Exploratory Data Analysis (EDA):

Step 1: Import Libraries

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

Step 2: Load Titanic Dataset

```
data = pd.read_csv('C:/Users/anubh/Downloads/Titanic-Dataset.csv')
data.head(10)
```

[14]:

	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
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

Step 3: Basic Data Exploration

a) .describe(), .info(), .value_counts()

[15]:
```
data.shape
```

[15]:
```
(891, 12)
```

[16]:
```
data.columns
```

[16]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

[17]:
```
data.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
```

[18]:
```
# Statistical summary of numerical columns
data.describe()
```

[18]:

	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

[25]:
```
# Value counts of 'Survived' column
data['Survived'].value_counts()
```

[25]:
```
Survived
0    549
1    342
Name: count, dtype: int64
```

[27]:
```
# Value counts of 'Pclass' column
data['Pclass'].value_counts()
```

[27]:
```
Pclass
3    491
1    216
2    184
Name: count, dtype: int64
```

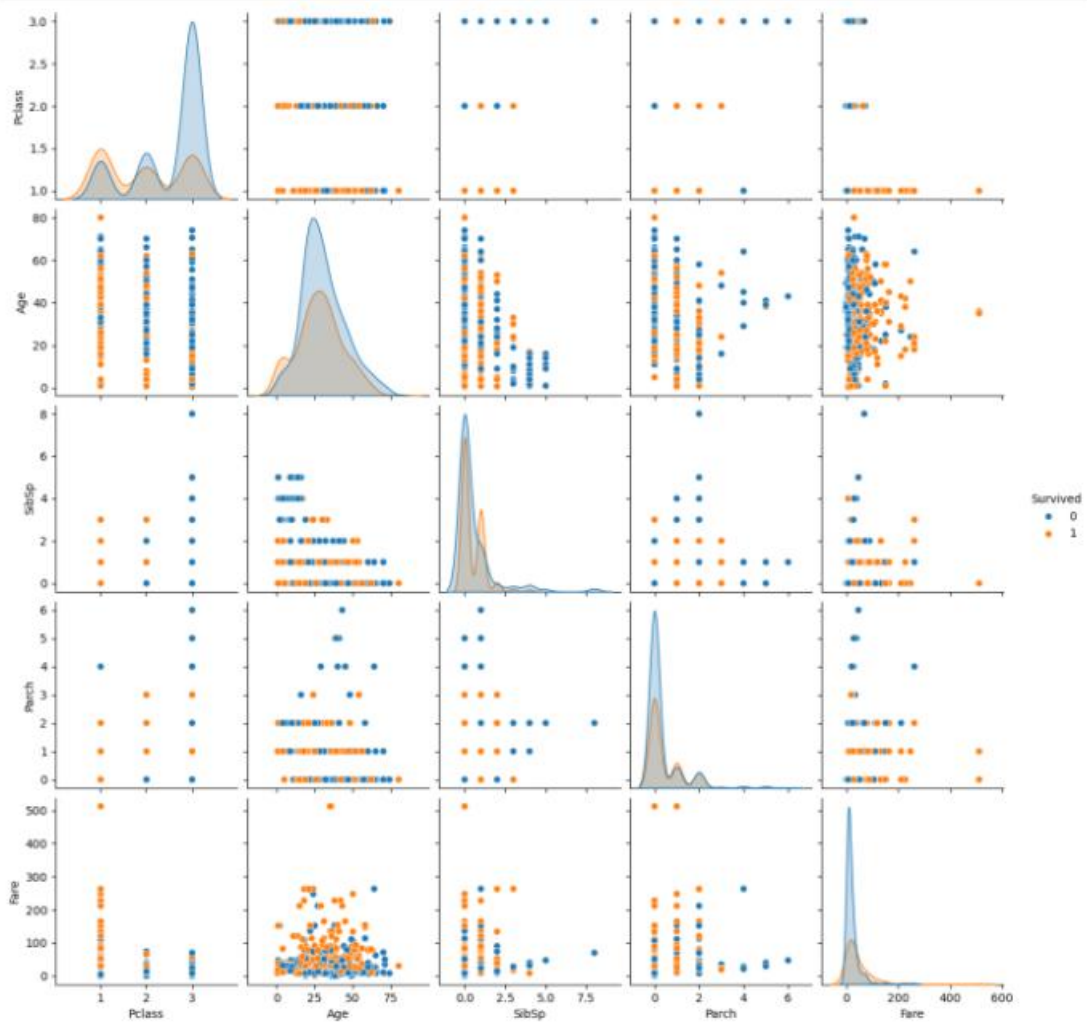
[29]:
```
# Value counts of 'Sex' column
data['Sex'].value_counts()
```

