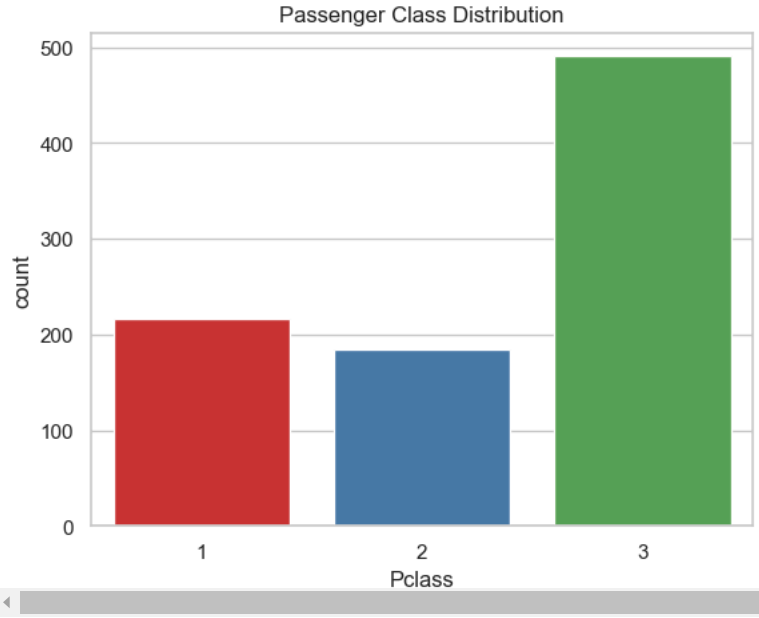



```
## Survived Count
sns.countplot(data=train, x='Survived', hue='Survived', palette='Set2', legend=False)
plt.title('Survival Count')
plt.show()
```

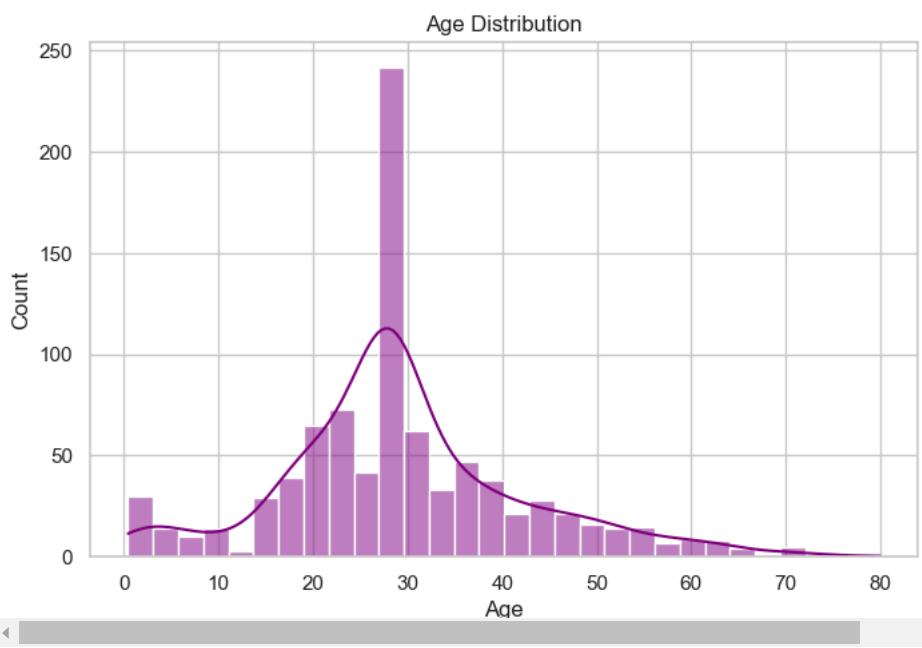



```
## Gender Distribution
sns.countplot(data=train, x='Sex', hue='Sex', palette={'male': '#72b6a1', 'female': '#e99675'}, legend=False)
plt.title('Gender Distribution')
plt.show()
```

