1 Titanic Data Analysis

1.0.1 import all required libraries

[3]:

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

1.0.2 import data using pandas

[8]:

```python
df=pd.read_csv("train.csv")
df
```

[8]:

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         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      0
..                                                 ...     ...   ...    ...
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
```

```
889                           Behr, Mr. Karl Howell    male  26.0      0
890                             Dooley, Mr. Patrick    male  32.0      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
..     ...               ...      ...    ...      ...
886      0            211536  13.0000    NaN        S
887      0            112053  30.0000    B42        S
888      2        W./C. 6607  23.4500    NaN        S
889      0            111369  30.0000   C148        C
890      0            370376   7.7500    NaN        Q

[891 rows x 12 columns]
```

1.0.3 Clean the Data

step-1 : check any null or duplicate values is there or not

[19]:

```
null=df.isnull().sum()
dup=df.duplicated().sum()
print("duplicates:",dup)
print("null values:",null)
```

```
duplicates:  0
null values: 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
```

[21]:

```
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
```

So there is no duplicate values
And there are Null values in age and cabin and embarked column we can drop beacause most of the values are null

[40]:

```python
df.fillna(df['Age'].mean(),inplace=True)
df.fillna(df['Embarked'].mode(),inplace=True)
#Drop the cabin table due to high null values
df.drop('Cabin',axis=1,inplace=True)
```

After Cleaning The Data

[43]:

```python
null=df.isnull().sum()
dup=df.duplicated().sum()
print("duplicates:",dup)
print("null values:",null)
```

```
duplicates:  0
null values: PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

1.1 Exploratory Data Analysis

1.1.1 Statistics

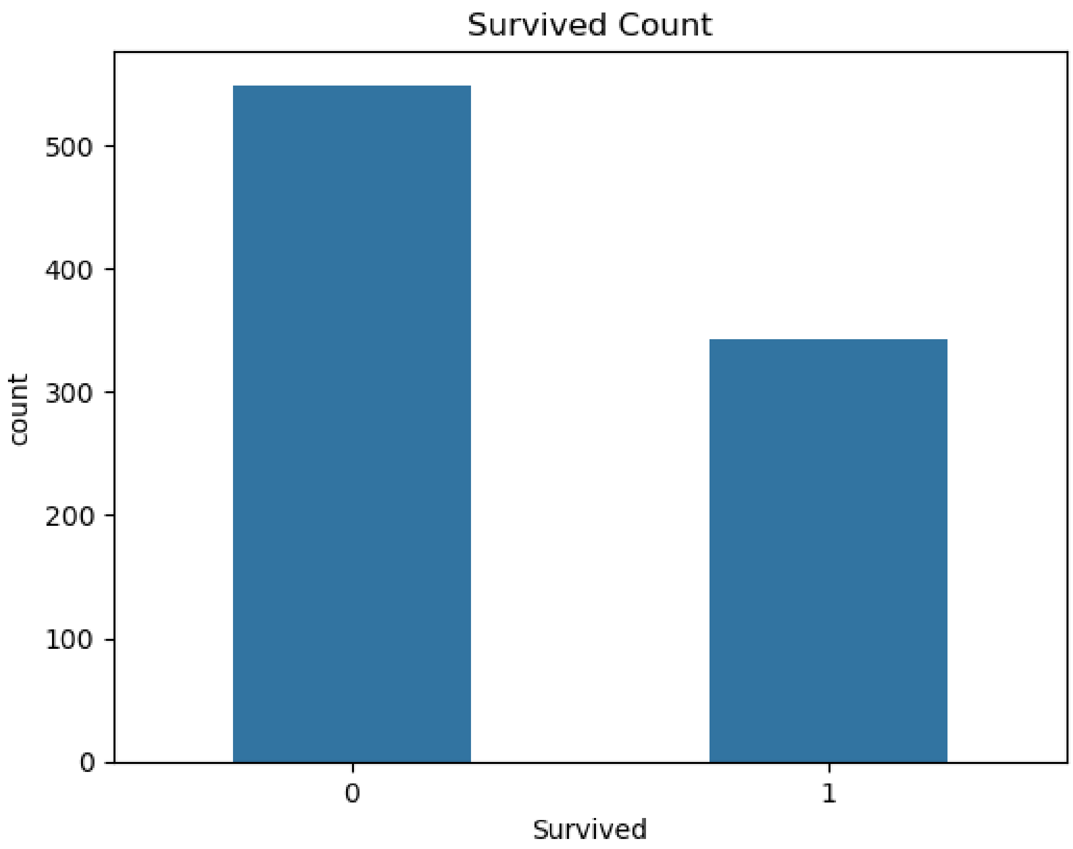
```
[47]: df.describe()
```

```
[47]:        PassengerId    Survived      Pclass         Age       SibSp  \
      count   891.000000  891.000000  891.000000  891.000000  891.000000
      mean    446.000000    0.383838    2.308642   29.699118    0.523008
      std     257.353842    0.486592    0.836071   13.002015    1.102743
      min       1.000000    0.000000    1.000000    0.420000    0.000000
      25%     223.500000    0.000000    2.000000   22.000000    0.000000
      50%     446.000000    0.000000    3.000000   29.699118    0.000000
      75%     668.500000    1.000000    3.000000   35.000000    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
```

