21BCE8974_Assignment-3_15th_September

September 19, 2023

1 Assignment- 15th September

Name : E.Naga Sai Tarun Ganesh Reg.No: 21BCE8974

1.1 Data Preprocessing on titanic dataset

[1]:
```
# Import the Libraries.
# Import the dataset
# Checking for Null Values.


# Data Visualization.
# Outlier Detection
#        Splitting Dependent and Independent variables
#        Encoding
#        Feature Scaling.
#        Splitting Data into Train and Test.
```

1.1.1 1.Import the libraries

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

1.1.2 2. Importing Dataset

[3]:
```
df=pd.read_csv("Titanic-Dataset.csv")
```

[4]:
```
df
```

[4]:
```
     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]
```

[5]:
```
df.head()
```

[5]:
```
   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
```

[6]:
```
df.shape
```

[6]:
```
(891, 12)
```

[7]:
```
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
```

[8]:
```
df.describe()
```

[8]:
```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    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
```

[9]:
```
df.describe(include="all")
```

[9]:
```
        PassengerId    Survived      Pclass                     Name   Sex  \
count    891.000000  891.000000  891.000000                      891   891
unique          NaN         NaN         NaN                      891     2
top             NaN         NaN         NaN  Braund, Mr. Owen Harris  male
freq            NaN         NaN         NaN                        1   577
mean     446.000000    0.383838    2.308642                      NaN   NaN
std      257.353842    0.486592    0.836071                      NaN   NaN
min        1.000000    0.000000    1.000000                      NaN   NaN
25%      223.500000    0.000000    2.000000                      NaN   NaN
50%      446.000000    0.000000    3.000000                      NaN   NaN
75%      668.500000    1.000000    3.000000                      NaN   NaN
max      891.000000    1.000000    3.000000                      NaN   NaN

                Age       SibSp       Parch  Ticket        Fare    Cabin  \
count   714.000000  891.000000  891.000000     891  891.000000      204
unique         NaN         NaN         NaN     681         NaN      147
top            NaN         NaN         NaN  347082         NaN  B96 B98
freq           NaN         NaN         NaN       7         NaN        4
mean     29.699118    0.523008    0.381594     NaN   32.204208      NaN
std      14.526497    1.102743    0.806057     NaN   49.693429      NaN
min       0.420000    0.000000    0.000000     NaN    0.000000      NaN
25%      20.125000    0.000000    0.000000     NaN    7.910400      NaN
50%      28.000000    0.000000    0.000000     NaN   14.454200      NaN
75%      38.000000    1.000000    0.000000     NaN   31.000000      NaN
max      80.000000    8.000000    6.000000     NaN  512.329200      NaN

       Embarked
count       889
unique        3
top           S
freq        644
mean        NaN
std         NaN
min         NaN
25%         NaN
```

```
50%          NaN
75%          NaN
max          NaN
```

[10]:
```
df.corr(numeric_only=True)
```

[10]:
```
             PassengerId  Survived    Pclass       Age     SibSp     Parch  \
PassengerId     1.000000 -0.005007 -0.035144  0.036847 -0.057527 -0.001652
Survived       -0.005007  1.000000 -0.338481 -0.077221 -0.035322  0.081629
Pclass         -0.035144 -0.338481  1.000000 -0.369226  0.083081  0.018443
Age             0.036847 -0.077221 -0.369226  1.000000 -0.308247 -0.189119
SibSp          -0.057527 -0.035322  0.083081 -0.308247  1.000000  0.414838
Parch          -0.001652  0.081629  0.018443 -0.189119  0.414838  1.000000
Fare            0.012658  0.257307 -0.549500  0.096067  0.159651  0.216225

                 Fare
PassengerId  0.012658
Survived     0.257307
Pclass      -0.549500
Age          0.096067
SibSp        0.159651
Parch        0.216225
Fare         1.000000
```

[11]:
```
df.corr(numeric_only=True).Survived.sort_values(ascending = False)
```

[11]:
```
Survived       1.000000
Fare           0.257307
Parch          0.081629
PassengerId   -0.005007
SibSp         -0.035322
Age           -0.077221
Pclass        -0.338481
Name: Survived, dtype: float64
```

1.1.3 3.Handling Missing/Null Values

[12]:
```
df.isnull().any() # Null Values are Present in Age,Cabin and Embarked Column
```

[12]:
```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
Parch          False
```

```
Ticket         False
Fare           False
Cabin           True
Embarked        True
dtype: bool
```

[28]:
```
df.isnull().sum().sort_values(ascending=False)
```

[28]:
```
Cabin          687
Age            177
Embarked         2
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
SibSp            0
Parch            0
Ticket           0
Fare             0
dtype: int64
```

