```
# 📌 Titanic Dataset – Exploratory Data Analysis (EDA)

## 1. Uploading Dataset

## 2. Loading and Displaying Data

## 3. Basic Data Exploration

## 4. Univariate Analysis

## 5. Bivariate Analysis

## 6. Multivariate Analysis

## 7. Handling Missing Values

## 8. Summary of Findings
```

```
# Uploading the dataset from your local system into Google Colab.
# After running this, it will ask you to choose the CSV file.
from google.colab import files
uploaded = files.upload()
```

Choose files datasets_11…98_train.csv

datasets_11657_16098_train.csv(text/csv) - 61194 bytes, last modified: 20/11/2025 - 100% done

```
Saving datasets_11657_16098_train.csv to datasets_11657_16098_train.csv
```

```
# Importing pandas to work with data tables.
import pandas as pd

# Reading the Titanic dataset into a DataFrame.

df = pd.read_csv("datasets_11657_16098_train.csv")
# Showing the first few rows to understand how the data looks.
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

Next steps: Generate code with df New interactive sheet

```
# Checking basic information about the dataset:
# - total rows
# - column names
# - data types
# - missing values in each column

df.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
```

```
# Getting statistical summary of all numerical columns.
# Helps understand mean, min, max, percentiles, etc.

df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
# Counting how many missing (null) values are present in each column.

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

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2

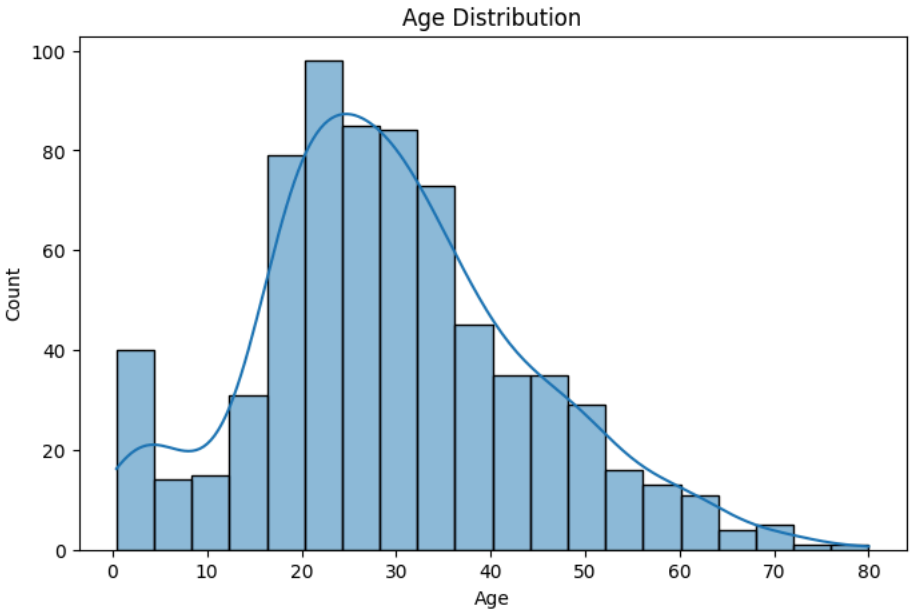
dtype: int64

```
# Checking if the dataset has any duplicate rows.
df.duplicated().sum()
```

