EXPLORATORY DATA ANALYSIS REPORT

DATASET : TITANIC DATASET

Step 1:

Install all necessary libraries

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns
```

Step 2:

load the dataset

```
df = pd.read_csv("C://Users//harsh//Downloads//titanic_dataset_500_clean.csv")
```

Step 3:

Basic Information

The .info() output shows:

- **The dataset contains X rows and Y columns.**
- **It lists each column name, the number of non-null (non-missing) values, and the data type (e.g., integer, float, object for text).**
- **Some columns may have missing values (less than total row count).**

Key takeaway:
This helps us quickly understand the structure of the dataset, detect missing data, and see which columns are numerical or categorical.

```
print("Basic Information:")
print(df.info())
```

```
Basic Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  500 non-null    int64
 1   Survived     500 non-null    int64
 2   Pclass       500 non-null    int64
 3   Name         500 non-null    object
 4   Sex          500 non-null    object
 5   Age          500 non-null    int64
 6   SibSp        500 non-null    int64
 7   Parch        500 non-null    int64
 8   Ticket       500 non-null    object
 9   Fare         500 non-null    float64
 10  Cabin        500 non-null    object
 11  Embarked     500 non-null    object
dtypes: float64(1), int64(6), object(5)
memory usage: 47.0+ KB
None
```

Basic Statistics

The .describe() output provides:

- **Count: Number of non-missing values for each numeric column.**
- **Mean, Min, Max: Average and range of the values.**
- **25%, 50%, 75%: Percentiles showing the spread and distribution.**

Key takeaway:
This summary helps identify typical values, outliers, and variability in the dataset. For example, if Fare has a very high max compared to its mean, it suggests some passengers paid much more than most others.

```
print("\nBasic Statistics:")
print(df.describe())
```

```
Basic Statistics:
       PassengerId    Survived      Pclass         Age       SibSp  \
count   500.000000  500.000000  500.000000  500.000000  500.000000
mean    250.500000    0.396000    2.264000   37.758000    2.038000
std     144.481833    0.489554    0.846012   20.062414    1.401601
min       1.000000    0.000000    1.000000    1.000000    0.000000
25%     125.750000    0.000000    1.000000   24.000000    1.000000
50%     250.500000    0.000000    3.000000   36.000000    2.000000
75%     375.250000    1.000000    3.000000   50.000000    3.000000
max     500.000000    1.000000    3.000000   80.000000    4.000000

            Parch        Fare
count  500.000000  500.000000
mean     1.450000   52.612458
std      1.142853   27.960898
min      0.000000    5.001100
25%      0.000000   29.651575
50%      1.000000   52.049200
75%      2.000000   77.740425
max      3.000000   99.793000
```

Step3:

Survival Count

The output of df['Survived'].value_counts() shows:

- **0 → Number of passengers who did not survive.**
- **1 → Number of passengers who survived.**

Interpretation:
This gives a quick overview of survival distribution. If the number of 0's is higher than 1's, it means more passengers did not survive compared to those who did.

Passenger Class Count

The output of df['Pclass'].value_counts() shows:

- **1 → Number of passengers in First Class.**
- **2 → Number of passengers in Second Class.**
- **3 → Number of passengers in Third Class.**

Interpretation:
This breakdown helps us see how many passengers traveled in each ticket class. It's also useful for comparing survival rates between classes later.

