assignment-15-sept

September 21, 2023

0.1 1.import the necessary libraries

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

0.2 2.import the dataset

[2]:
```
ds= pd.read_csv(r"D:\smartbridge\vitmorningslot\archive\Titanic-Dataset.csv")
```

[3]:
```
ds
```

[3]:
```
     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]
```

```
[4]: ds.head()
```

```
[4]:    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      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
```

```
[5]: ds.tail()
```

```
[5]:      PassengerId  Survived  Pclass                                      Name  \
     886          887         0       2                     Montvila, Rev. Juozas
     887          888         1       1              Graham, Miss. Margaret Edith
     888          889         0       3  Johnston, Miss. Catherine Helen "Carrie"
     889          890         1       1                     Behr, Mr. Karl Howell
```

```
890          891         0       3                       Dooley, Mr. Patrick

        Sex   Age  SibSp  Parch      Ticket   Fare Cabin Embarked
886    male  27.0      0      0      211536  13.00   NaN        S
887  female  19.0      0      0      112053  30.00   B42        S
888  female   NaN      1      2  W./C. 6607  23.45   NaN        S
889    male  26.0      0      0      111369  30.00  C148        C
890    male  32.0      0      0      370376   7.75   NaN        Q
```

[6]:

```
ds.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
```

0.3 Dropping Unnecessary features

[7]:

```
ds.drop(['PassengerId','Name','Ticket'],axis=1,inplace=True)
ds.head()
```

[7]:

```
   Survived  Pclass     Sex   Age  SibSp  Parch     Fare Cabin Embarked
0         0       3    male  22.0      1      0   7.2500   NaN        S
1         1       1  female  38.0      1      0  71.2833   C85        C
2         1       3  female  26.0      0      0   7.9250   NaN        S
3         1       1  female  35.0      1      0  53.1000  C123        S
4         0       3    male  35.0      0      0   8.0500   NaN        S
```

0.4 3.Handling Null Values

[8]:
```
ds.isnull().sum()
```

[8]:
```
Survived      0
Pclass        0
Sex           0
Age         177
SibSp         0
Parch         0
Fare          0
Cabin       687
Embarked      2
dtype: int64
```

[9]:
```
ds['Age'].fillna(ds['Age'].median(),inplace=True)
```

[10]:
```
ds['Embarked'].fillna(ds['Embarked'].mode()[0],inplace =True)
```

[11]:
```
ds.drop(columns=['Cabin'],inplace=True)
```

[12]:
```
ds.isnull().sum()
```

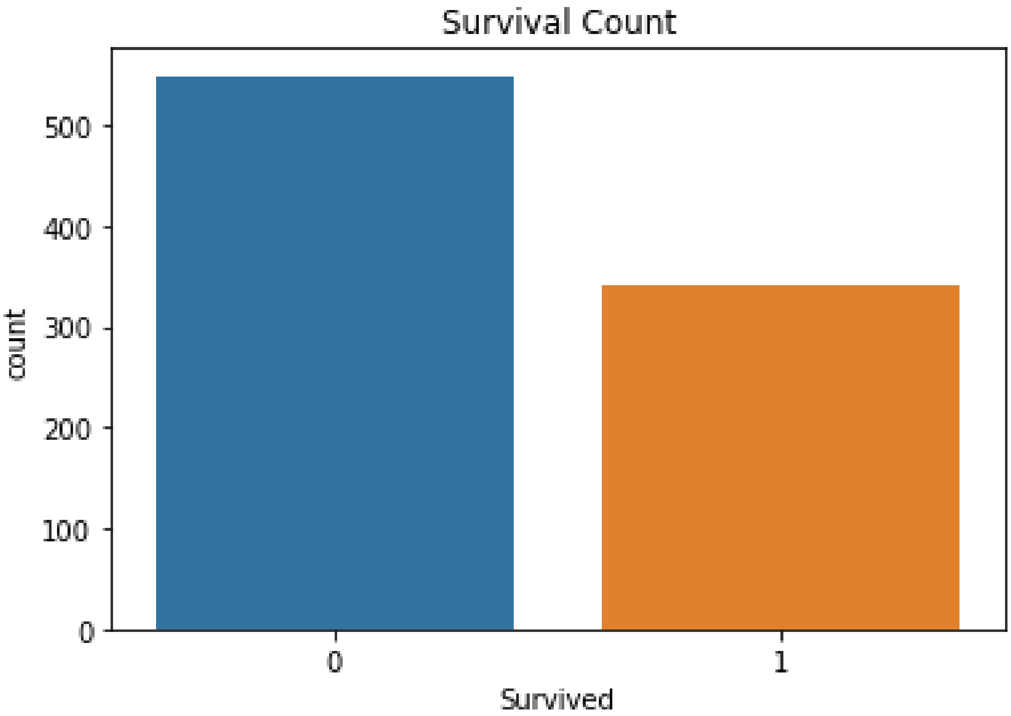
[12]:
```
Survived    0
Pclass      0
Sex         0
Age         0
SibSp       0
Parch       0
Fare        0
Embarked    0
dtype: int64
```

[13]:
```
ds.tail()
```

[13]:
```
     Survived  Pclass     Sex   Age  SibSp  Parch   Fare Embarked
886         0       2    male  27.0      0      0  13.00        S
887         1       1  female  19.0      0      0  30.00        S
888         0       3  female  28.0      1      2  23.45        S
889         1       1    male  26.0      0      0  30.00        C
890         0       3    male  32.0      0      0   7.75        Q
```