[29]:
```
Sex
male      577
female    314
Name: count, dtype: int64
```

Step 4: Visualizations

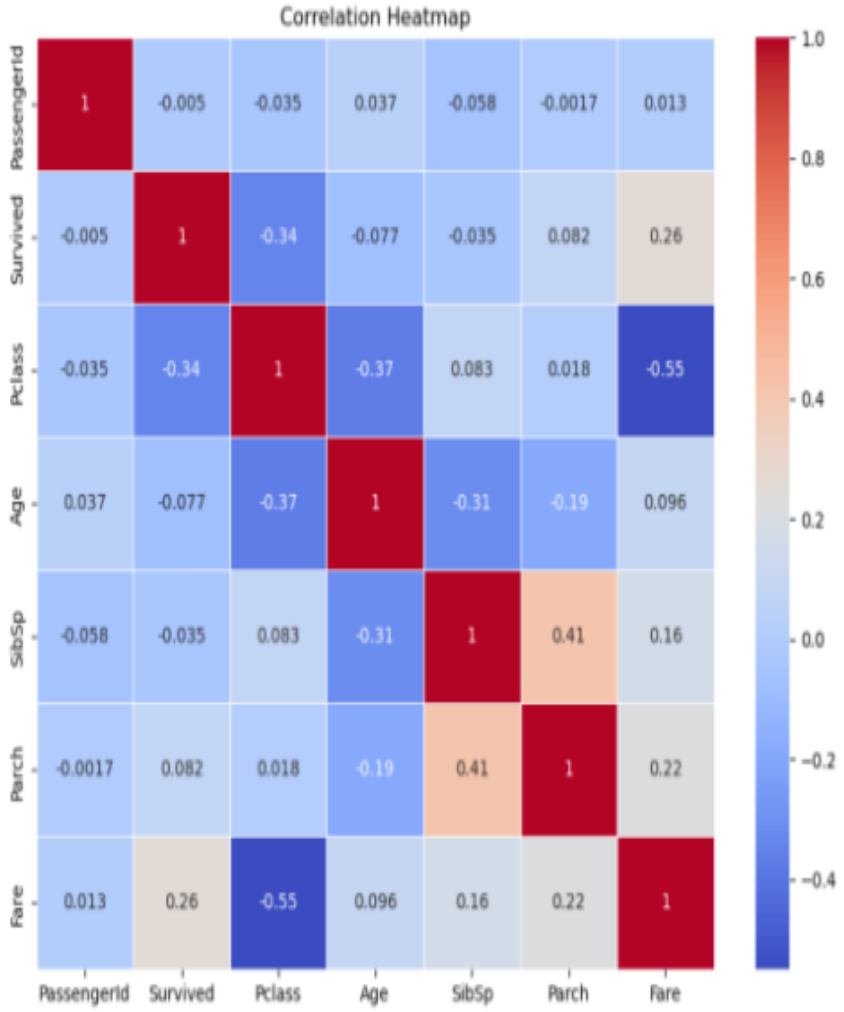
a) Pairplot (Relationship overview)

```
[36]: # Plot pairwise relationships (select only numerical columns for better results)
sns.pairplot(data[['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']], hue='Survived')
plt.show()
```



b) Heatmap (Correlation between variables):

