

Exploratory Data Analysis on titanic dataset

1. Import necessary libraries.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

2. Read external csv file and store as dataframe.

```
df = pd.read_csv("titanic.csv")
```

3. Return first 5 rows from dataframe.

```
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... Heikkinen, Miss. Laina	female	38.0	1	0	PC 17599 STON/O2. 3101282	71.2833 7.9250	C85 NaN	C S
2	3	1	3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	26.0	0	0	113803	53.1000	C123	S
3	4	1	1	Allen, Mr. William Henry	male	35.0	1	0	373450	8.0500	NaN	S

4. Remove unnecessary columns that aren't useful for EDA.

```
df.drop(['PassengerId', 'Name', 'Ticket', 'Fare',
'Cabin', 'Embarked'], axis=1, inplace=True)
df.head()
```

Survived	Pclass	Sex	Age	SibSp	Parch
0	0	3	male	22.0	1
1	1	1	female	38.0	1
2	1	3	female	26.0	0
3	1	1	female	35.0	1
4	0	3	male	35.0	0

Insights:

- Survived, Sex – binary categorical variable.
- Pclass – Ordinal categorical variable.

- Age – Continuous numerical variable.
 - SibSp, Parch – Discrete numerical variable.
5. Check for null values; if they are present in significant amount, impute them with statistical value or if they are in small count, simply drop them from dataframe.

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

Survived	0
Pclass	0
Sex	0
Age	177
SibSp	0
Parch	0
dtype:	int64

The Age column contains a large number of missing values, so these need to be imputed using an appropriate statistical measure. Analysis shows that each class corresponds to a distinct age range, causing the average age to vary significantly across classes. Therefore, imputing missing age values with the mean age of the corresponding class is an effective approach and helps improve the accuracy of the analysis. This can be implemented using the groupby(), transform(), andfillna() functions.

```
df.groupby('Pclass')[['Age']].mean()
```

Pclass	
1	38.233441
2	29.877630
3	25.140620

```
Name: Age, dtype: float64
```

```
df['Age'].fillna(df.groupby('Pclass')[['Age']].transform('mean'))
```

Check for null values,

Survived	0
Pclass	0
Sex	0
Age	0

```
SibSp      0  
Parch      0  
dtype: int64
```

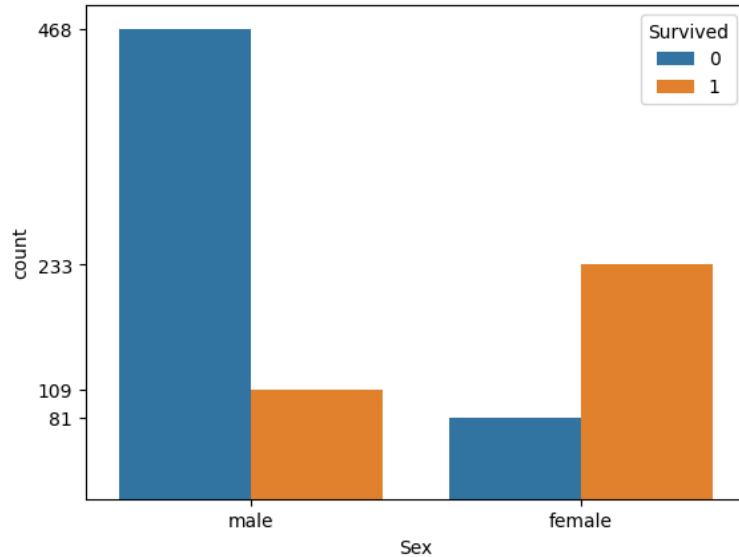
Shape of dataframe