[12]:
```
sum(df.Cabin.isnull())
```

[12]:
```
687
```

[13]:
```
sum(df.Age.isnull())
```

[13]:
```
177
```

[14]:
```
df["Age"].fillna(df["Age"].mean(),inplace=True)
```

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

[15]:
```
2
```

[16]:
```
df["Embarked"].fillna(df["Embarked"].mode()[0],inplace=True)
```

[19]:
```
df.describe()
```

[19]:

	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	

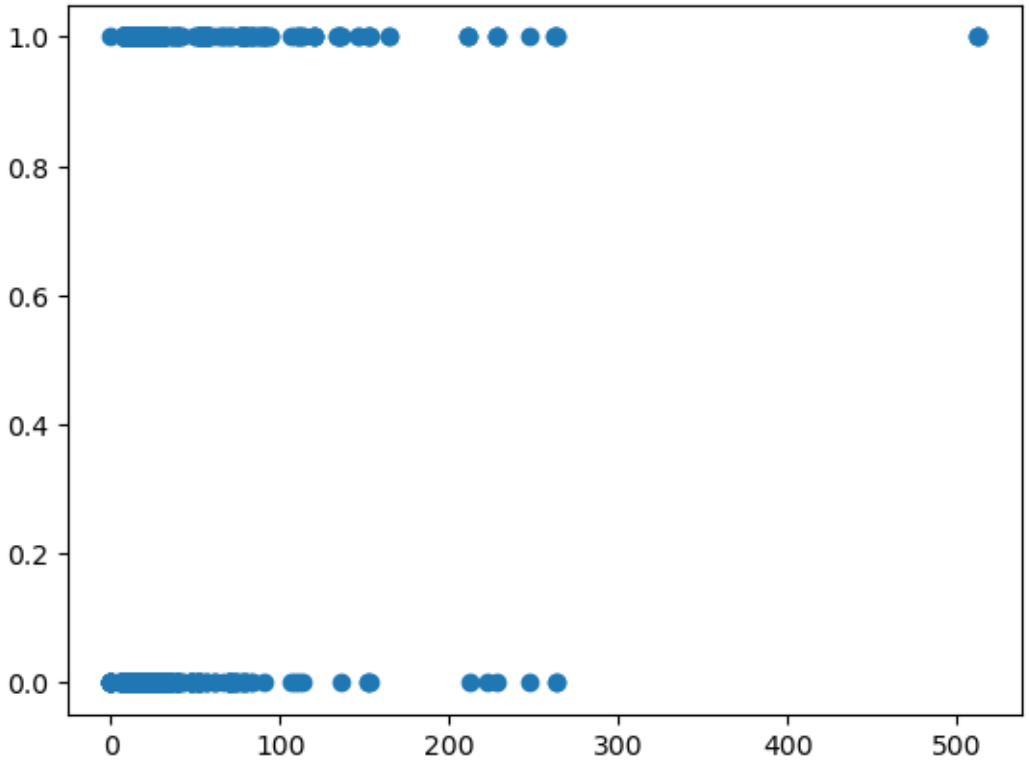
```
max     891.000000    1.000000    3.000000   80.000000    8.000000
```

```
             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
```

1.1.4 4. Data Visualisation

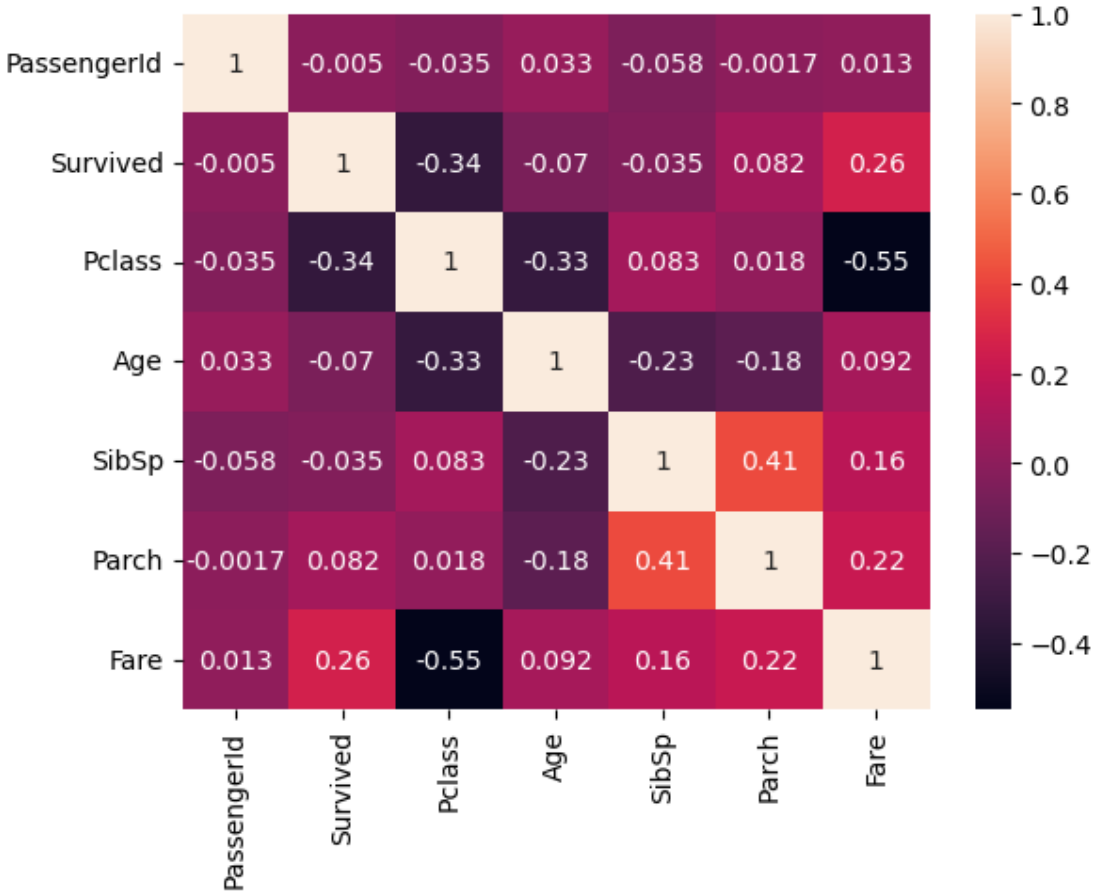
[20]:
```
plt.scatter(df["Fare"],df["Survived"])
```

