CHRIST
(DEEMED TO BE UNIVERSITY)
DELHI-NCR, INDIA

# AI/ML Programming

# MCA-475

## CIA – 01

*BY*

Himanshu Heda (24225013)

**SUBMITTED TO**

Dr. Manjula Shannhog

## SCHOOL OF SCIENCES

## 2024-25

## Importing the Libraries

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

✓ 13.3s Python

## Importing the CSV file to Perform some analysis

```
df = pd.read_csv('./Dataset/train.csv')
```

✓ 0.1s Python

## Performing few commmands like head, tail, info, describe, shape, columns, dtypes

```
df.head()
```

✓ 0.0s Python

| | 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 |

```
df.tail()
```

✓ 0.0s Python

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 886 | 887 | 0 | 2 | Montvila, Rev. Juozas | male | 27.0 | 0 | 0 | 211536 | 13.00 | NaN | S |
| 887 | 888 | 1 | 1 | Graham, Miss. Margaret Edith | female | 19.0 | 0 | 0 | 112053 | 30.00 | B42 | S |
| 888 | 889 | 0 | 3 | Johnston, Miss. Catherine Helen "Carrie" | female | NaN | 1 | 2 | W./C. 6607 | 23.45 | NaN | S |
| 889 | 890 | 1 | 1 | Behr, Mr. Karl Howell | male | 26.0 | 0 | 0 | 111369 | 30.00 | C148 | C |
| 890 | 891 | 0 | 3 | Dooley, Mr. Patrick | male | 32.0 | 0 | 0 | 370376 | 7.75 | NaN | Q |

```
df.info()
```

✓ 0.1s Python

```
<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
```

```
df.info()
```

[5] ✓ 0.1s Python

```
<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
```

```
df.dtypes
```

[7] ✓ 0.0s Python

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

# Is use to Replace the Exisiting Values in the DataFrame

```
df.rename(columns={'PassengerId': 'P_ID', 'Pclass': 'P_Class'}, inplace=True)
df.head()
```

[4] ✓ 0.0s Python

| | P_ID | Survived | P_Class | 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 |

Generate + Code + Markdown

```
df.sample()
```

✓ 0.0s Python

| | P_ID | Survived | P_Class | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 371 | 372 | 0 | 3 | Wiklund, Mr. Jakob Alfred | male | 18.0 | 1 | 0 | 3101267 | 6.4958 | NaN | S |

```
df.shape
```

✓ 0.0s Python

```
(891, 12)
```

```
df.columns
```

✓ 0.0s Python

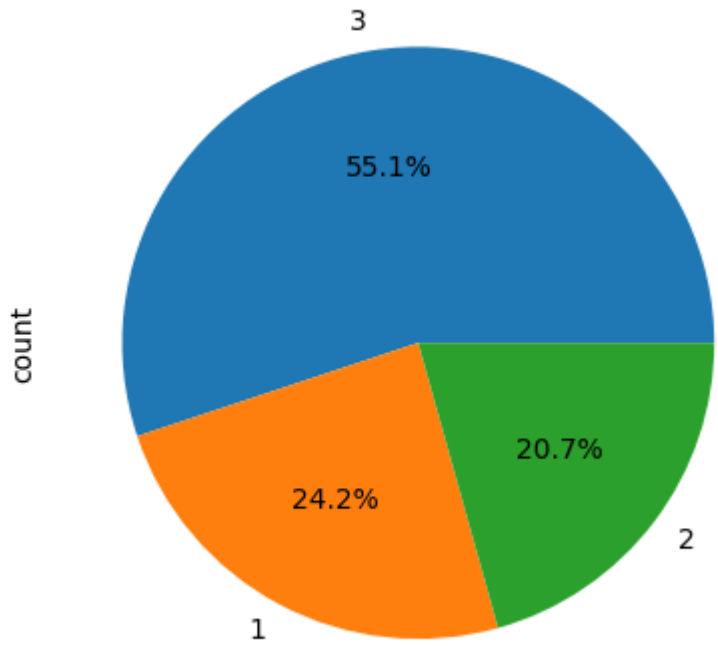
```
Index(['P_ID', 'Survived', 'P_Class', 'Name', 'Sex', 'Age', 'SibSp', 'Parch',
       'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

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

✓ 0.0s Python

```
P_ID          0
Survived      0
P_Class       0
Name          0
Sex           0
Age         177
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin       687
Embarked      2
dtype: int64
```

