TITANIC DATA SET ANALYSIS

Titanic

HAVING A QUICK VIEW OF DATASET

Introduction to Titanic Dataset The Titanic dataset provides information about passengers aboard the Titanic, including details on their demographics, ticket fare, and survival status. The primary goal of this analysis is to explore the relationships between various features and the likelihood of survival during the disaster.

Key Columns in the Dataset: PassengerId: Unique ID for each passenger. Survived: Whether the passenger survived (1) or not (0). Pclass: Passenger's class (1st, 2nd, or 3rd). Name: Passenger's full name. Sex: Passenger's gender (male or female). Age: Passenger's age (some missing values). SibSp: Number of siblings or spouses aboard the Titanic. Parch: Number of parents or children aboard the Titanic. Ticket: Ticket number. Fare: Fare paid by the passenger (one missing value). Cabin: Cabin number (many missing values). Embarked: Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton).

What We'll Do in This Analysis:

Data Exploration: We'll load the dataset and perform a preliminary exploration to understand its structure, check for missing values, and summarize key statistics. Data Cleaning: We'll handle missing values, particularly in columns like Age, Embarked, and Cabin, and prepare the dataset for further analysis. Feature Analysis: We'll explore key features such as passenger class, gender, age, and fare, and analyze how they correlate with survival. Visualization: We will create visual representations of the data to uncover trends, such as: Survival rates by gender and class Age and fare distribution. Correlation heatmaps of numeric features. Survival rates based on family size (using SibSp and Parch). The overall objective is to find the trends and relationships that affected survival during the Titanic disaster.

```python
#importing important python libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# importing and reading the csv file
df= pd.read_csv(r'C:\Users\Windows\Desktop\projects\tested.csv')
```

EDA

```python
df.head()
```

```
   PassengerId  Survived  Pclass  \
0          892         0       3
1          893         1       3
2          894         0       2
3          895         0       3
4          896         1       3

                                           Name     Sex   Age  SibSp
Parch  \
0                              Kelly, Mr. James    male  34.5      0
0
1              Wilkes, Mrs. James (Ellen Needs)  female  47.0      1
0
2                     Myles, Mr. Thomas Francis    male  62.0      0
0
3                              Wirz, Mr. Albert    male  27.0      0
0
4  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0      1
1

    Ticket     Fare Cabin Embarked
0   330911   7.8292   NaN        Q
1   363272   7.0000   NaN        S
2   240276   9.6875   NaN        Q
3   315154   8.6625   NaN        S
4  3101298  12.2875   NaN        S
```

```
df.shape
```

```
(418, 12)
```

```
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
```

```python
# checking for null values
df.isnull().sum().sort_values(ascending= False)
```

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

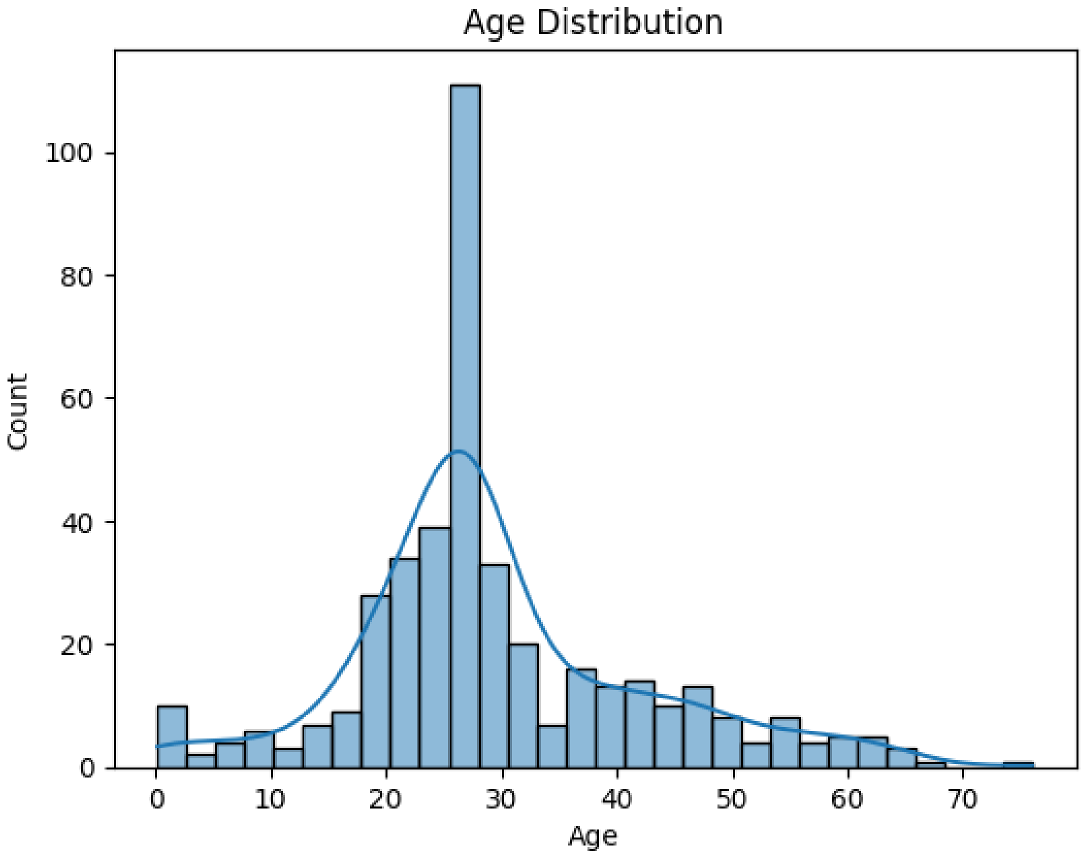
```python
# Handling Missing Values
# Filling missing Age values with the median
df['Age']=df['Age'].fillna(df['Age'].median())

# Droping the Cabin column
df.drop(columns=['Cabin'], inplace=True)

df['Embarked']=df['Embarked'].fillna(df['Embarked'].mode()[0])

#Handling Categorical Data
#Coverting sex to numeric
df['Sex'] = df['Sex'].map({'male': 0, 'female': 1})

# Embarked to numeric
df['Embarked'] = df['Embarked'].map({'C': 0, 'Q': 1, 'S': 2})
```