[20]:
```
<matplotlib.collections.PathCollection at 0x14d8542ca90>
```



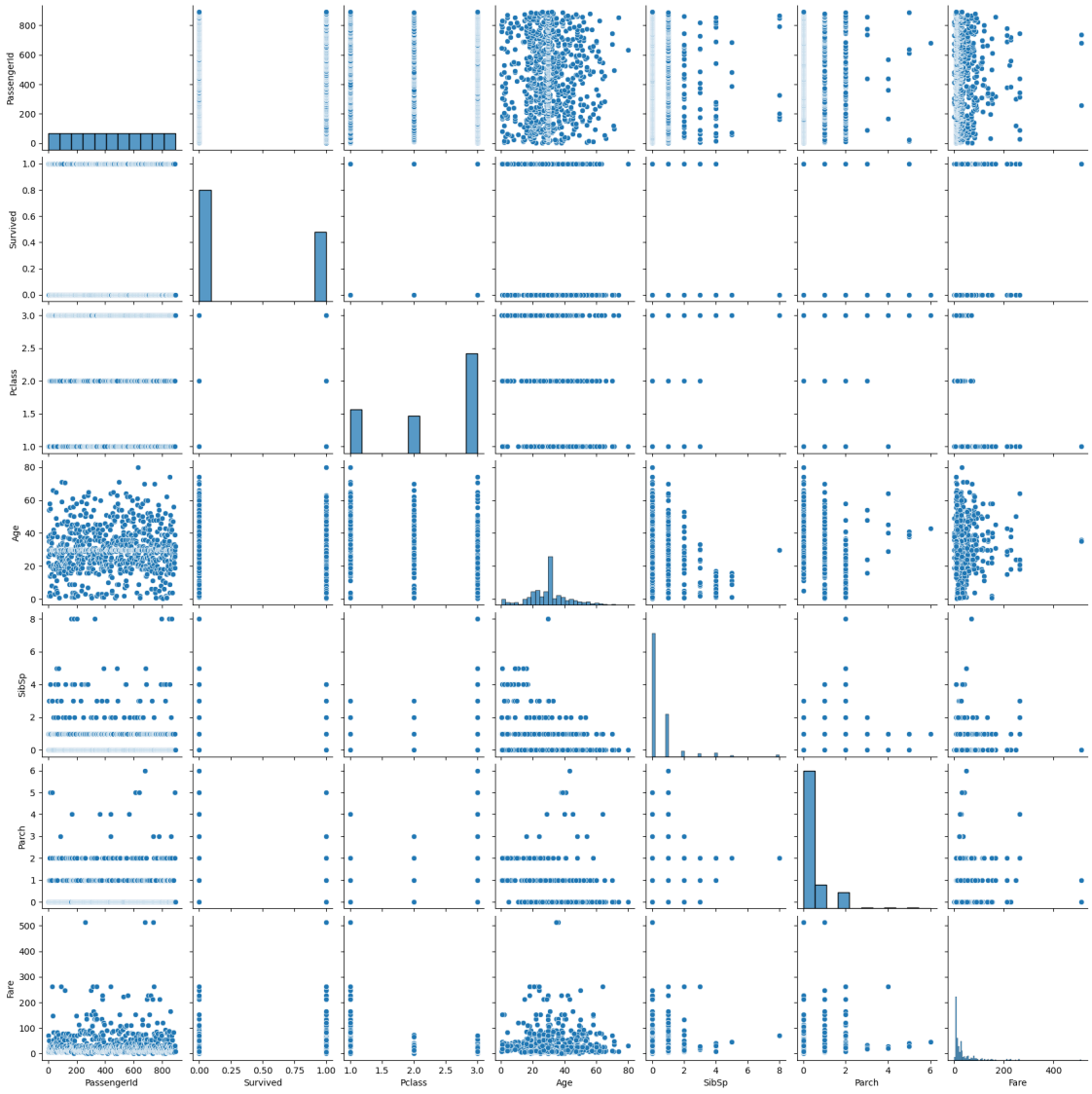
[21]:
```
sns.heatmap(df.corr(numeric_only=True),annot=True)
```