```
print("\nSurvival Count:")
print(df['Survived'].value_counts())
print("\nPassenger Class Count:")
print(df['Pclass'].value_counts())
```

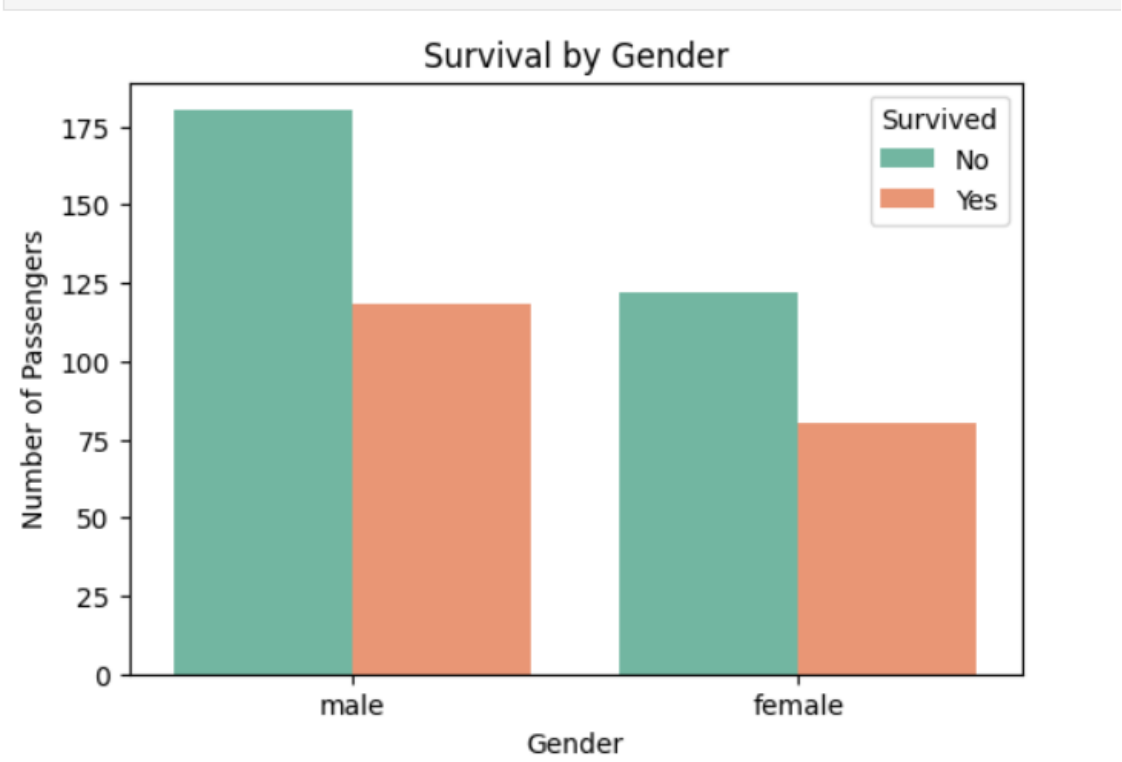
```
Survival Count:
Survived
0    302
1    198
Name: count, dtype: int64