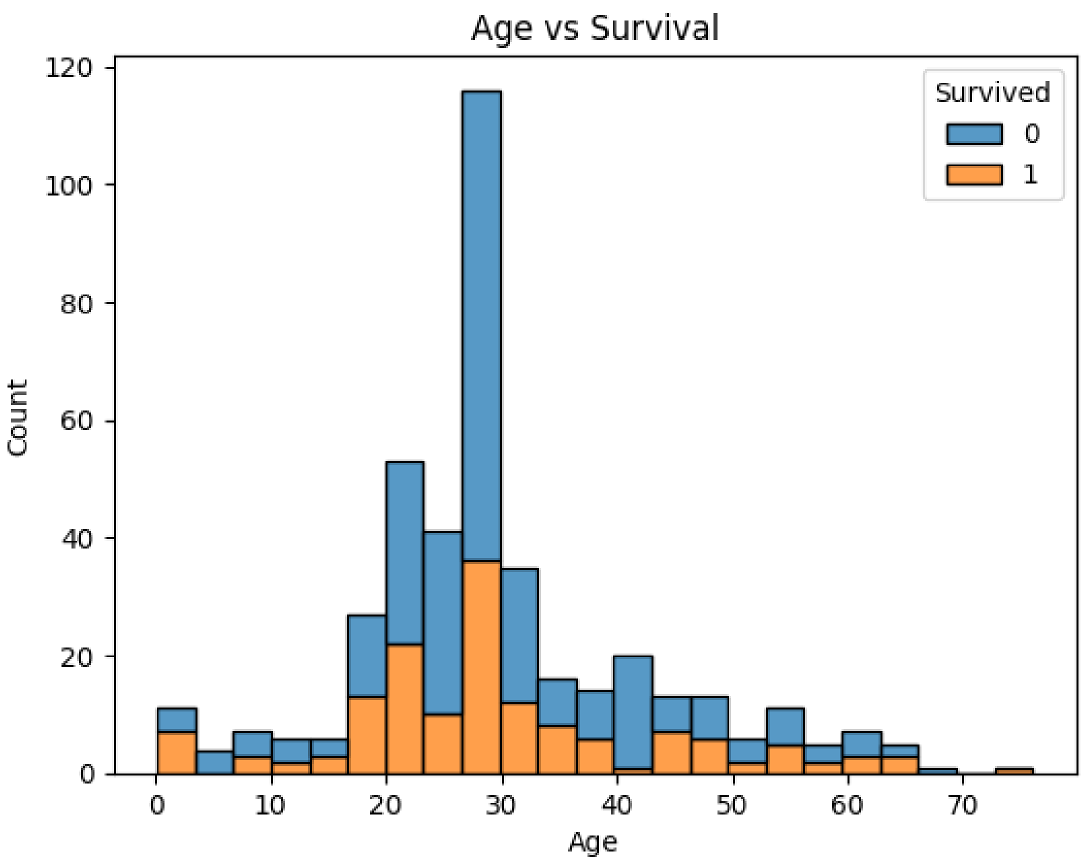
What the data tells ?? (Visualisation)

Age Distribution and Survival Rate

```python
sns.histplot(df['Age'].dropna(), bins=30, kde=True)
plt.title("Age Distribution")
plt.show()

# Age vs. Survival
sns.histplot(data=df, x='Age', hue='Survived', multiple='stack', 
kde=False)
```

```
plt.title("Age vs Survival")
plt.show()
```