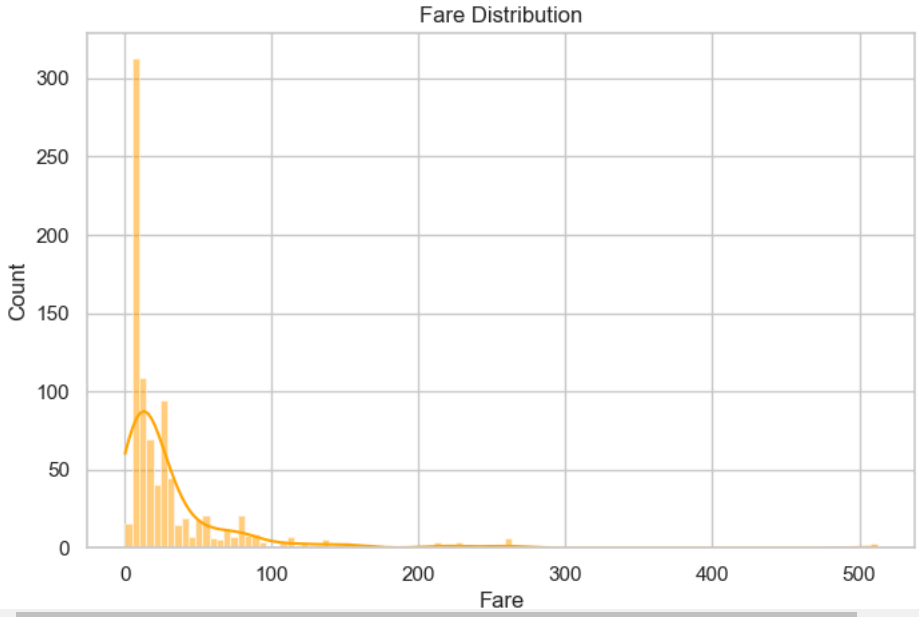

```
## Passenger Class
sns.countplot(data=train, x='Pclass', hue='Pclass', palette='Set1', legend=False)
plt.title('Passenger Class Distribution')
plt.show()
```




```
## Age Distribution
plt.figure(figsize=(8,5))
sns.histplot(train['Age'].dropna(), kde=True, color='purple')
plt.title('Age Distribution')
plt.show()
```

