Task 2

EDA- Exploratory Data Analysis

Problem Statement :

Perform data cleaning and exploratory data analysis (EDA) on a dataset of your choice, such as the Titanic dataset from Kaggle. Explore the relationships between variables and identify patterns and trends in the data.

Sample Dataset :- https://www.kaggle.com/c/titanic/data

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
```

In [7]:
```
df=pd.read_excel("C:\\Users\\abhis\\Downloads\\Titanic.xlsx")
df
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN
...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN

891 rows × 12 columns

In [10]:
```
print(df.shape)
```

```
(891, 12)
```

In [11]:
```
print(df.columns)
```

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

In [12]:
```
print(df.size)
```

```
10692
```

In [13]:
```
print(df.dtypes)
```

```
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
```

In [14]:
```
df.describe()
```

Out[14]:

	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

In [15]:
```
df.isnull().sum()
```

Out[15]:
```
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
```

In [22]:
```
df.drop(["PassengerId","Name","Ticket","Cabin"],axis=1,inplace=True)
```

In [23]:
```
df.isnull().sum()
```

Out[23]:
```
Survived      0
Pclass        0
Sex           0
Age         177
SibSp         0
Parch         0
Fare          0
Embarked      2
dtype: int64
```

In [28]:
```
df.Age.median()
```

Out[28]:
```
28.0
```

In [40]:
```
df.Age.fillna(df.Age.median(),inplace=True)
```

In [52]:
```
df.Embarked.fillna(df.Embarked.mode()[0],inplace=True)
```

In [49]:
```
df.Embarked.mode()
```

Out[49]:
```
0    S
Name: Embarked, dtype: object
```

In [53]:
```
df.isnull().sum()
```

Out[53]:
```
Survived    0
Pclass      0
Sex         0
Age         0
SibSp       0
Parch       0
Fare        0
Embarked    0
dtype: int64
```

In [57]:
```
df.Survived.value_counts()
```

Out[57]:
```
0    549
1    342
Name: Survived, dtype: int64
```

In [58]:
```
df.Pclass.value_counts()
```

Out[58]:
```
3    491
1    216
2    184
Name: Pclass, dtype: int64
```

In [59]:
```
df.Sex.value_counts()
```

Out[59]:
```
male      577
female    314
Name: Sex, dtype: int64
```

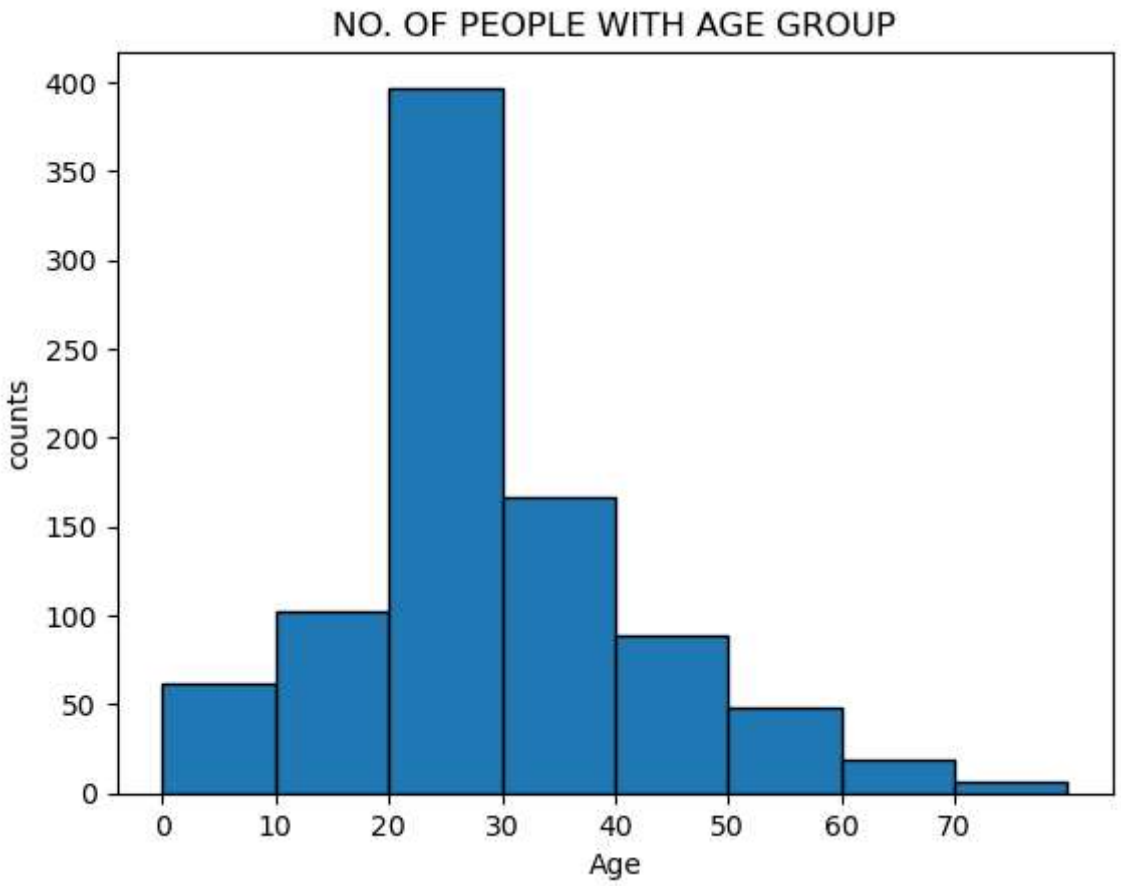
In [60]:
```
df.Embarked.value_counts()
```

Out[60]:
```
S    646
C    168
Q     77
Name: Embarked, dtype: int64
```

NO. of people with different age groups

In [69]:
```
plt.hist(x=df.Age, edgecolor='black', bins=[0,10,20,30,40,50,60,70,80])
plt.xlabel("Age")
plt.xticks(range(0,80,10))
plt.ylabel("counts")
plt.title("NO. OF PEOPLE WITH AGE GROUP")
plt.show
```

