Titanic Data Analysis Project

✨ Exploratory Data Analysis using Pandas, Matplotlib & Seaborn

Name: SOUJANYA S P
Course: Data Analytics / Data Science
Tool Used: Python (Jupyter Notebook)
Date: April 14, 2025

This project involves a comprehensive exploratory data analysis of the Titanic dataset to understand the key factors that affected passenger survival. We use Python libraries to explore, visualize, and interpret the data for valuable insights.

Introduction

This Exploratory Data Analysis (EDA) is performed on the Titanic dataset to uncover patterns and relationships between passenger attributes and survival. The goal is to understand the data using various descriptive statistics and visualizations.

We use Pandas for data manipulation, and Matplotlib and Seaborn for data visualization. The analysis includes univariate, bivariate, and multivariate techniques to find trends, correlations, and key insights that help us interpret the survival factors of Titanic passengers.

```
# Step 1: Importing libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Set the theme for seaborn
sns.set_theme(style="whitegrid")

# Optional: For displaying plots inside the notebook
%matplotlib inline
```

```
from google.colab import files

# Prompt the user to upload a file
uploaded = files.upload()
```

Choose Files train.csv

- **train.csv**(text/csv) - 61194 bytes, last modified: 4/14/2025 - 100% done

```
# Step 2: Load the dataset
df = pd.read_csv('train.csv')

# Display the first few rows
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
# Shape of the dataset
print("Shape of dataset:", df.shape)

# Check data types and non-null counts
df.info()
```

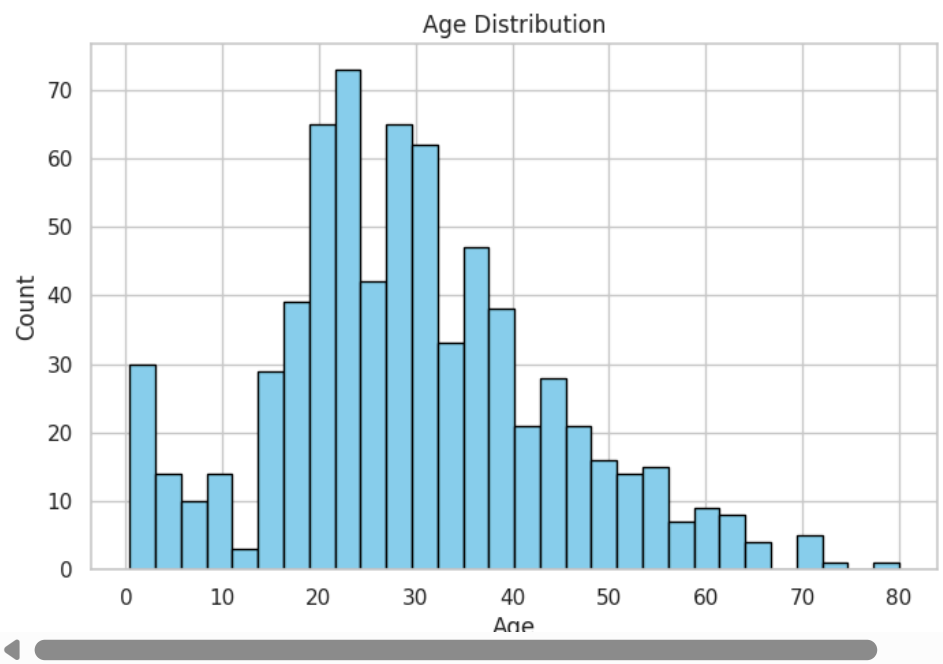
```
# Summary statistics for numeric columns
df.describe()

# Count of missing values in each column
df.isnull().sum()

# Count of survivors vs non-survivors
df['Survived'].value_counts()
```

```
Shape of dataset: (891, 12)
<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
```

