```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as snr
        from sklearn.preprocessing import OneHotEncoder
```

```
In [3]: data=pd.read_csv("C:\\Users\\Lenovo\\Desktop\\Intern\\archive\\tested.csv")
        data.head(10)
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabir
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Na
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	Na
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	Na
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	Na
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	Na
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	Na
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	Na

In [5]:

```
data_new=data.drop(['PassengerId','Name','Cabin','Ticket'],axis=1)
data_new.head(10)
```

Out[5]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S
5	0	3	male	NaN	0	0	8.4583	Q
6	0	1	male	54.0	0	0	51.8625	S
7	0	3	male	2.0	3	1	21.0750	S
8	1	3	female	27.0	0	2	11.1333	S
9	1	2	female	14.0	1	0	30.0708	C

In [6]:

```
data_new.describe().round(3)
```

Out[6]:

	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000	891.000	714.000	891.000	891.000	891.000
mean	0.384	2.309	29.699	0.523	0.382	32.204
std	0.487	0.836	14.526	1.103	0.806	49.693
min	0.000	1.000	0.420	0.000	0.000	0.000
25%	0.000	2.000	20.125	0.000	0.000	7.910
50%	0.000	3.000	28.000	0.000	0.000	14.454
75%	1.000	3.000	38.000	1.000	0.000	31.000
max	1.000	3.000	80.000	8.000	6.000	512.329

In [7]:

```
data_new.isnull().sum()
```

Out[7]:

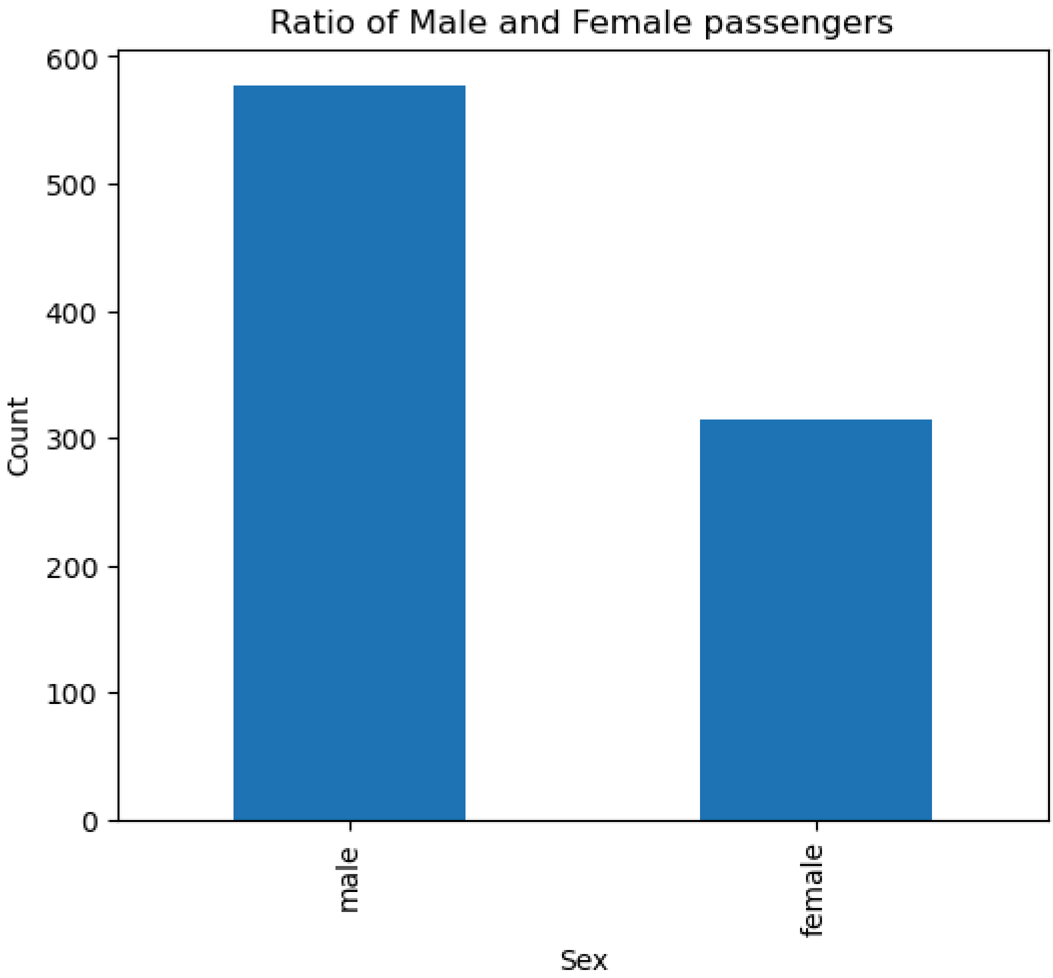
```
Survived      0
Pclass        0
Sex           0
Age         177
SibSp         0
Parch         0
Fare          0
Embarked      2
dtype: int64
```

In [8]:

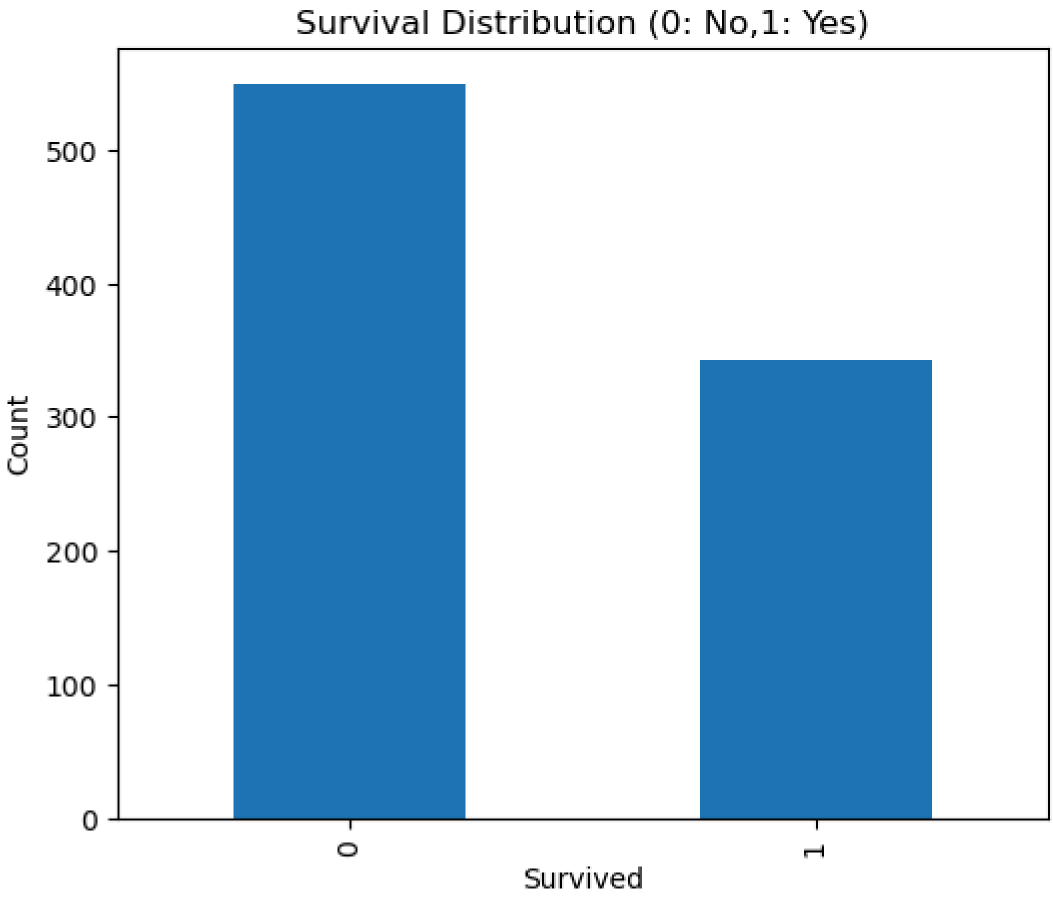
```
data_new['Age']=data_new['Age'].fillna(data_new['Age'].mean())
data_new['Fare']=data_new['Fare'].fillna(data_new['Fare'].mean())
data_new.isnull().sum()
```