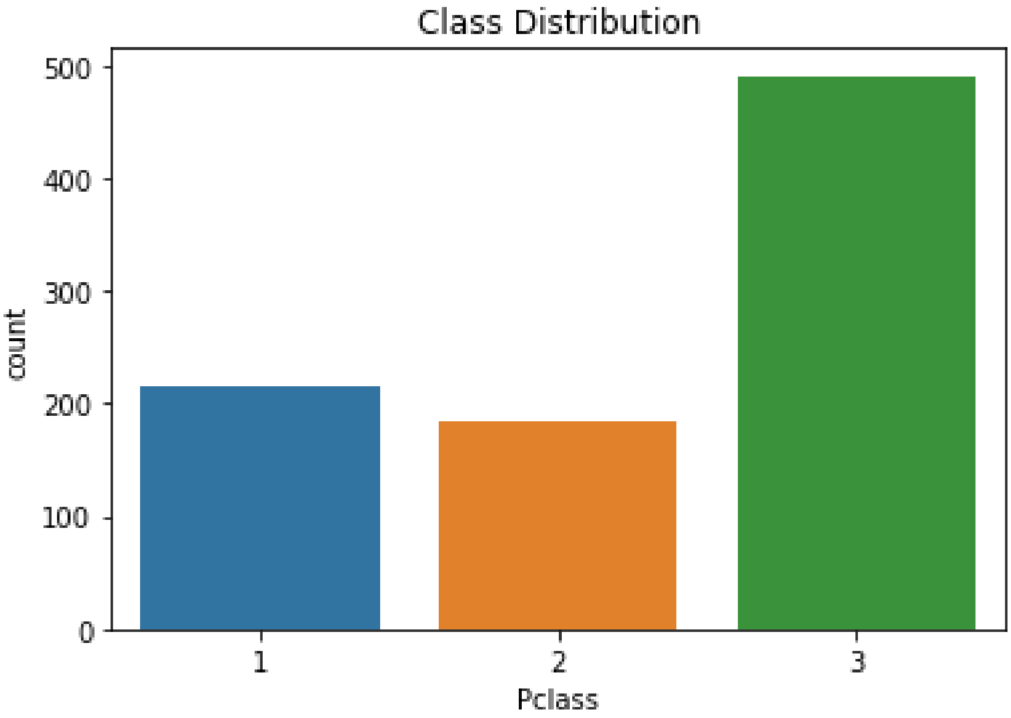
0.5 4.Data Visualisation

[14]:
```
sns.countplot(x='Survived', data=ds)
plt.title('Survival Count')
plt.show()
```



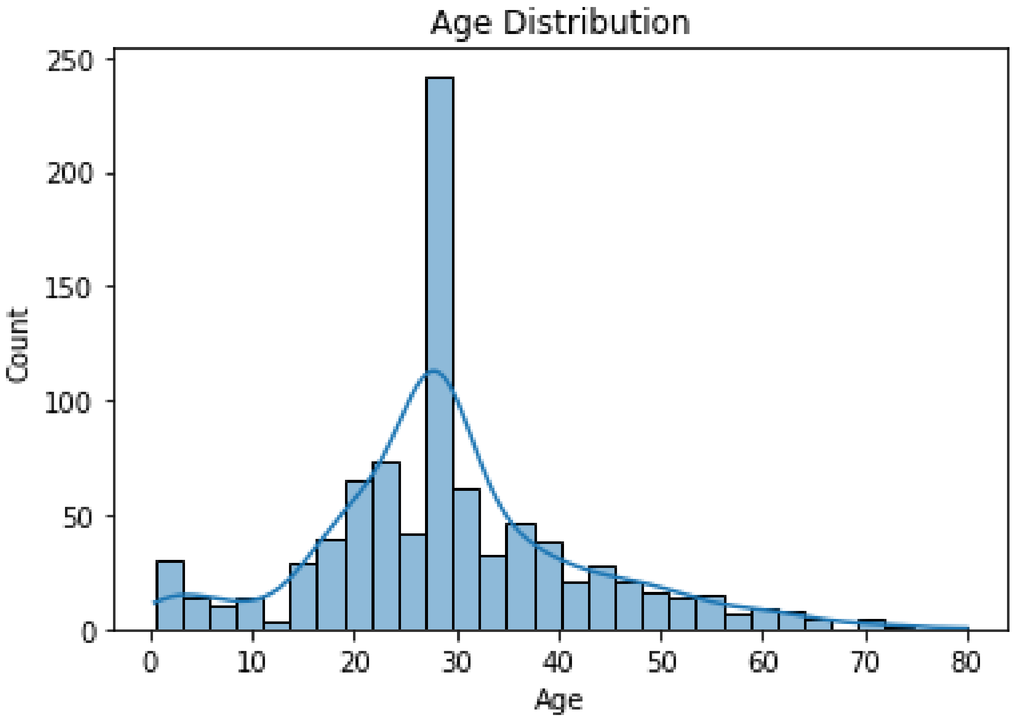
from this plot we can say survived(1) is less compared to death(0)

[15]:
```
sns.countplot(x='Pclass', data=ds)
plt.title('Class Distribution')
plt.show()
```

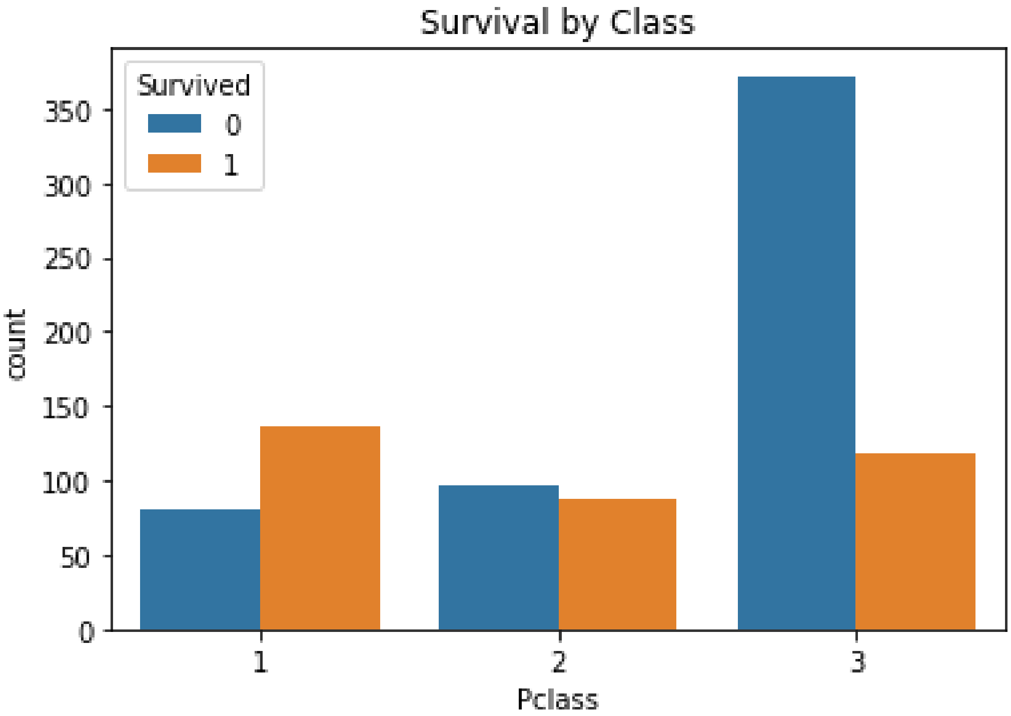


from this distribution we can say class -3 members present were more in the ship

[16]:
```
sns.histplot(ds['Age'], kde=True)
plt.title('Age Distribution')
plt.show()
```