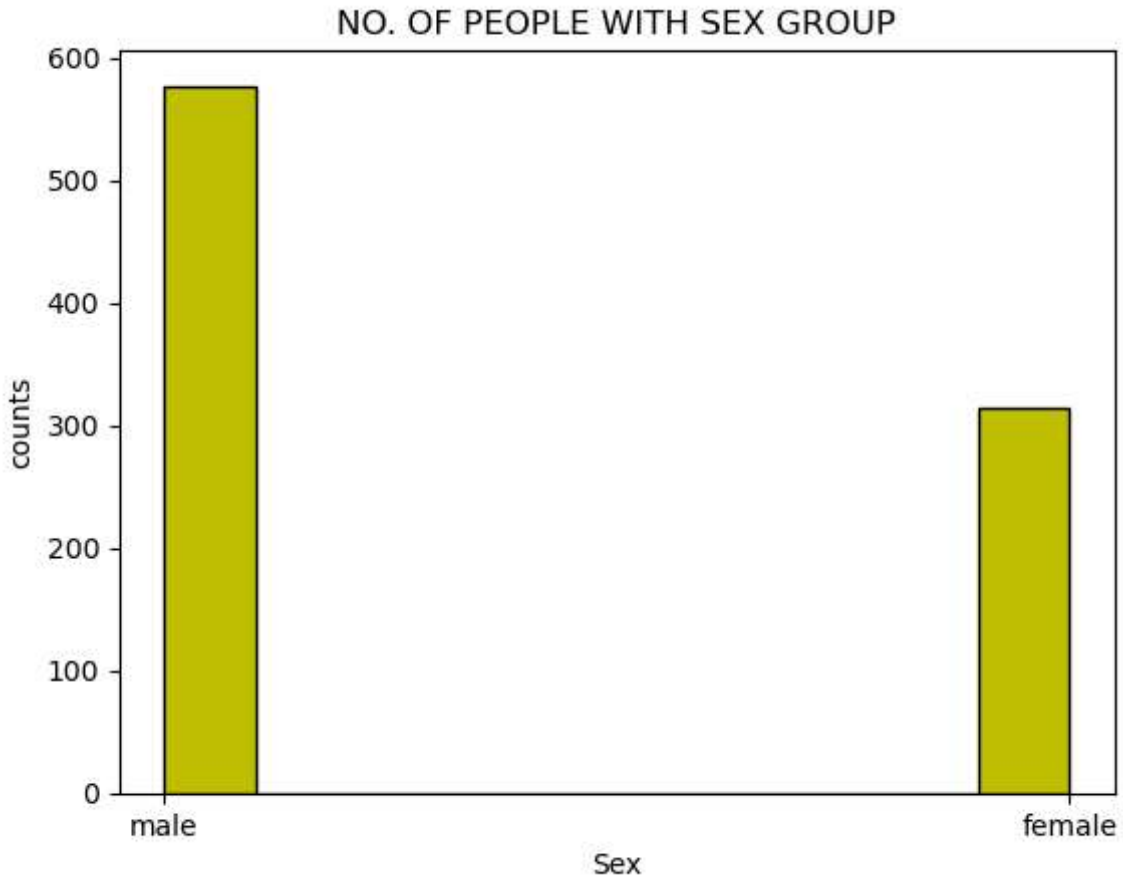
Out[69]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



NO. of people with different sex groups

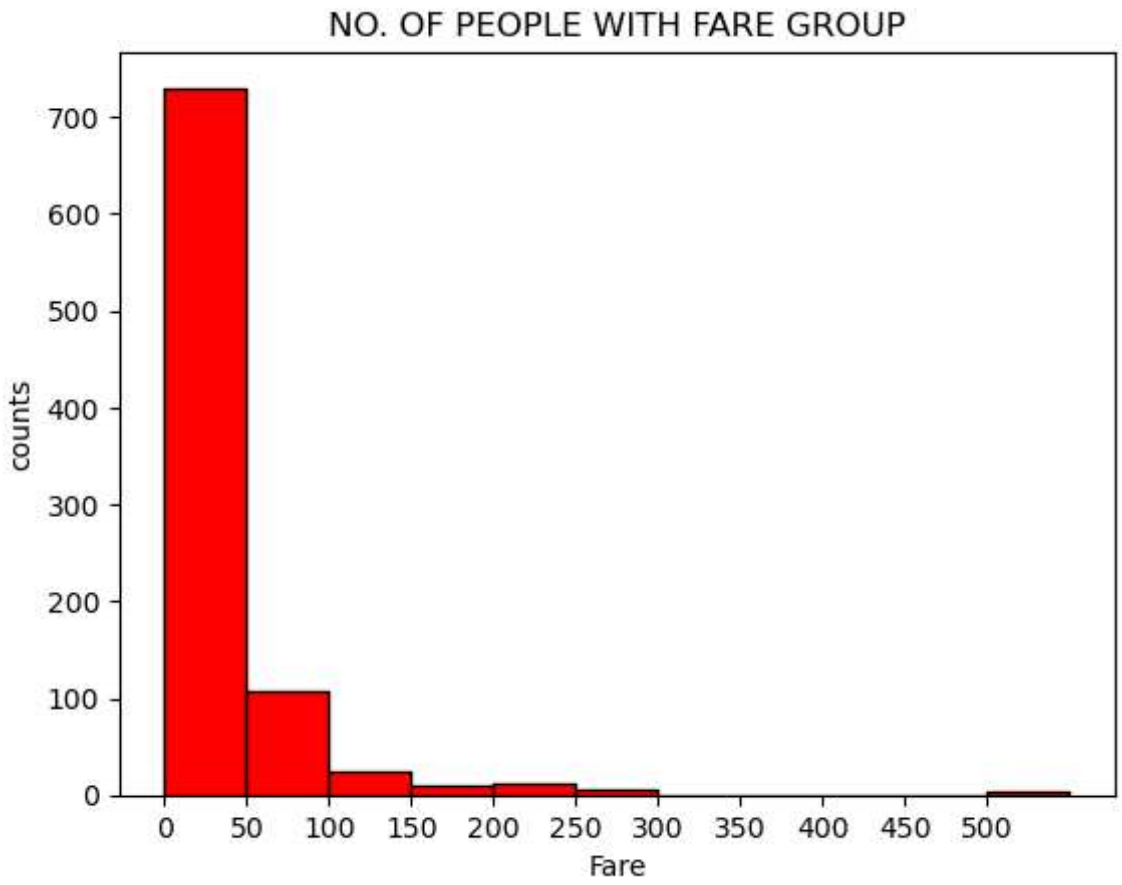
In [79]:
```
plt.hist(x=df.Sex, edgecolor='black', color='y')
plt.xlabel("Sex")
plt.ylabel("counts")
plt.title("NO. OF PEOPLE WITH SEX GROUP")
plt.show
```