```
df['P_Class'].value_counts().plot(kind='pie', autopct='%1.1f%%')
```



```
df.nunique()
```

✓ 0.0s Python

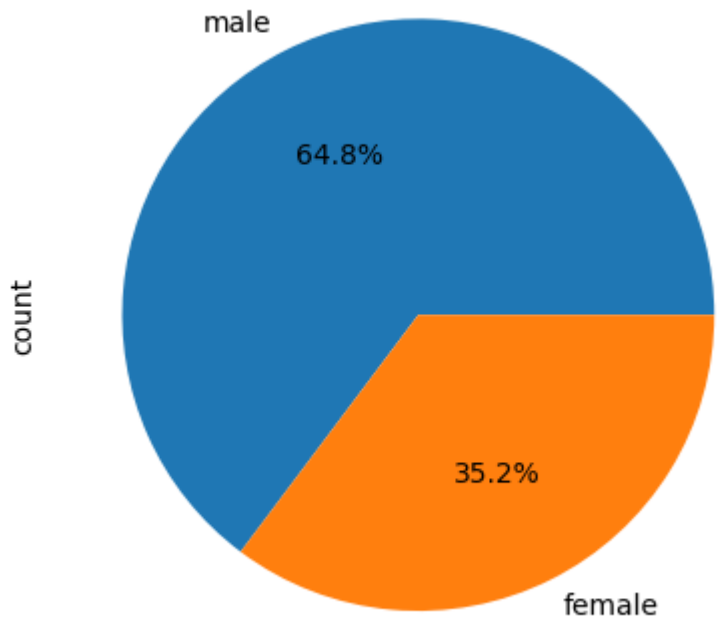
```
P_ID        891
Survived      2
P_Class       3
Name        891
Sex           2
Age          88
SibSp         7
Parch         7
Ticket      681
Fare        248
Cabin       147
Embarked      3
dtype: int64
```

```
df['Sex'].value_counts()
```

✓ 0.0s Python

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

```
df['Sex'].value_counts().plot(kind='pie', autopct='%1.1f%%')
```



```
df.sort_values('Age', ascending=False).head()
```

[27] ✓ 0.0s Python

| | P_ID | Survived | P_Class | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 630 | 631 | 1 | 1 | Barkworth, Mr. Algernon Henry Wilson | male | 80.0 | 0 | 0 | 27042 | 30.0000 | A23 | S |
| 851 | 852 | 0 | 3 | Svensson, Mr. Johan | male | 74.0 | 0 | 0 | 347060 | 7.7750 | NaN | S |
| 493 | 494 | 0 | 1 | Artagaveytia, Mr. Ramon | male | 71.0 | 0 | 0 | PC 17609 | 49.5042 | NaN | C |
| 96 | 97 | 0 | 1 | Goldschmidt, Mr. George B | male | 71.0 | 0 | 0 | PC 17754 | 34.6542 | A5 | C |
| 116 | 117 | 0 | 3 | Connors, Mr. Patrick | male | 70.5 | 0 | 0 | 370369 | 7.7500 | NaN | Q |