```
Out[8]: Survived    0
        Pclass      0
        Sex         0
        Age         0
        SibSp       0
        Parch       0
        Fare        0
        Embarked    2
        dtype: int64
```

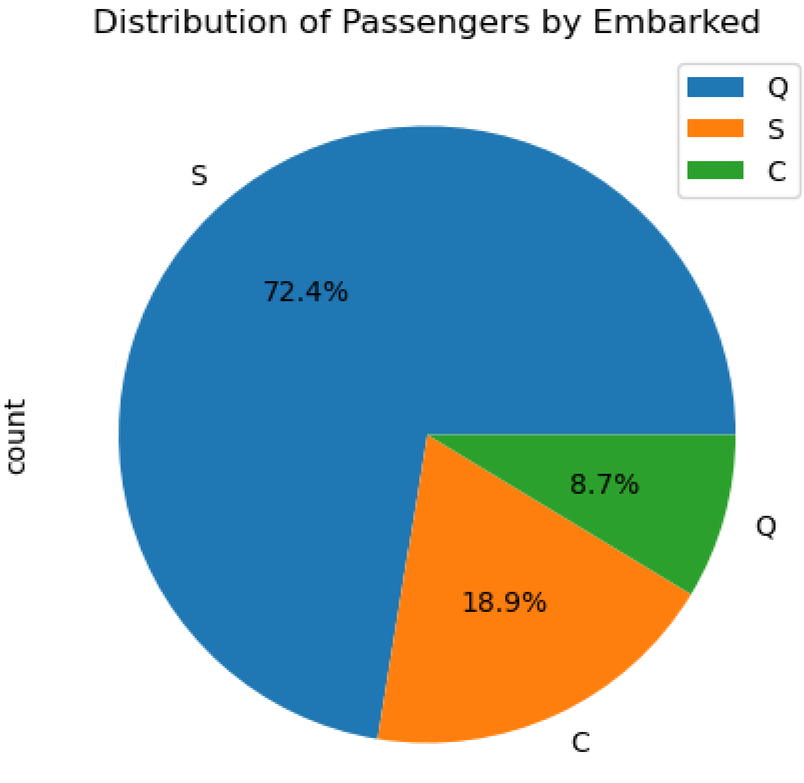
```
In [9]: ax=data_new['Sex'].value_counts().plot(kind='bar', x='Survived', figsize=(6,5))
        plt.title("Ratio of Male and Female passengers")
        plt.ylabel("Count")
        plt.xlabel("Sex")
        plt.show()
```



```
In [10]: ax=data_new['Survived'].value_counts().plot(kind='bar', x='Survived', figsize=(6,5)
         plt.title("Survival Distribution (0: No,1: Yes)")
         plt.ylabel("Count")
         plt.xlabel("Survived")
         plt.show()
```



```
In [12]: ax=data_new['Embarked'].value_counts().plot(kind='pie',figsize=(6,5),autopct='%1.1f
         plt.title("Distribution of Passengers by Embarked")
         plt.legend(["Q", "S", "C"])
         plt.show()
```

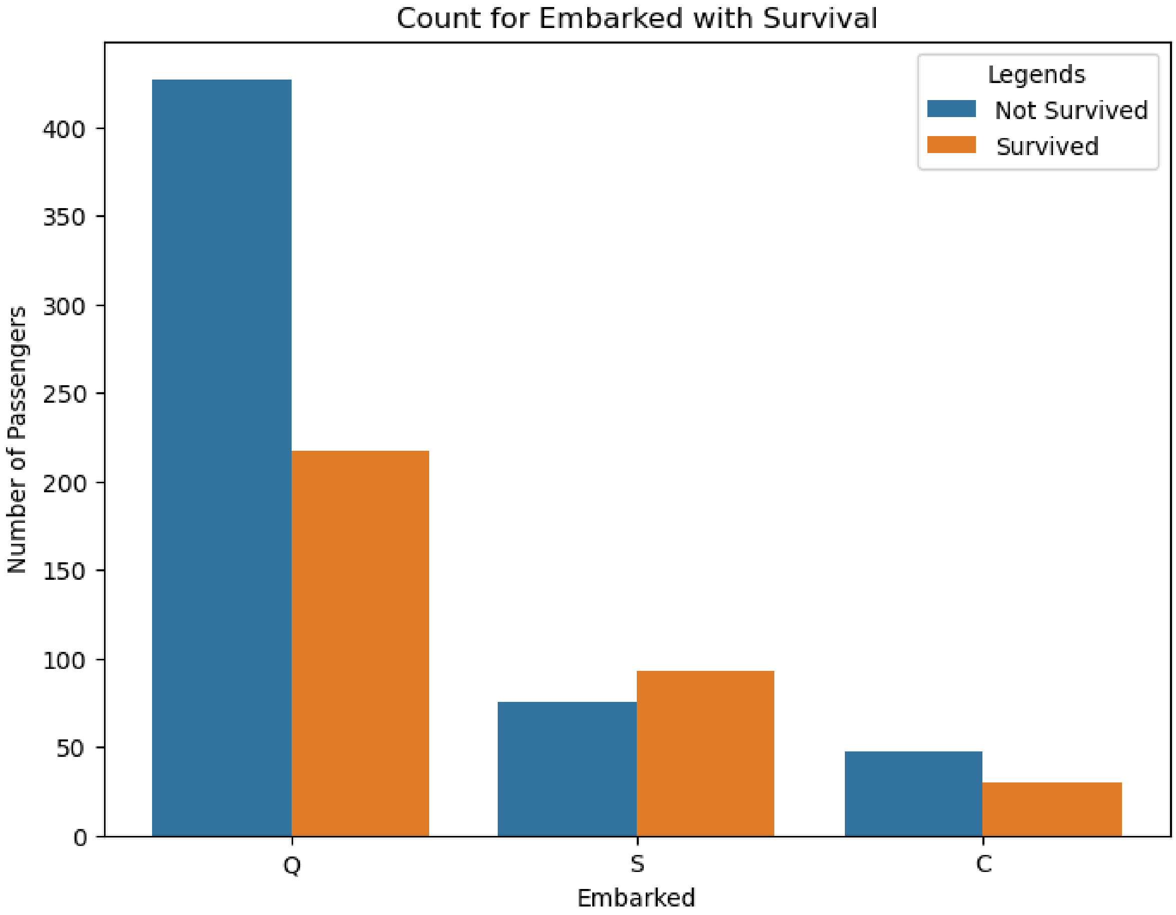


