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

```
#importing liberaries..
data=pd.read_csv(r"C:\Users\avani\Downloads\train.csv")
data.head()
```

Out[25]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500

```
data.shape
```

Out[6]: (891, 12)

```
data.dtypes
```

Out[7]:
```
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
```

```
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
```

```
data.nunique()
```

Out[9]:
```
PassengerId    891
Survived         2
Pclass           3
Name           891
Sex              2
Age             88
SibSp            7
Parch            7
Ticket         681
Fare           248
Cabin          147
Embarked         3
dtype: int64
```

```
data.describe()
```

Out[10]:

	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

```
#data cleanig
data.isnull().sum()
```

Out[15]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

```
data.isnull().sum(axis=1).sort_values(ascending=False)
```

Out[14]:
```
1      0
571    0
577    0
581    0
583    0
      ..
327    0
329    0
331    0
332    0
889    0
Length: 183, dtype: int64
```

```
data.dropna(inplace=True)
```

```
data['Survived'].value_counts()
```

Out[21]:
```
Survived
1    123
0     60
Name: count, dtype: int64
```

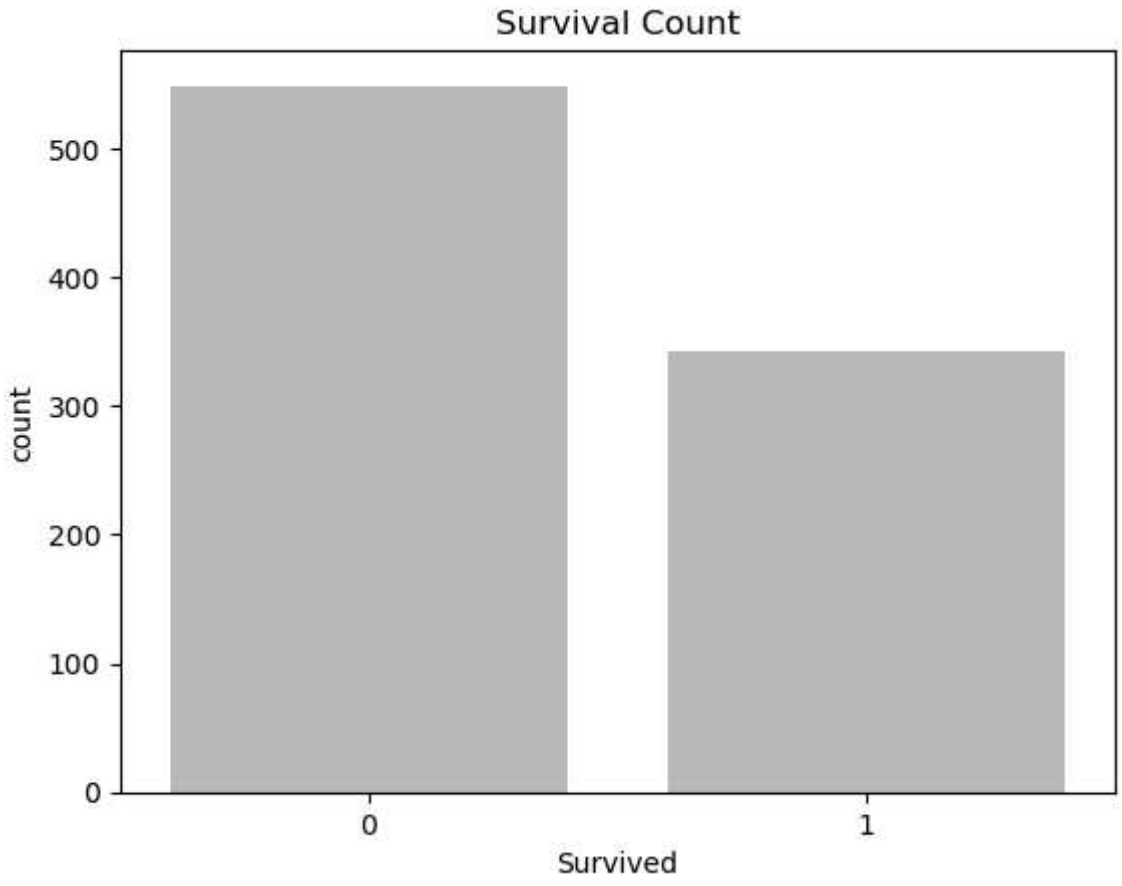
```
len(data[data.duplicated()])
```

Out[17]: 0

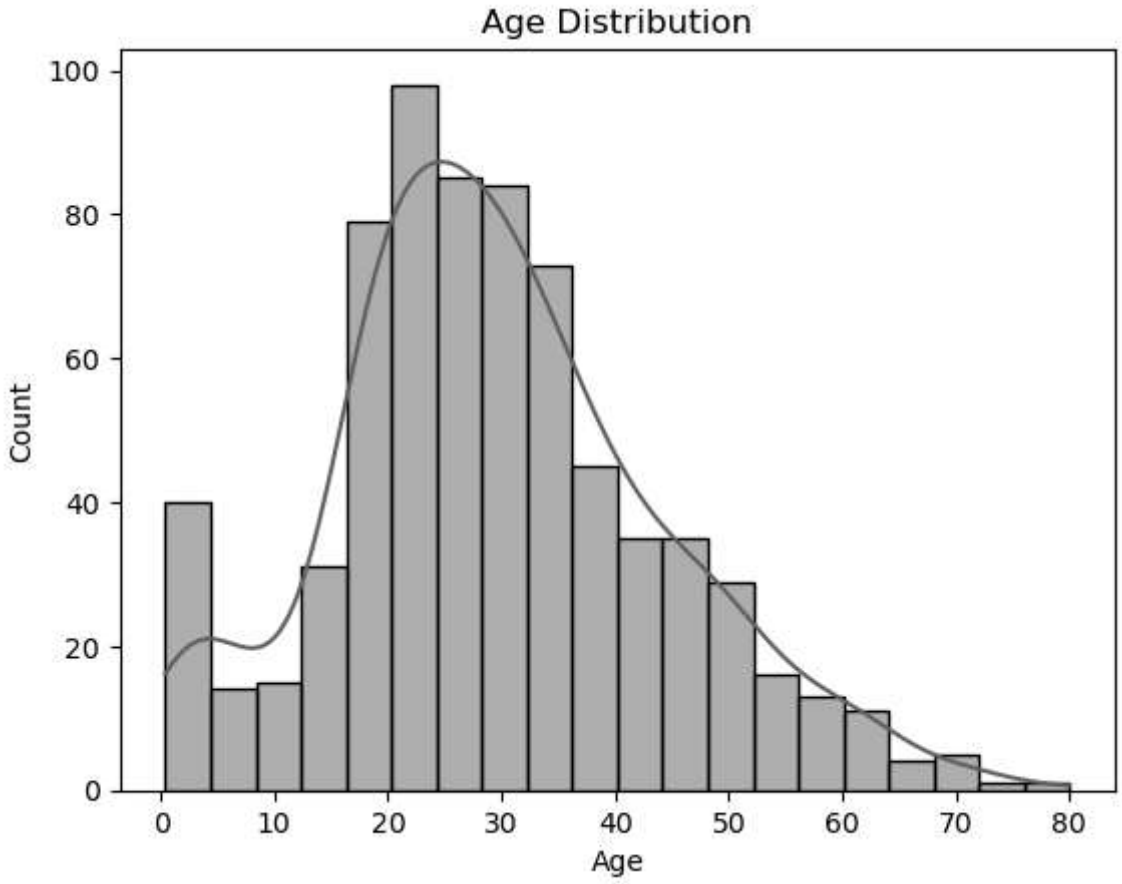
```
len(data.Name.unique())
```

Out[18]: 183

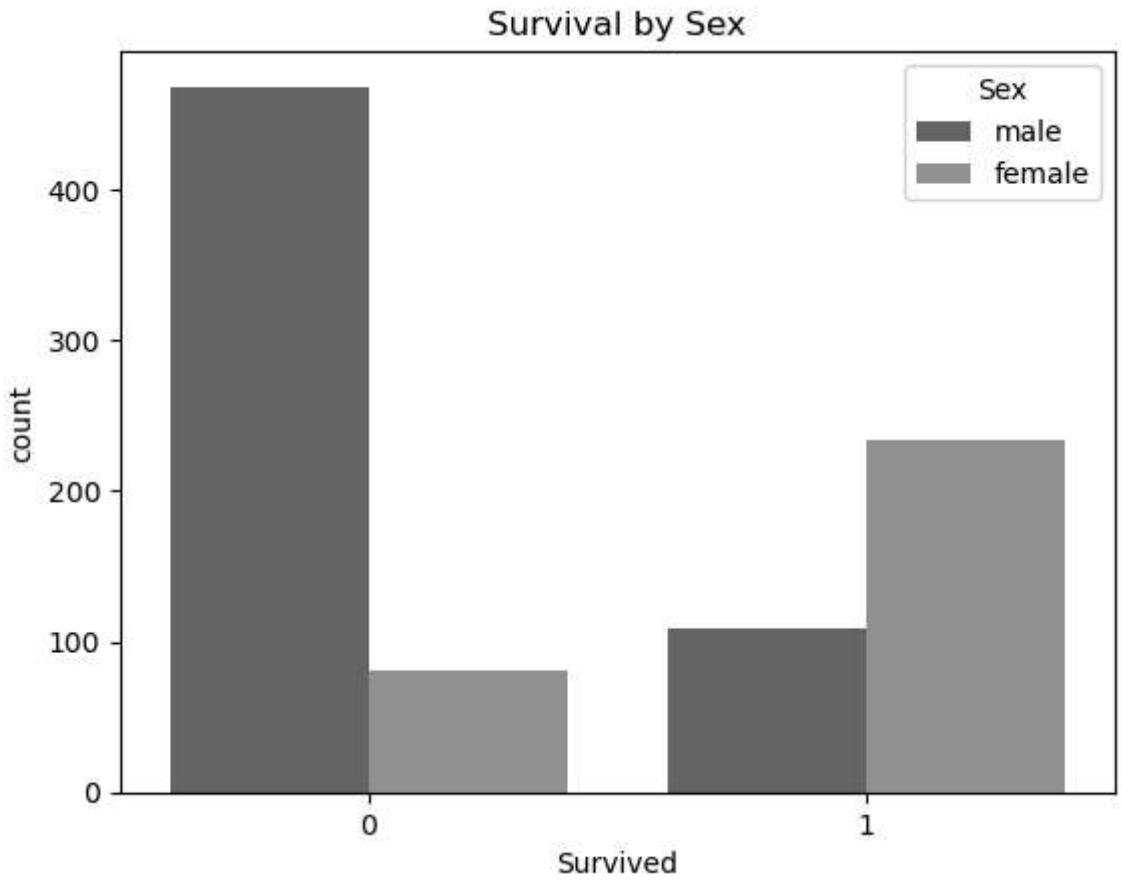
```
#data analysis (univarite analysis)
# Countplot of Survived
sns.countplot(color="skyblue",x='Survived',data=data)
plt.title("Survival Count")
plt.show()
```