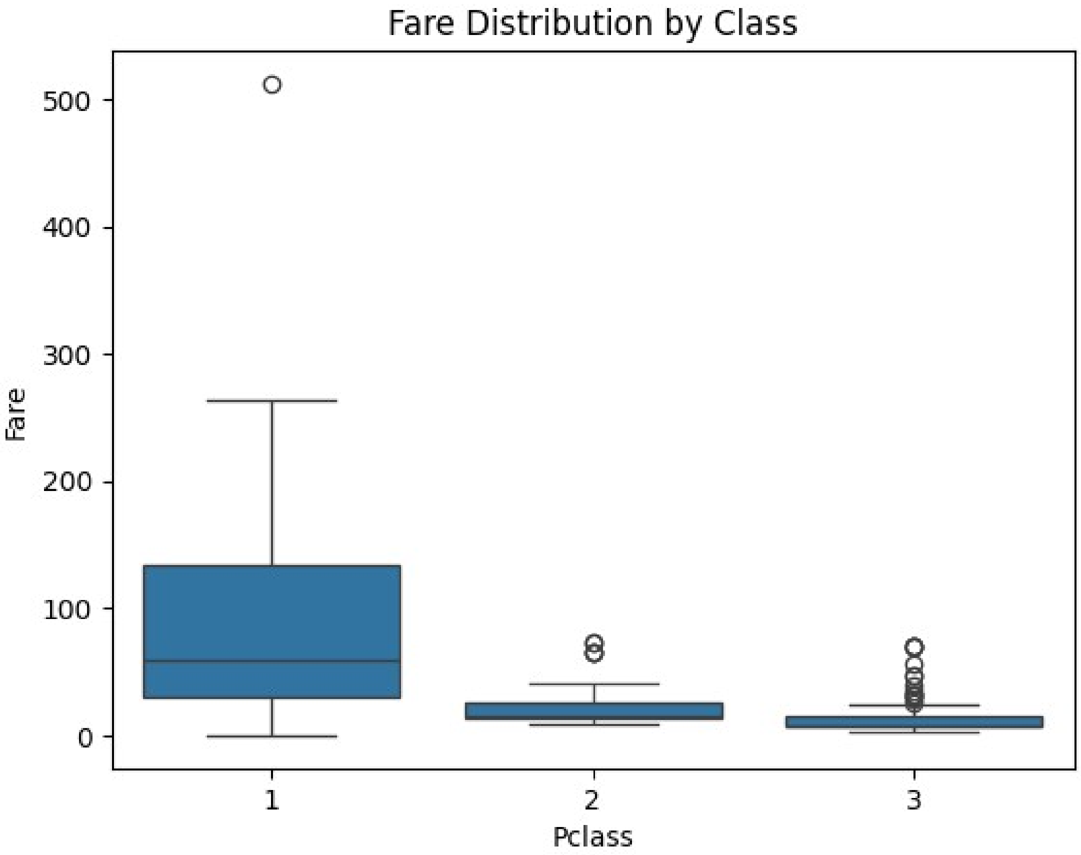




-Age Distribution Most passengers were between the ages of 20 and 40, with a noticeable peak around 30 years old. There's a long tail in the distribution, indicating there were both children and older passengers onboard. Age vs. Survival Younger children had higher survival rates, while passengers in the middle-age range (30-50) had lower survival rates. Children and younger passengers likely received more attention during rescue operations.

Fare Distribution by Class:

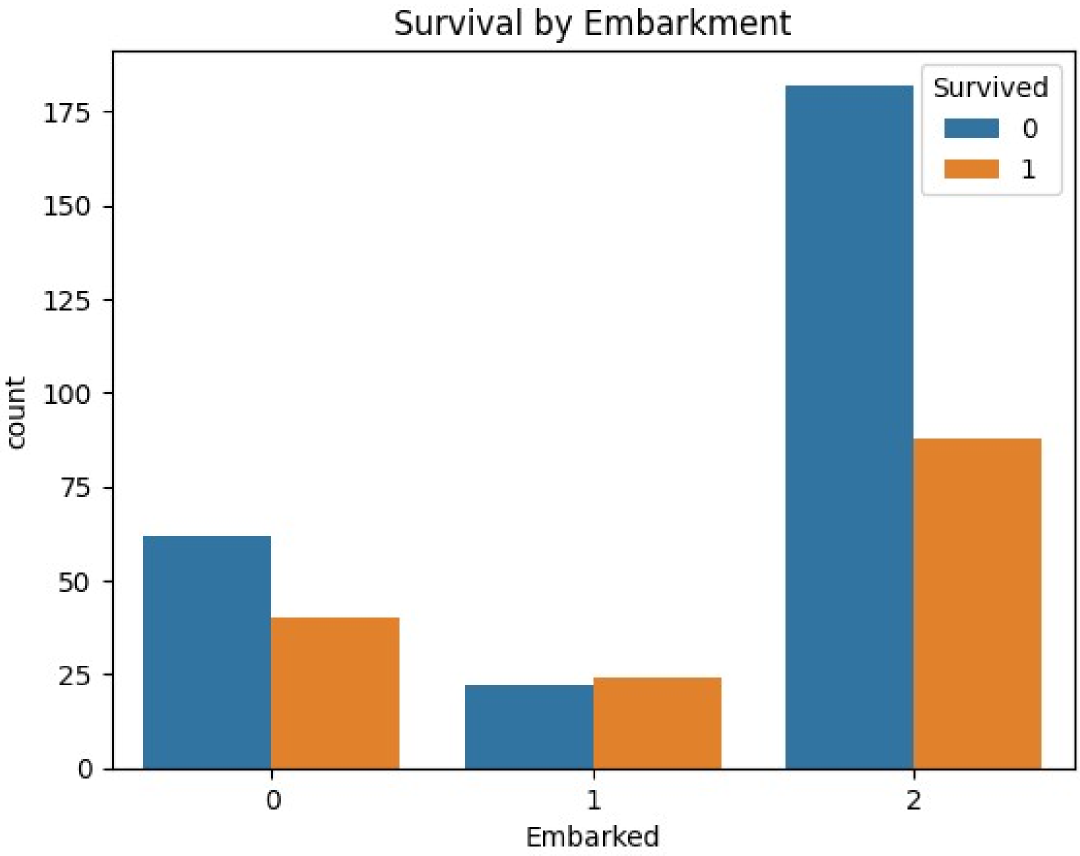
```
sns.boxplot(x='Pclass', y='Fare', data=df)
plt.title("Fare Distribution by Class")
plt.show()
```