```
df.describe(include='all')
```

[28] ✓ 0.0s Python

| | P_ID | Survived | P_Class | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 891.000000 | 891.000000 | 891.000000 | 891 | 891 | 714.000000 | 891.000000 | 891.000000 | 891 | 891.000000 | 204 | 889 |
| unique | NaN | NaN | NaN | 891 | 2 | NaN | NaN | NaN | 681 | NaN | 147 | 3 |
| top | NaN | NaN | NaN | Braund, Mr. Owen Harris | male | NaN | NaN | NaN | 347082 | NaN | B96 B98 | S |
| freq | NaN | NaN | NaN | 1 | 577 | NaN | NaN | NaN | 7 | NaN | 4 | 644 |
| mean | 446.000000 | 0.383838 | 2.308642 | NaN | NaN | 29.699118 | 0.523008 | 0.381594 | NaN | 32.204208 | NaN | NaN |
| std | 257.353842 | 0.486592 | 0.836071 | NaN | NaN | 14.526497 | 1.102743 | 0.806057 | NaN | 49.693429 | NaN | NaN |
| min | 1.000000 | 0.000000 | 1.000000 | NaN | NaN | 0.420000 | 0.000000 | 0.000000 | NaN | 0.000000 | NaN | NaN |
| 25% | 223.500000 | 0.000000 | 2.000000 | NaN | NaN | 20.125000 | 0.000000 | 0.000000 | NaN | 7.910400 | NaN | NaN |
| 50% | 446.000000 | 0.000000 | 3.000000 | NaN | NaN | 28.000000 | 0.000000 | 0.000000 | NaN | 14.454200 | NaN | NaN |
| 75% | 668.500000 | 1.000000 | 3.000000 | NaN | NaN | 38.000000 | 1.000000 | 0.000000 | NaN | 31.000000 | NaN | NaN |
| max | 891.000000 | 1.000000 | 3.000000 | NaN | NaN | 80.000000 | 8.000000 | 6.000000 | NaN | 512.329200 | NaN | NaN |

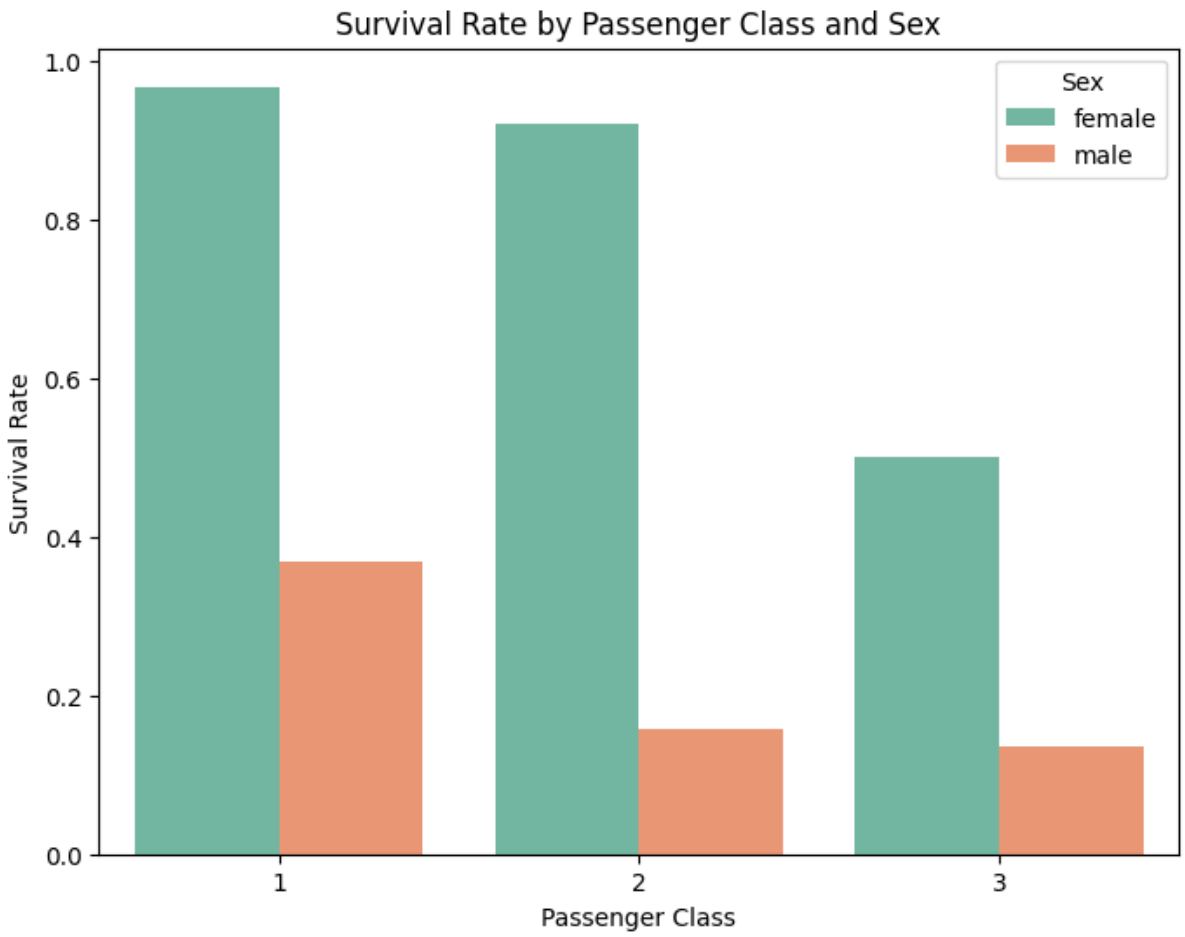
## Perform the analysis on the basis of which gender is sitting in which class

```
# Groupby analysis: Survival rate by P_Class and Sex
grouped = df.groupby(['P_Class', 'Sex'])['Survived'].mean().reset_index()
print('Survival rate by P_Class and Sex:')
print(grouped)
```

[1] ✓ 0.0s Python

```
Survival rate by P_Class and Sex:
   P_Class     Sex  Survived
0        1  female  0.968085
1        1    male  0.368852
2        2  female  0.921053
3        2    male  0.157407
4        3  female  0.500000
5        3    male  0.135447
```

```
# Plotting the survival rate by P_Class and Sex

plt.figure(figsize=(8,6))
sns.barplot(data=grouped, x='P_Class', y='Survived', hue='Sex',
palette='Set2')
plt.title('Survival Rate by Passenger Class and Sex')
plt.xlabel('Passenger Class')
plt.ylabel('Survival Rate')
plt.legend(title='Sex')
plt.show()
```



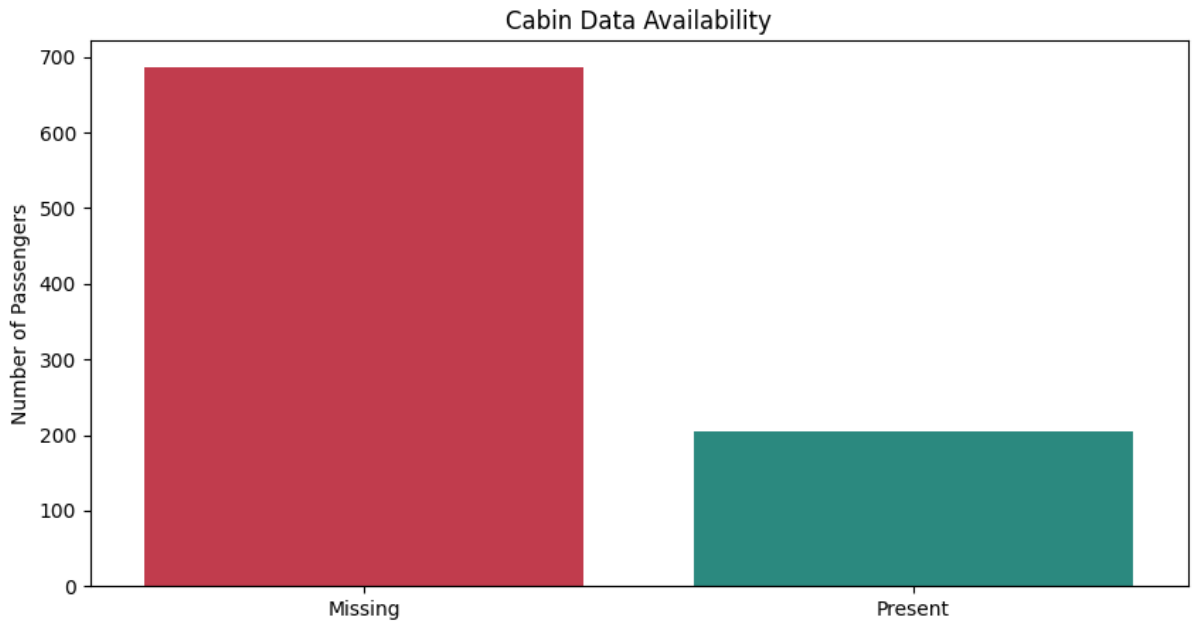

## Perform the analysis on the basis of Cabin and Passenger