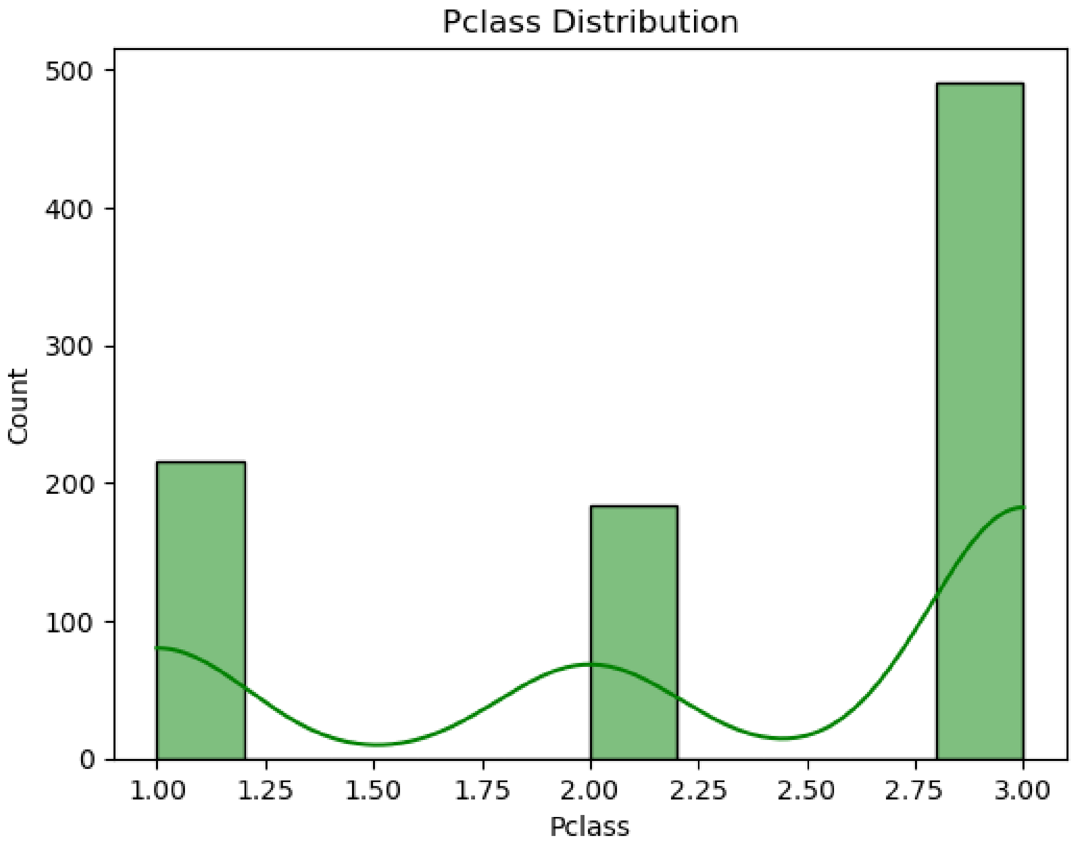
1. Survived Count

```
[54]: sns.countplot(x='Survived',data=df,width=0.5)
      plt.title('Survived Count')
      plt.show()
```