[21]:
```
<Axes: >
```



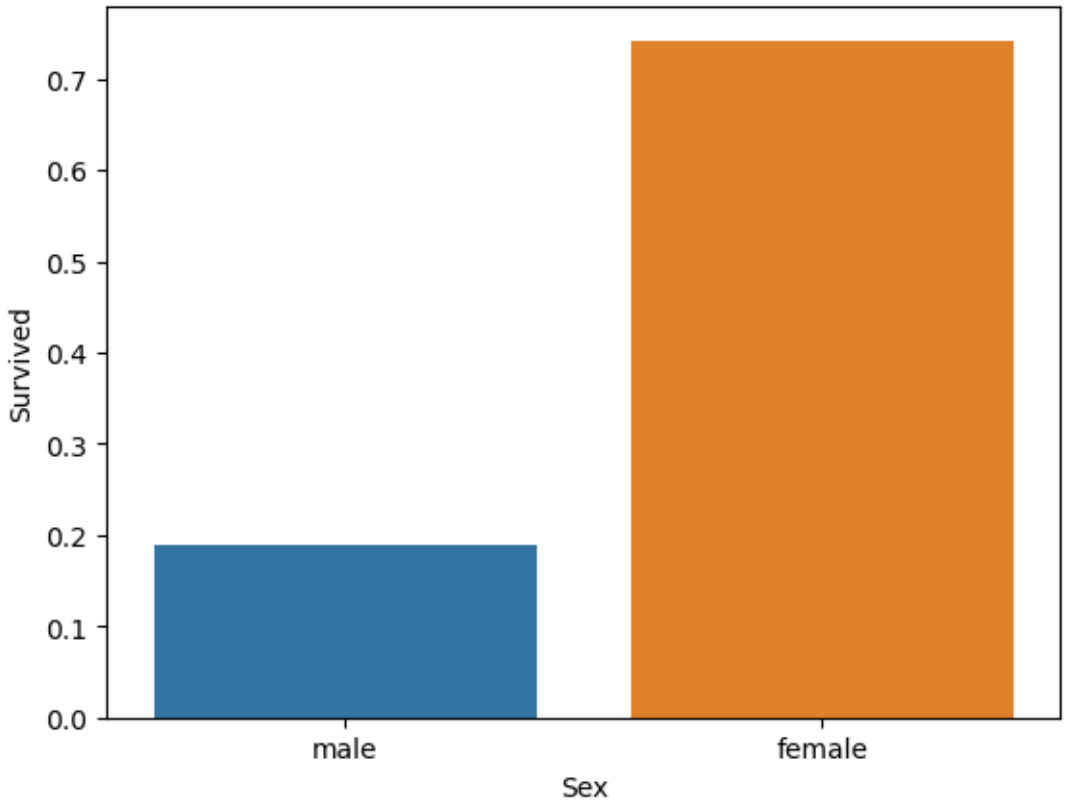
[22]:
```
sns.pairplot(df)
```

[22]: <seaborn.axisgrid.PairGrid at 0x14d85e18810>



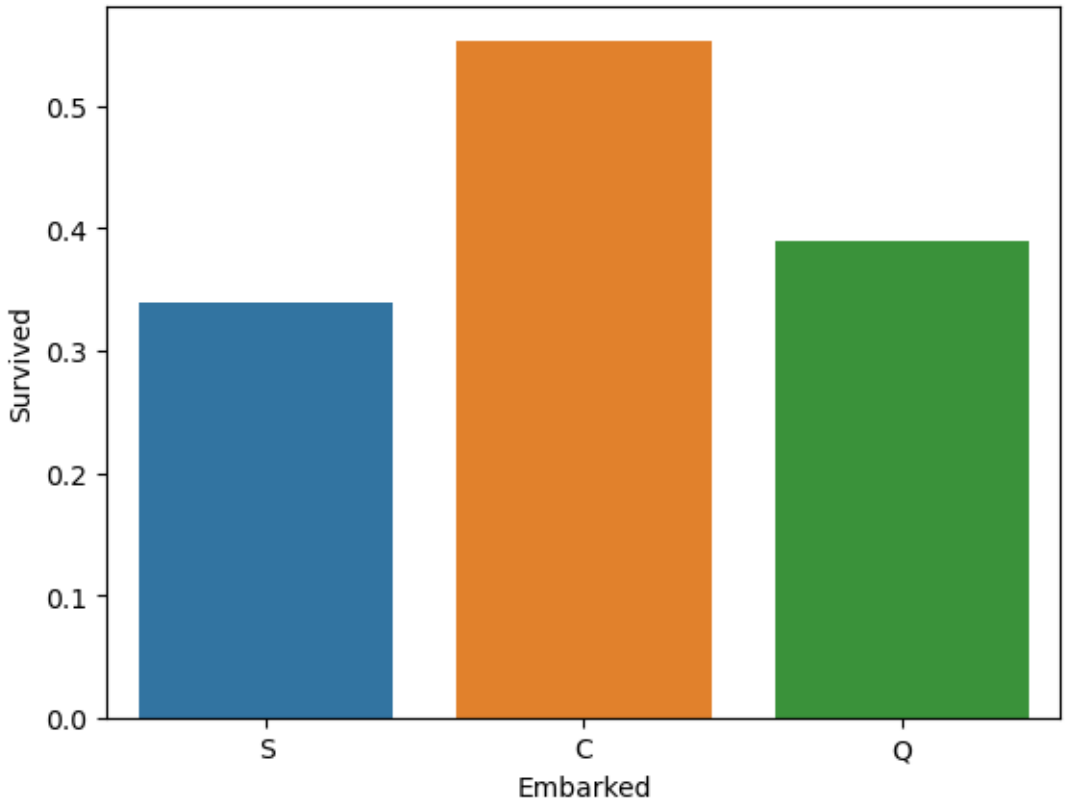
[24]:
```