```
df.shape  
(891, 6)
```

Visualizations:

1. Countplot:

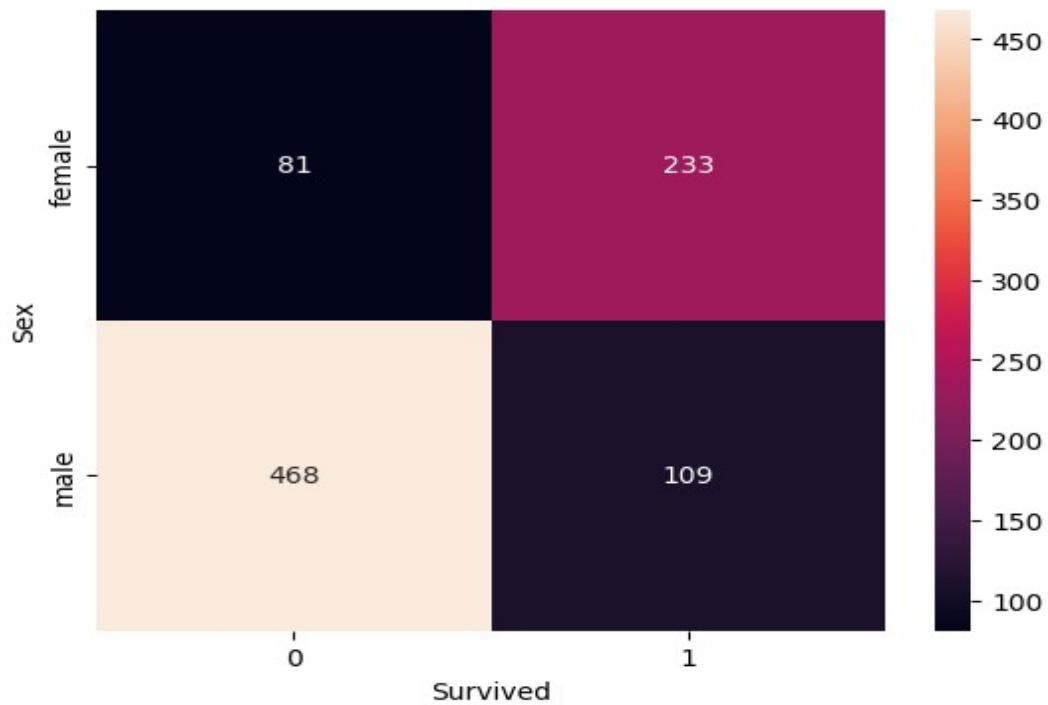
```
sns.countplot(data=df, x='Sex', hue='Survived')  
plt.yticks(df.groupby(['Sex', 'Survived'])['Survived'].count())  
plt.show()
```



Insight: Female passengers survived more than male.

2. Heatmaps:

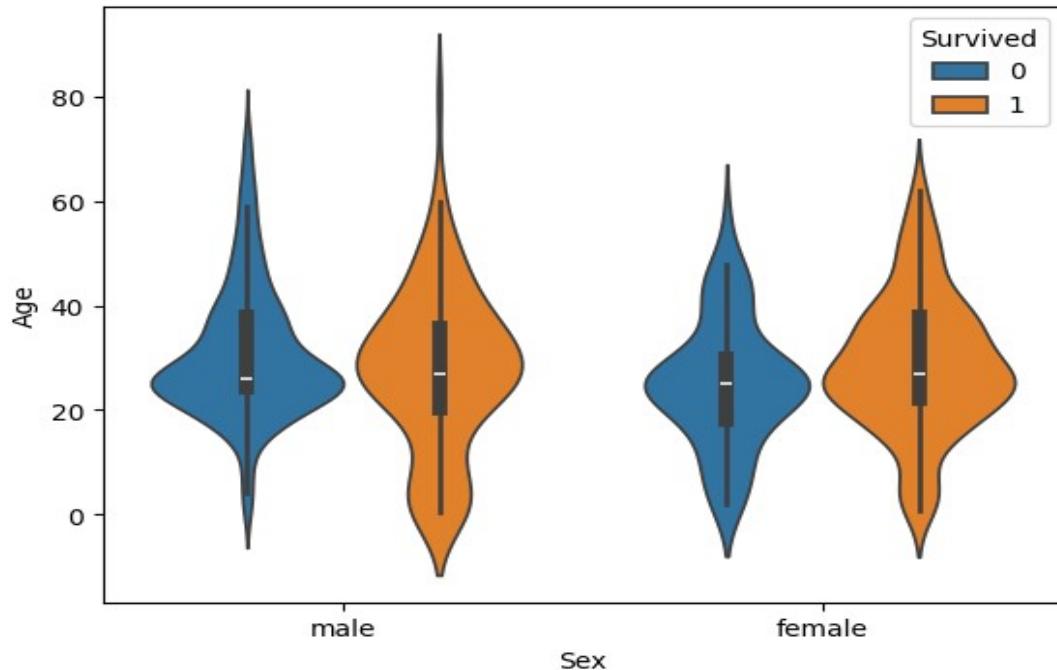
```
sns.heatmap(df.groupby(['Sex', 'Survived']).size().unstack()  
, annot=True, fmt='d')  
plt.show()
```



Insight: Heatmap exhibits light spectrum. Here, light shade shows most count whereas dark shows less count. Male has highest death count so it's in light shade.

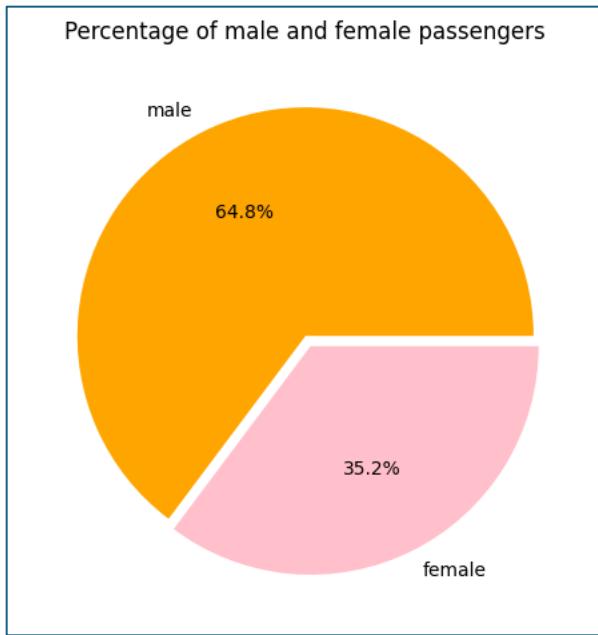
3. Violin plot:

```
sns.violinplot(data=df, x='Sex', y='Age', hue='Survived')
```



4. Pie Chart

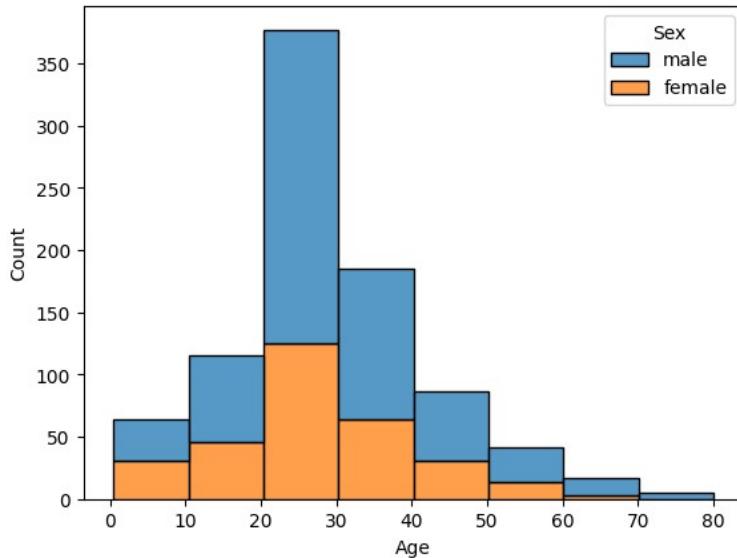
```
plt.pie(x=df['Sex'].value_counts(), labels=['male',  
'female'], autopct="%0.1f%%", colors=['orange', 'pink'],  
explode=[0, 0.05])  
plt.title('Percentage of male and female passengers')  
plt.tight_layout()
```



Insight : Male passengers are more in number than female.

5. Hist plot