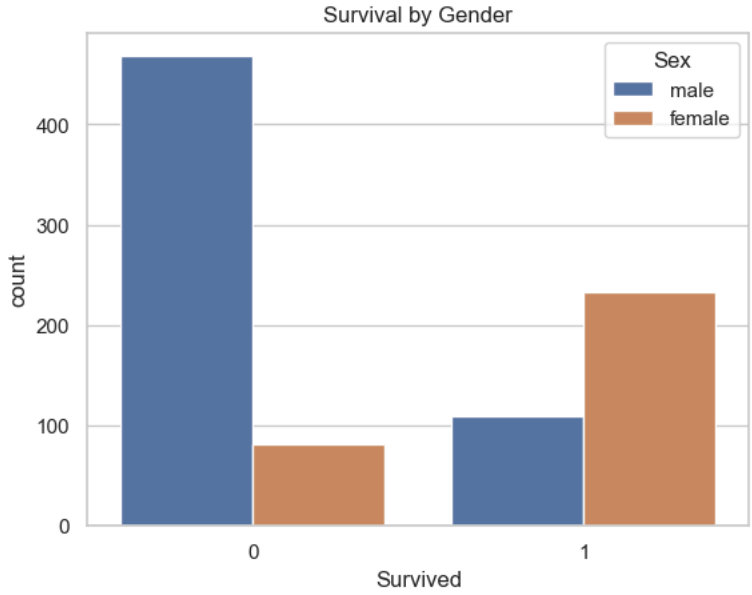


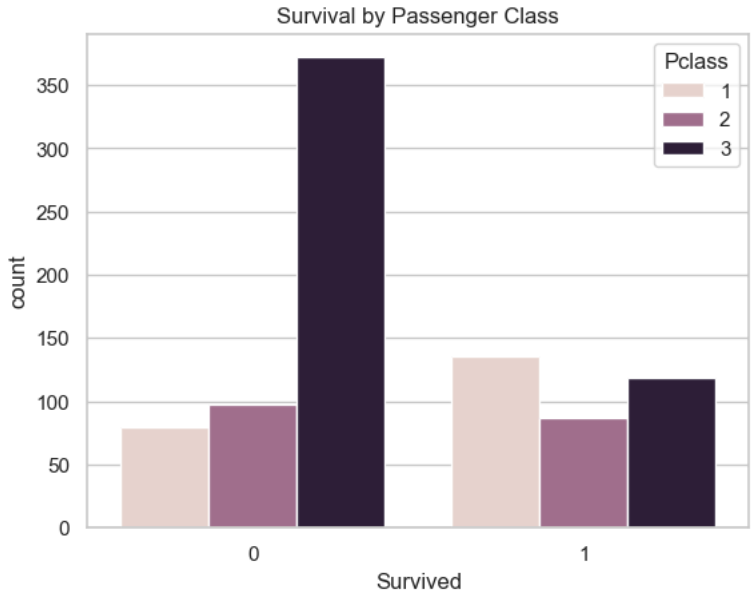

```
## Fare Distribution
plt.figure(figsize=(8,5))
sns.histplot(train['Fare'], kde=True, color='orange')
plt.title('Fare Distribution')
plt.show()
```