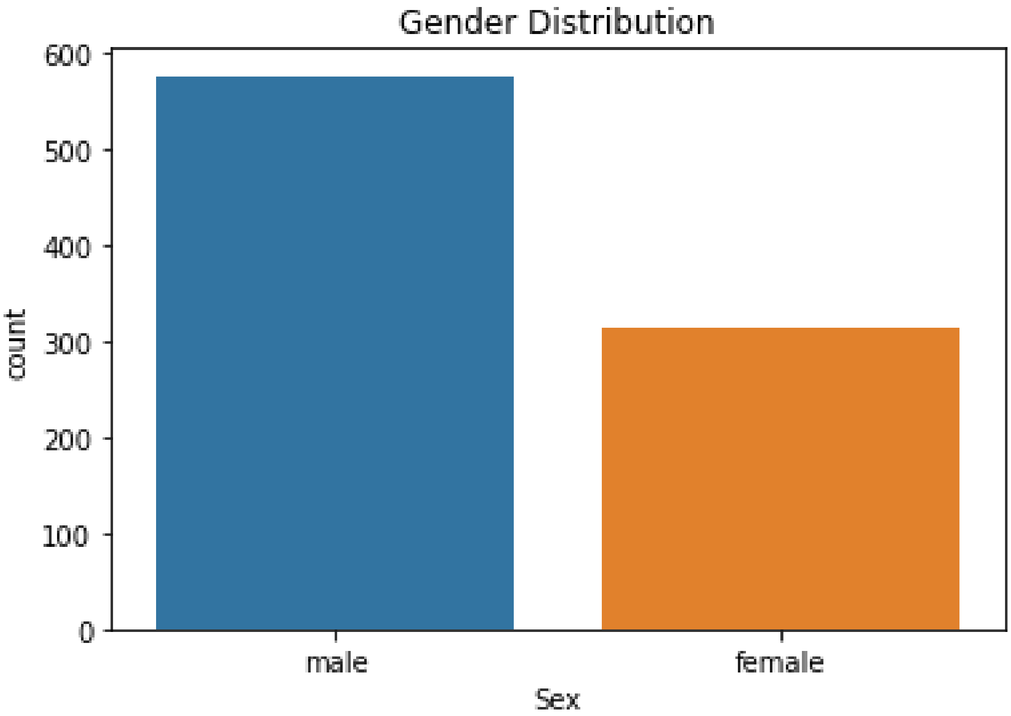


[17]:
```
sns.countplot(x='Pclass', hue='Survived', data=ds)
plt.title('Survival by Class')
plt.show()
```



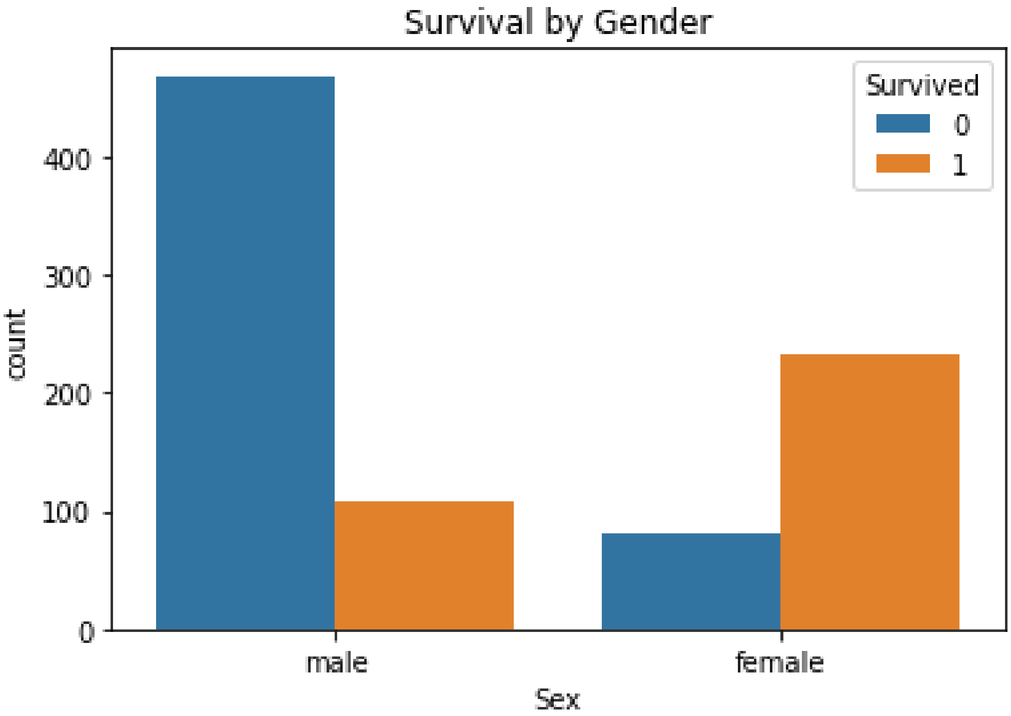
[18]:

```
sns.countplot(x='Sex', data=ds)
plt.title('Gender Distribution')
plt.show()
```



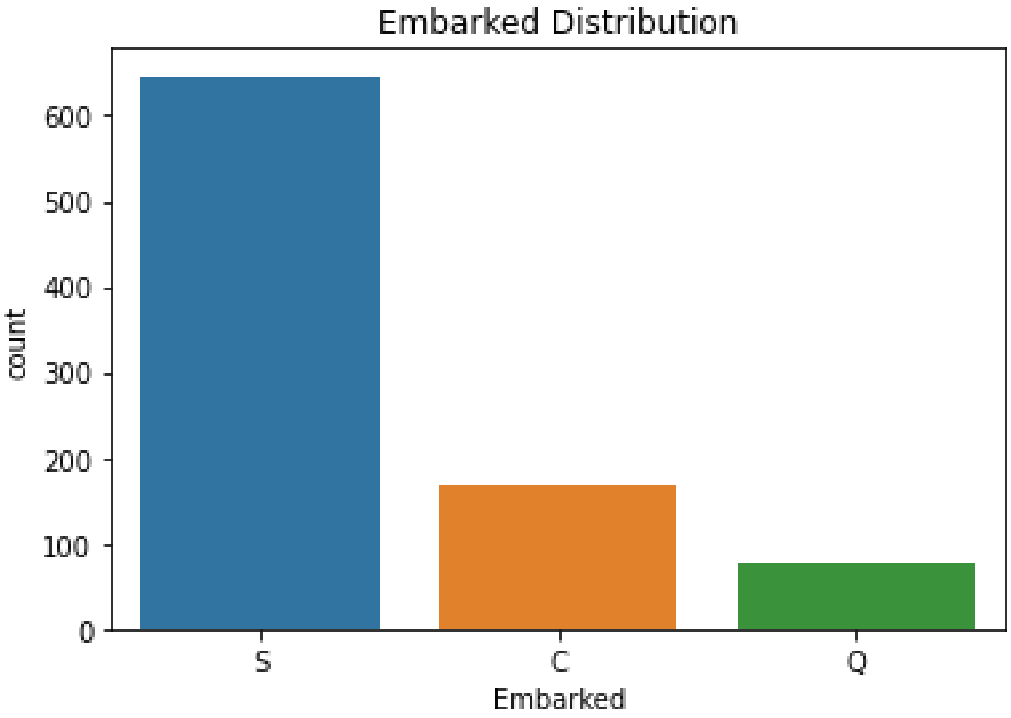
[19]:

```
sns.countplot(x='Sex', hue='Survived', data=ds)
plt.title('Survival by Gender')
plt.show()
```