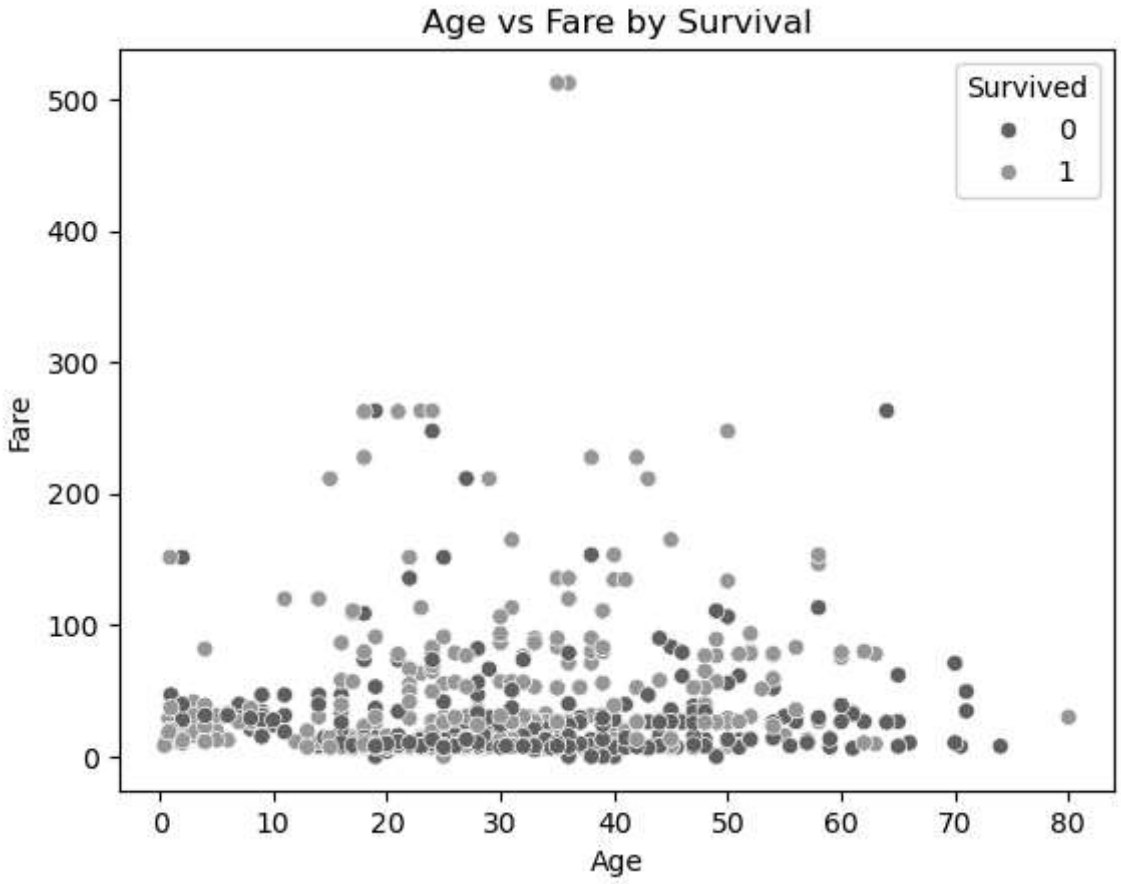

```
# Distribution of Age
sns.histplot(data['Age'].dropna(),kde=True)
plt.title("Age Distribution")
plt.show()
```




```
#Bivariate Analysis
# Survival by Sex
sns.countplot(x='Survived', hue='Sex', data=data)