The median fare for 1st class passengers was significantly higher than for those in 2nd and 3rd class. The fare distribution for 1st class is more spread out, indicating variability in ticket prices depending on the location of the cabins.

# Embarked vs Survival:

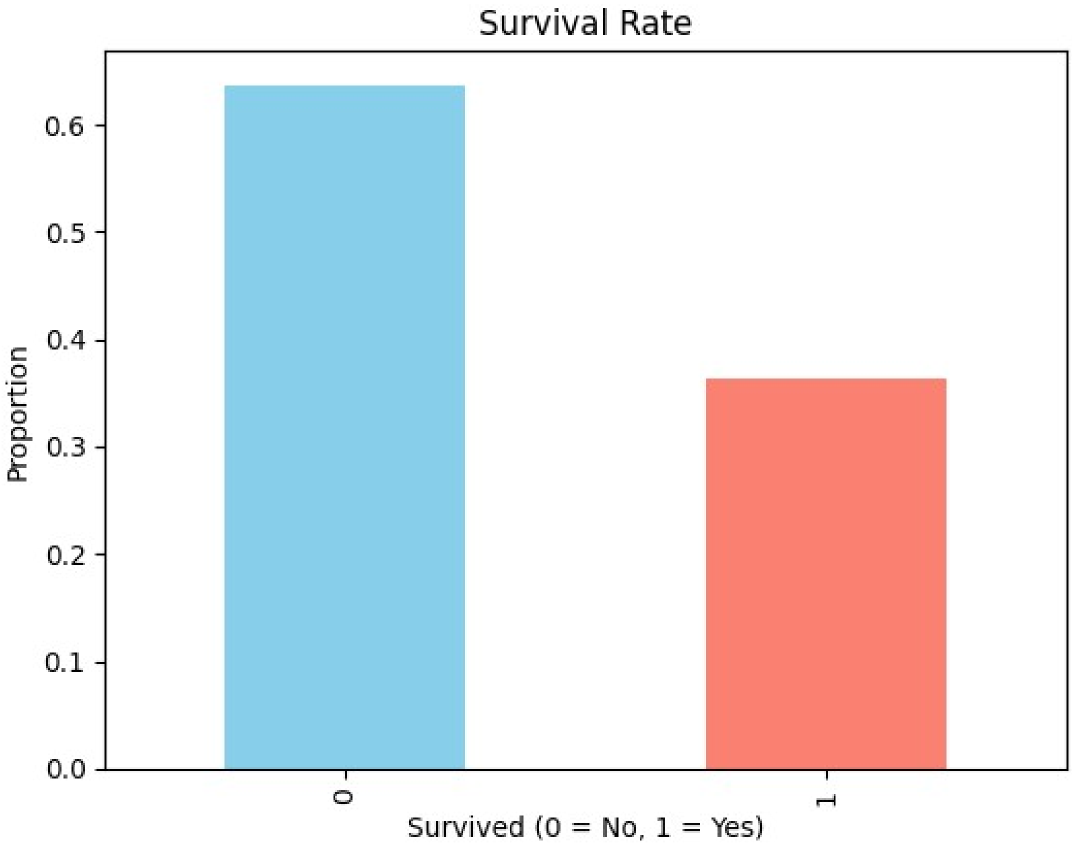
```
sns.countplot(x='Embarked', hue='Survived', data=df)
plt.title("Survival by Embarkment")
plt.show()
```