```
df['Cabin'].unique()
```

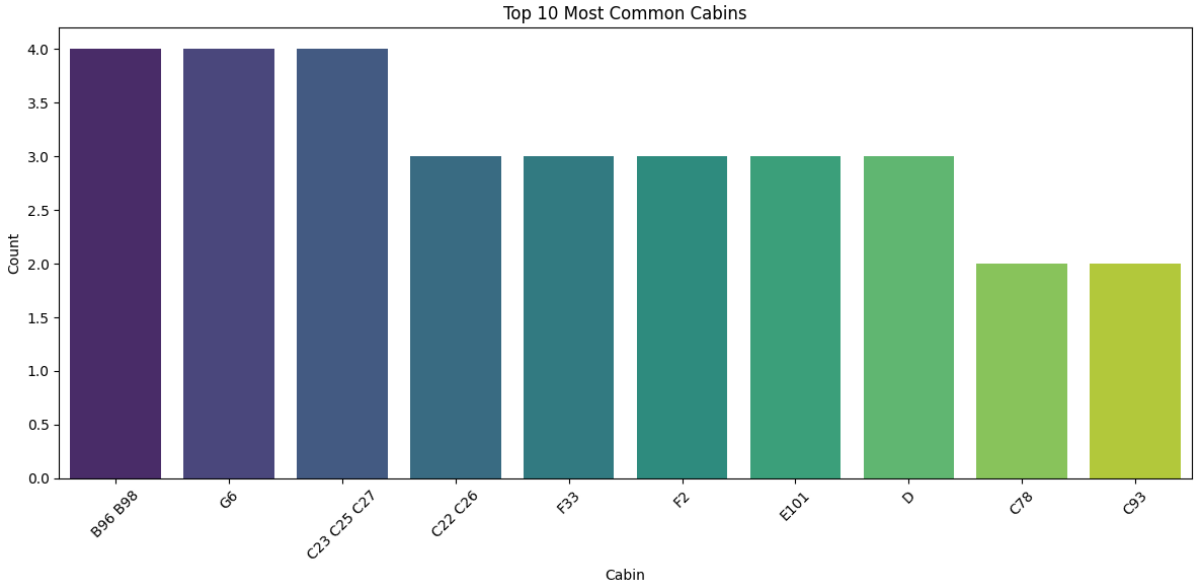
[13] ✓ 0.0s Python

```
array([nan, 'C85', 'C123', 'E46', 'G6', 'C103', 'D56', 'A6',
       'C23 C25 C27', 'B78', 'D33', 'B30', 'C52', 'B28', 'C83', 'F33',
       'F G73', 'E31', 'A5', 'D10 D12', 'D26', 'C110', 'B58 B60', 'E101',
       'F E69', 'D47', 'B86', 'F2', 'C2', 'E33', 'B19', 'A7', 'C49', 'F4',
       'A32', 'B4', 'B80', 'A31', 'D36', 'D15', 'C93', 'C78', 'D35',
       'C87', 'B77', 'E67', 'B94', 'C125', 'C99', 'C118', 'D7', 'A19',
       'B49', 'D', 'C22 C26', 'C106', 'C65', 'E36', 'C54',
       'B57 B59 B63 B66', 'C7', 'E34', 'C32', 'B18', 'C124', 'C91', 'E40',
       'T', 'C128', 'D37', 'B35', 'E50', 'C82', 'B96 B98', 'E10', 'E44',
       'A34', 'C104', 'C111', 'C92', 'E38', 'D21', 'E12', 'E63', 'A14',
       'B37', 'C30', 'D20', 'B79', 'E25', 'D46', 'B73', 'C95', 'B38',
       'B39', 'B22', 'C86', 'C70', 'A16', 'C101', 'C68', 'A10', 'E68',
       'B41', 'A20', 'D19', 'D50', 'D9', 'A23', 'B50', 'A26', 'D48',
       'E58', 'C126', 'B71', 'B51 B53 B55', 'D49', 'B5', 'B20', 'F G63',
       'C62 C64', 'E24', 'C90', 'C45', 'E8', 'B101', 'D45', 'C46', 'D30',
       'E121', 'D11', 'E77', 'F38', 'B3', 'D6', 'B82 B84', 'D17', 'A36',
       'B102', 'B69', 'E49', 'C47', 'D28', 'E17', 'A24', 'C50', 'B42',
       'C148'], dtype=object)
```