Survived	count
0	549
1	342

dtype: int64

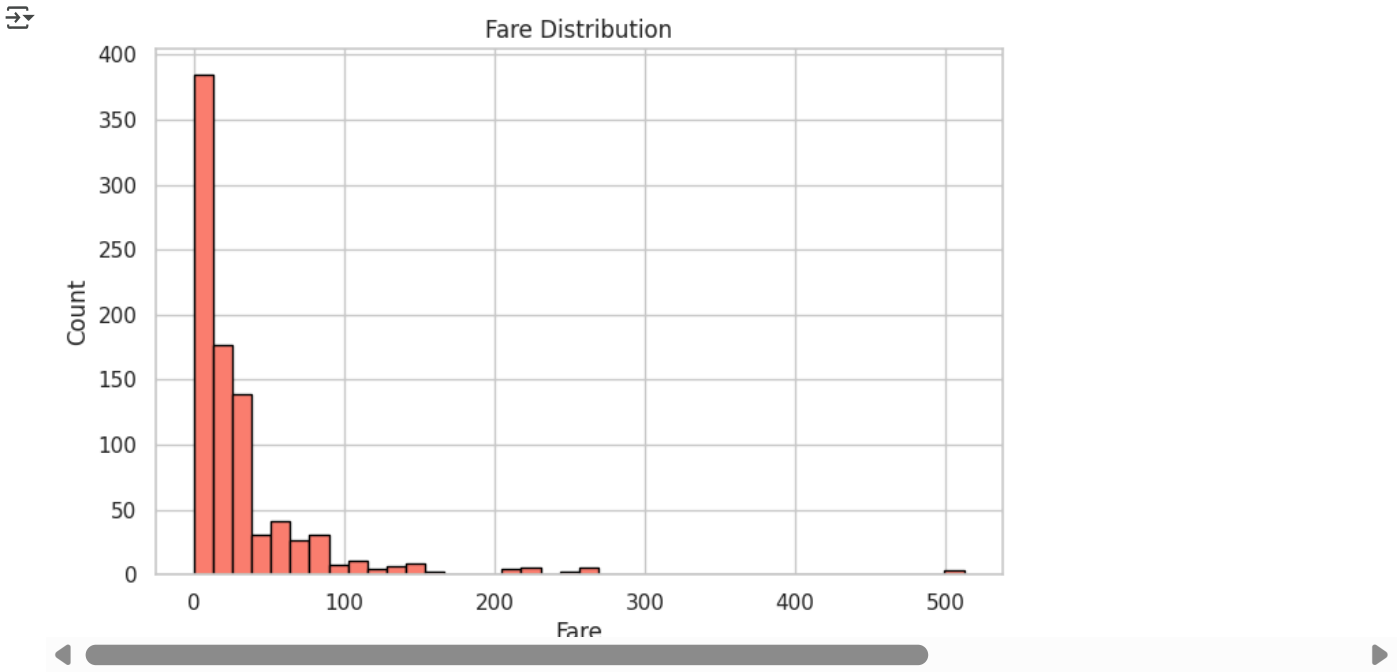
```
plt.figure(figsize=(8, 5))
df['Age'].hist(bins=30, color='skyblue', edgecolor='black')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```



📌 Observation: Younger passengers were more common.