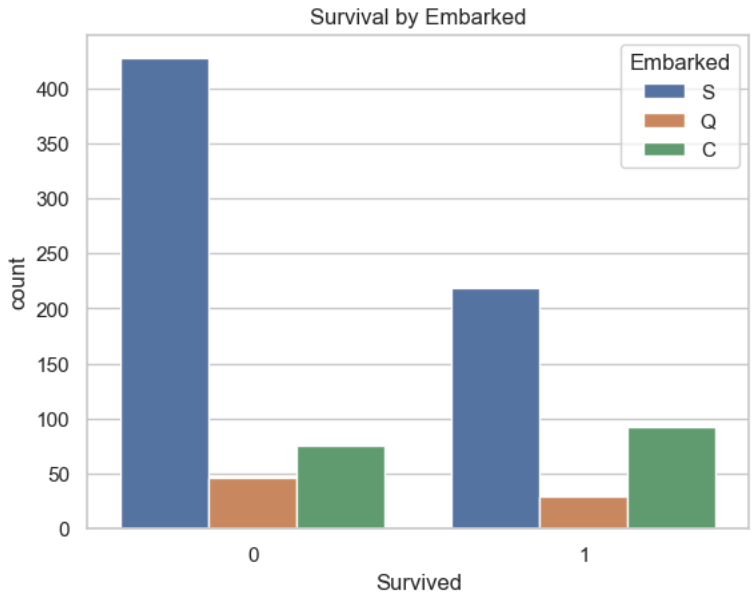


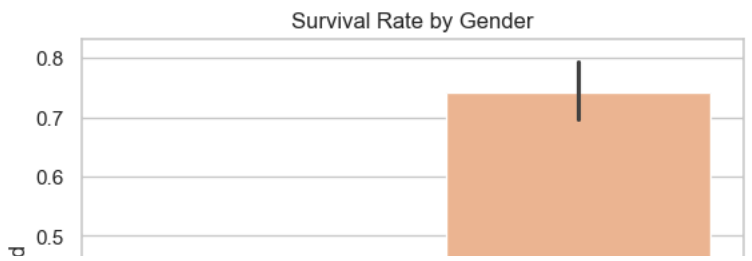
4. Bivariate Analysis

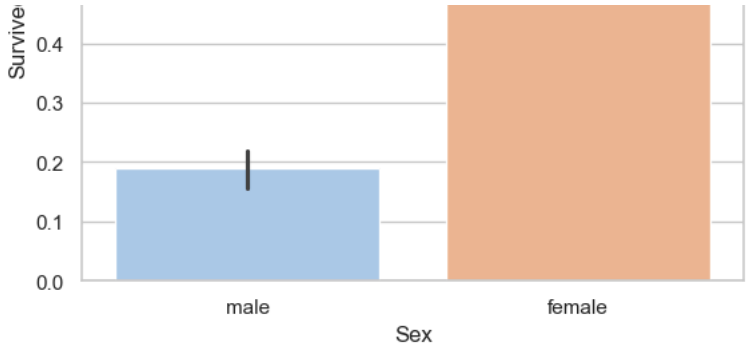
```
# 1. Categorical vs. Survived (Countplots)
# Survival by Gender
sns.countplot(x='Survived', hue='Sex', data=train)
plt.title('Survival by Gender')
plt.show()

# Survival by Passenger Class
sns.countplot(x='Survived', hue='Pclass', data=train)
plt.title('Survival by Passenger Class')
plt.show()

# Survival by Embarked
sns.countplot(x='Survived', hue='Embarked', data=train)
plt.title('Survival by Embarked')
plt.show()

# 2. Survival Rate by Gender (Barplot)
# Now using hue='Sex' to avoid FutureWarning
sns.barplot(data=train, x='Sex', y='Survived', hue='Sex', palette='pastel', legend=False)
plt.title('Survival Rate by Gender')
plt.show()

# 3. Survival Rate by Passenger Class (Barplot)
# Now using hue='Pclass' to avoid FutureWarning
sns.barplot(data=train, x='Pclass', y='Survived', hue='Pclass', palette='muted', legend=False)
plt.title('Survival Rate by Passenger Class')
plt.show()

# 4. Age vs Survival (Histogram)
plt.figure(figsize=(10,6))
sns.histplot(data=train, x='Age', hue='Survived', multiple='stack', palette='cool')
plt.title('Age vs Survival')
plt.show()
```