```
# Graphical presentation of Cabin data
plt.figure(figsize=(10,5))
cabin_missing = df['Cabin'].isnull().sum()
cabin_present = df['Cabin'].notnull().sum()
sns.barplot(x=['Missing', 'Present'], y=[cabin_missing, cabin_present],
palette=['#d7263d', '#1b998b'])
plt.title('Cabin Data Availability')
plt.ylabel('Number of Passengers')
plt.show()
```

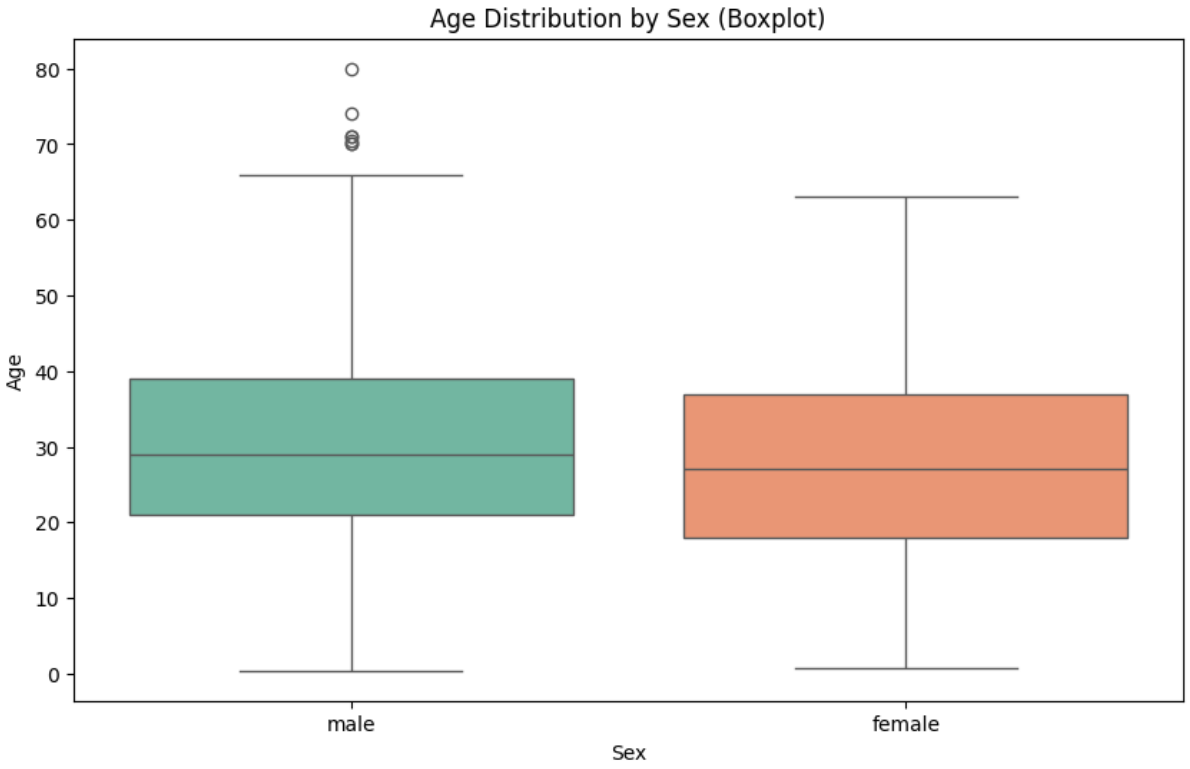



```
# Most common cabins (excluding missing)
cabin_counts = df['Cabin'].value_counts().head(10)
plt.figure(figsize=(12,6))
sns.barplot(x=cabin_counts.index, y=cabin_counts.values, palette='viridis')
plt.title('Top 10 Most Common Cabins')
plt.xlabel('Cabin')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```