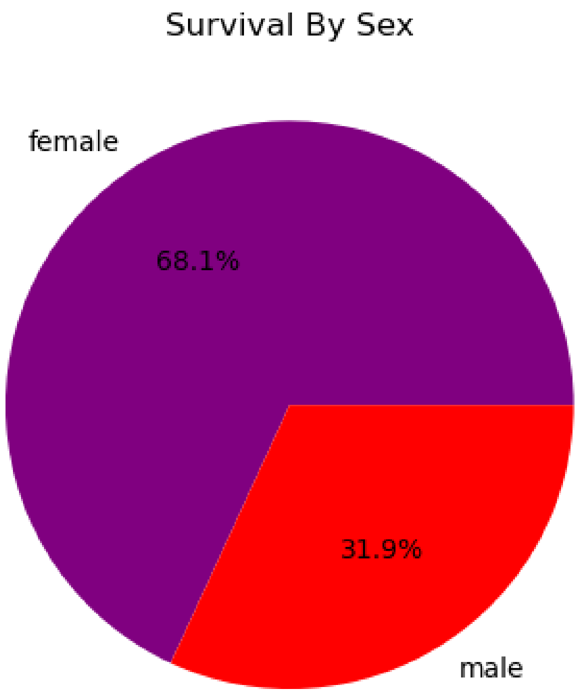
2. Pclass Distribution

[105]:
```
sns.histplot(x="Pclass",bins=10,data=df,kde=True,color='g')
plt.title("Pclass  Distribution")
plt.show()
```



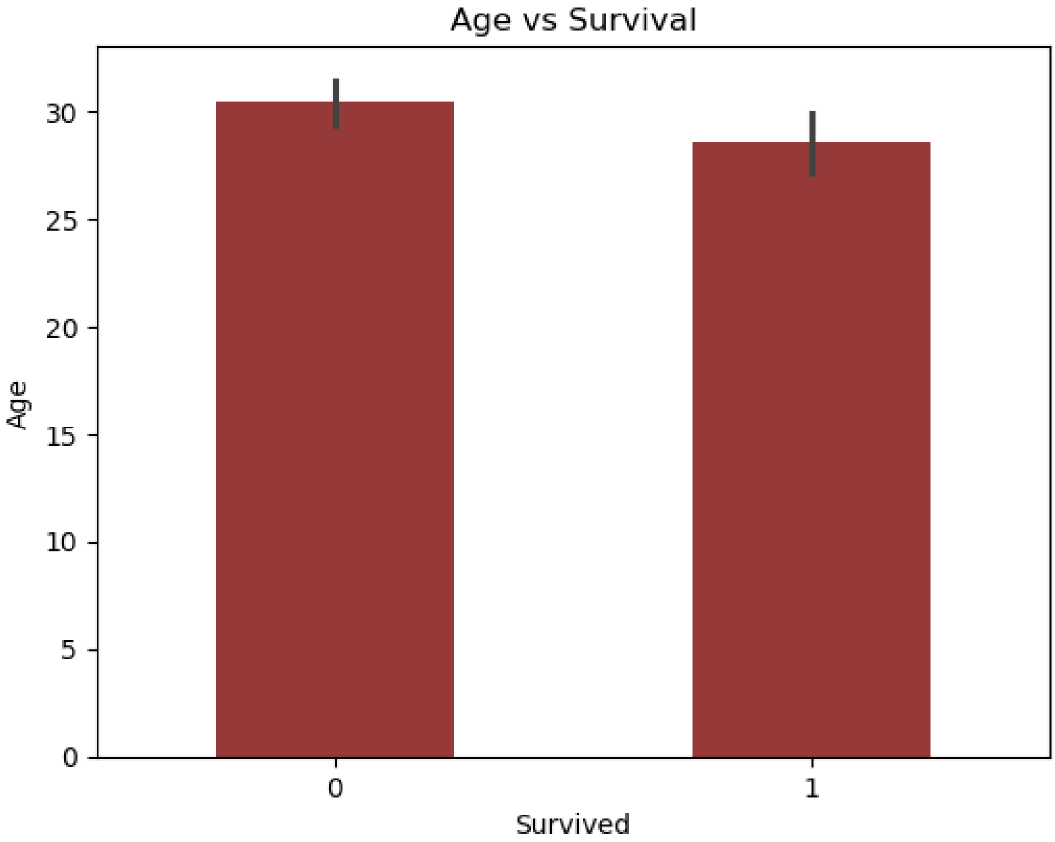
3.Survival By Sex

```
[95]: data=df.groupby('Sex')['Survived'].sum().reset_index()
plt.pie(data["Survived"],labels=data["Sex"],autopct='%1.1f%%',colors=['purple','red'])
plt.title('Survival By Sex')
plt.show()
```