sns.barplot(x=df["Sex"],y=df["Survived"],errorbar=('ci',0))
```

[24]: <Axes: xlabel='Sex', ylabel='Survived'>



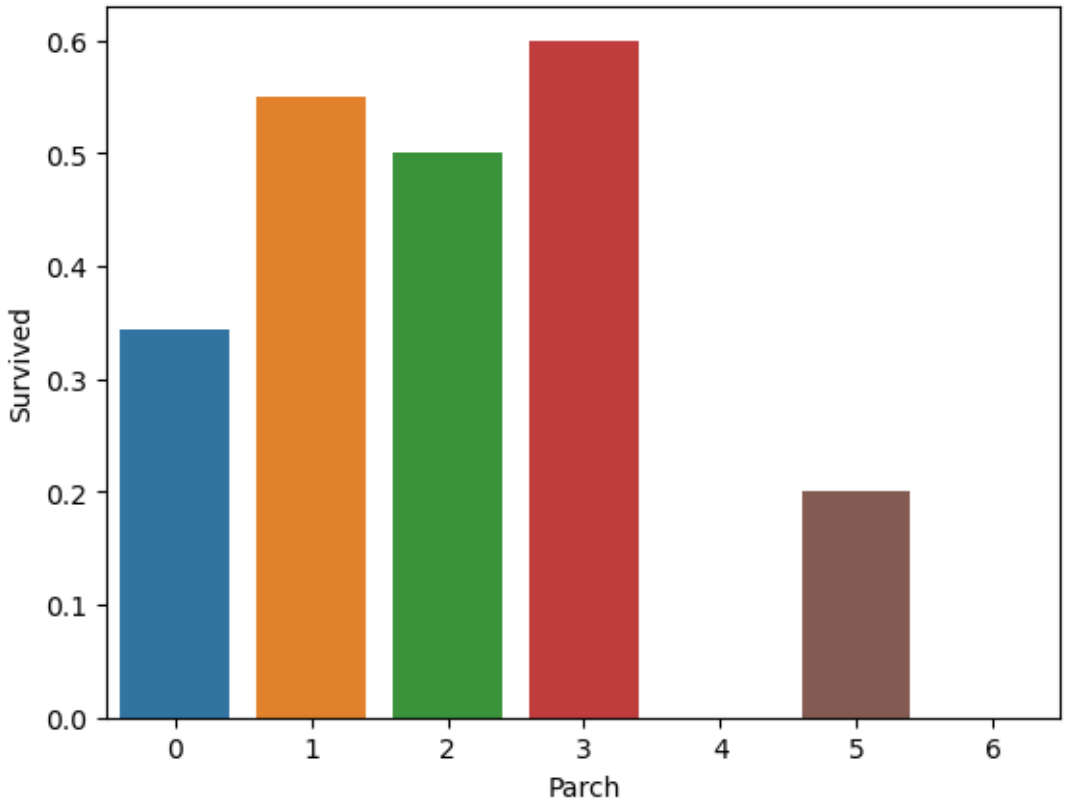
[25]:
```
sns.barplot(x=df["Embarked"],y=df["Survived"],errorbar=('ci',0))
```