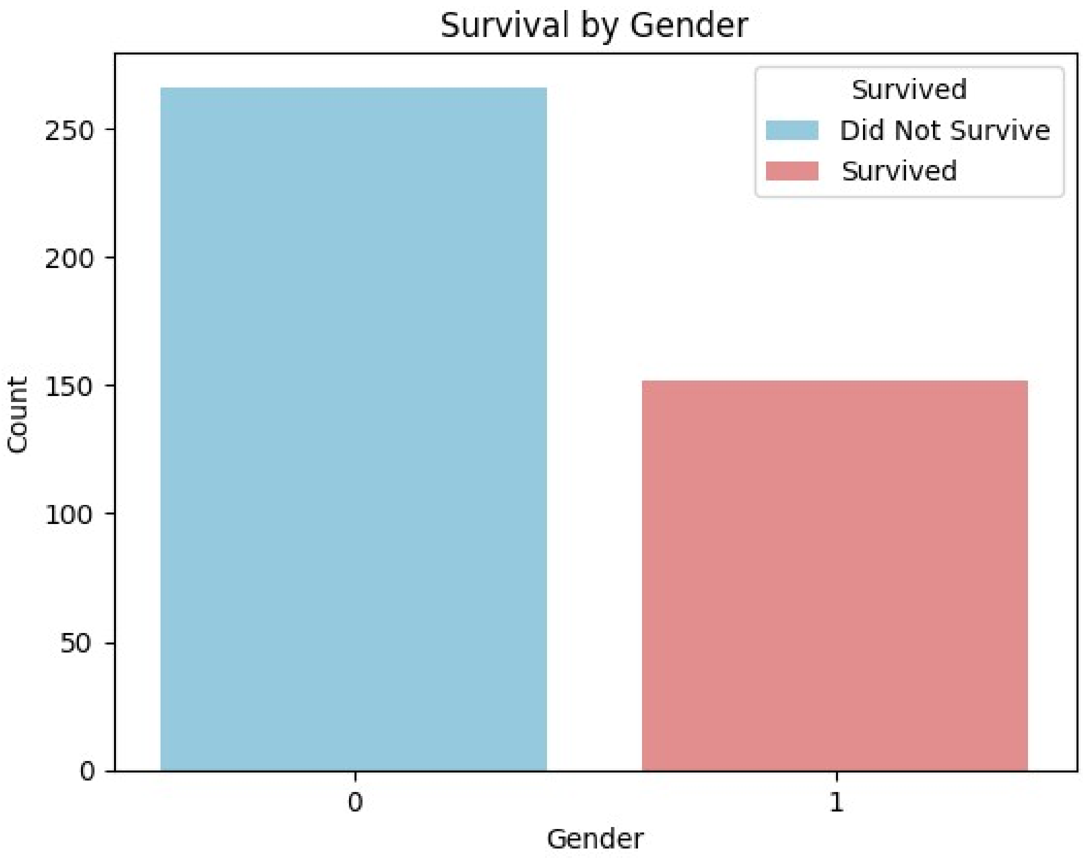


-Passengers who embarked from Cherbourg (C) had a higher survival rate compared to those from Southampton (S) and Queenstown (Q). This might suggest that the class distribution or priority access to lifeboats differed by embarkation point.

Survival Rate by Gender

```
df['Survived'].value_counts(normalize=True).plot(kind='bar',
color=['skyblue', 'salmon'])
plt.title("Survival Rate")
plt.xlabel("Survived (0 = No, 1 = Yes)")
plt.ylabel("Proportion")
plt.show()

# Gender-wise survival rate
sns.countplot(x='Sex', hue='Survived', data=df, palette=['skyblue',
'lightcoral'])
plt.title("Survival by Gender")
plt.xlabel("Gender")
plt.ylabel("Count")
plt.legend(title='Survived', loc='upper right', labels=['Did Not
Survive', 'Survived'])
plt.show()
```

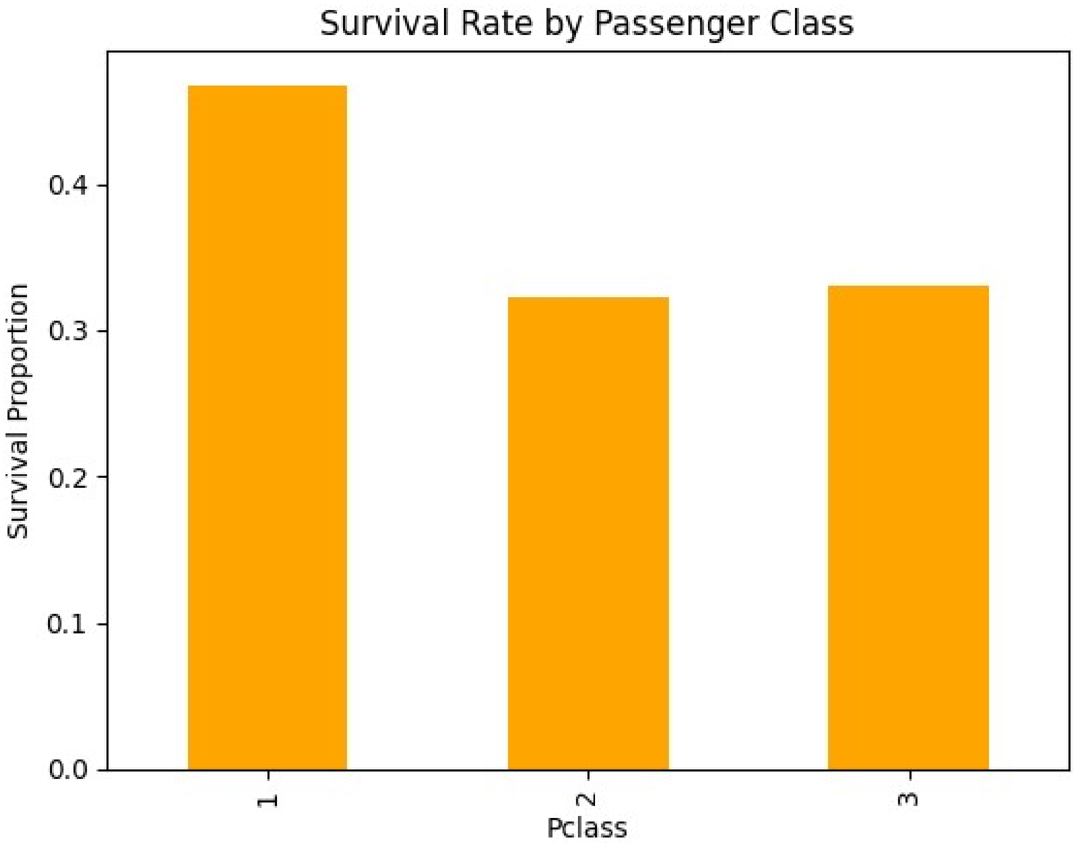

Survival Rate
Proportion
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
1
Survived (0 = No, 1 = Yes)



-The survival rate of passengers is approximately 38%. This shows that more than half of the passengers (about 62%) did not survive. -Females: Around 74% survival rate. Males: Around 18% survival rate.