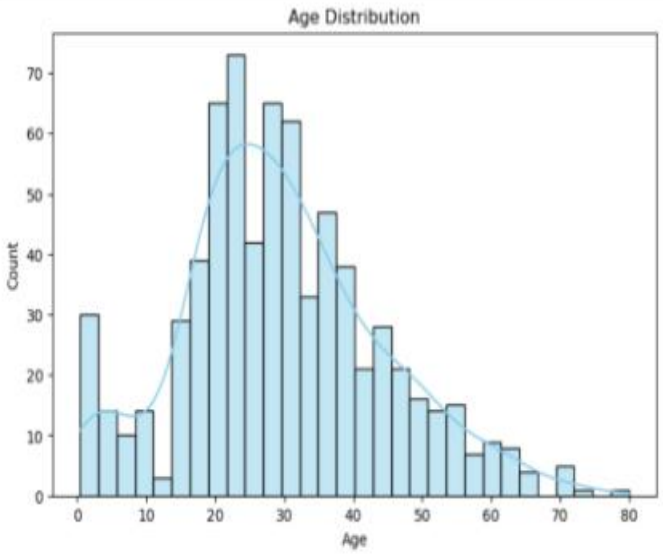
```
[39]: # Correlation matrix only for numeric columns
plt.figure(figsize=(10,8))
corr_matrix = data.select_dtypes(include=[np.number]).corr()  # Only numeric columns
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```



🎯 Step 5: Plot Histograms, Boxplots, Scatterplots

a) Histogram of Age

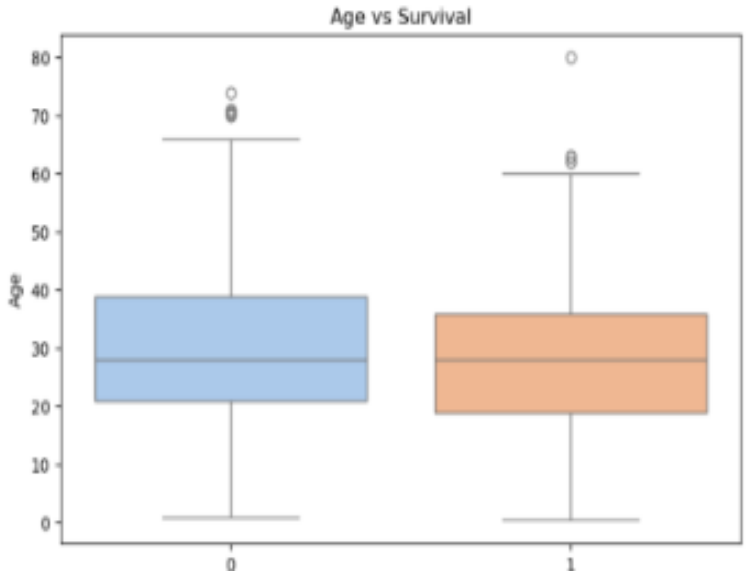
```
[41]: plt.figure(figsize=(8,5))
      sns.histplot(data['Age'].dropna(), kde=True, bins=30, color='skyblue')
      plt.title('Age Distribution')
      plt.xlabel('Age')
      plt.ylabel('Count')
      plt.show()
```



b) Boxplot of Age vs Survived

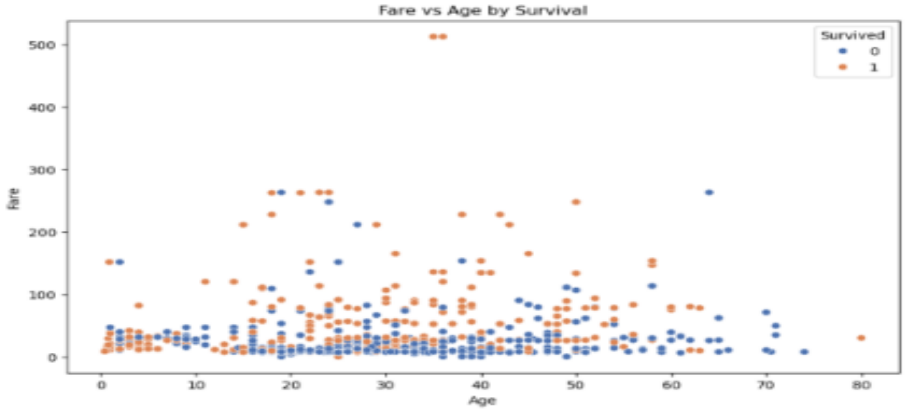
```
[43]: plt.figure(figsize=(8,5))
sns.boxplot(x='Survived', y='Age', data=data, palette='pastel')
plt.title('Age vs Survival')
plt.xlabel('Survived (0 = No, 1 = Yes)')
plt.ylabel('Age')
plt.show()
```

```
C:\Users\anubh\AppData\Local\Temp\ipykernel_26944\3758484003.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 sam
e effect.

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



c) Scatterplot: Fare vs Age colored by Survival

```
[45]: plt.figure(figsize=(10,6))
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=data, palette='deep')
plt.title('Fare vs Age by Survival')
plt.xlabel('Age')
plt.ylabel('Fare')
plt.show()
```



✍️ Step 6: Observations for Each Visual

Visual	Observation
Pairplot	Survived passengers mostly belonged to Pclass 1 and had higher fares.
Heatmap	'Fare' and 'Pclass' are negatively correlated. Survival slightly correlates with Fare and Pclass.
Histogram (Age)	Majority of passengers are between 20-40 years. Very young and very old are fewer.
Boxplot (Age vs Survived)	Younger passengers survived more compared to older ones.
Scatterplot (Fare vs Age)	Passengers paying higher fare (luxury cabins) had better survival chances.

🧾 Step 7: Final Summary of Findings

- ✅ Most survivors were from 1st class (Pclass 1).
- ✅ Female passengers had a higher survival rate than males.
- ✅ Younger passengers and children had slightly better survival rates.
- ✅ Passengers who paid higher fares had greater chances of survival.
- ✅ Survival is closely related to **Pclass**, **Fare**, **Sex**, and **Age**.