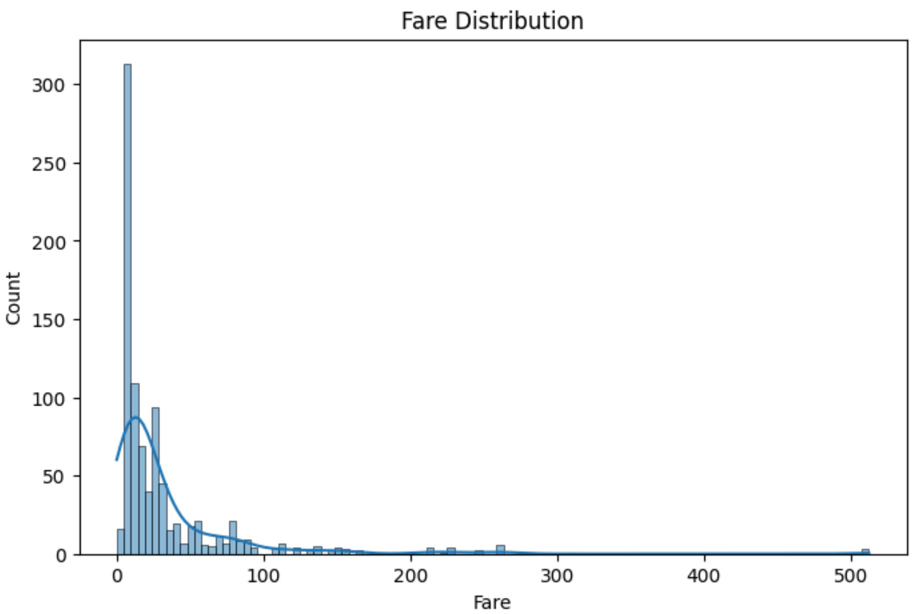
```
np.int64(0)
```

```
# Importing visualization libraries to create graphs.

import matplotlib.pyplot as plt
import seaborn as sns
```

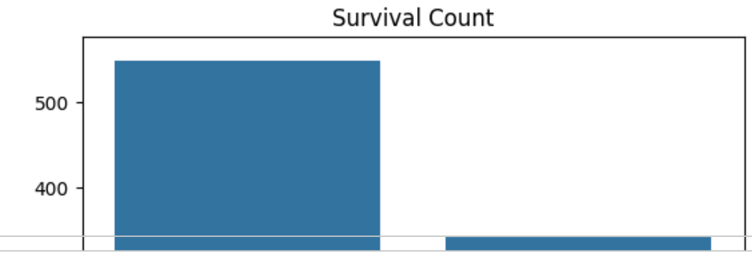
```
# Plotting a histogram to see how passenger ages are distributed.

plt.figure(figsize=(8,5))
sns.histplot(df['Age'], kde=True)
plt.title("Age Distribution")
plt.show()
```



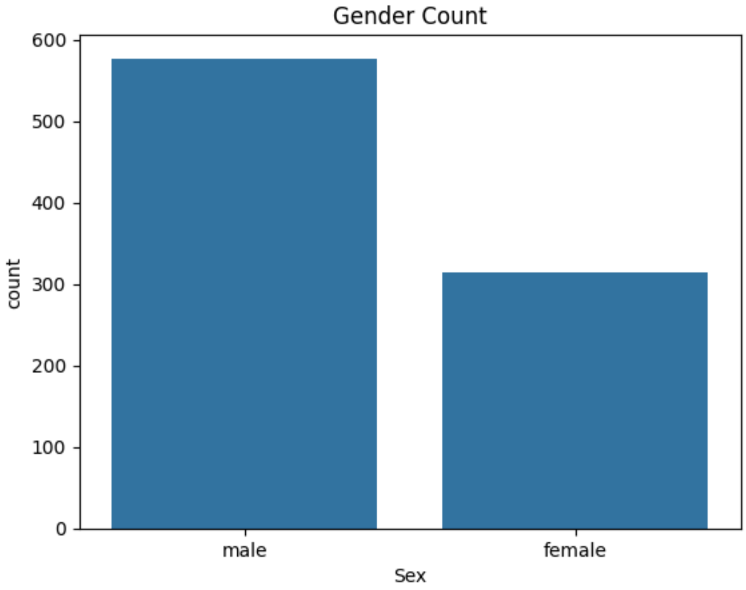

```
# Plotting fare distribution to check how much passengers paid.

plt.figure(figsize=(8,5))
sns.histplot(df['Fare'], kde=True)
plt.title("Fare Distribution")
plt.show()
```