Out[79]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



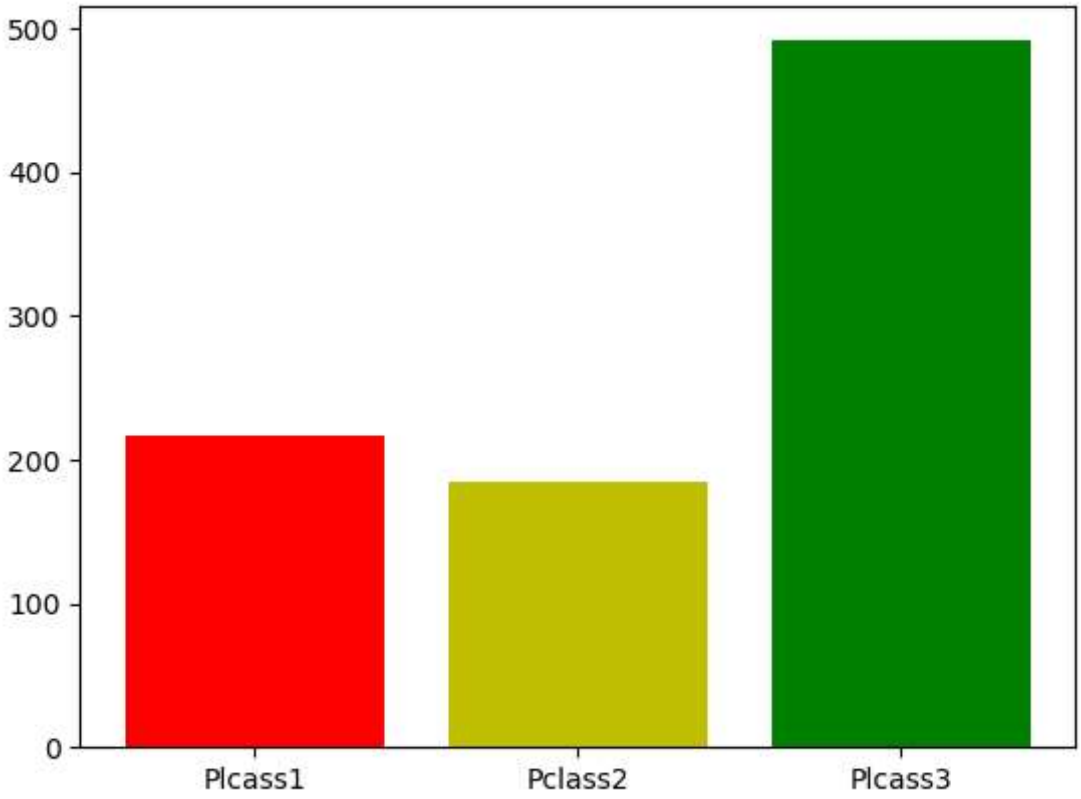
In [186…

```
plt.hist(x=df.Fare, edgecolor='black', color='r', bins=[0,50,100,150,200,250,300,350,4
plt.xticks(range(0,550,50))
plt.xlabel("Fare")
plt.ylabel("counts")
plt.title("NO. OF PEOPLE WITH FARE GROUP")
plt.show()
```



NO. of people with different class groups

In [95]:
```
plt.bar(['Plcass1','Pclass2','Plcass3'],[216,184,491],color=['r','y','g'])
plt.show()
```

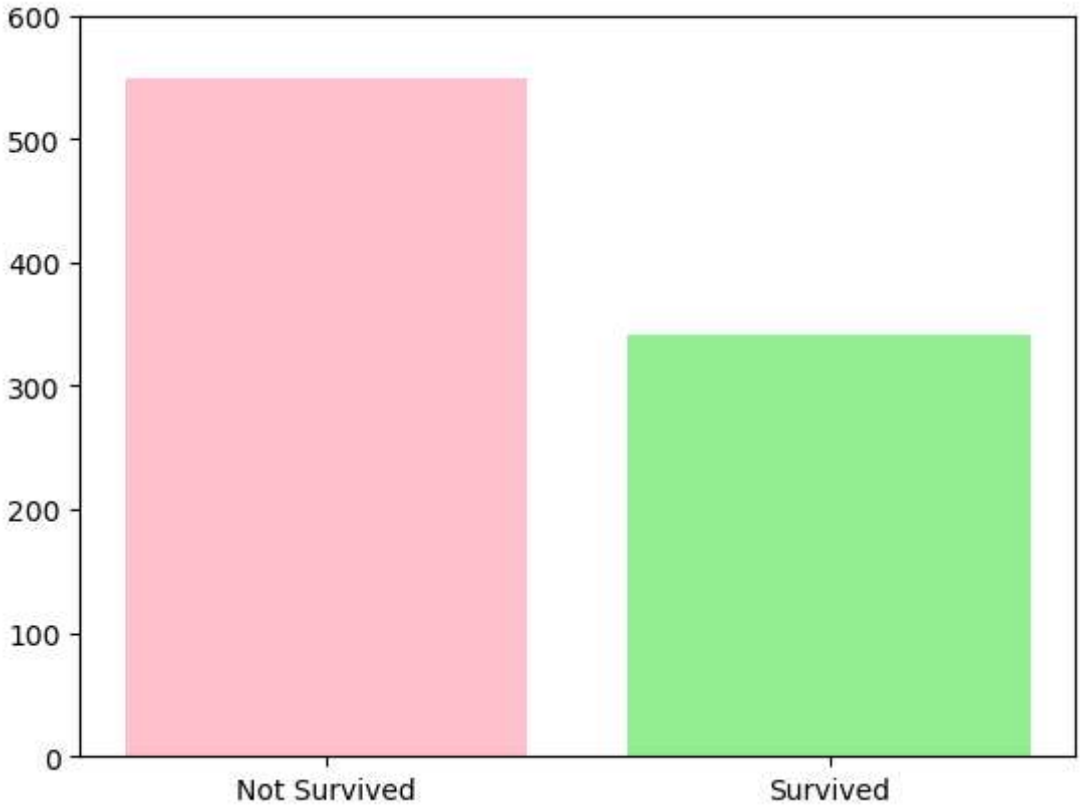


After Accident data report

NO. of people survived after an accident

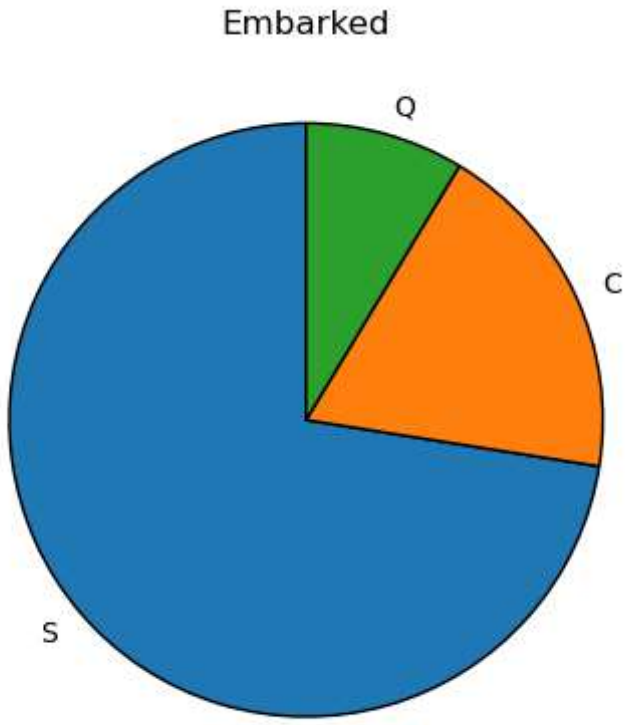
In [100…

```
plt.bar(['Not Survived','Survived'],[549,342],color=['pink','lightgreen'])
plt.yticks(range(0,700,100))
plt.show()
```



In [187…

```
plt.pie(labels=['S','C','Q'],x=[646,168,77],wedgeprops={'edgecolor':'black'},startangl
plt.title("Embarked")
plt.show()
```