## Perform the analysis on the basis of Age and Sex using Boxplot and Violinplot

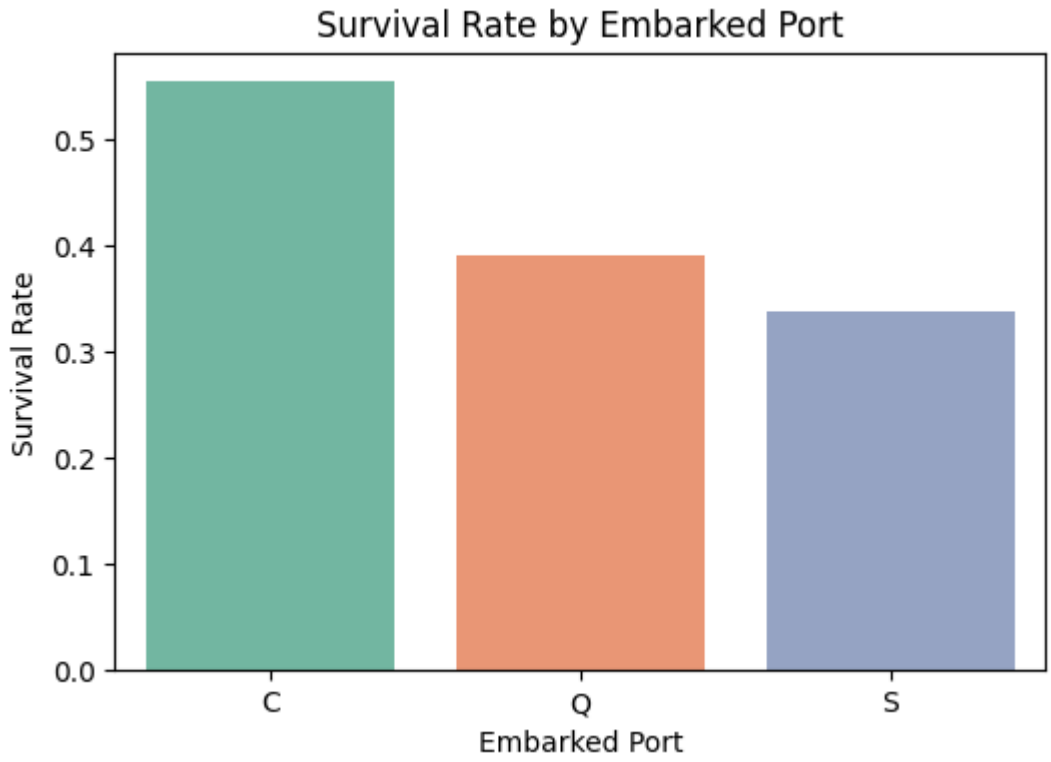
```
# Age distribution by Sex: Boxplot and Violinplot
plt.figure(figsize=(10,6))
sns.boxplot(data=df, x='Sex', y='Age', palette='Set2')
plt.title('Age Distribution by Sex (Boxplot)')
plt.xlabel('Sex')
plt.ylabel('Age')
plt.show()

plt.figure(figsize=(10,6))
sns.violinplot(data=df, x='Sex', y='Age', palette='Set2')
plt.title('Age Distribution by Sex (Violinplot)')
plt.xlabel('Sex')
plt.ylabel('Age')
plt.show()
```


Age Distribution by Sex (Boxplot)
Age
Sex
male
female
0
10
20
30
40
50
60
70
80


Age Distribution by Sex (Violinplot)
Age
Sex
male
female
0
20
40
60
80

## Perform the analysis on the basis of how much Survivals on the Embarked Port using Barplot

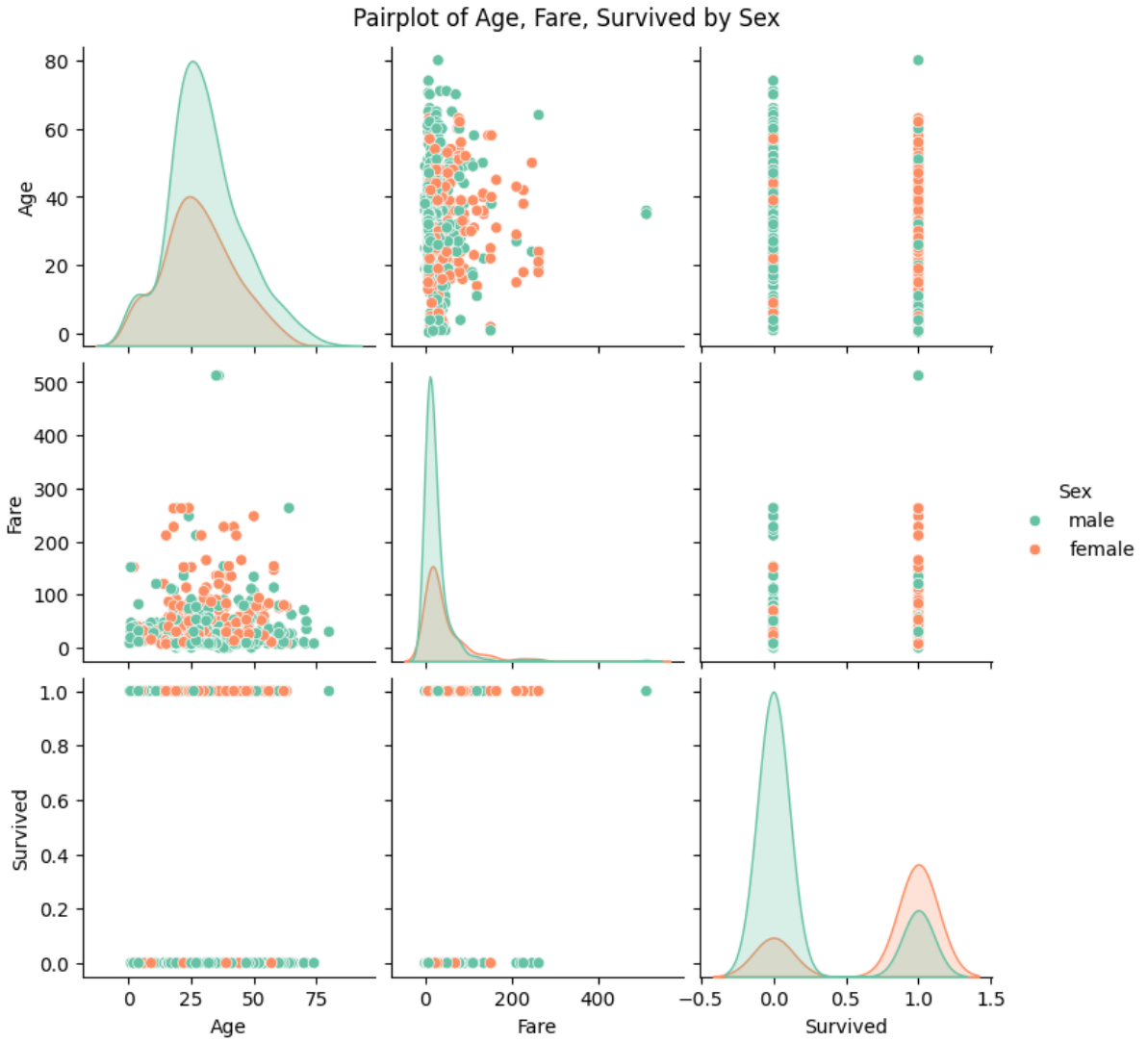
```
# Survival rate by Embarked port
embarked_survival = df.groupby('Embarked')['Survived'].mean().reset_index()
print('Survival rate by Embarked port:')
print(embarked_survival)
plt.figure(figsize=(6,4))
sns.barplot(data=embarked_survival, x='Embarked', y='Survived',
palette='Set2')
plt.title('Survival Rate by Embarked Port')
plt.ylabel('Survival Rate')
plt.xlabel('Embarked Port')
plt.show()
```