```
plt.figure(figsize=(8, 5))
df['Fare'].hist(bins=40, color='salmon', edgecolor='black')
plt.title('Fare Distribution')
```

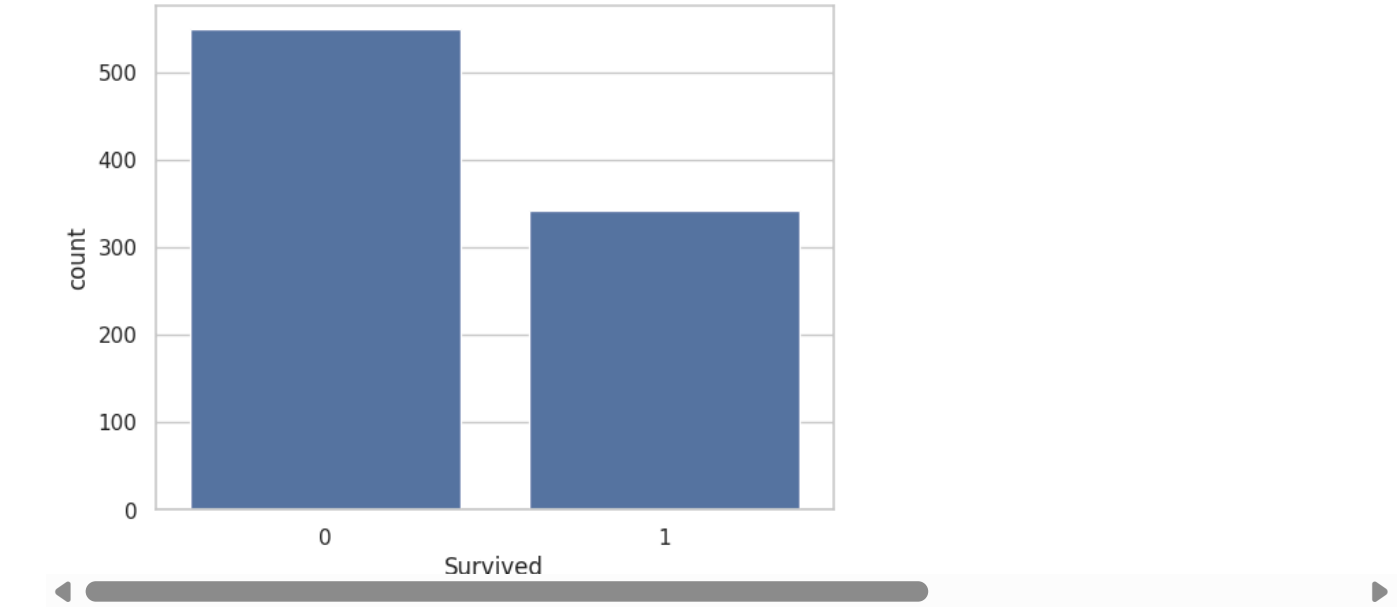
```
plt.xlabel('Fare')
plt.ylabel('Count')
plt.show()
```



📌 Observation: Most passengers paid low fares, with a few outliers paying high.

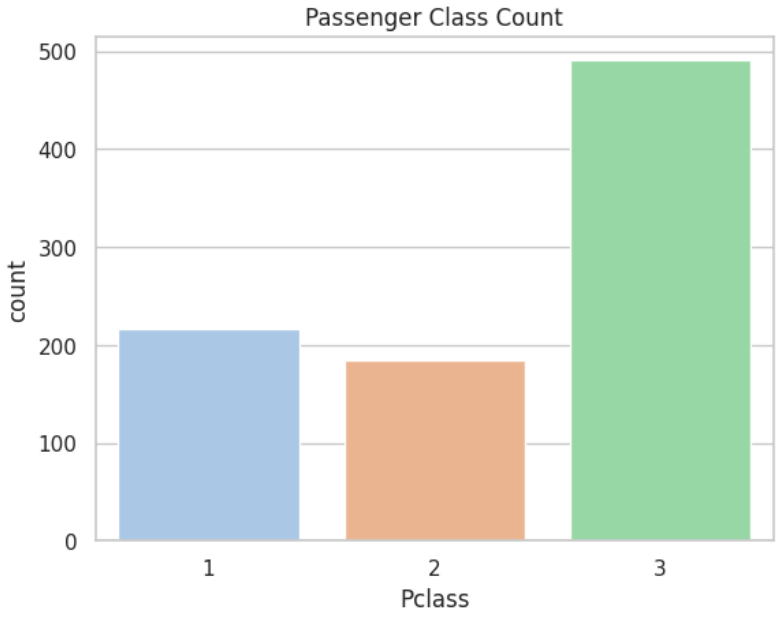
```
sns.countplot(x='Survived', data=df)
```