plt.title("Survival by Sex")
plt.show()
```

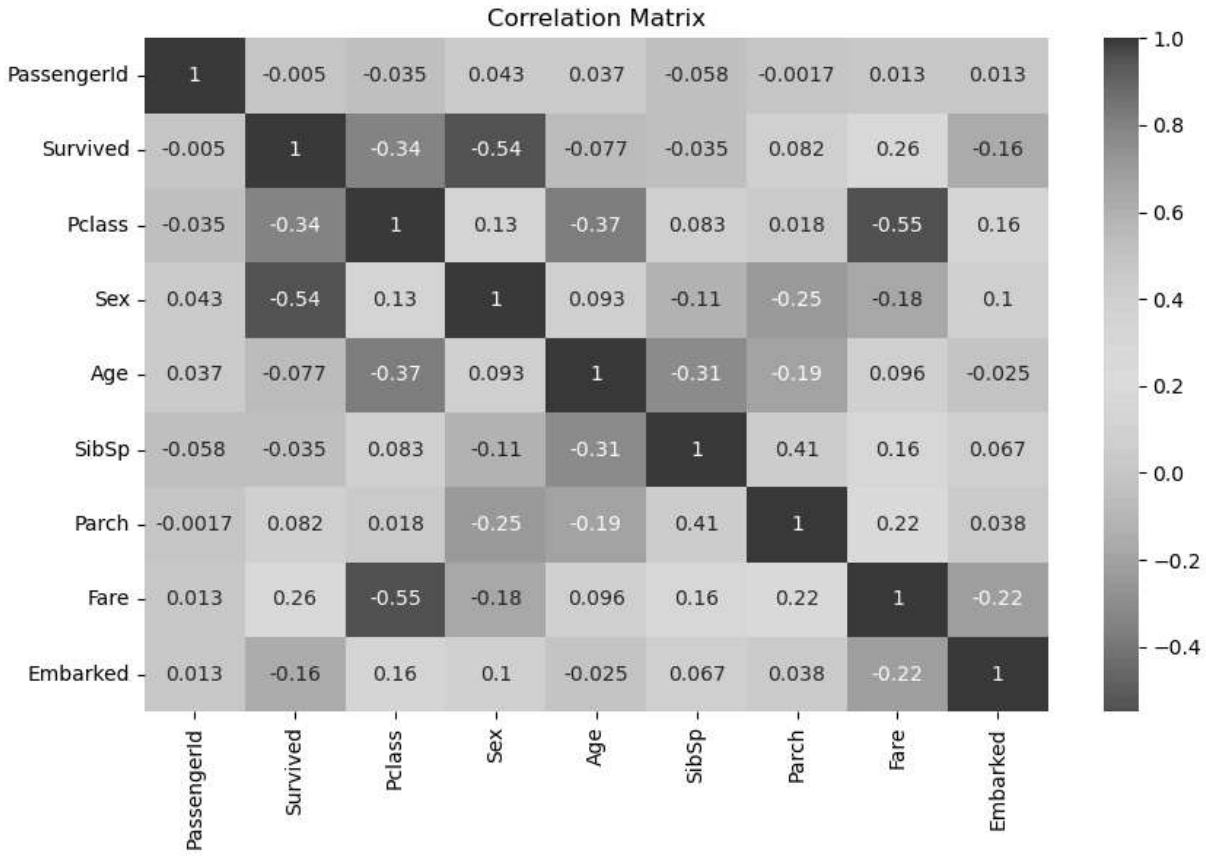



```
# Age vs Fare
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=data)
plt.title("Age vs Fare by Survival")
plt.show()
```