## Perform the analysis on the basis of which Age group , whats's the Fare and How much Survived using Pairplot

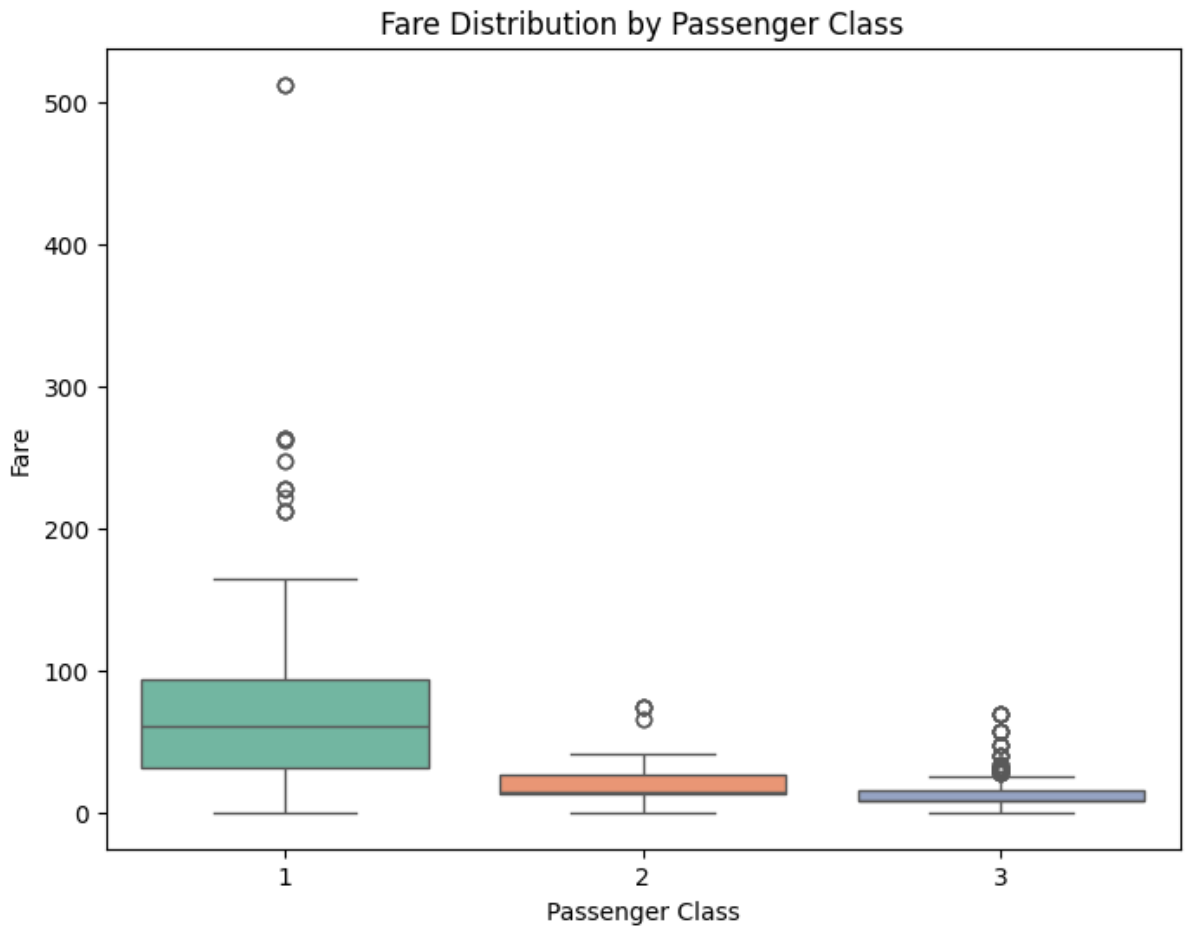


```
# Pairplot for selected features
sns.pairplot(df, vars=['Age', 'Fare', 'Survived'], hue='Sex', palette='Set2')
plt.suptitle('Pairplot of Age, Fare, Survived by Sex', y=1.02)
plt.show()
```