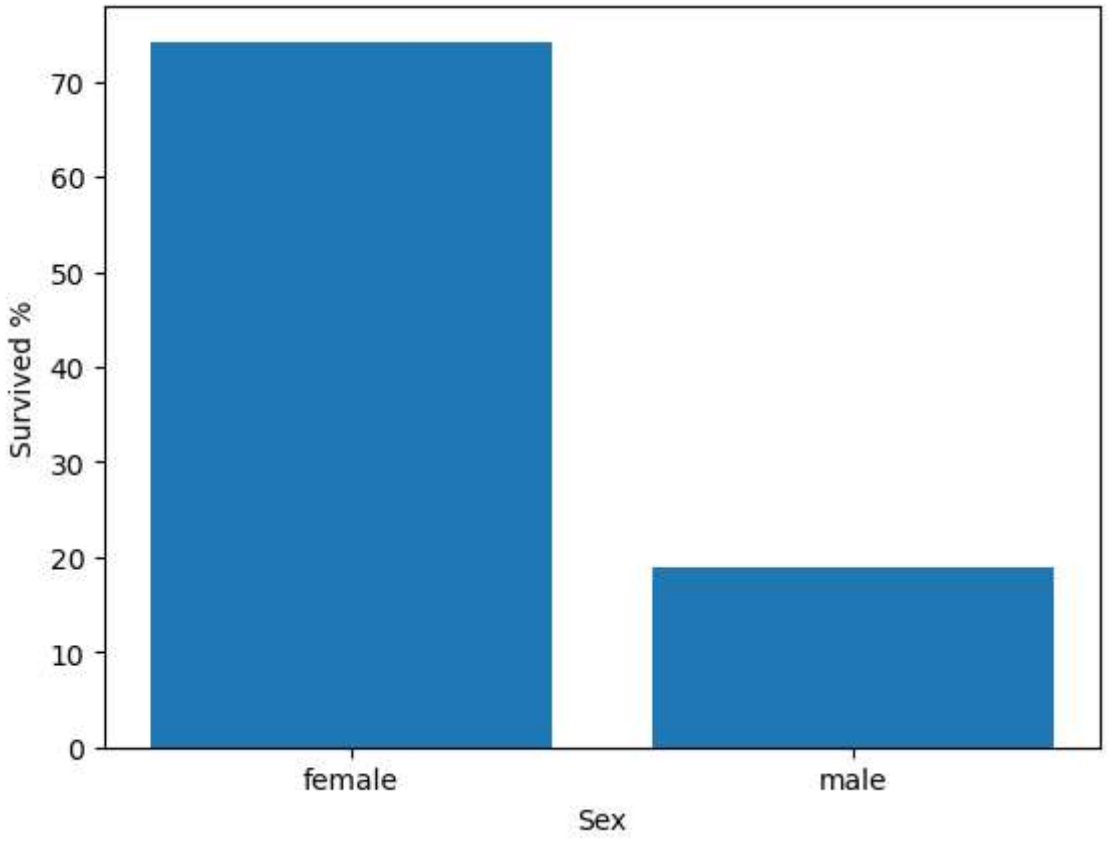


In [120…

```
df_sex=df.groupby(by='Sex')['Survived'].mean()*100
df_sex
```

Out[120]:
```
Sex
female    74.203822
male      18.890815
Name: Survived, dtype: float64
```

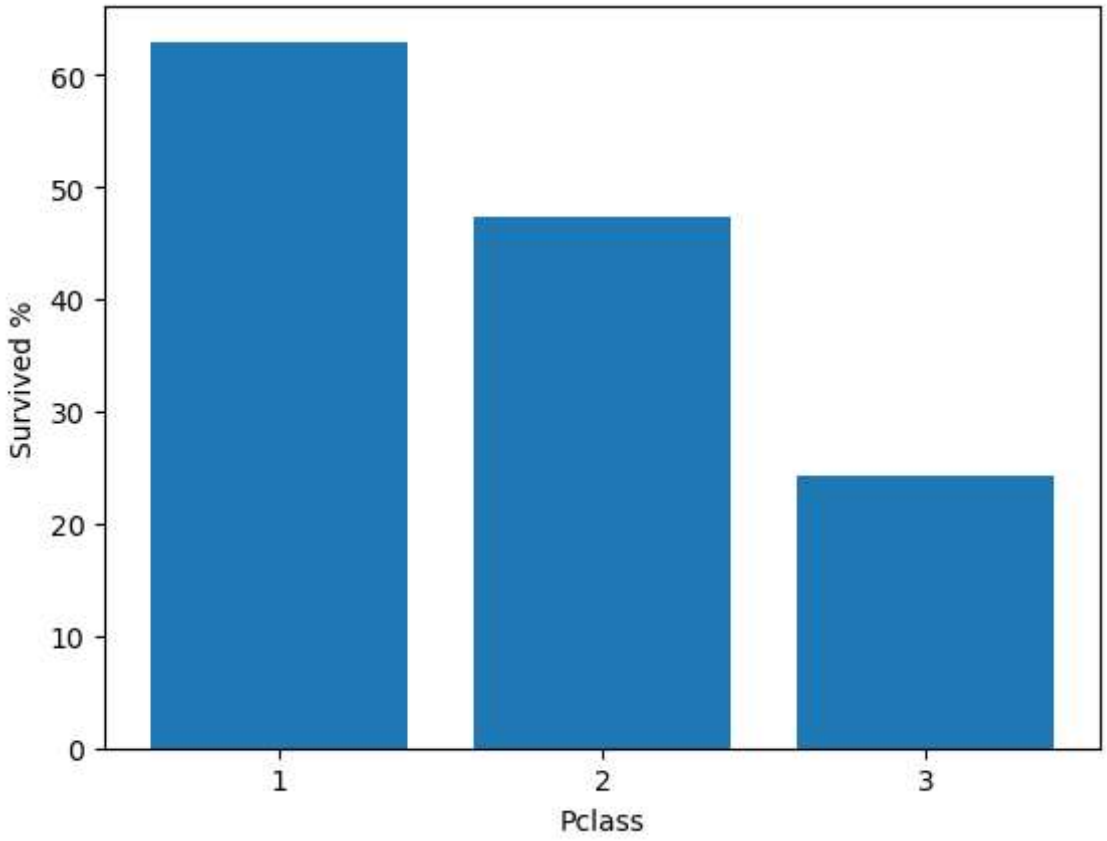
In [130…
```
plt.bar(df_sex.index,df_sex.values)
plt.xlabel("Sex")
plt.ylabel("Survived %")
plt.show()
```



In [118…
```
df_class=df.groupby(by='Pclass')['Survived'].mean()*100
df_class
```

Out[118]:
```
Pclass
1    62.962963
2    47.282609
3    24.236253
Name: Survived, dtype: float64
```