[25]:
```
<Axes: xlabel='Embarked', ylabel='Survived'>
```



[26]:
```
sns.barplot(x=df["Parch"],y=df["Survived"],errorbar=('ci',0))
```

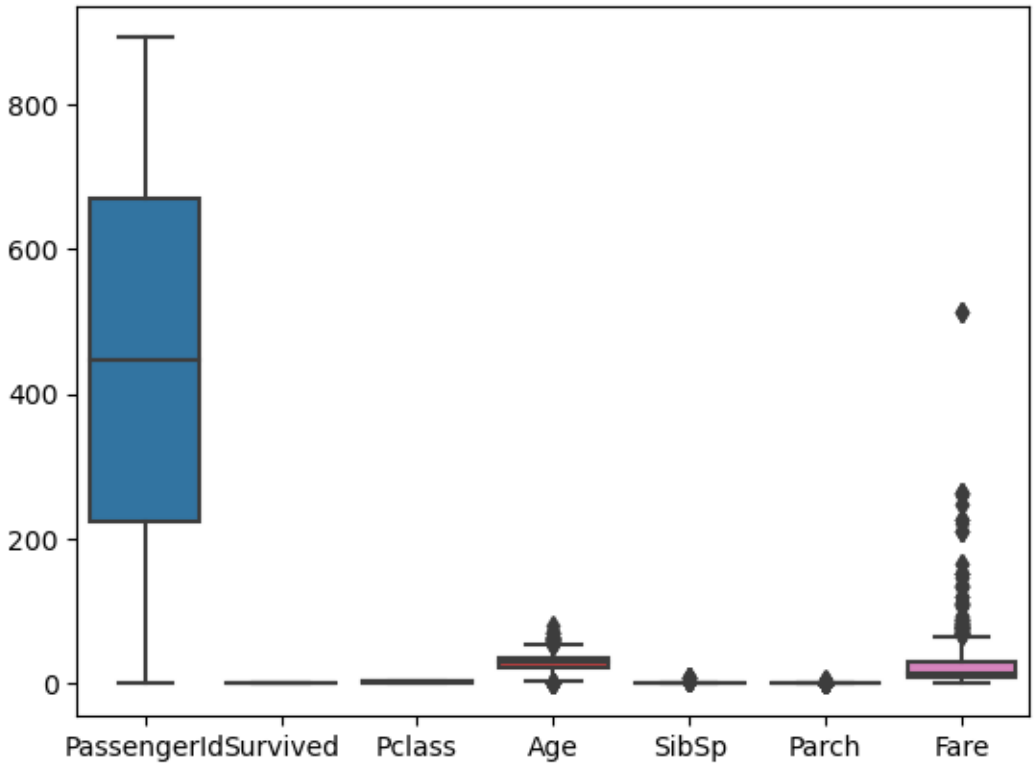
[26]:
```
<Axes: xlabel='Parch', ylabel='Survived'>
```