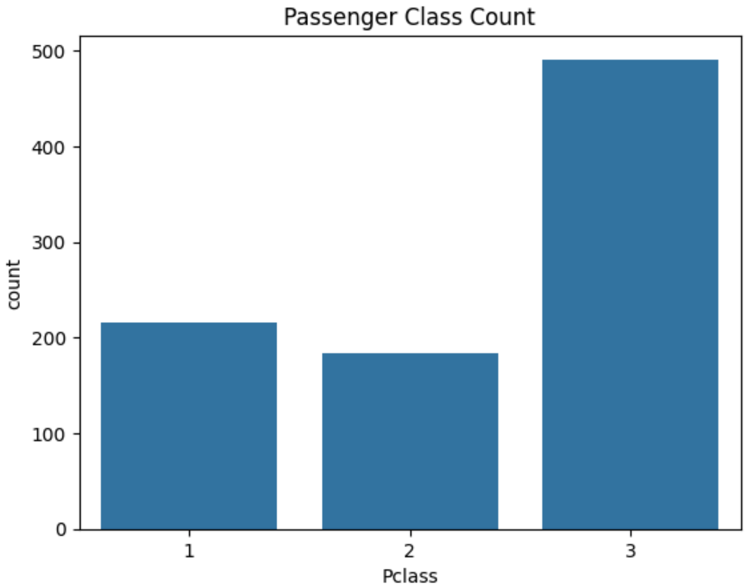

```
# Counting how many passengers survived vs did not survive.

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




```
# Counting number of male vs female passengers.
sns.countplot(x='Sex', data=df)
plt.title("Gender Count")
plt.show()
```

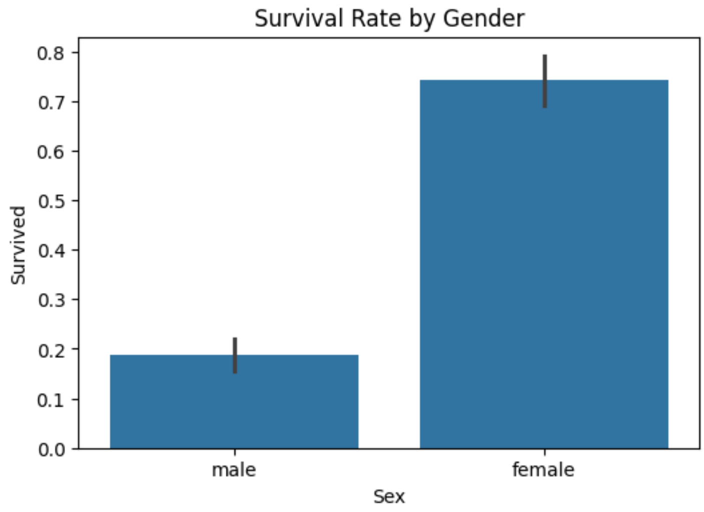



```
# Counting number of passengers in each class (1st, 2nd, 3rd).
sns.countplot(x='Pclass', data=df)
plt.title("Passenger Class Count")
plt.show()
```

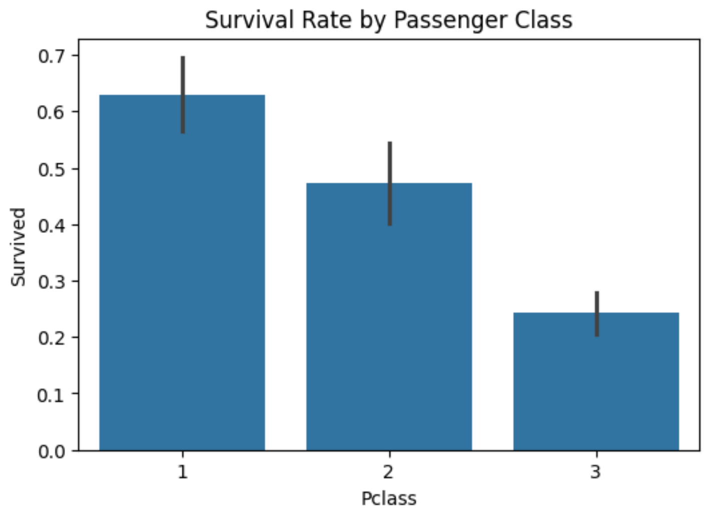