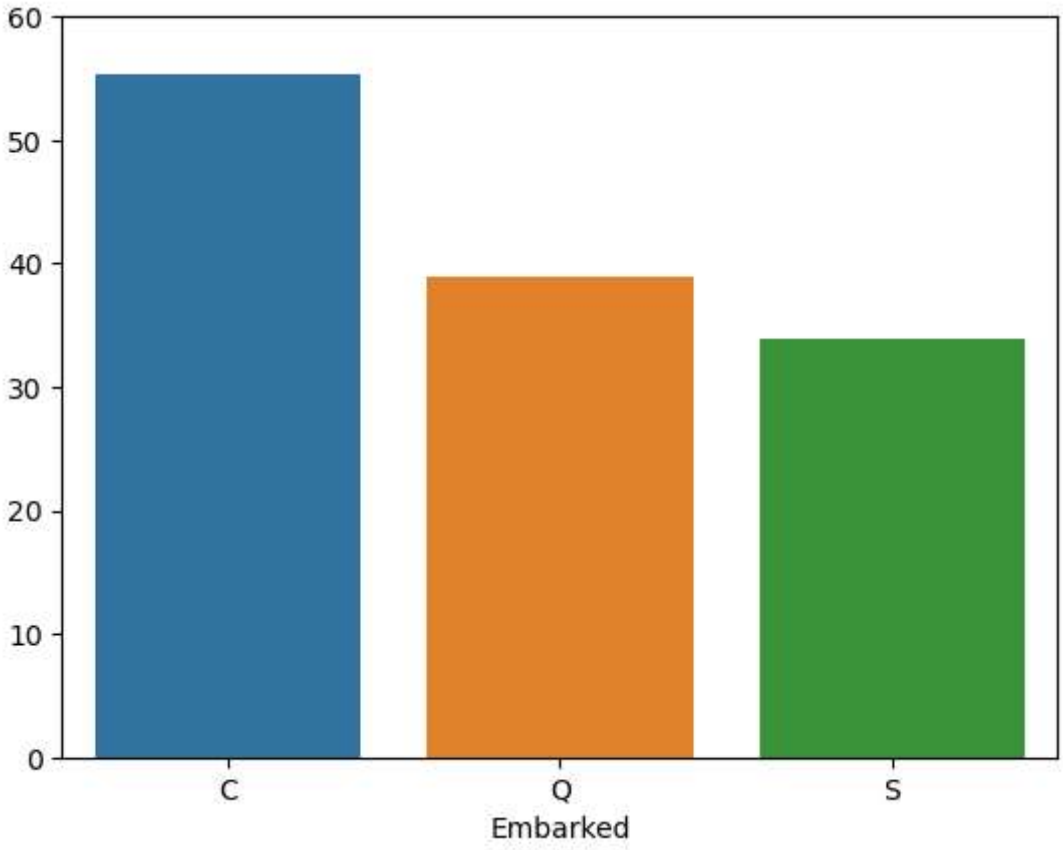
In [139…
```
plt.bar(df_class.index,df_class.values)
plt.xlabel("Pclass")
plt.ylabel("Survived %")
plt.xticks(range(1,4,1))
plt.show()
```



In [134…
```
df_emb=df.groupby(by='Embarked')['Survived'].mean()*100
df_emb
```

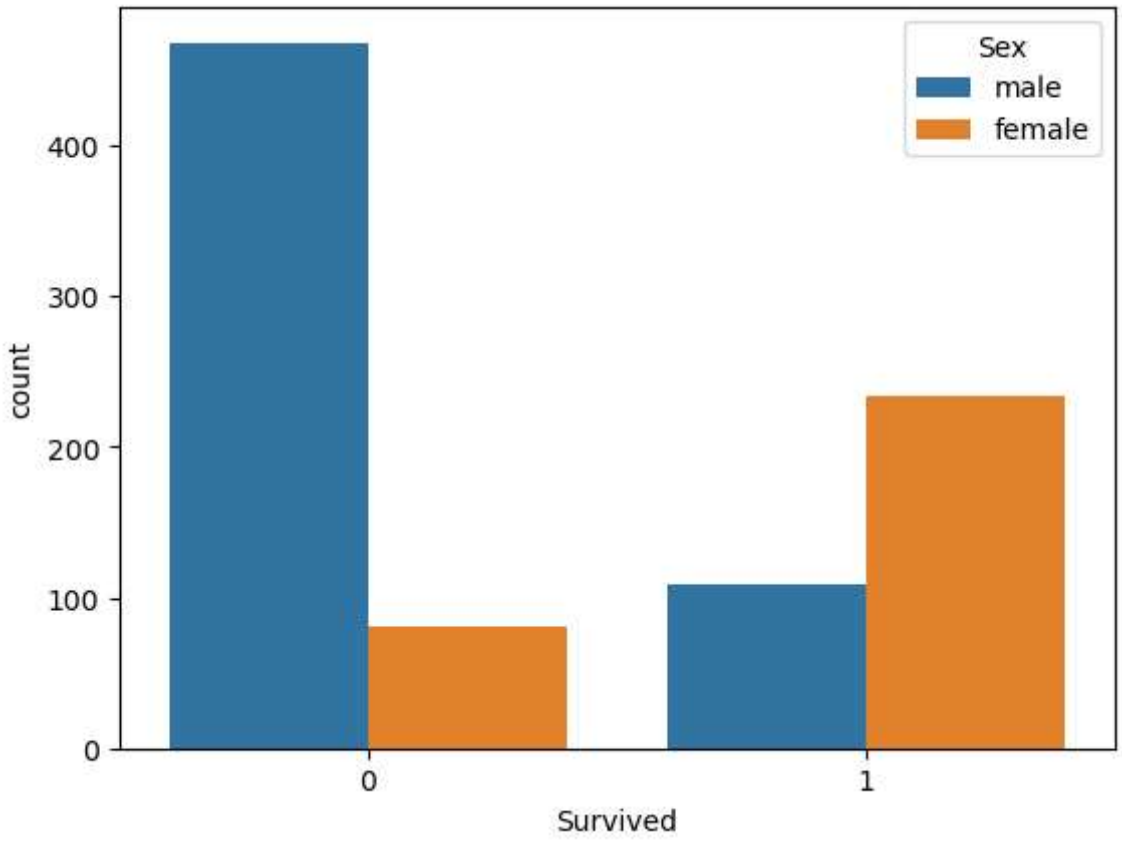
Out[134]:
```
Embarked
C    55.357143
Q    38.961039
S    33.900929
Name: Survived, dtype: float64
```

In [138…
```
sb.barplot(x=df_emb.index,y=df_emb.values)
plt.yticks(range(0,70,10))
plt.show()
```