Survival Rate by Passenger Class

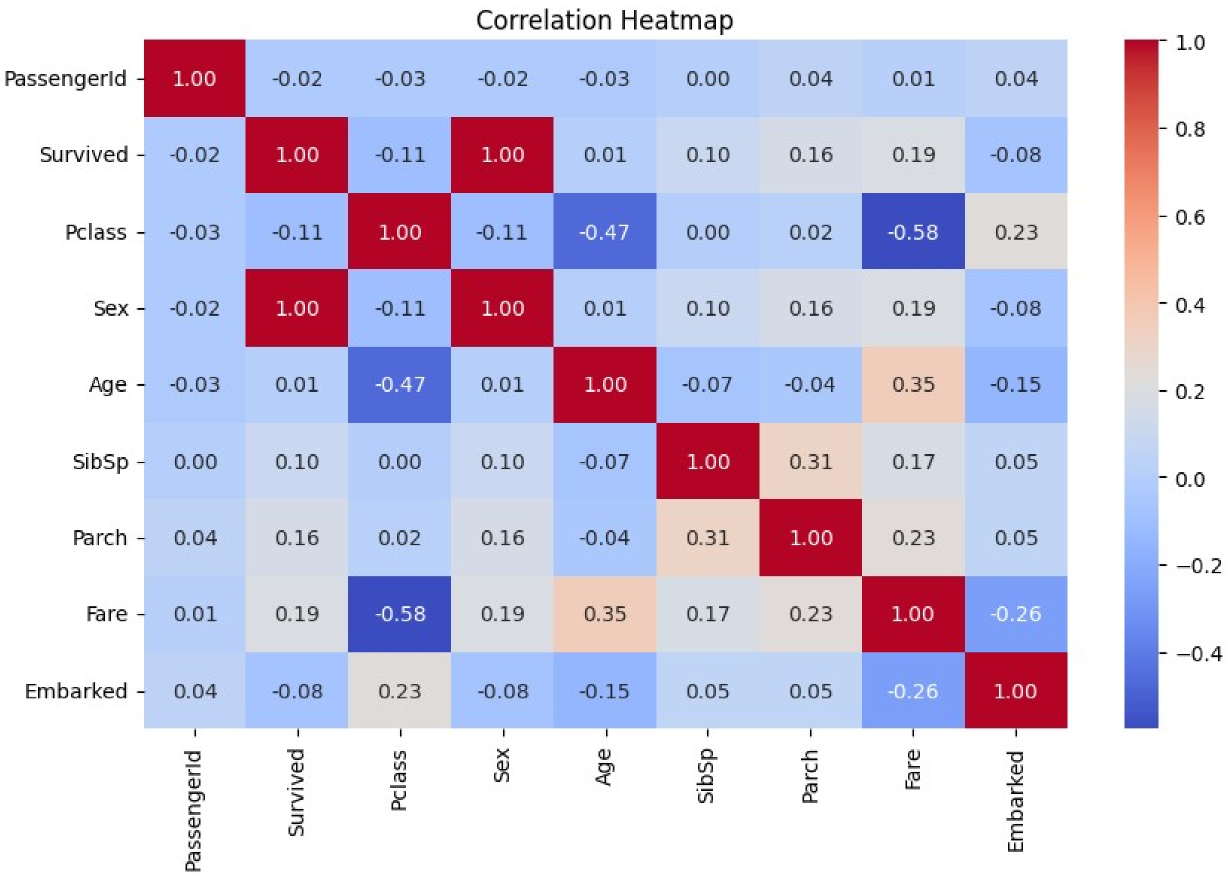
```
df.groupby('Pclass')['Survived'].mean().plot(kind='bar',
color='orange')
plt.title("Survival Rate by Passenger Class")
plt.ylabel("Survival Proportion")
plt.show()
```



Passengers in 1st class had a significantly higher survival rate (63%) than those in 3rd class (24%). This indicates that passengers in higher classes had better access to lifeboats or were prioritized.