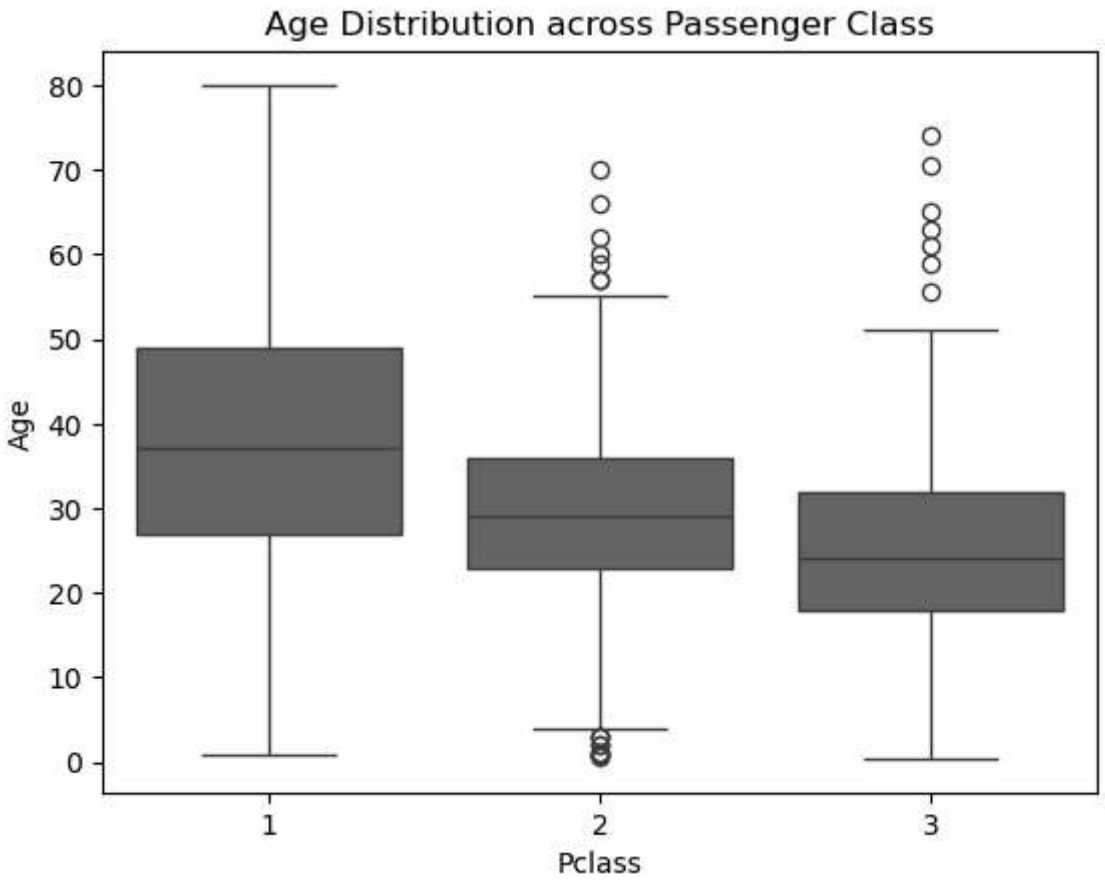



```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data['Sex'] = le.fit_transform(data['Sex'])
data['Embarked'] = le.fit_transform(data['Embarked'].astype(str))
```

```
#correlation heatmap
plt.figure(figsize=(10,6))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```



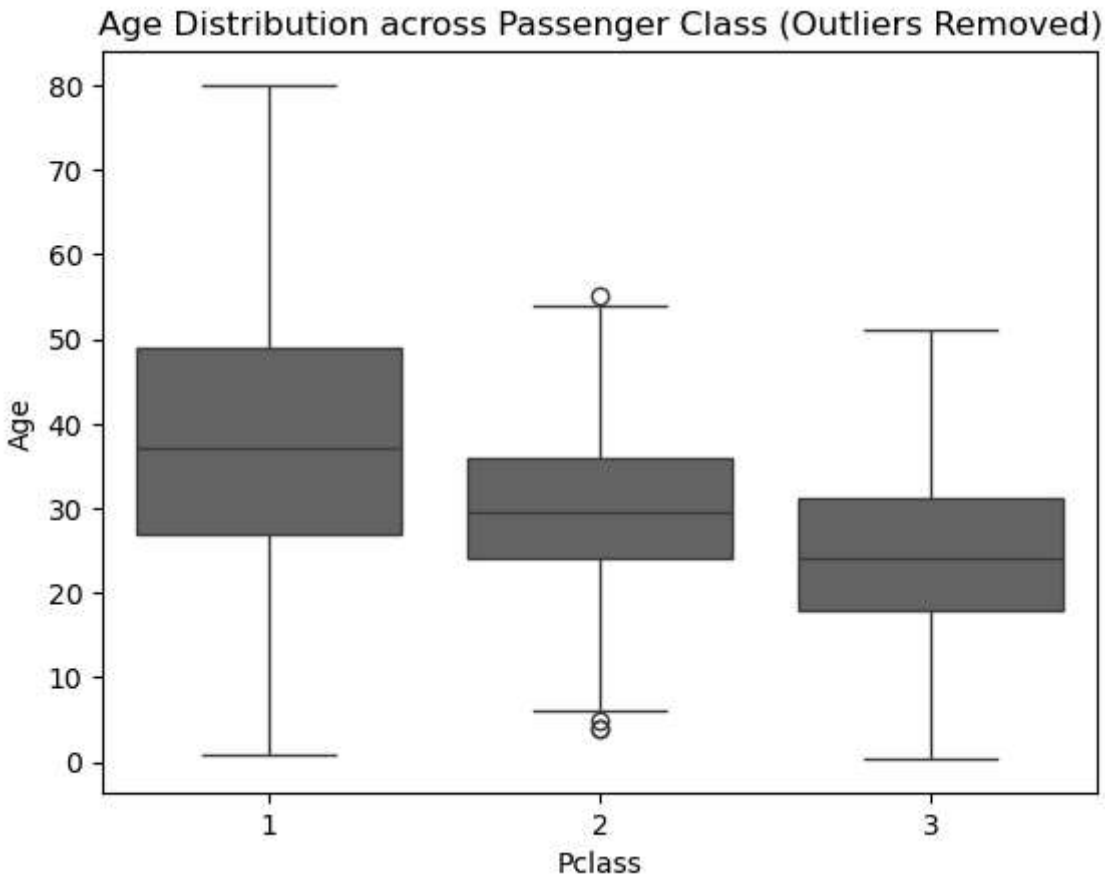