```
<Axes: xlabel='Survived', ylabel='count'>
```




```
sns.countplot(x='Pclass', data=df, palette='pastel')
plt.title('Passenger Class Count')
plt.show()
```

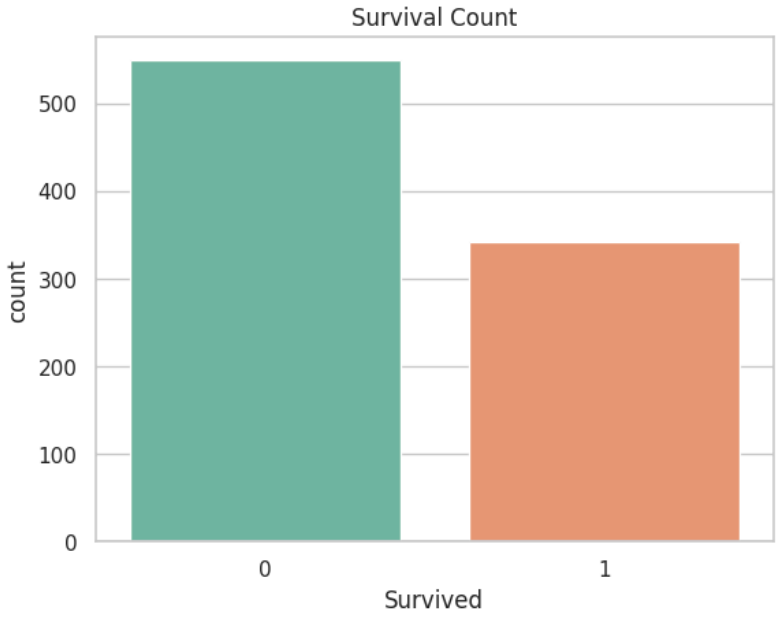
```
<ipython-input-8-93bf8ec33ce9>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set

  sns.countplot(x='Pclass', data=df, palette='pastel')
```



📌 Observation: Majority of passengers were in 3rd class.

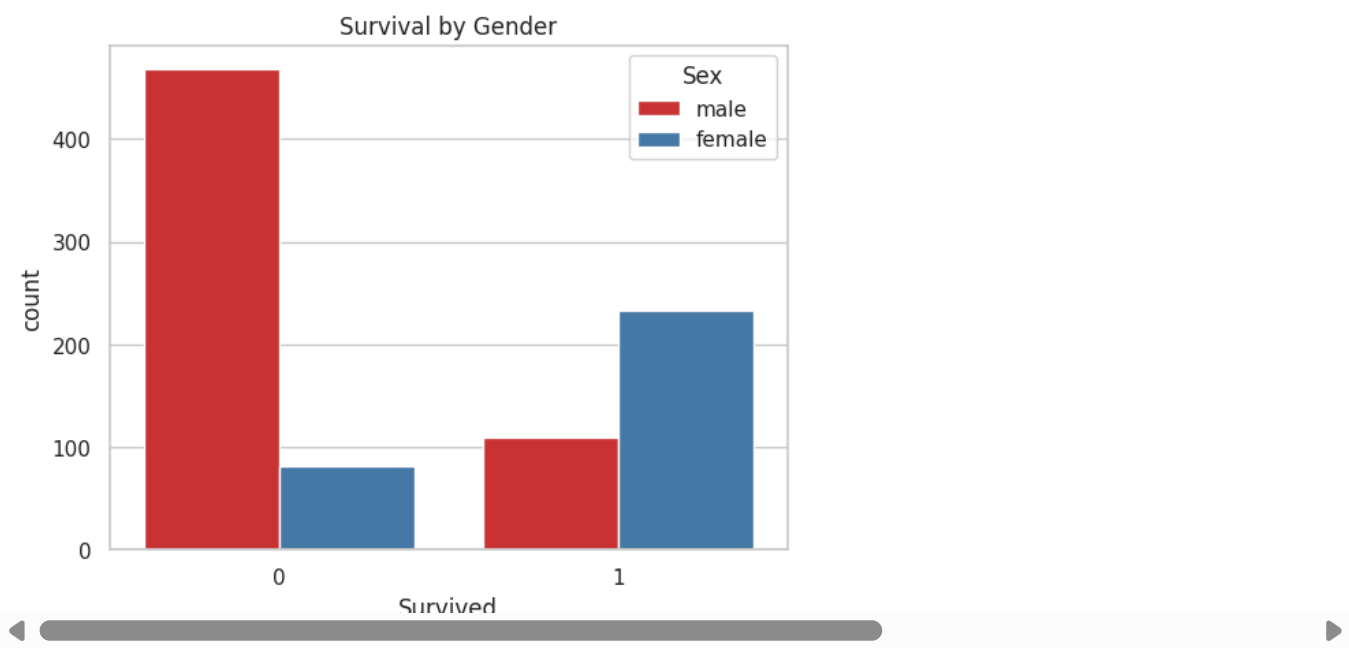
```
sns.countplot(x='Survived', data=df, palette='Set2')
plt.title('Survival Count')
plt.show()
```