Passenger Class Count:
Pclass
3    262
1    130
2    108
Name: count, dtype: int64
```

Step4:

1. Bar Chart – Survival by Gender

- **Observation:**
  - **A significantly higher proportion of females survived compared to males.**
  - **Most males did not survive.**
- **Finding:**
  - **Gender played a major role in survival chances, with females having a much better survival rate. This could be due to the "women and children first" evacuation practice.**

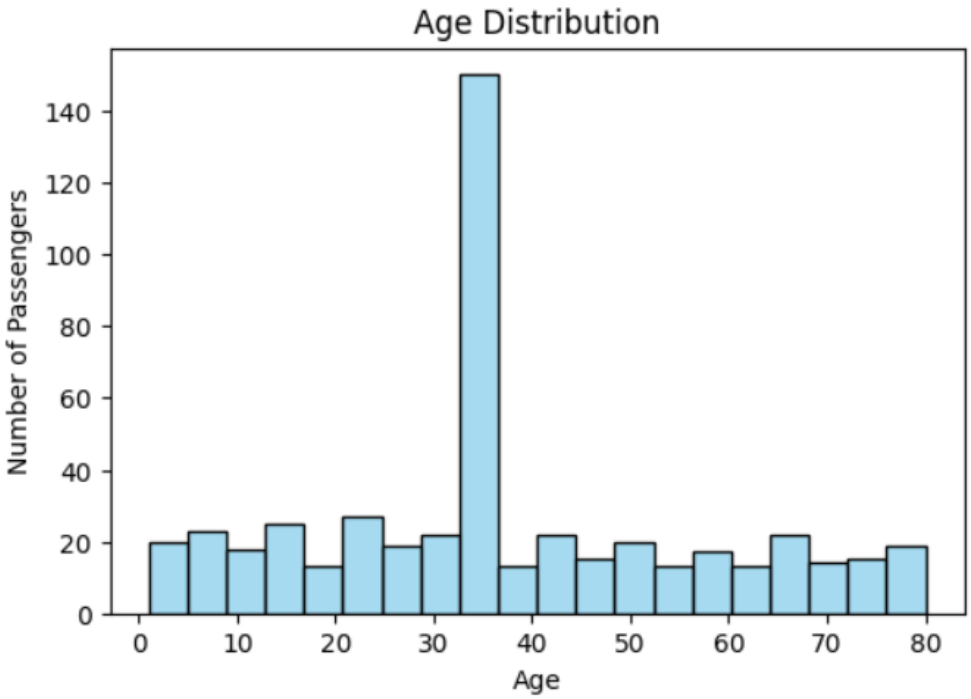
```
plt.figure(figsize=(6,4))
sns.countplot(data=df, x="Sex", hue="Survived", palette="Set2")
plt.title("Survival by Gender")
plt.xlabel("Gender")
plt.ylabel("Number of Passengers")
plt.legend(title="Survived", labels=["No", "Yes"])
plt.show()
```



2. Histogram – Age Distribution

- **Observation:**
  - **The age distribution is concentrated between 20–40 years.**
  - **There are fewer children and elderly passengers.**
- **Finding:**
  - **Most passengers were young adults, which could impact survival patterns when analyzed with other factors like gender and class.**