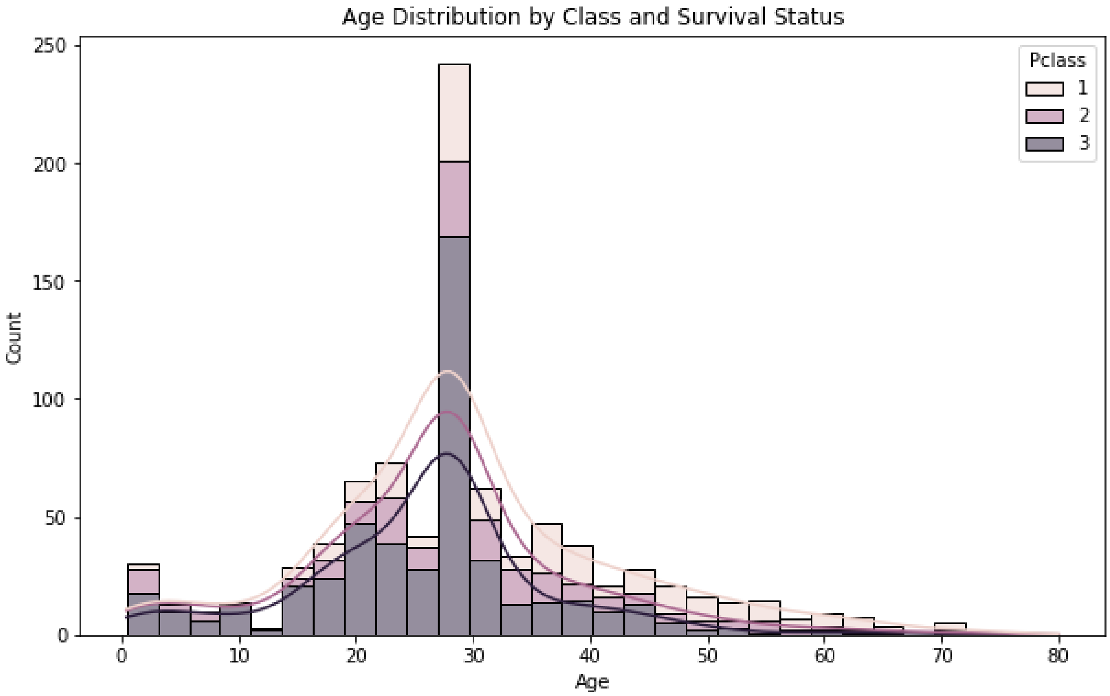
[20]:

```
sns.countplot(x='Embarked', data=ds)
plt.title('Embarked Distribution')
plt.show()
```

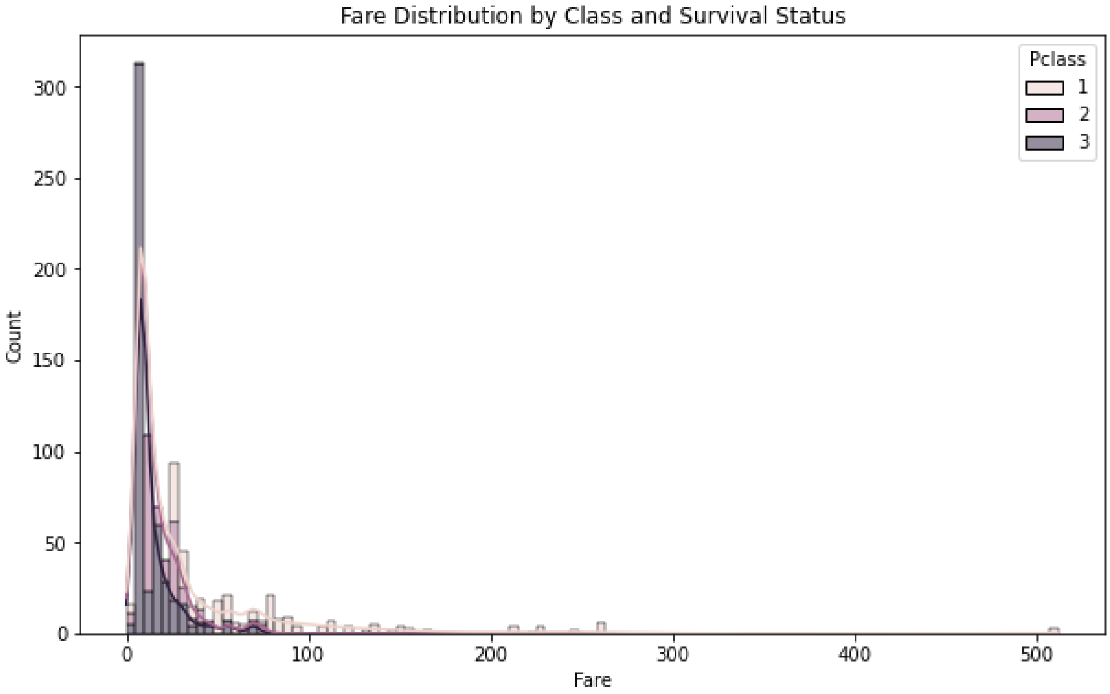


[21]:

```
plt.figure(figsize=(10, 6))
sns.histplot(data=ds, x='Age', hue='Pclass', multiple='stack', kde=True)
plt.title('Age Distribution by Class and Survival Status')
plt.show()
```