1.1.5 5.Outlier Detection

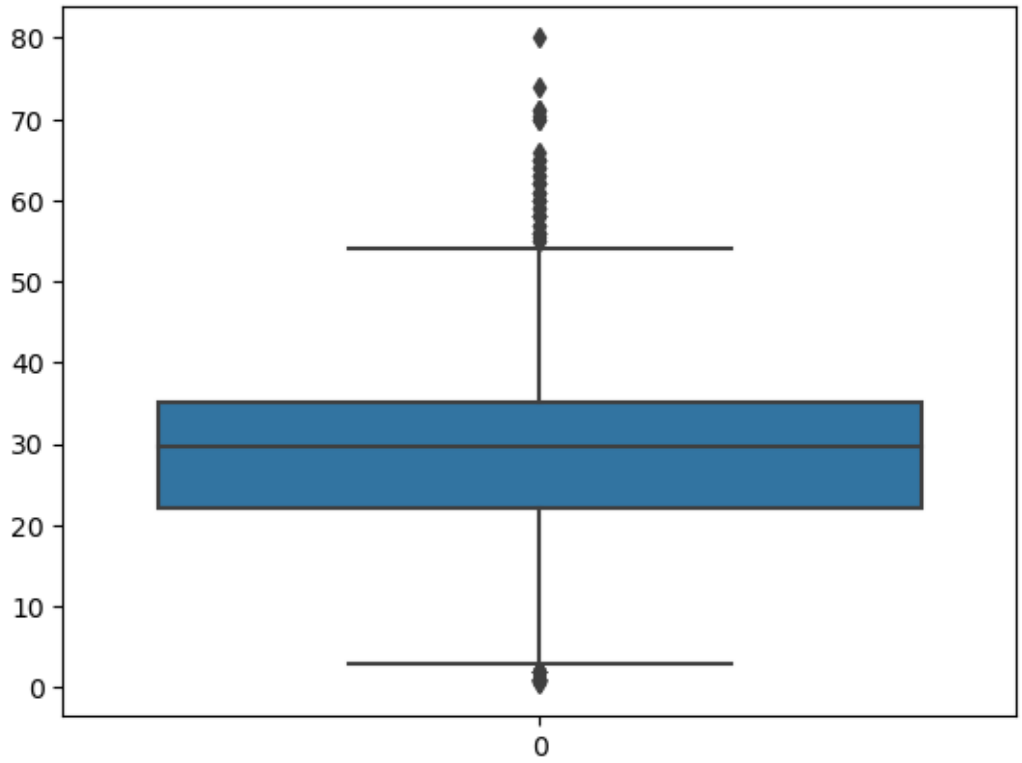
[27]:
```
sns.boxplot(df)
```

[27]:
```
<Axes: >
```



[29]:
```
sns.boxplot(df.Age)
```

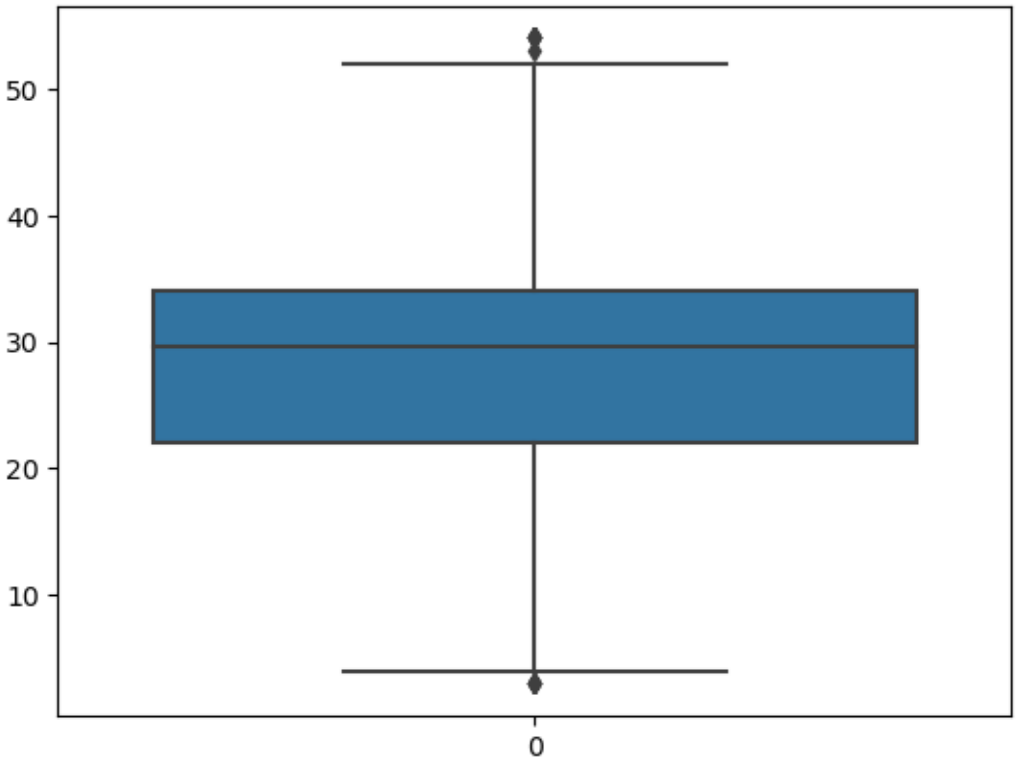
[29]: <Axes: >



[30]:
```
Q1 = df['Age'].quantile(0.25)
Q3 = df['Age'].quantile(0.75)

IQR = Q3 - Q1

threshold = 1.5 * IQR

df = df[(df['Age'] >= Q1 - threshold) & (df['Age'] <= Q3 + threshold)]
```