```
plt.figure(figsize=(6,4))
sns.histplot(df["Age"].dropna(), bins=20, kde=False, color="skyblue", edgecolor="black")
plt.title("Age Distribution")
plt.xlabel("Age")
plt.ylabel("Number of Passengers")
plt.show()
```

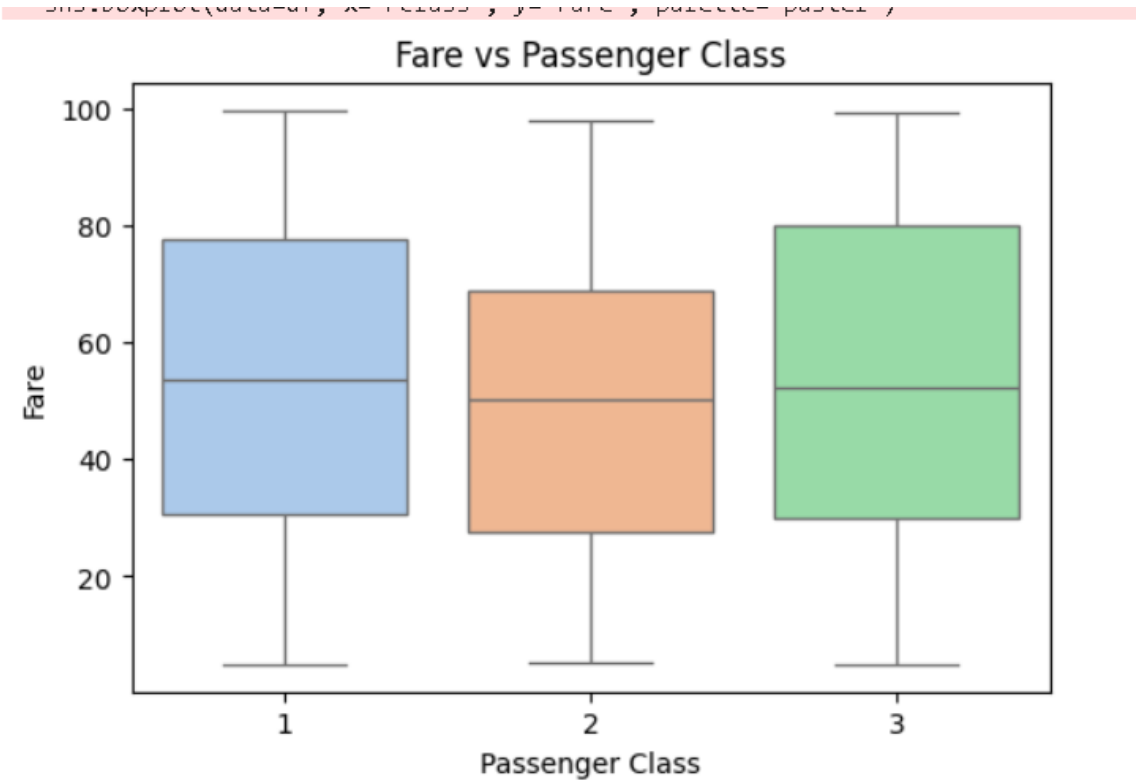


3. Box Plot – Fare vs. Passenger Class

- **Observation:**
  - **First-class passengers paid the highest fares, with a wider spread in prices.**
  - **Third-class fares were the lowest and had less variation.**
- **Finding:**

  Fare is strongly related to passenger class. Wealthier passengers likely had better cabins and possibly better access to lifeboats, affecting survival.

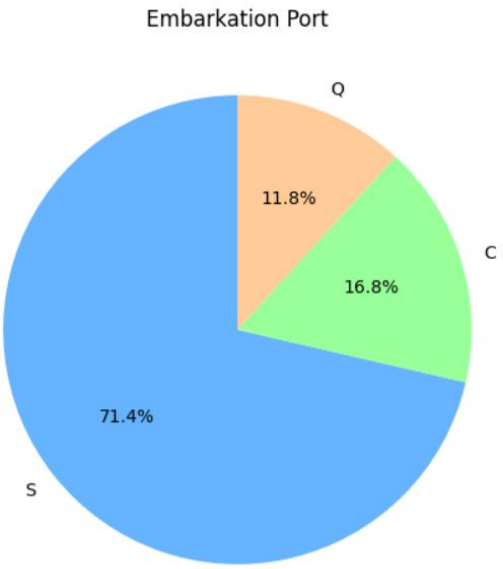
```
plt.figure(figsize=(6,4))
sns.boxplot(data=df, x="Pclass", y="Fare", palette="pastel")
plt.title("Fare vs Passenger Class")
plt.xlabel("Passenger Class")
plt.ylabel("Fare")
plt.show()
```



4. Pie Chart – Embarkation Port

- **Observation:**
  - **Most passengers boarded from Southampton, followed by Cherbourg and Queenstown.**
- **Finding:**
  - **Southampton was the main boarding location for Titanic passengers, which could reflect socio-economic and geographical factors.**