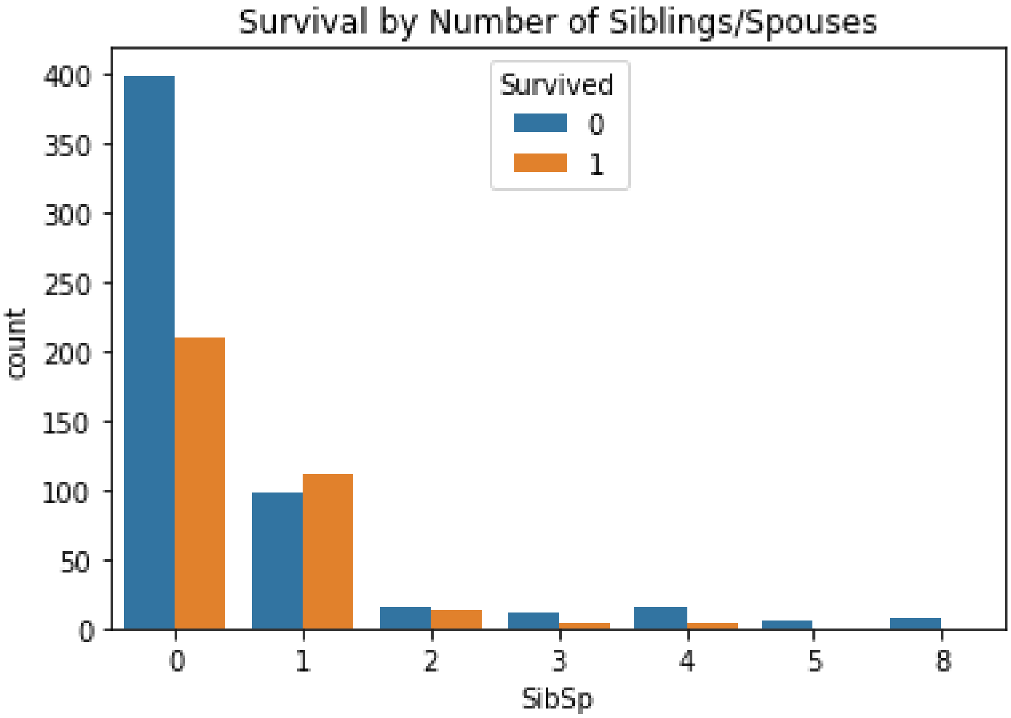


[22]:
```
plt.figure(figsize=(10, 6))
sns.histplot(data=ds, x='Fare', hue='Pclass', multiple='stack', kde=True)
plt.title('Fare Distribution by Class and Survival Status')
plt.show()
```



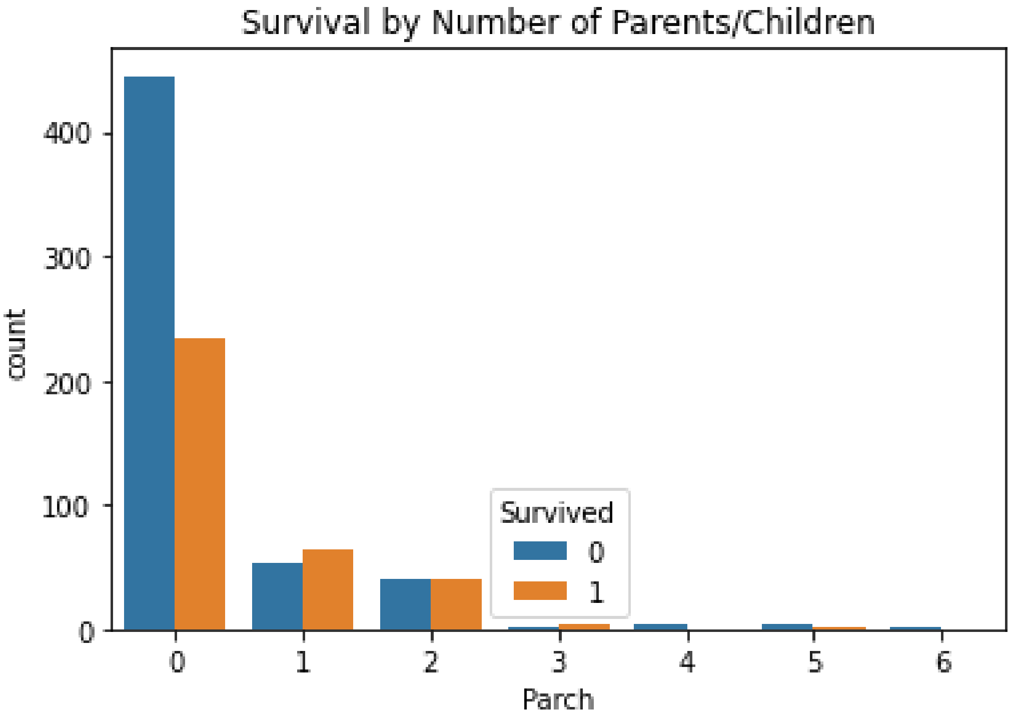
[23]:

```
sns.countplot(x='SibSp', hue='Survived', data=ds)
plt.title('Survival by Number of Siblings/Spouses')
plt.show()
```



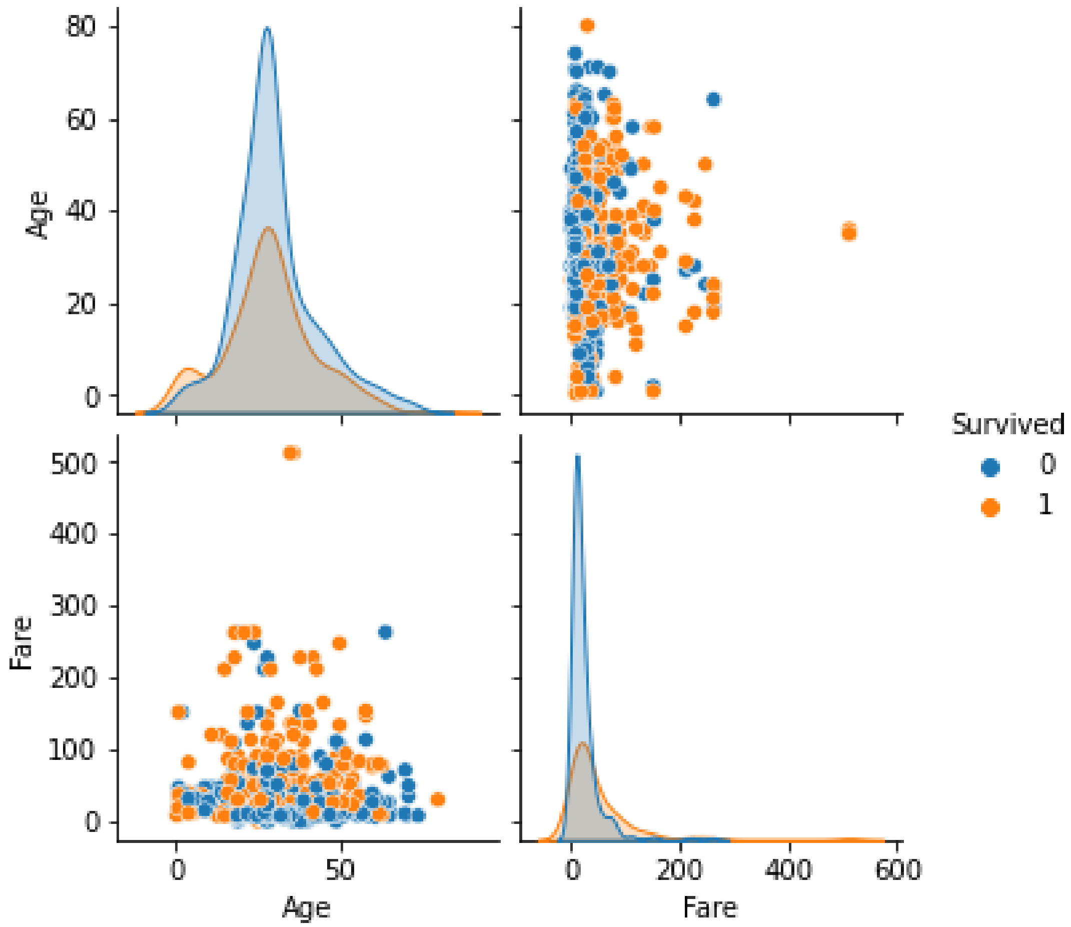
[24]:

```
sns.countplot(x='Parch', hue='Survived', data=ds)
plt.title('Survival by Number of Parents/Children')
plt.show()
```



```
[25]: sns.pairplot(ds[['Age', 'Fare', 'Survived']], hue='Survived')
```

```
[25]: <seaborn.axisgrid.PairGrid at 0x13c79234190>
```