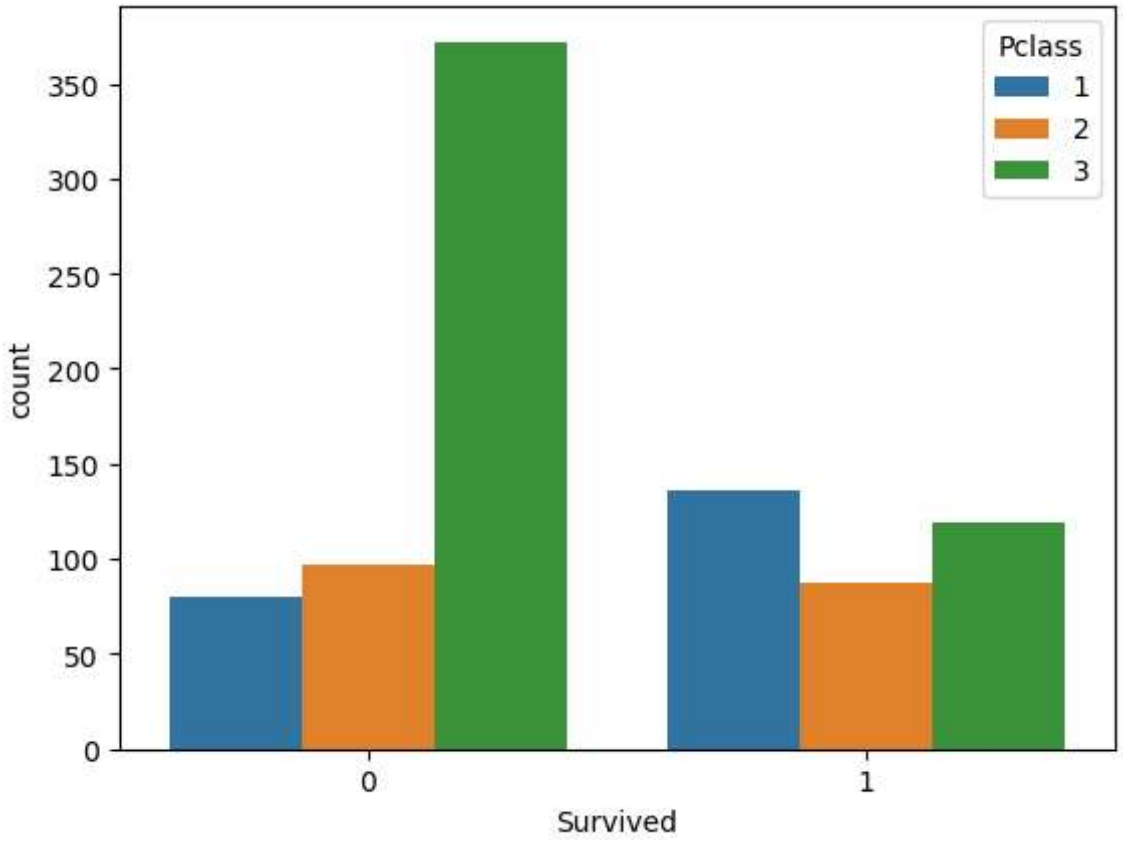
In [168…

```
sb.countplot(x=df.Survived, hue=df.Sex)
plt.show()
```



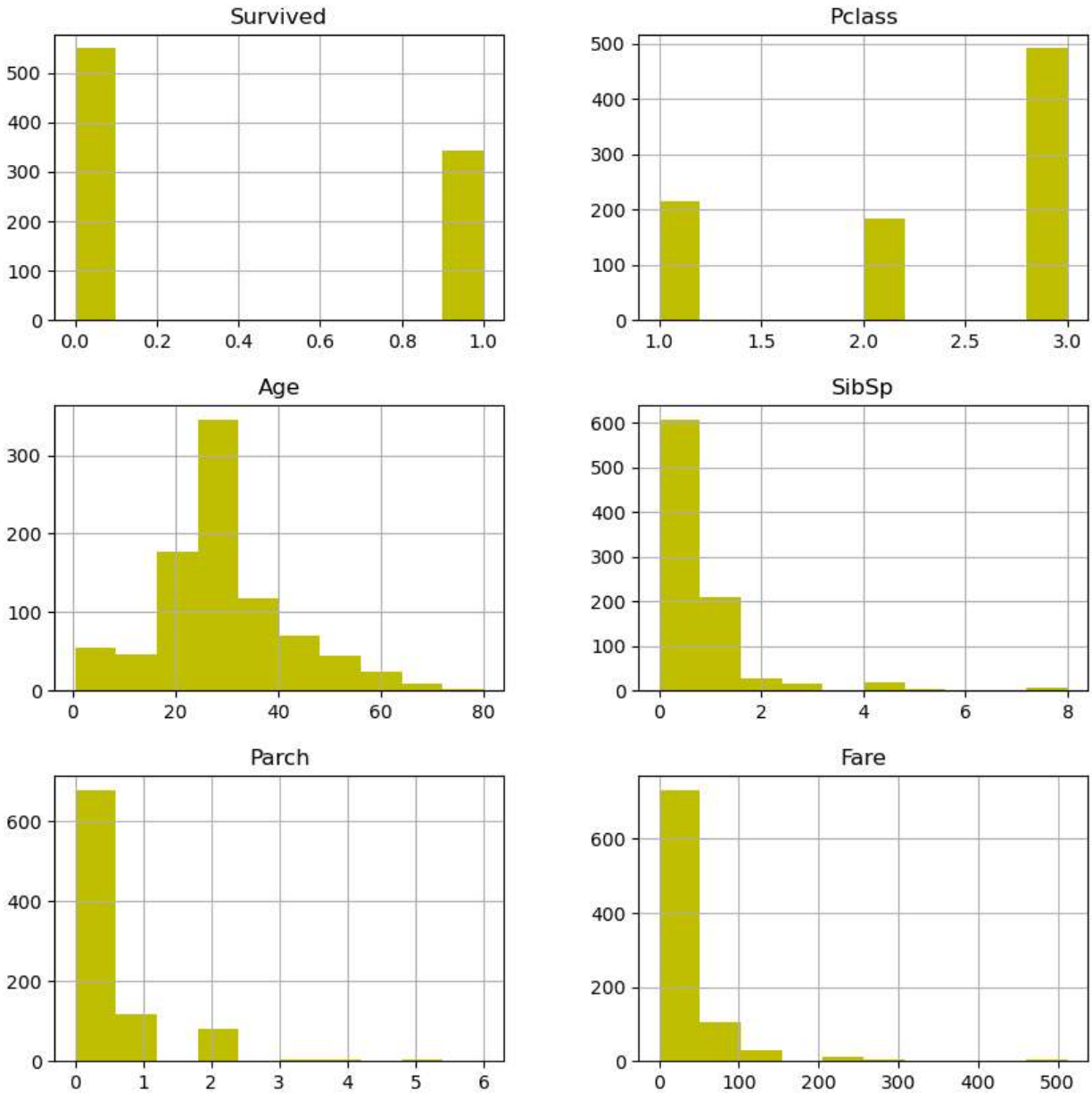
In [169…

```
sb.countplot(x=df.Survived, hue=df.Pclass)
plt.show()
```