```
plt.figure(figsize=(6,6))
df["Embarked"].value_counts().plot.pie(autopct="%1.1f%%", startangle=90,
colors=["#66b3ff", "#99ff99", "#ffcc99"])
plt.title("Embarkation Port")
plt.ylabel("")
plt.show()
```



4.Correlation Analysis Report

Objective:
To explore the relationships between the numerical variables in the dataset using a correlation heatmap.

---

1. Methodology

- **Used the Pearson correlation coefficient to measure the linear relationship between variables.**
- **Generated a heatmap to visualize the correlations:**
  - **Red tones indicate a stronger positive correlation.**
  - **Blue tones indicate a weaker or negative correlation.**
  - **Values range from -1 to +1.**

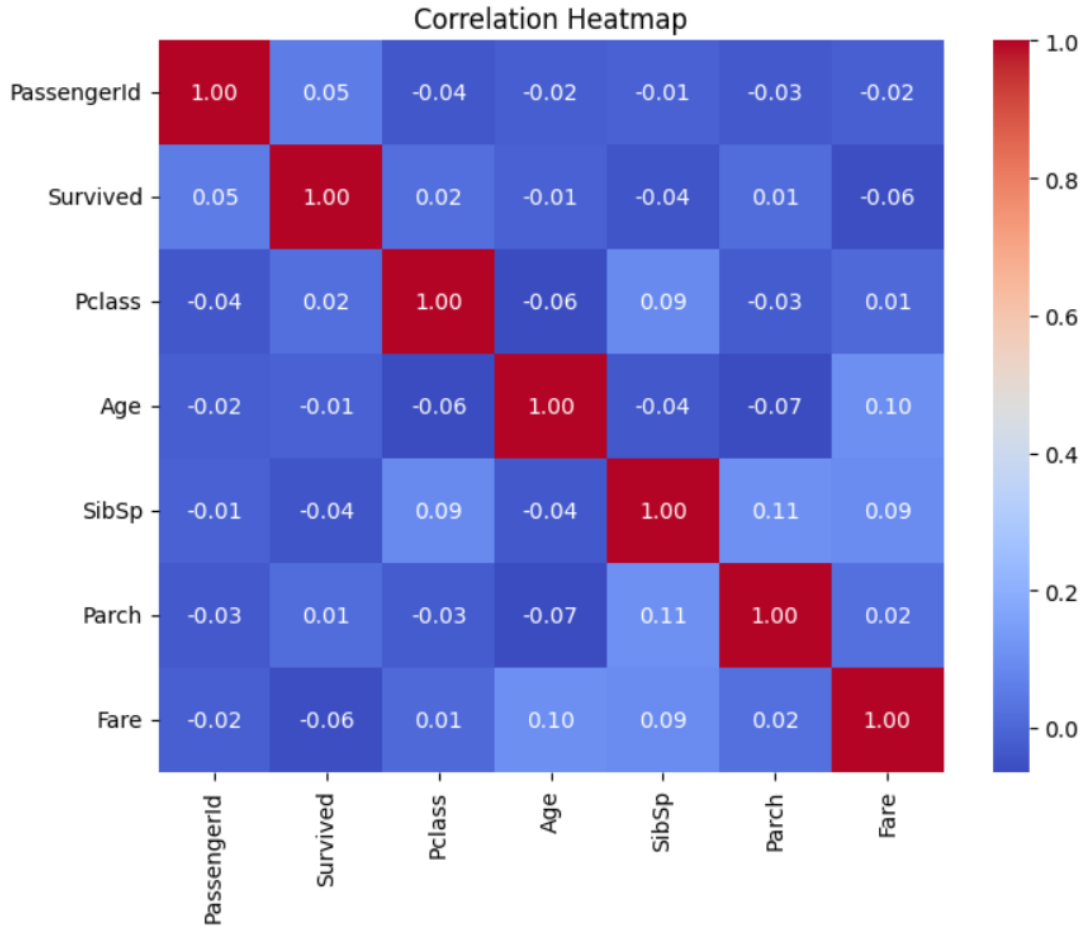
---

2. Key Findings

- **Survived vs Pclass → 0.02**
  Minimal relationship. Passenger class did not strongly influence survival in this dataset.
- **Fare vs Age → 0.10**
  Slight positive relationship. Older passengers tended to pay slightly higher fares.
- **SibSp vs Parch → 0.11**
  Weak positive correlation. Passengers traveling with siblings/spouses often also traveled with parents/children.

Most correlations are close to 0
This indicates that the numerical variables in this dataset are largely independent.

```
plt.figure(figsize=(8, 6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```



Summary of Findings:

- **Survival chances were higher for females than males.**
- **The passenger age distribution was dominated by young adults.**
- **Higher-class passengers paid significantly more for tickets and might have had better chances of survival.**
- **Most passengers boarded from Southampton.**