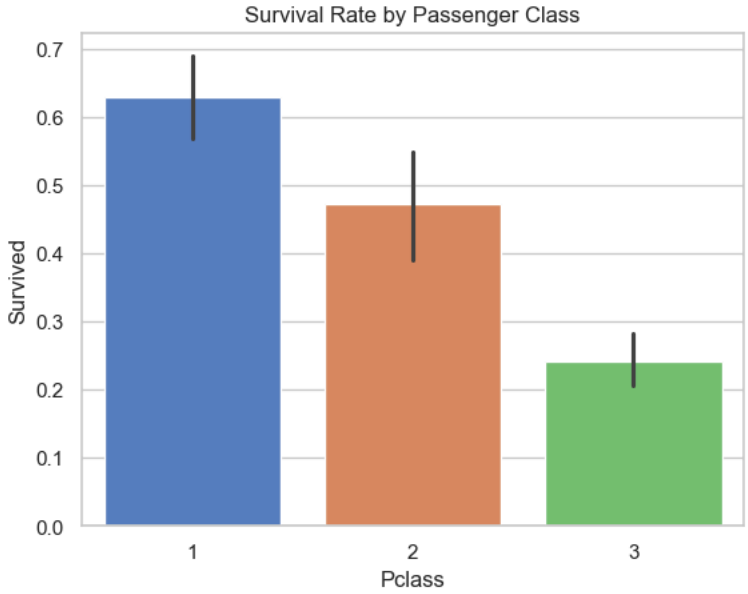

Survival by Gender
Sex
male
female
count
Survived
0
1
0
100
200
300
400

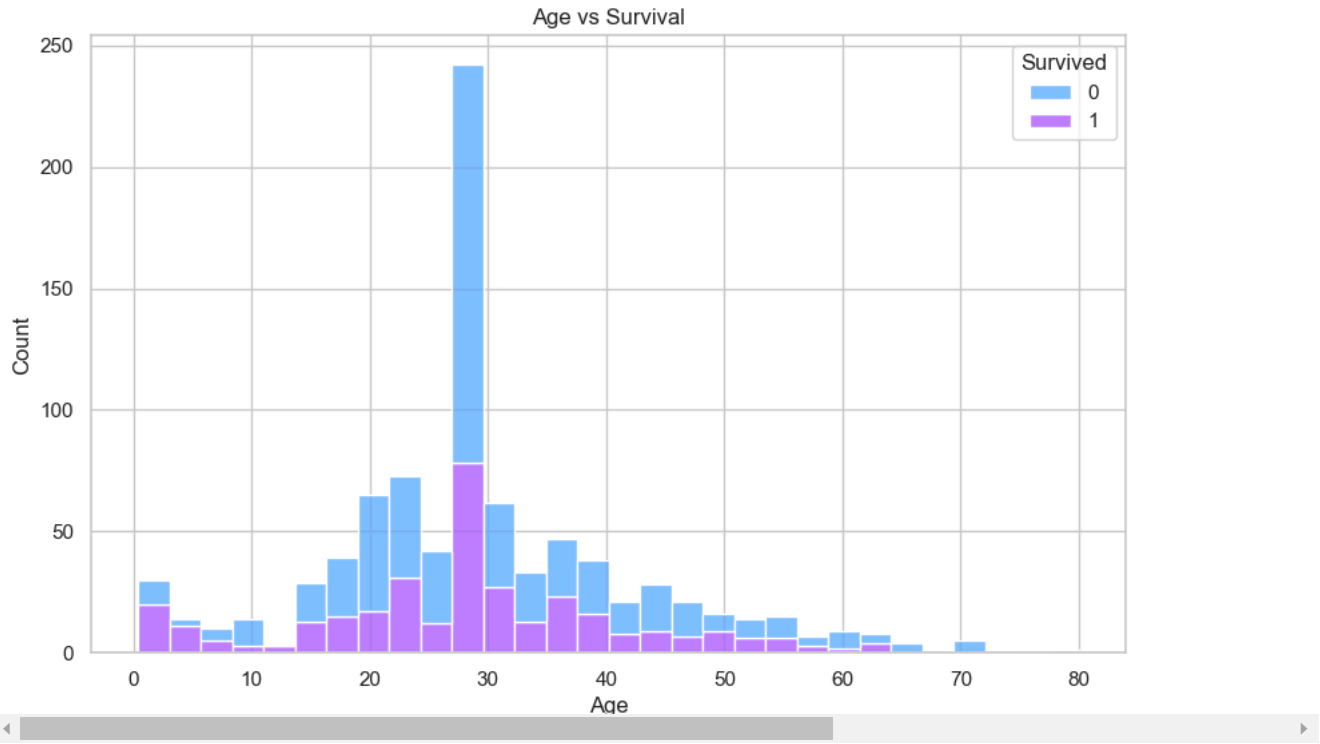

Survival by Passenger Class
Pclass
1
2
3
count
Survived
0
1
0
50
100
150
200
250
300
350


Survival by Embarked
Embarked
S
Q
C
count
Survived
0
1
0
50
100
150
200
250
300
350
400