```
# Checking survival rate based on gender.

plt.figure(figsize=(6,4))
sns.barplot(x='Sex', y='Survived', data=df)
plt.title("Survival Rate by Gender")
```

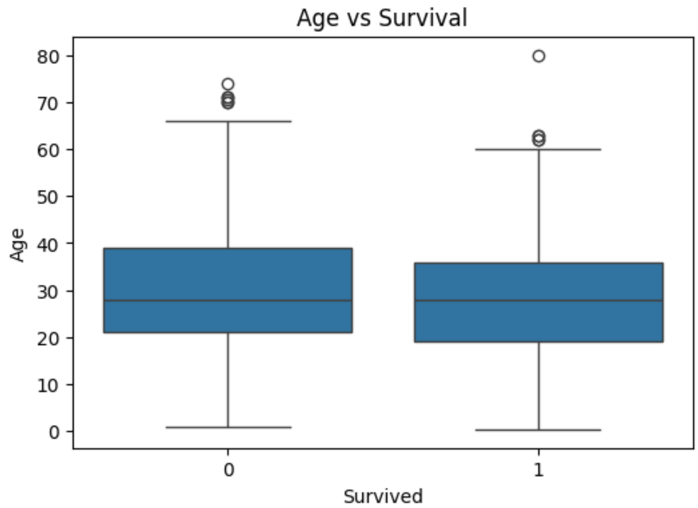
```
plt.show()
```



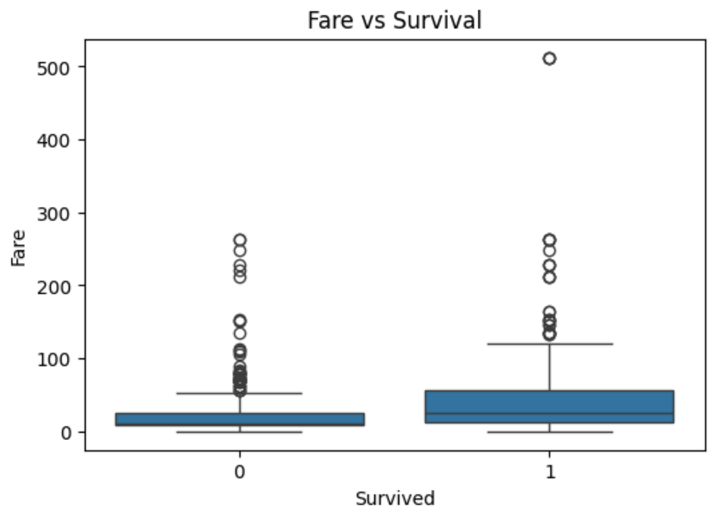

```
# Checking survival rate for each passenger class.
plt.figure(figsize=(6,4))
sns.barplot(x='Pclass', y='Survived', data=df)
plt.title("Survival Rate by Passenger Class")
plt.show()
```




```
# Comparing the ages of survivors vs non-survivors.



plt.figure(figsize=(6,4))
sns.boxplot(x='Survived', y='Age', data=df)
plt.title("Age vs Survival")
plt.show()
```




```
# Comparing fares paid by survivors vs non-survivors.

plt.figure(figsize=(6,4))
sns.boxplot(x='Survived', y='Fare', data=df)
plt.title("Fare vs Survival")
plt.show()
```




```
# Selecting only numeric columns to check correlation between them.