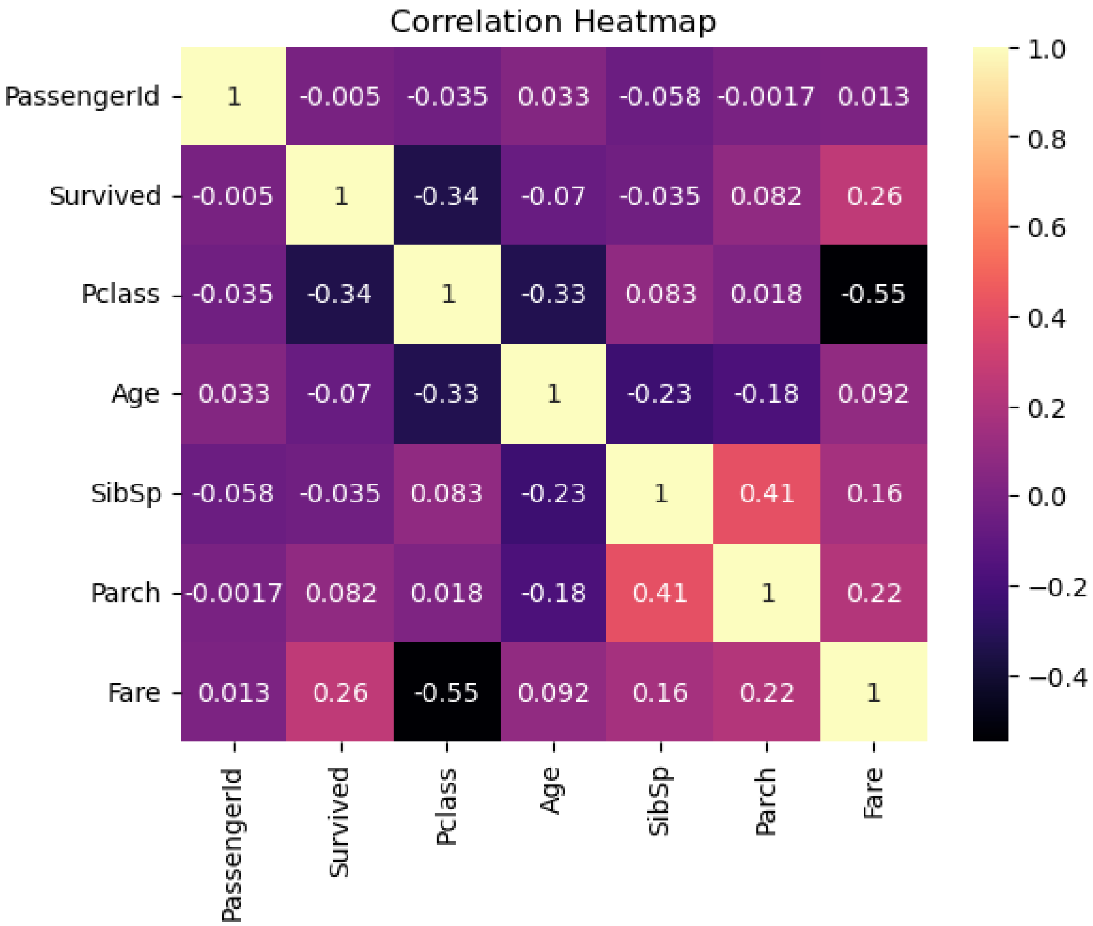
Age Vs Survival

[103]:
```
sns.barplot(x="Survived", y="Age", data=df,width=0.5,color="brown")
plt.title("Age vs Survival")
plt.show()
```



Correlation Analysis

[116]:
```
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="magma")
plt.title("Correlation  Heatmap")
plt.show()
```



1.1.2 Conclusion

**Key Insights Females had a higher survival rate.

1st class passengers had better chances of survival.

Younger passengers had slightly higher survival rates.

Strong correlation between Pclass and Fare