```
sns.histplot(data = df, x='Age', hue='Sex', bins=8,  
multiple='stack') plt.show()
```



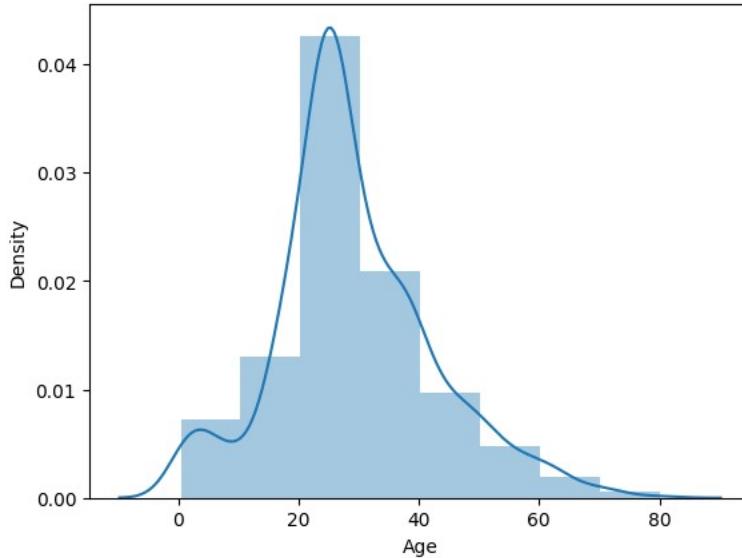
Insights:

- There are highest count of passengers in the 20-30 age range.
 - Within the range, males are in more number.
- The data is right skewed as the tail approaches towards higher ages.
- There are fewer old-aged passengers.

Skewness of data can be clearly illustrated with help of density plot.

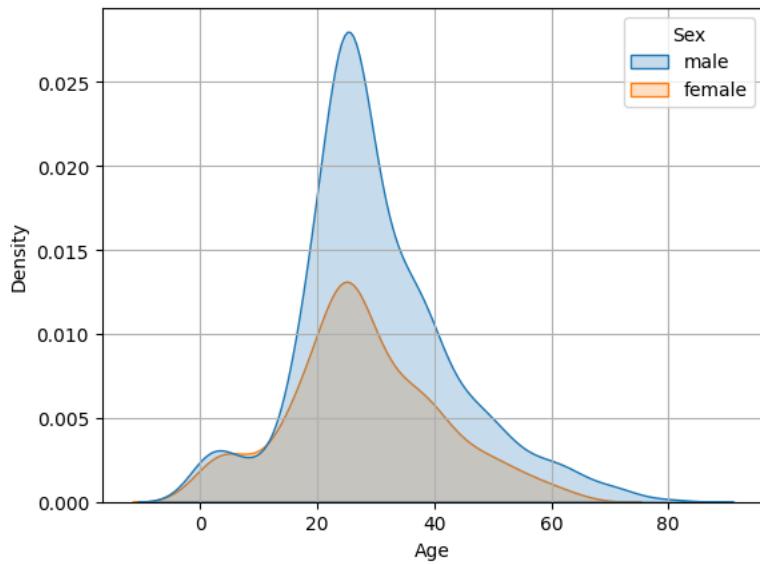
6. Dist plot:

```
sns.distplot(df['Age'], bins=8)
```



Similar type of graph can be achieved using kdeplot.

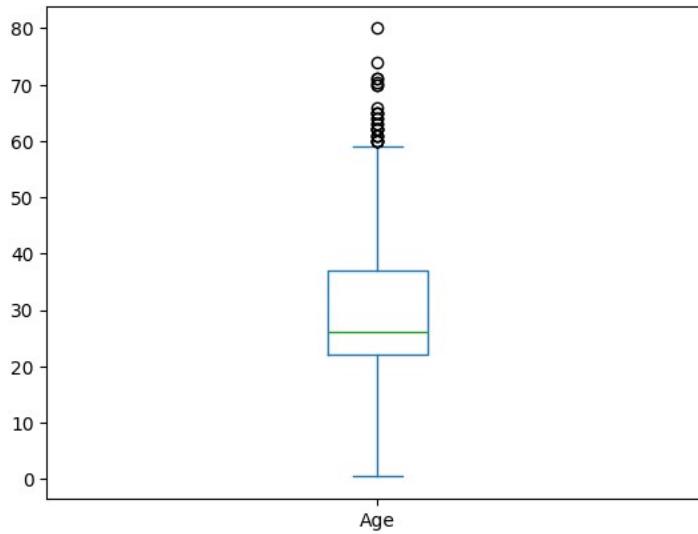
```
sns.kdeplot(data = df, x='Age', hue='Sex', fill=True)
plt.grid()
plt.show()
```



Insight: Male has high density of age.

7. Box plot : To find Outliers

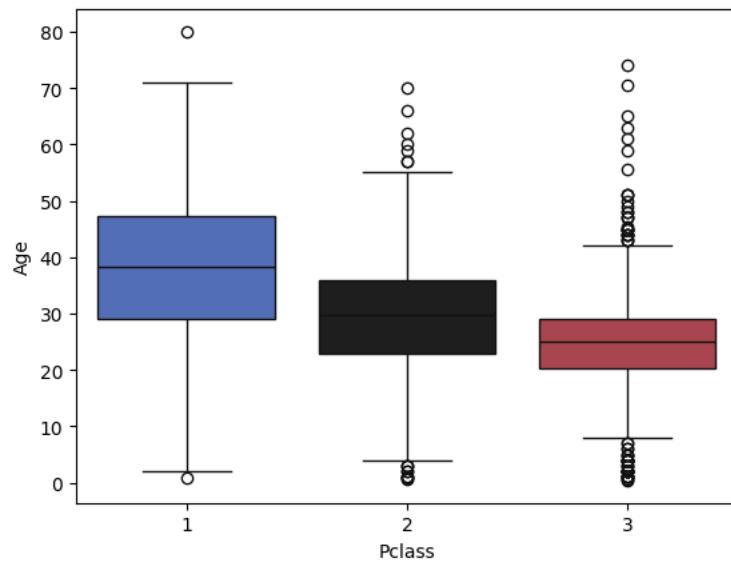
```
df[ 'Age' ].plot(kind= 'box')
plt.show()
```



Insight: There are outliers present above the maximum value.