```
<ipython-input-9-d6a60f00d962>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set

  sns.countplot(x='Survived', data=df, palette='Set2')
```



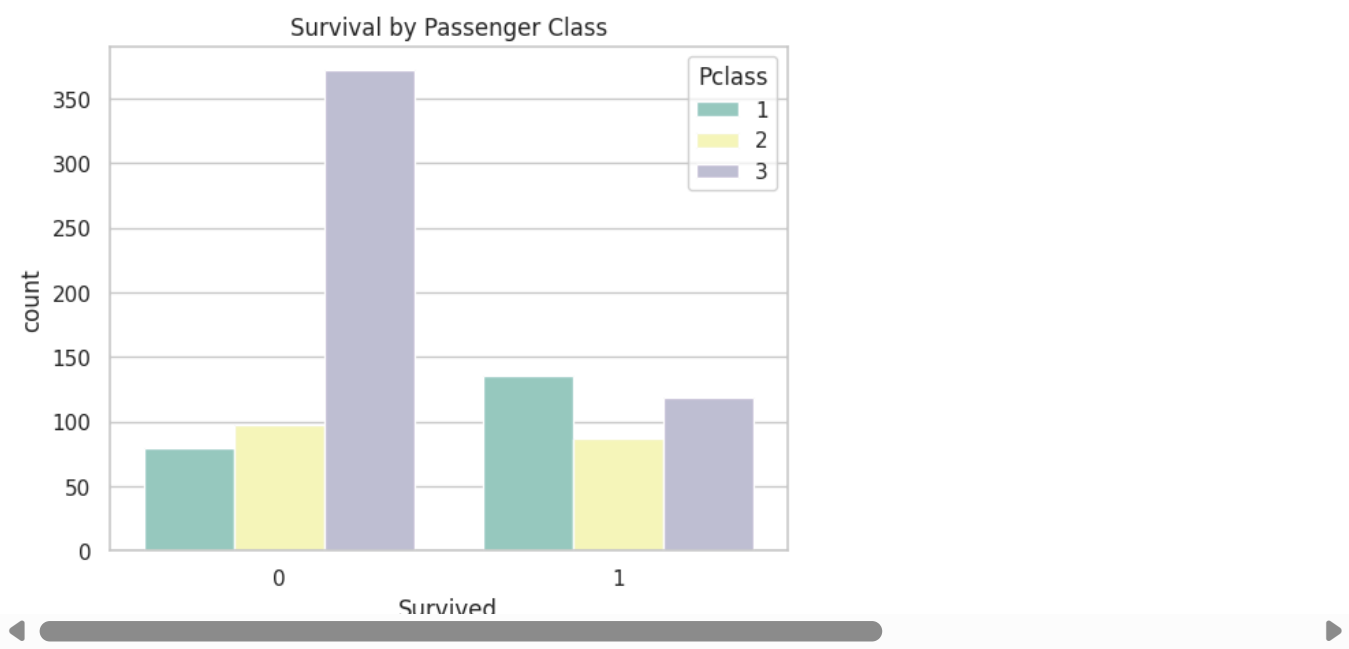
📌 Observation: More people did not survive than those who did.