Survival Rate by Gender
0.5
0.6
0.7
0.8

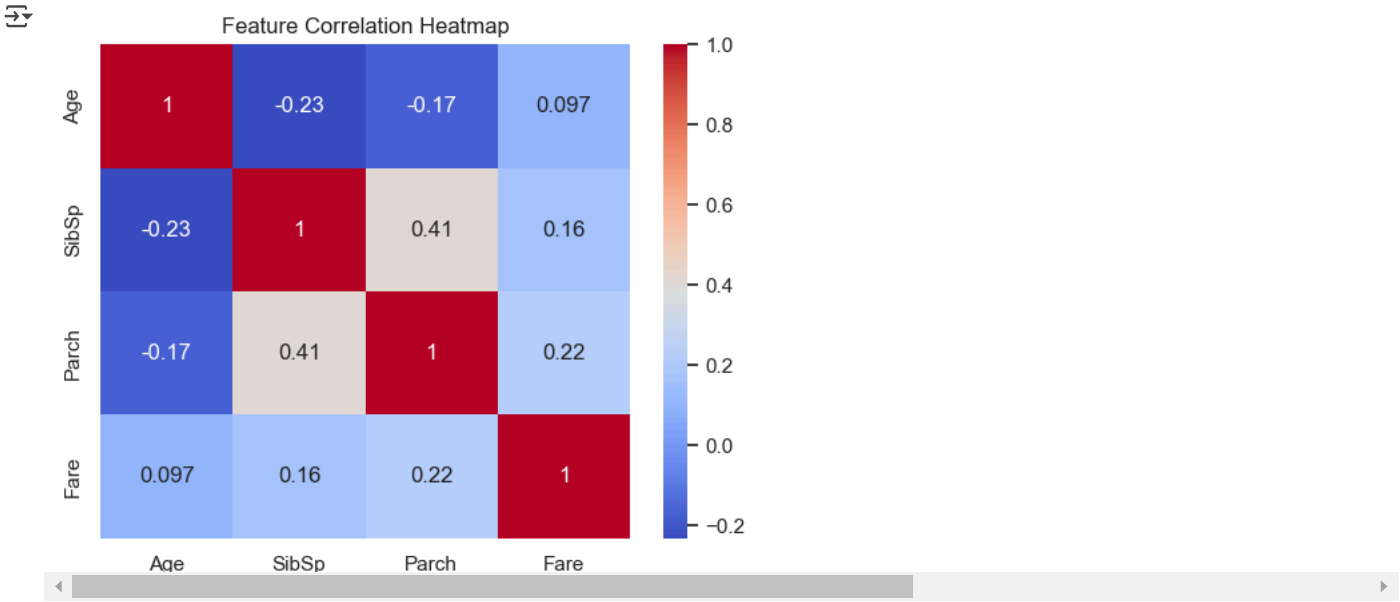

Survive
0.4
0.3
0.2
0.1
0.0
male
female
Sex


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

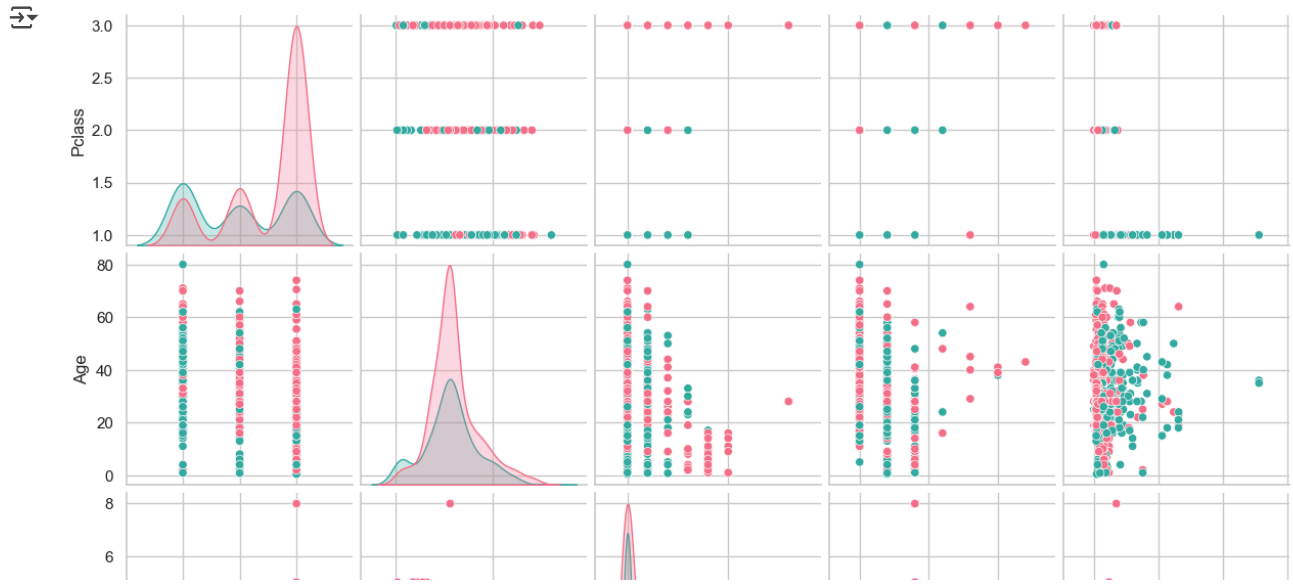

Age vs Survival
Survived
0
1
Count
250
200
150
100
50
0
0
10
20
30
40
50
60
70
80
Age

5. Multivariate Analysis

```
## Heatmap (Correlation)
corr_matrix = train[['Age','SibSp','Parch','Fare']].corr()
sns.heatmap(corr_matrix, annot = True, cmap = 'coolwarm')
plt.title('Feature Correlation Heatmap')
plt.show()
```




```
sns.pairplot(train[['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']], hue='Survived', palette='husl')
plt.show()
```




```
# FacetGrid for Survival based on Pclass and Sex
sns.catplot(
    data=train,
    x="Sex",
    kind="count",
    col="Pclass",
    order=["female", "male"],
    hue="Sex",
    palette="coolwarm",
    legend=False
)
plt.suptitle("Survival Count Based on Sex and Passenger Class", fontsize=16)
plt.show()
```