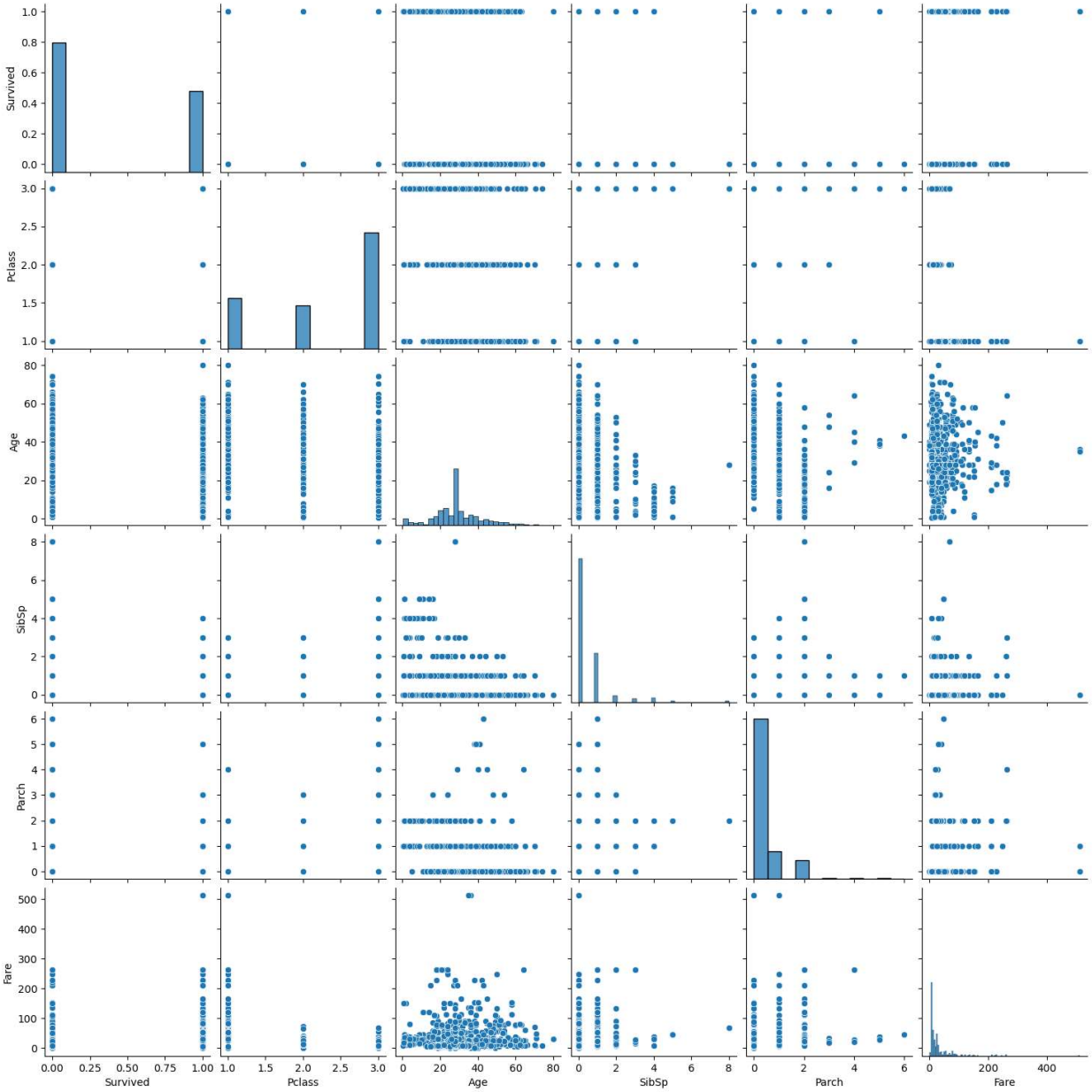
In [176…

```
df.hist(color='y',figsize=(10,10))
plt.show()
```



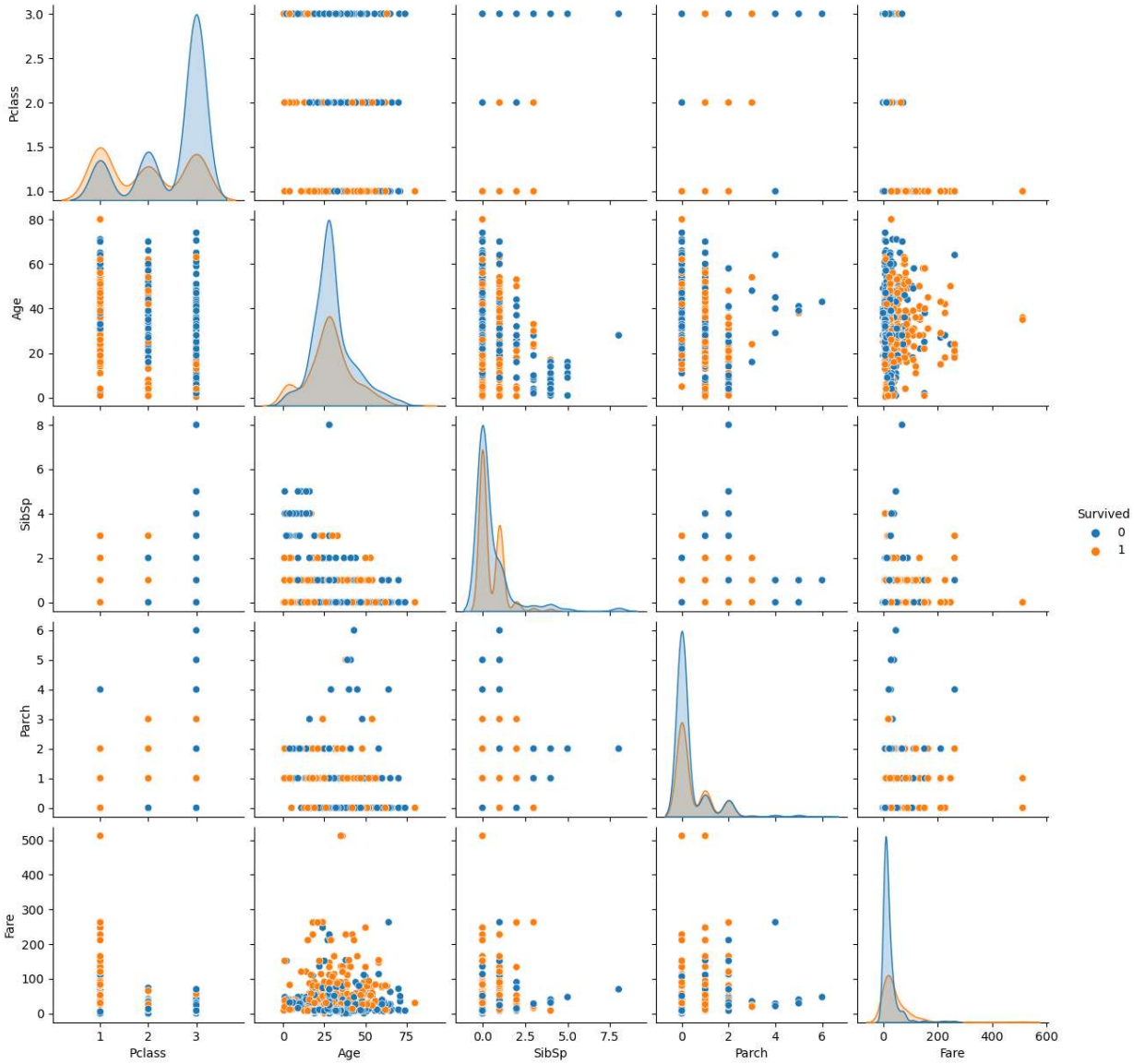
In [173…

```
sb.pairplot(df)
plt.show()
```