```
sns.countplot(x='Survived', hue='Sex', data=df, palette='Set1')
plt.title('Survival by Gender')
plt.show()
```



📌 Observation: Females had a much higher survival rate than males.

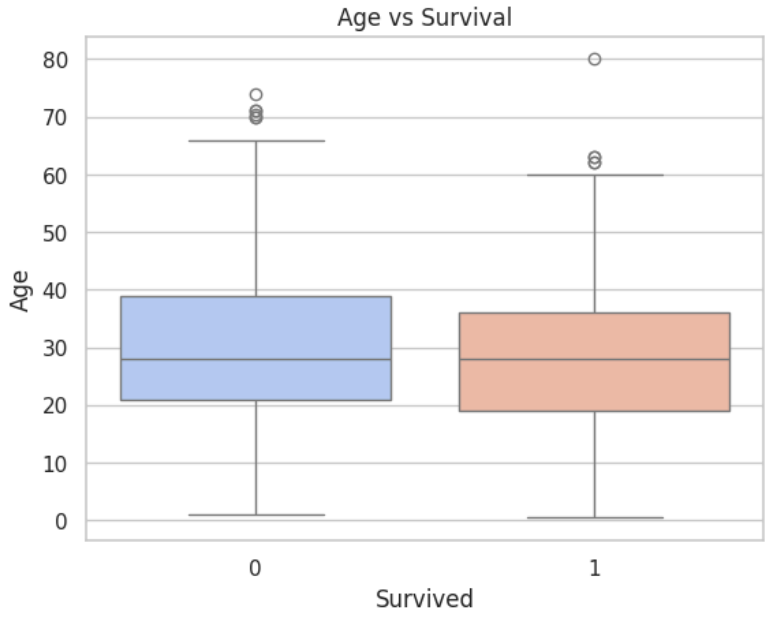
```
sns.countplot(x='Survived', hue='Pclass', data=df, palette='Set3')
plt.title('Survival by Passenger Class')
plt.show()
```



📌 Observation: Passengers in 1st class survived more than those in 3rd.

```
sns.boxplot(x='Survived', y='Age', data=df, palette='coolwarm')
plt.title('Age vs Survival')
plt.show()
```

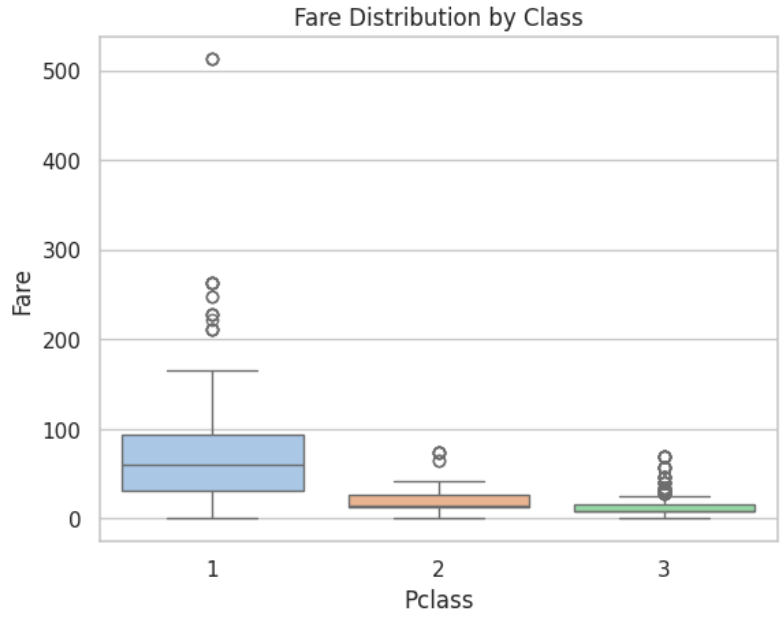
```
<ipython-input-12-922f5eeed45b>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set