[26]:

```
corr =ds.corr()
corr
```

[26]:

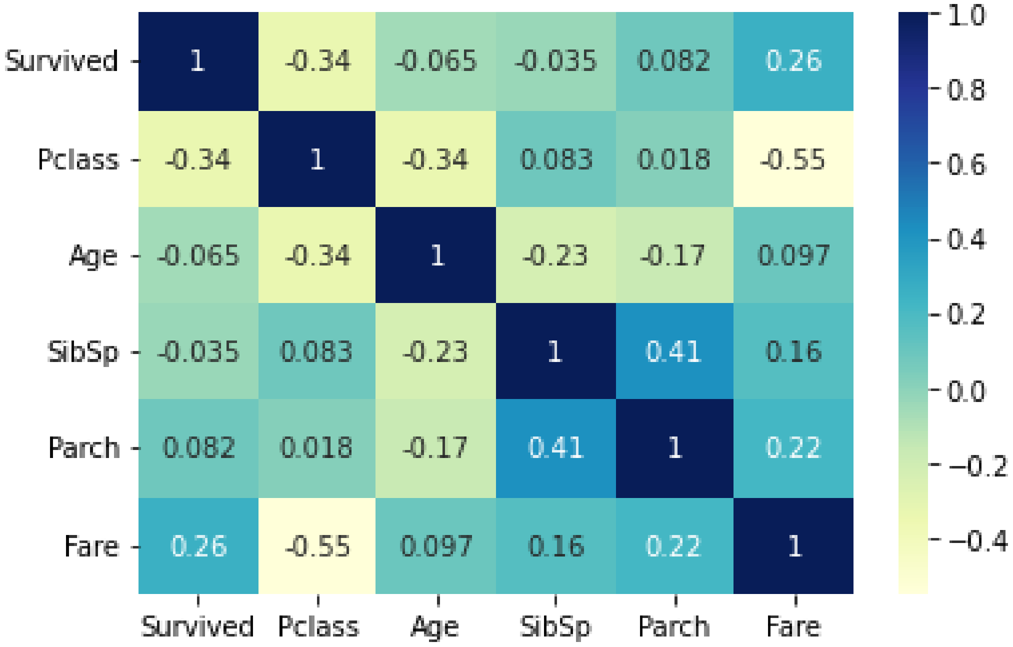
```
          Survived    Pclass       Age     SibSp     Parch      Fare
Survived  1.000000 -0.338481 -0.064910 -0.035322  0.081629  0.257307
Pclass   -0.338481  1.000000 -0.339898  0.083081  0.018443 -0.549500
Age      -0.064910 -0.339898  1.000000 -0.233296 -0.172482  0.096688
SibSp    -0.035322  0.083081 -0.233296  1.000000  0.414838  0.159651
Parch     0.081629  0.018443 -0.172482  0.414838  1.000000  0.216225
Fare      0.257307 -0.549500  0.096688  0.159651  0.216225  1.000000
```

[27]:

```
sns.heatmap(corr,annot=True,cmap="YlGnBu")
```

[27]:

```
<AxesSubplot:>
```



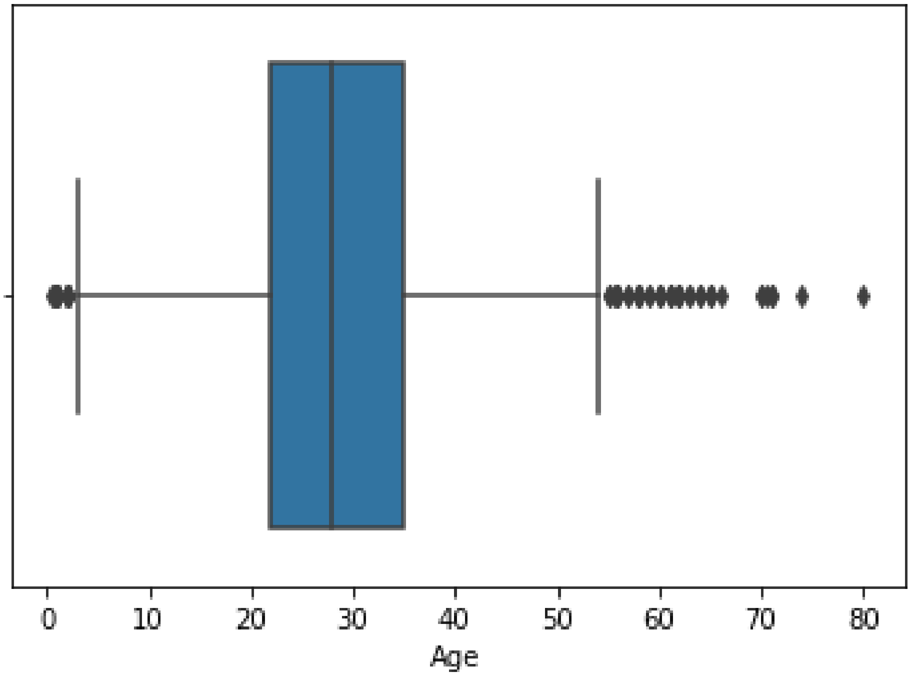
[28]:

```
sns.boxplot(ds.Age)
```

```
C:\Users\Sayani Roy Choudhury\anaconda3\lib\site-
packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will
be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
```