```
In [14]: _, ax=plt.subplots(figsize=(8,6))
         snr.countplot(data = data_new, x = "Embarked", hue = "Survived", ax = ax)
```

```
ax.set_title("Count for Embarked with Survival")
ax.set_xlabel("Embarked")
ax.set_xticklabels(["Q", "S", "C"])
ax.set_ylabel("Number of Passengers")
ax.legend(title = "Legends", labels = ["Not Survived", "Survived"])
plt.plot()
```

Out[14]:

```
[]
```



In [15]:

```
data_new['Sex']= data_new['Sex'].apply({'male':1, 'Female':0}.get)
data_new['Embarked']= data_new['Embarked'].apply({'S':1, 'Q':2, 'C':3}.get)
data_new.head()
```

Out[15]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	1.0	22.0	1	0	7.2500	1.0
1	1	1	NaN	38.0	1	0	71.2833	3.0
2	1	3	NaN	26.0	0	0	7.9250	1.0
3	1	1	NaN	35.0	1	0	53.1000	1.0
4	0	3	1.0	35.0	0	0	8.0500	1.0

In [19]:

```
x=data_new.drop(['Survived'],axis=1)
y=data_new['Survived']
```

In [28]:

```
from sklearn.model_selection import train_test_split
x_train, x_test,y_train,y_test=train_test_split(x, y, train_size=0.8)
```

In [26]:

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=5)
```

In [33]:

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
x_train = imputer.fit_transform(x_train)
x_test = imputer.transform(x_test)
```

In [34]:

```
knn.fit(x_train,y_train)
```

Out[34]:

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

In [35]:

```
predictions=knn.predict(x_test)
print(predictions)
```

```
[0 1 1 0 0 0 0 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 1 0 1
 1 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1 1 1 0
 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0
 0 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 0 1 0 0 0 1 0 1
 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0]
```

In [36]:

```
from sklearn.metrics import confusion_matrix, accuracy_score
ac=accuracy_score(y_test,predictions)
cm=confusion_matrix(y_test,predictions)
```

In [37]:

```
print(cm)
```

```
[[91 24]
 [37 27]]
```

In [38]:

```
print(ac)
```

```
0.659217877094972
```

In [39]:

```
from sklearn.tree import DecisionTreeClassifier
tree=DecisionTreeClassifier()
```

In [40]:

```
tree.fit(x_train,y_train)
```

Out[40]:

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

In [41]:

```
predictions=tree.predict(x_test)
print(predictions)
```

```
[0 1 1 1 0 0 0 1 0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1
 1 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 1 0 0 0 1 0 1 1 1 1 1
 0 1 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0 1 0 1 0
 0 1 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 1
 0 0 0 1 1 1 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 1 0 1 1 1 0 1 0 0 1]
```

In [42]:

```
from sklearn.metrics import confusion_matrix, accuracy_score
ac=accuracy_score(y_test,predictions)
cm=confusion_matrix(y_test,predictions)
```

In [43]:

```
print(cm)
```

```
[[84 31]
 [31 33]]
```

In [44]:

```
print(ac)
```

```
0.6536312849162011
```

In [45]:
```
from sklearn.svm import SVC
svm=SVC()
```

In [46]:
```
svm.fit(x_train,y_train)
```

Out[46]:
```
▾ SVC
SVC()
```

In [47]:
```
predictions=svm.predict(x_test)
print(predictions)
```

```
[0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1
 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0
 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0
 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1
 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0]
```

In [48]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
ac=accuracy_score(y_test,predictions)
cm=confusion_matrix(y_test,predictions)
```

In [49]:
```
print(cm)
```

```
[[93 22]
 [50 14]]
```

In [50]:
```
print(ac)
```

```
0.5977653631284916
```