  sns.boxplot(x='Survived', y='Age', data=df, palette='coolwarm')
```



📌 Observation: Younger passengers had slightly higher survival.

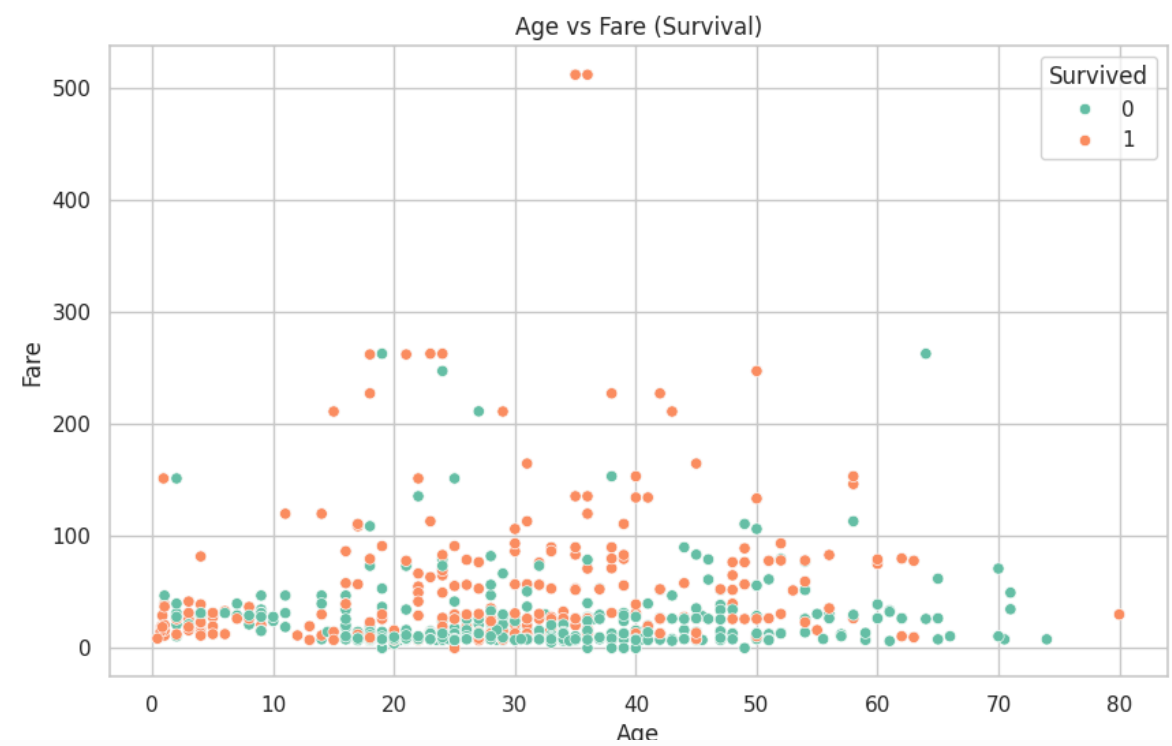
```
sns.boxplot(x='Pclass', y='Fare', data=df, palette='pastel')
plt.title('Fare Distribution by Class')
plt.show()
```