```
 #. Boxplots & Pairplots
# Boxplot of Age vs Pclass
sns.boxplot(x='Pclass', y='Age', data=data)
plt.title("Age Distribution across Passenger Class")
plt.show()
```



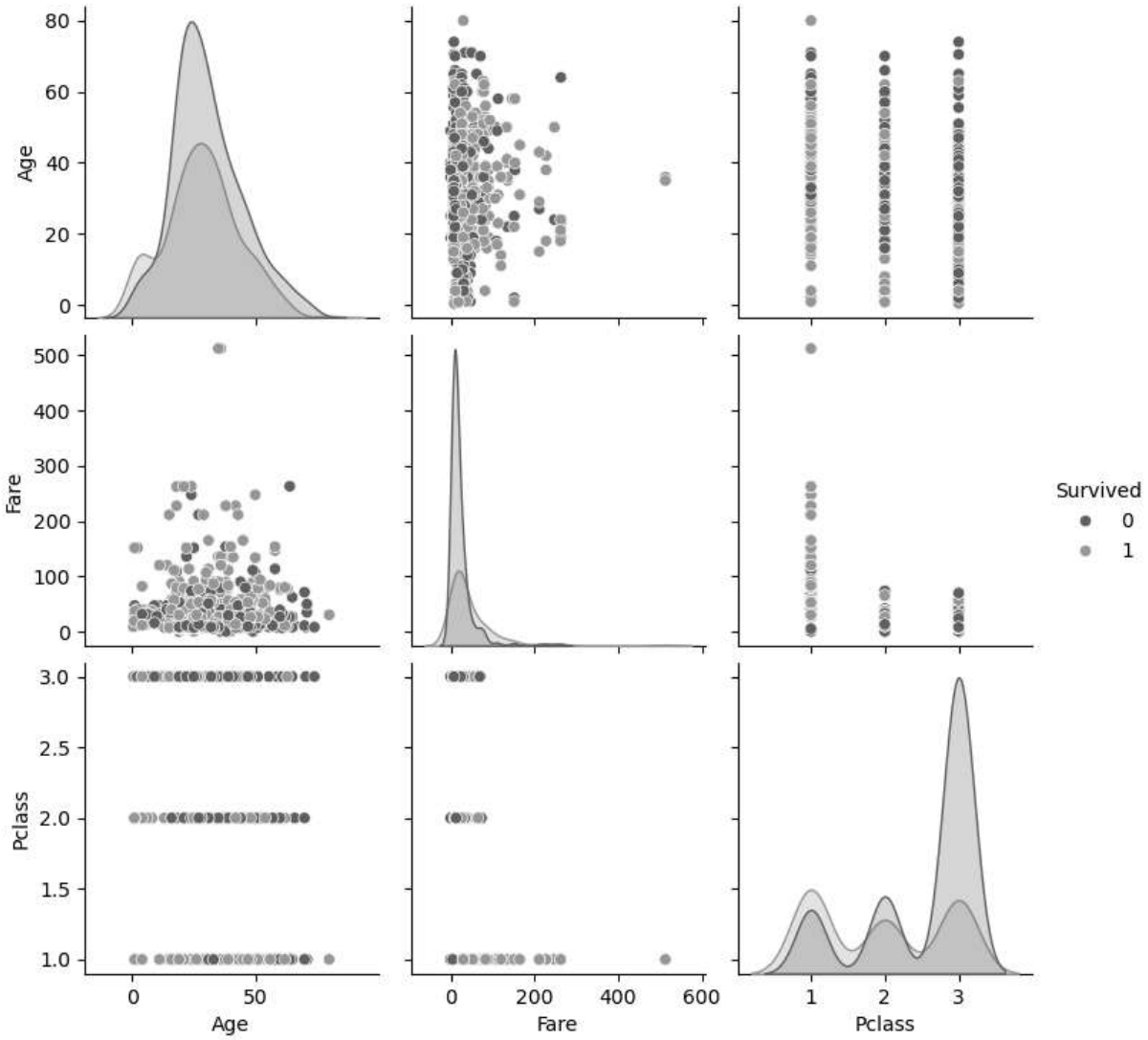

```
  def remove_outliers_iqr(data, group_col, target_col):
    cleaned_data = pd.DataFrame()

    for group in data[group_col].unique():
        group_data = data[data[group_col] == group]
        Q1 = group_data[target_col].quantile(0.25)
        Q3 = group_data[target_col].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        filtered_data = group_data[
            (group_data[target_col] >= lower_bound) & (group_data[target_col] <= upper_bound)
        ]
        cleaned_data = pd.concat([cleaned_data, filtered_data])

    return cleaned_data
```

```
data_cleaned = remove_outliers_iqr(data, 'Pclass', 'Age')
```

```
sns.boxplot(x='Pclass', y='Age', data=data_cleaned)
plt.title("Age Distribution across Passenger Class (Outliers Removed)")
plt.show()
```




```
# Pairplot of selected features
sns.pairplot(data[['Age', 'Fare', 'Pclass', 'Survived']], hue='Survived')
plt.show()
```


Age
Fare
Pclass
Survived
0
1
0
50
0
200
400
600
1
2
3