[28]:

```
<AxesSubplot:xlabel='Age'>
```



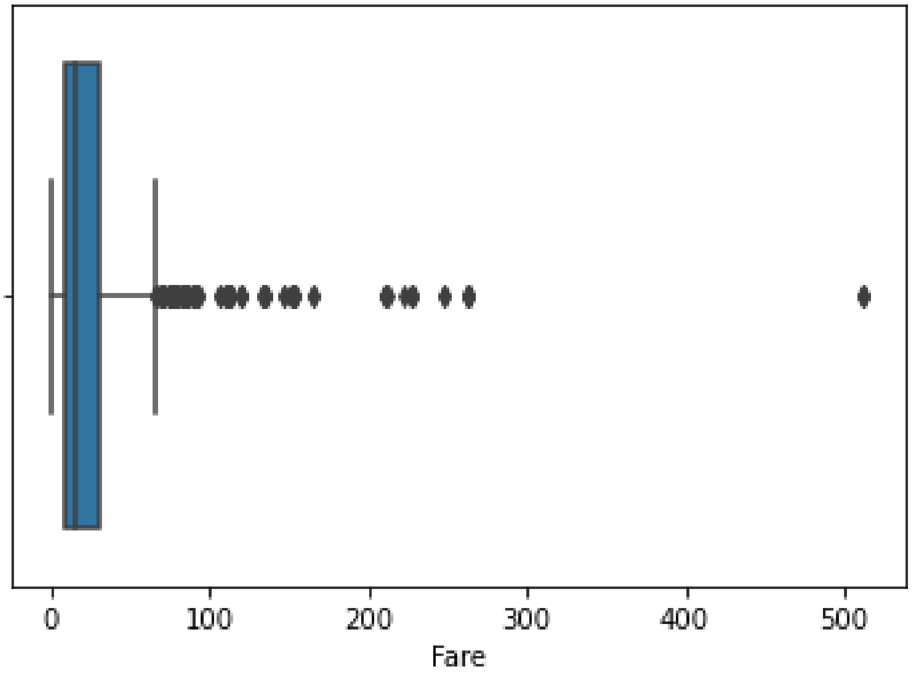
[29]:

```
sns.boxplot(ds.Fare)
```

```
C:\Users\Sayani Roy Choudhury\anaconda3\lib\site-
packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will
be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
```

[29]:

```
<AxesSubplot:xlabel='Fare'>
```



0.6 outlier removal by replacement with median

[30]:
```
upper_limit = ds['Age'].mean() + 3* ds['Age'].std() # Right from the mean
lower_limit = ds['Age'].mean() - 3* ds['Age'].std() # Left from the mean
print(upper_limit)
print(lower_limit)
```

```
68.42067214450208
-9.697507161337093
```

[31]:
```
quant=ds['Age'].quantile(q=[0.75,0.25])
```

[32]:
```
q3=quant.loc[0.75]
q3
```

[32]:
```
35.0
```

[33]:
```
q1=quant.loc[0.25]
q1
```

[33]:
```
22.0
```

[34]:
```
IQR=q3-q1#inter quantile
IQR
```

```
[34]: 13.0
```

```
[35]: maxwhisker=q3+1.5*IQR
      maxwhisker
```

```
[35]: 54.5
```

```
[36]: minwhisker=q1-1.5*IQR
      minwhisker
```

```
[36]: 2.5
```

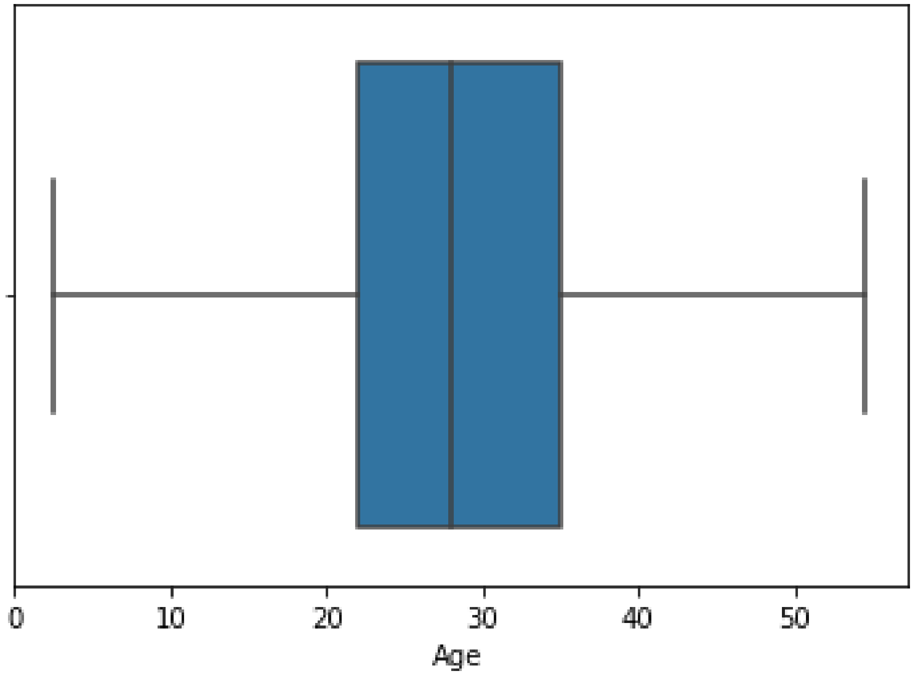
```
[37]: ds['Age']=np.where(ds.Age>54.5,54.5,ds.Age)
```

```
[38]: ds['Age']=np.where(ds.Age<2.5,2.5,ds.Age)
```

```
[39]: sns.boxplot(ds.Age)
```

```
C:\Users\Sayani Roy Choudhury\anaconda3\lib\site-
packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will
be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
```

```
[39]: <AxesSubplot:xlabel='Age'>
```



[40]:
```
upper_limit = ds['Fare'].mean() + 3* ds['Fare'].std() # Right from the mean
lower_limit = ds['Fare'].mean() - 3* ds['Fare'].std() # Left from the mean
print(upper_limit)
print(lower_limit)
```

```
181.2844937601173
-116.87607782296811
```