numeric_df = df[['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']]
numeric_df.head()
```

	Survived	Pclass	Age	SibSp	Parch	Fare
0	0	3	22.0	1	0	7.2500
1	1	1	38.0	1	0	71.2833
2	1	3	26.0	0	0	7.9250
3	1	1	35.0	1	0	53.1000
4	0	3	35.0	0	0	8.0500

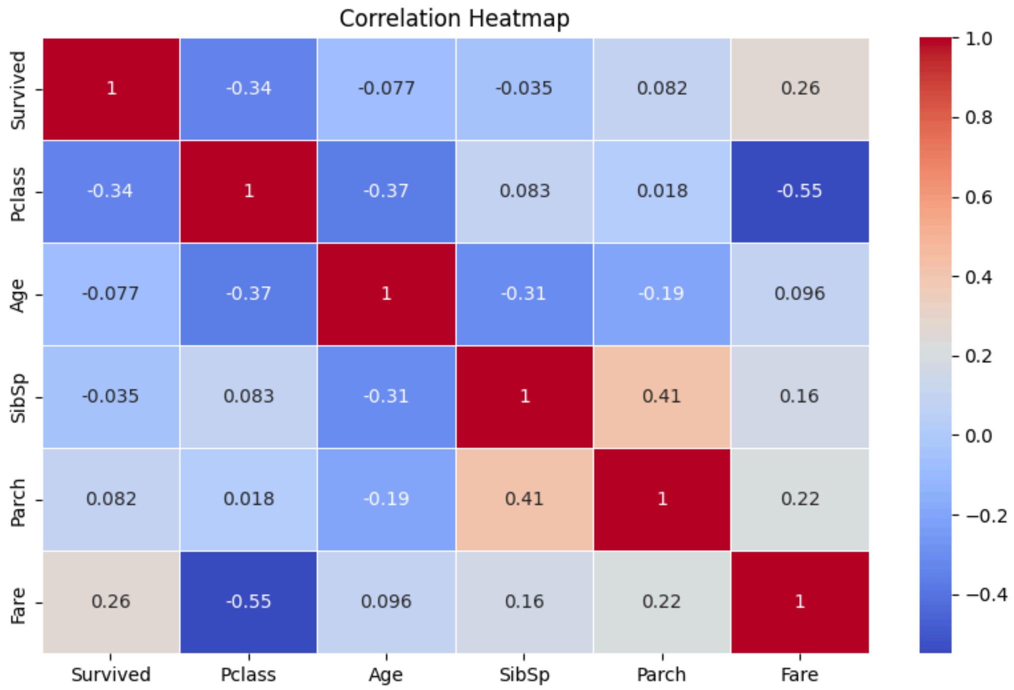
Next steps: Generate code with numeric_df New interactive sheet

```
# Creating a correlation matrix to measure how features relate to each other.

corr = numeric_df.corr()
corr
```

	Survived	Pclass	Age	SibSp	Parch	Fare
Survived	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

Next steps: Generate code with corr New interactive sheet

```
# Plotting the correlation matrix as a heatmap for easy visualization.

plt.figure(figsize=(10,6))
sns.heatmap(corr, annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```




```
# Creating pairplots to visualize relationships between multiple variables at once.

sns.pairplot(df[['Survived','Pclass','Age','Fare','SibSp','Parch']])
plt.show()
```

```
# Filling missing ages with the median age so the column has no null values.

df['Age'] = df['Age'].fillna(df['Age'].median())
```

```
# Filling missing 'Embarked' values with the most common category.
df['Embarked'] = df['Embarked'].fillna(df['Embarked'].mode()[0])
```

```
# Dropping the 'Cabin' column because it has too many missing values.
df = df.drop(columns=['Cabin'])
```

```
# 📌 Final Summary of EDA Findings

# - Most passengers were in 3rd class.
# - Majority of passengers were male.
# - Females had a much higher survival rate.
# - 1st class passengers survived more than 2nd and 3rd class.
# - Younger passengers survived slightly more than older ones.
```