```
<ipython-input-16-f303b85cb13b>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set

  sns.boxplot(x='Pclass', y='Fare', data=df, palette='pastel')
```



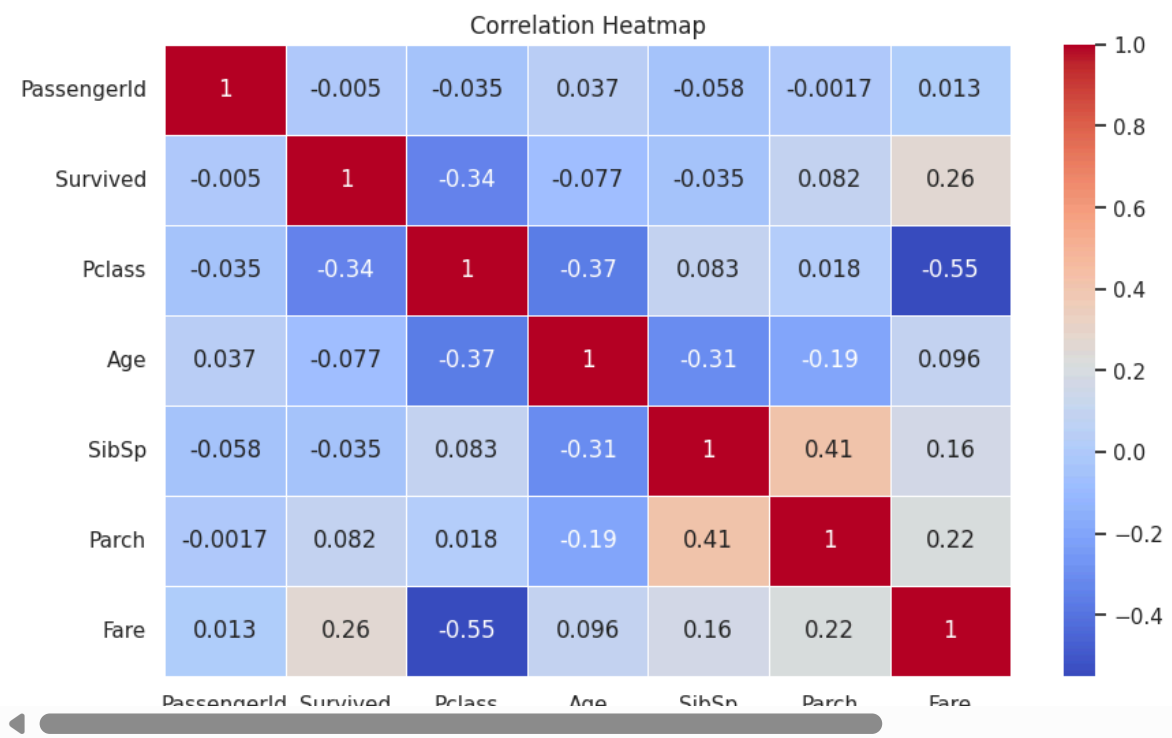
📌 Observation: Higher class passengers paid more.

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=df, palette='Set2')
plt.title('Age vs Fare (Survival)')
plt.show()
```



📌 Observation: High fare and middle-aged passengers had higher survival chances.

```
# Keep only numeric columns for correlation
numeric_df = df.select_dtypes(include='number')

# Now create the heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```



📌 Observation:

- Fare and Pclass show a negative correlation (makes sense – higher class means higher fare).
- Survived has positive correlation with Fare and slight negative with Pclass.

```
# Optional: drop NaN values to avoid errors
```

```
pairplot_df = df[['Survived', 'Age', 'Fare', 'Pclass']].dropna()

sns.pairplot(pairplot_df, hue='Survived', palette='husl')
plt.suptitle('Pairplot of Age, Fare, and Class by Survival', y=1.02)
plt.show()
```

📌 Observation:

- Higher fare is more associated with survival.
- Many younger and middle-aged people survived.
- 1st class passengers show a dense cluster among survivors.

🪦 **Conclusion**

From the analysis, we observed that several key factors influenced survival probability on the Titanic:

- Gender played a major role — females had significantly higher survival chances.