Correlation Heatmap

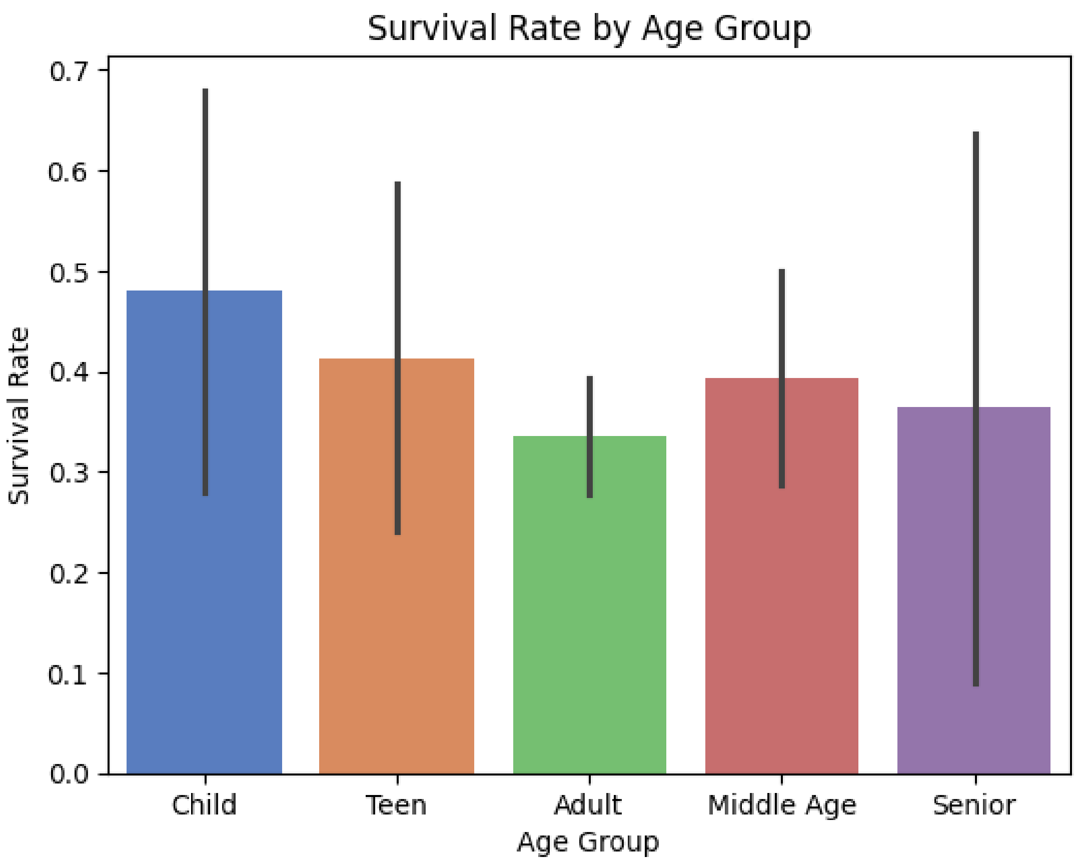
```
numeric_df = df.select_dtypes(include=[np.number])

# Generating the correlation heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```



Survival Rate by Age Group

```
df['AgeGroup'] = pd.cut(df['Age'], bins=[0, 12, 18, 35, 60, 100],
labels=['Child', 'Teen', 'Adult', 'Middle Age', 'Senior'])