[41]:
```
quant=ds['Fare'].quantile(q=[0.75,0.25])
```

[42]:
```
q3=quant.loc[0.75]
q3
```

[42]:
```
31.0
```

[43]:
```
q1=quant.loc[0.25]
q1
```

[43]:
```
7.9104
```

[44]:
```
IQR=q3-q1#inter quantile
IQR
```

[44]:
```
23.0896
```

[45]:
```
maxwhisker=q3+1.5*IQR
maxwhisker
```

[45]:
```
65.6344
```

[46]:
```
minwhisker=q1-1.5*IQR
minwhisker
```

[46]:
```
-26.724
```

[47]:
```
ds['Fare']=np.where(ds.Fare>65.6344,65.6344,ds.Fare)
```

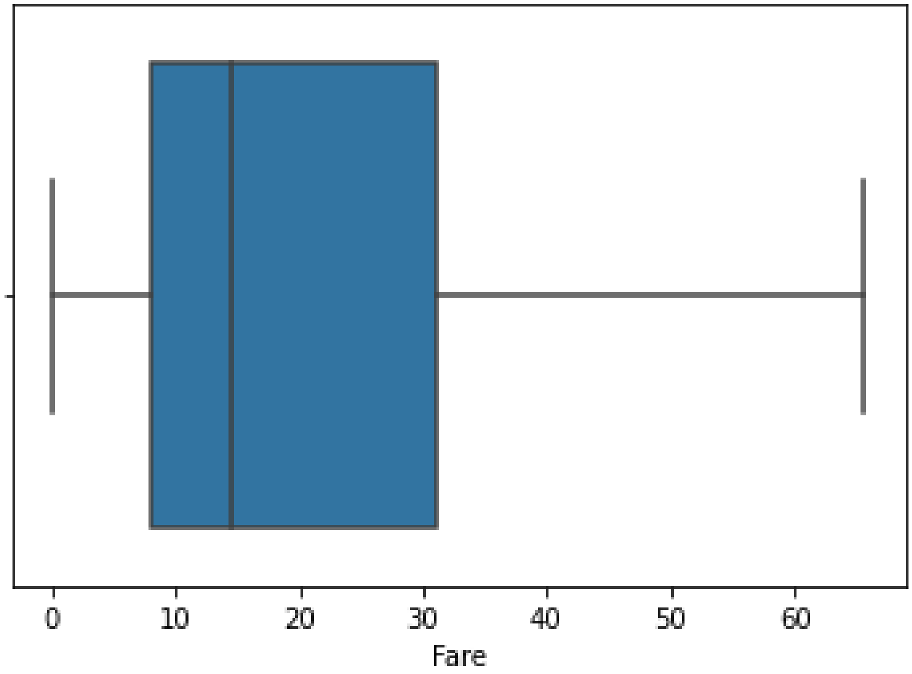
[48]:
```
ds['Fare']=np.where(ds.Fare<-26.724,-26.724,ds.Fare)
```

[49]:
```
sns.boxplot(ds.Fare)
```

```
C:\Users\Sayani Roy Choudhury\anaconda3\lib\site-
packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable
as a keyword arg: x. From version 0.12, the only valid positional argument will
be `data`, and passing other arguments without an explicit keyword will result
```

```
in an error or misinterpretation.
  warnings.warn(
```

[49]:
```
<AxesSubplot:xlabel='Fare'>
```



0.7 Spliting dependent and independent variables

[50]:
```
x=ds.drop(columns=["Survived"],axis=1)
y=ds["Survived"]
```

[51]:
```
x.head()
```

[51]:
```
   Pclass     Sex   Age  SibSp  Parch     Fare Embarked
0       3    male  22.0      1      0   7.2500        S
1       1  female  38.0      1      0  65.6344        C
2       3  female  26.0      0      0   7.9250        S
3       1  female  35.0      1      0  53.1000        S
4       3    male  35.0      0      0   8.0500        S
```

0.8 Encoding

[52]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
[53]: x["Sex"].nunique()
```

```
[53]: 2
```

```
[54]: x["Embarked"].nunique()
```

```
[54]: 3
```

```
[55]: x["Sex"]=le.fit_transform(x["Sex"])
```

```
[56]: x["Embarked"]=le.fit_transform(x["Embarked"])
```

```
[57]: x.head()
```

```
[57]:    Pclass  Sex   Age  SibSp  Parch     Fare  Embarked
      0       3    1  22.0      1      0   7.2500         2
      1       1    0  38.0      1      0  65.6344         0
      2       3    0  26.0      0      0   7.9250         2
      3       1    0  35.0      1      0  53.1000         2
      4       3    1  35.0      0      0   8.0500         2
```

0.9 Train test split

```
[58]: from sklearn.model_selection  import train_test_split
      x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

```
[59]: x_train.shape,y_train.shape,x_test.shape,y_test.shape
```

```
[59]: ((623, 7), (623,), (268, 7), (268,))
```

```
[ ]:
```

0.10 Feature Scaling

```
[60]: from sklearn.preprocessing import StandardScaler
      sc=StandardScaler()
```

```
[61]: x_train = sc.fit_transform(x_train)
      x_test = sc.fit_transform(x_test)
```

```
[62]: x_train
```

```
[62]: array([[-1.5325562 ,  0.72592065,  1.80447949, …, -0.47299765,
               0.08932336,  0.56710989],
             [-1.5325562 , -1.37756104,  1.63879184, …, -0.47299765,
               1.98540026, -2.03075381],
             [ 0.84844757,  0.72592065, -2.21344609, …,  1.93253327,
               1.0765501 ,  0.56710989],
             …,
             [ 0.84844757,  0.72592065, -0.10092851, …, -0.47299765,
              -0.82351937, -0.73182196],
             [ 0.84844757, -1.37756104,  0.5618221 , …, -0.47299765,
              -0.35456483,  0.56710989],
             [-0.34205431,  0.72592065,  2.09443288, …,  0.72976781,
               0.69330237,  0.56710989]])
```

```
[63]: x_test
```

```
[63]: array([[ 0.77963055,  0.76537495, -0.05174687, …, -0.47809977,
              -0.40150209, -1.76531134],
             [ 0.77963055,  0.76537495, -0.05174687, …, -0.47809977,
              -0.74607117,  0.63014911],
             [ 0.77963055,  0.76537495, -1.79564727, …,  0.87064484,
               0.33003698, -0.56758111],
             …,
             [ 0.77963055,  0.76537495, -0.13478974, …, -0.47809977,
              -0.40170659, -1.76531134],
             [ 0.77963055, -1.30654916, -0.88217563, …, -0.47809977,
              -0.74877454,  0.63014911],
             [-1.64991582,  0.76537495, -0.05174687, …, -0.47809977,
               0.25999892, -1.76531134]])
```

0.11 Name -Sayani Roy Choudhury

Registration no.-21BCE10336

```
[ ]:
```