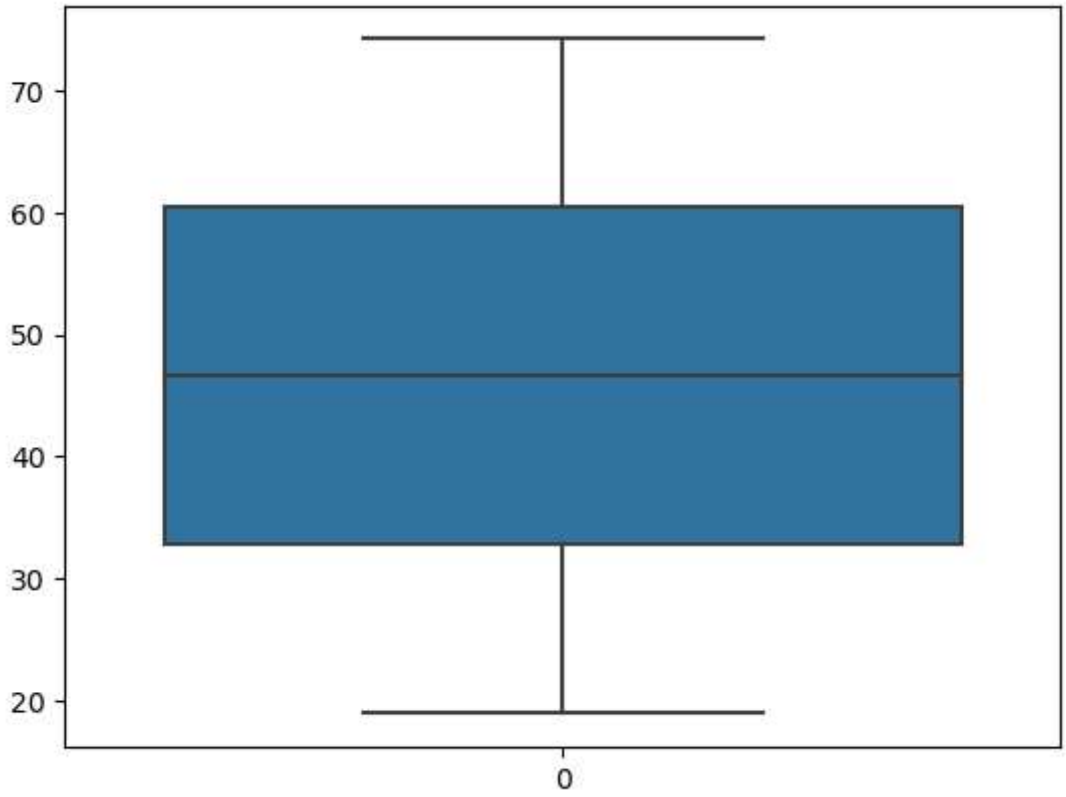
In [174…

```
sb.pairplot(df, hue='Survived')
plt.show()
```



In [184…

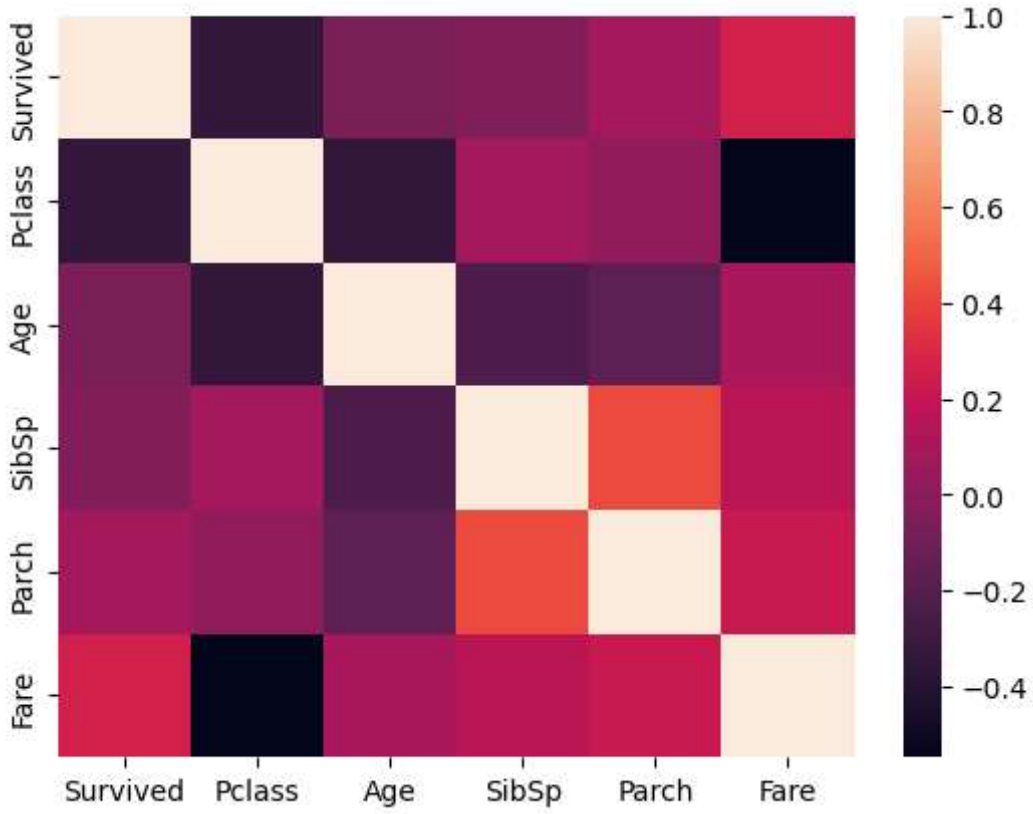
```
sb.boxplot(df_sex)
plt.show()
```



In [185…

```
sb.heatmap(df.corr())
plt.show()
```

```
C:\Users\abhis\AppData\Local\Temp\ipykernel_6360\2200381900.py:1: FutureWarning: The
default value of numeric_only in DataFrame.corr is deprecated. In a future version, i
t will default to False. Select only valid columns or specify the value of numeric_on
ly to silence this warning.
  sb.heatmap(df.corr())
```



Conclusion

The sinking of the Titanic is an undoubtedly tragic and historically significant event. The dataset provided encompasses a range of features pertaining to the passengers who were aboard the Titanic. These features include PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, and Embarked. Through analyzing these features, we are able to ascertain the survival rate of the passengers, the influence of Pclass and embarked location on their survival, the distribution of passengers based on age and fare, the survival rate based on gender, and the impact of having siblings, spouses, parents, and children on the passengers' chances of survival, among other insights.

This dataset serves as an excellent resource for conducting Exploratory Data Analysis.