Problem Statement:

You are given a dataset containing information about the passengers of the Titanic. Your task is to perform exploratory data analysis (EDA) on the dataset using the following methods: ● Load the dataset(Titanic.csv) into a pandas dataframe and print the first 5 rows using the head() method. ● Use the info() method to get information about the dataset. In particular, pay attention to the data types of each column and whether there are any missing values. ● Use the describe() method to describe the dataset. Pay attention to the range of values for each numeric column and whether there are any outliers. ● Use the groupby() method to group the data based on the Pclass column and calculate the mean fare for each class. ● Use the value_counts() method to get the frequency count of the Embarked column. ● Create a new column in the dataframe called AgeRange that categorises passengers into age ranges: "Child" for ages 0-12, "Teen" for ages 13-19, "Adult" for ages 20-59, and "Senior" for ages 60 and above. ● Use the pivot_table() method to create a pivot table that shows the survival rate of passengers based on their sex, class, and age range. ● Create a bar chart that shows the total number of passengers in each age range. ● Create a scatter plot that shows the relationship between age and fare. Color the points based on whether the passenger survived or not.

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

# Load the dataset into a pandas DataFrame
df=pd.read_csv("Titanic.csv")

# Print the first 5 rows using the head() method
df.head(5)
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age
SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                             Heikkinen, Miss. Laina  female  26.0
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                           Allen, Mr. William Henry    male  35.0
```

```
   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

```
# Get information about the dataset using the info() method
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
```

Exploratory Data Analysis

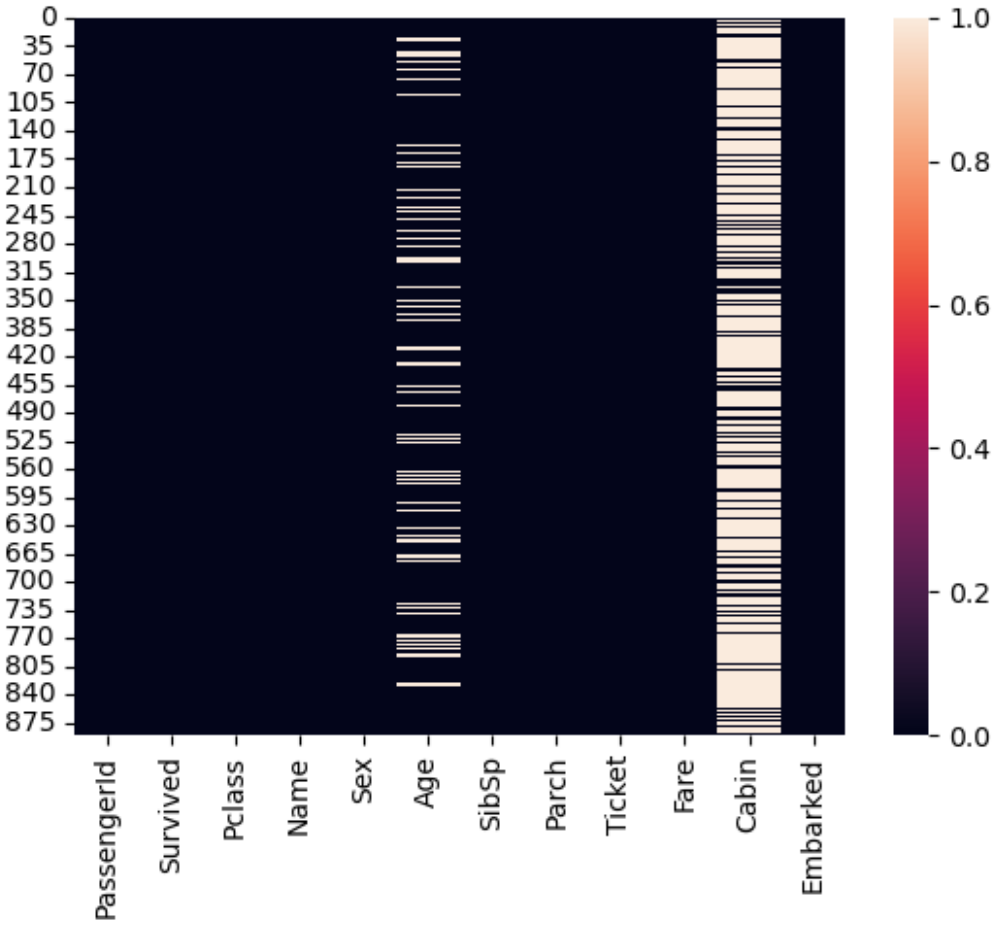
Missing Data

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

```
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
```

```
# Visualize missing data
sns.heatmap(df.isnull());
```




```
# Deal with missing data
# Filling missing Age values with median
median_age = df['Age'].median()
df['Age'].fillna(median_age, inplace=True)

# Filling missing Embarked values with mode
mode_embarked = df['Embarked'].mode()[0]
df['Embarked'].fillna(mode_embarked, inplace=True)
df.drop('Cabin', axis=1,inplace=True)
df.isna().sum()
```

```
PassengerId    0
Survived       0
Pclass         0
```

```
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

```
# Describe the dataset using the describe() method
df.describe()
```

```
       PassengerId    Survived      Pclass         Age        SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

            Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

```
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          891 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  Embarked     891 non-null    object
 11  AgeRange     891 non-null    category
dtypes: category(1), float64(2), int64(5), object(4)
memory usage: 77.8+ KB
```

```
# Group the data based on the Pclass column and calculate the mean
fare for each class
df.groupby('Pclass')['Fare'].mean()
```

```
Pclass
1    84.154687
2    20.662183
3    13.675550
Name: Fare, dtype: float64
```

```
# Get the frequency count of the Embarked column using value_counts()
method
df["Embarked"].value_counts()
```

```
S    646
C    168
Q     77
Name: Embarked, dtype: int64
```

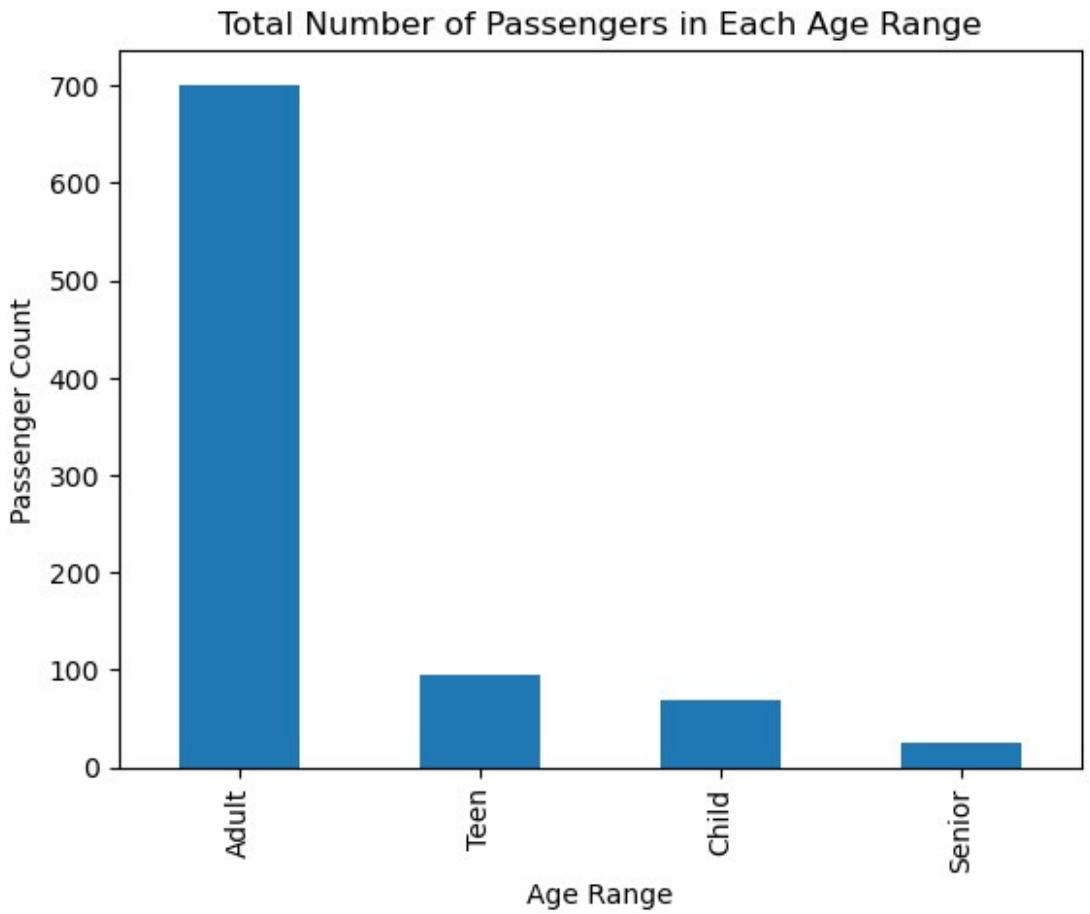
```
# Create a new column 'AgeRange' categorizing passengers into age
ranges
bins = [0, 12, 19, 59, 150]
labels = ['Child', 'Teen', 'Adult', 'Senior']
df['AgeRange'] = pd.cut(df['Age'], bins=bins, labels=labels)
```

```
# Create a pivot table showing the survival rate based on sex, class,
and age range
df.pivot_table('Survived', index=['Sex', 'Pclass', 'AgeRange'],
aggfunc='mean')
```

```
                        Survived
Sex    Pclass AgeRange
female 1      Child     0.000000
               Teen      1.000000
               Adult     0.974026
               Senior    1.000000
       2      Child     1.000000
               Teen      1.000000
               Adult     0.900000
       3      Child     0.478261
               Teen      0.590909
               Adult     0.479592
               Senior    1.000000
male   1      Child     1.000000
               Teen      0.250000
               Adult     0.386139
```

```
              Senior    0.142857
       2      Child     1.000000
              Teen      0.100000
              Adult     0.070588
              Senior    0.250000
       3      Child     0.360000
              Teen      0.078947
              Adult     0.125000
              Senior    0.000000
```

```
# Create a bar chart showing the total number of passengers in each
age range
age_range_count = df['AgeRange'].value_counts()
age_range_count.plot(kind='bar', xlabel='Age Range', ylabel='Passenger
Count')
plt.title('Total Number of Passengers in Each Age Range')
plt.show()
```




```
# Create a scatter plot showing the relationship between age and fare,
colored by survival
plt.scatter(df['Age'], df['Fare'], c=df['Survived'], cmap='viridis')
```

```
plt.xlabel('Age')
plt.ylabel('Fare')
plt.title('Relationship between Age and Fare (Colored by Survival)')
plt.colorbar(label='Survived')
plt.show()
```