# Plot survival rate by age group
sns.barplot(x='AgeGroup', y='Survived', data=df, palette='muted')
plt.title("Survival Rate by Age Group")
plt.xlabel("Age Group")
plt.ylabel("Survival Rate")
plt.show()
```

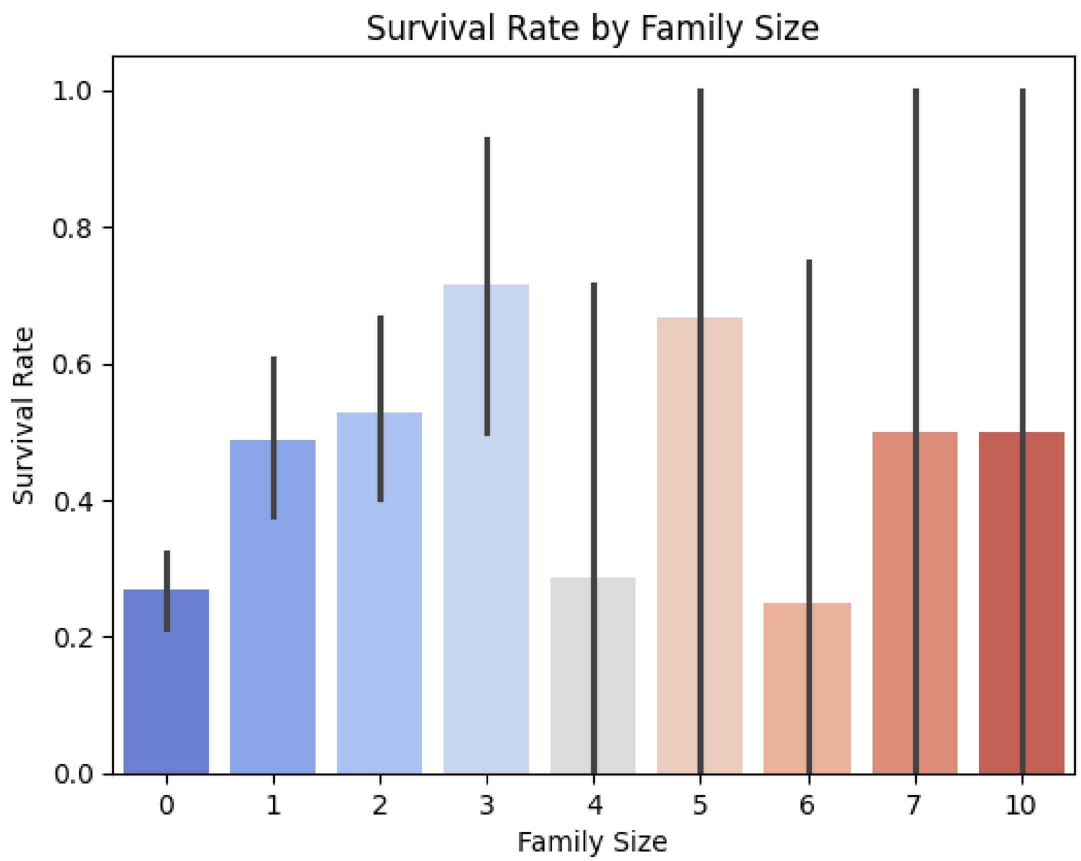
```
C:\Users\Windows\AppData\Local\Temp\ipykernel_9456\1867470614.py:4:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='AgeGroup', y='Survived', data=df, palette='muted')
```



# Survival Rate by Family Size:

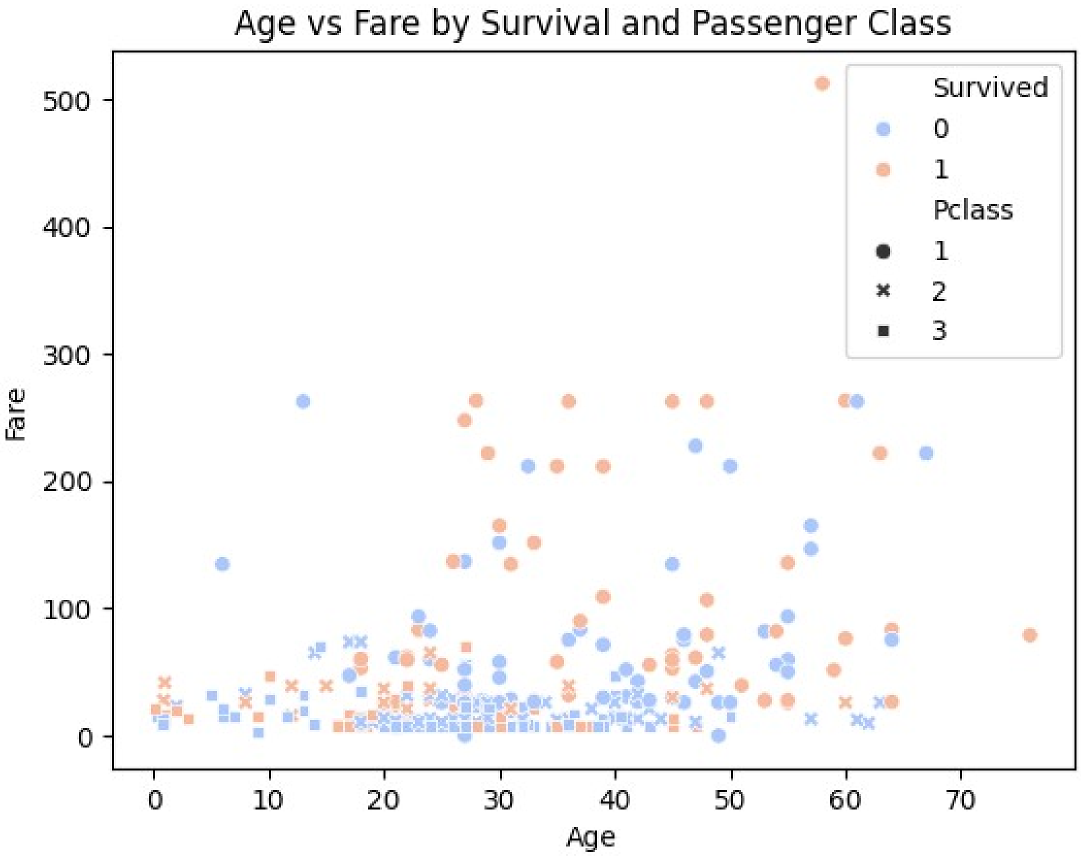
```
df['FamilySize'] = df['SibSp'] + df['Parch']
sns.barplot(x='FamilySize', y='Survived', data=df, palette='coolwarm')
plt.title("Survival Rate by Family Size")
plt.xlabel("Family Size")
plt.ylabel("Survival Rate")
plt.show()
```

```
C:\Users\Windows\AppData\Local\Temp\ipykernel_9456\1543103290.py:2:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x='FamilySize', y='Survived', data=df,
palette='coolwarm')
```



Passenger Class vs. Age vs. Survival

```
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=df,
palette='coolwarm', style='Pclass')
plt.title("Age vs Fare by Survival and Passenger Class")
plt.xlabel("Age")
plt.ylabel("Fare")
plt.show()
```



In this project, I conducted an extensive Exploratory Data Analysis (EDA) on the Titanic dataset, focusing on data cleaning, handling missing values and visualizing survival trends across various features, providing key insights into how factors like gender, age, class, and family size influenced survival outcomes

THANK YOU !!!