```
sns.boxplot(data=df, x='Pclass', y='Age', palette='icefire')
```

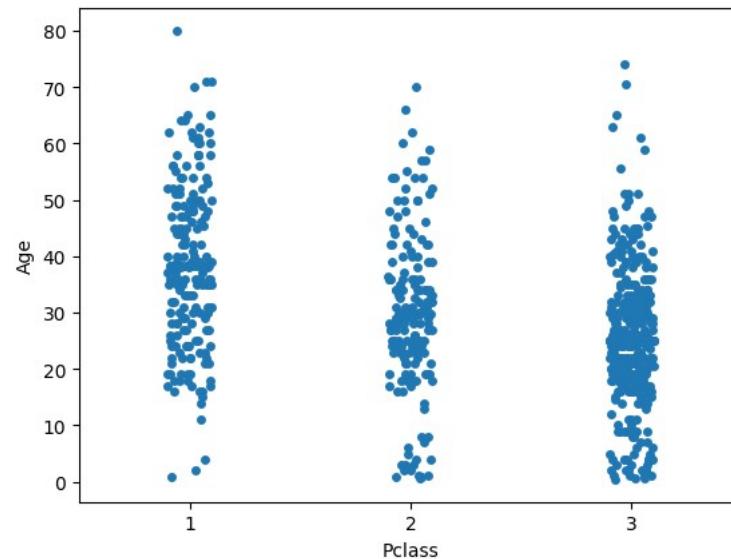
```
plt.show()
```



Insight: Pclass 1 has most age variation with less outliers.

7. Strip plot:

```
sns.stripplot(data=df, x= 'Pclass' , y= 'Age' , jitter=True)  
plt.show()
```

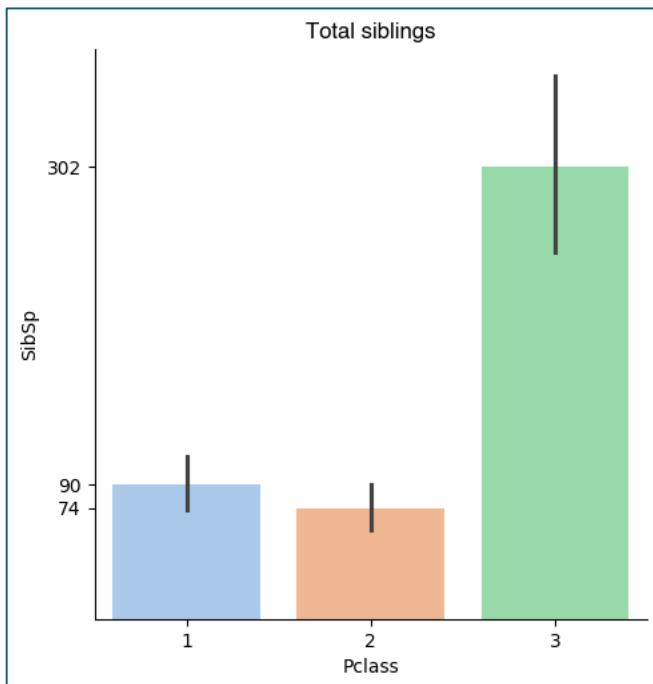


Insights:

- Pclass 3 is densely populated with young passengers.
- Pclass 1 has evenly spread passengers above 15 age.
- In Pclass 2, age is scatter above 35.

8. Cat plot:

```
sns.catplot(data=df, x='Pclass', y='SibSp', kind='bar',
estimator=np.sum, palette='pastel')
plt.yticks(df.groupby('Pclass')['SibSp'].sum())
plt.title('Total siblings', fontname='Helvetica')
plt.show()
```



Insight: Class 3 has highest siblings followed by class 1 and 2.