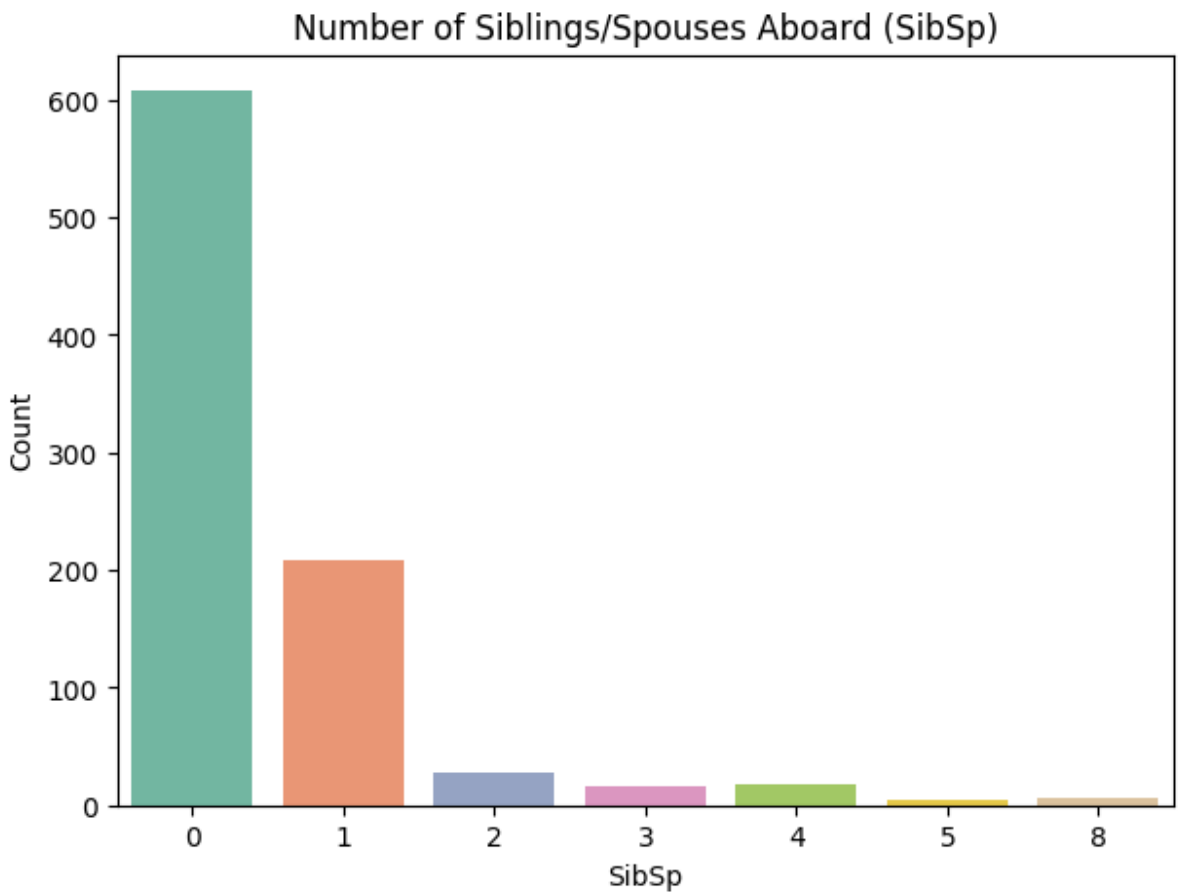

Perform the analysis on the basis of Passenger Class and how much Fare it is paying using Boxplot

```
# Distribution of Fare by Passenger Class
plt.figure(figsize=(8,6))
sns.boxplot(data=df, x='P_Class', y='Fare', palette='Set2')
plt.title('Fare Distribution by Passenger Class')
plt.xlabel('Passenger Class')
plt.ylabel('Fare')
plt.show()
```




## Perform the analysis on the basis of Counting the SibSp(Siblings/Spouses) using Countplot

```
# Countplot for number of siblings/spouses aboard (SibSp)
plt.figure(figsize=(7,5))
sns.countplot(data=df, x='SibSp', palette='Set2')
plt.title('Number of Siblings/Spouses Aboard (SibSp)')
plt.xlabel('SibSp')
plt.ylabel('Count')
plt.show()
```


Number of Siblings/Spouses Aboard (SibSp)
Count
600
500
400
300
200
100
0
0
1
2
3
4
5
8
SibSp