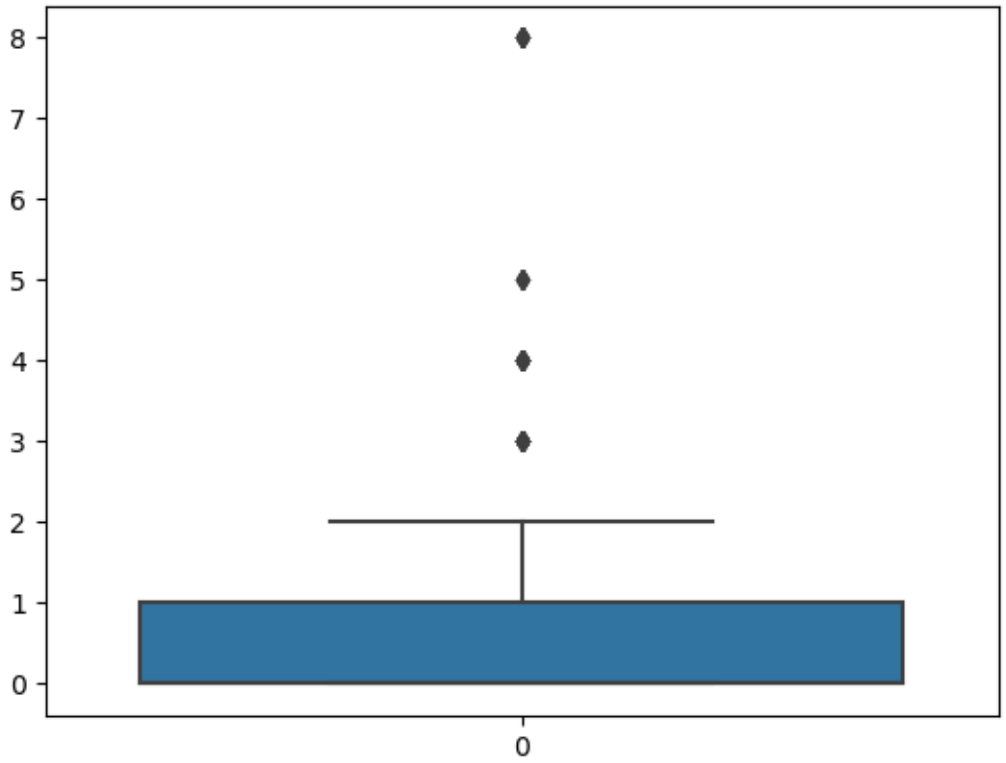
[31]:
```
sns.boxplot(df.Age)
```

[31]: <Axes: >



[32]:
```
sns.boxplot(df.SibSp)
```

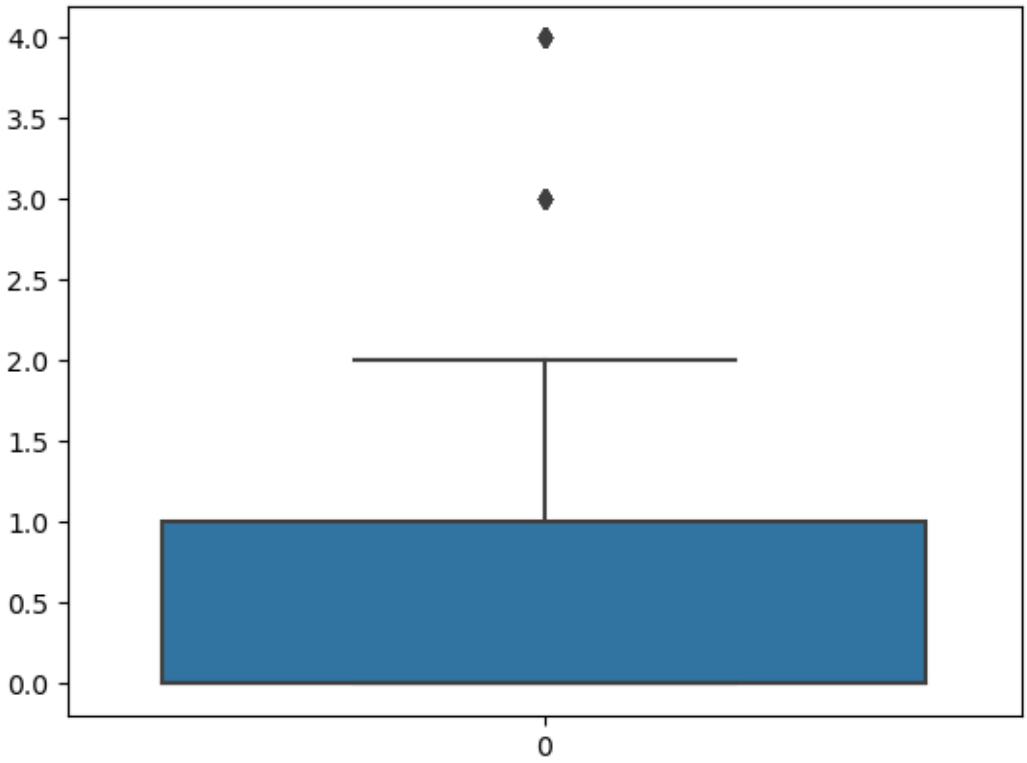
[32]: <Axes: >



[34]:
```
p99 = df.SibSp.quantile(0.99)
df = df[df.SibSp < p99]
```

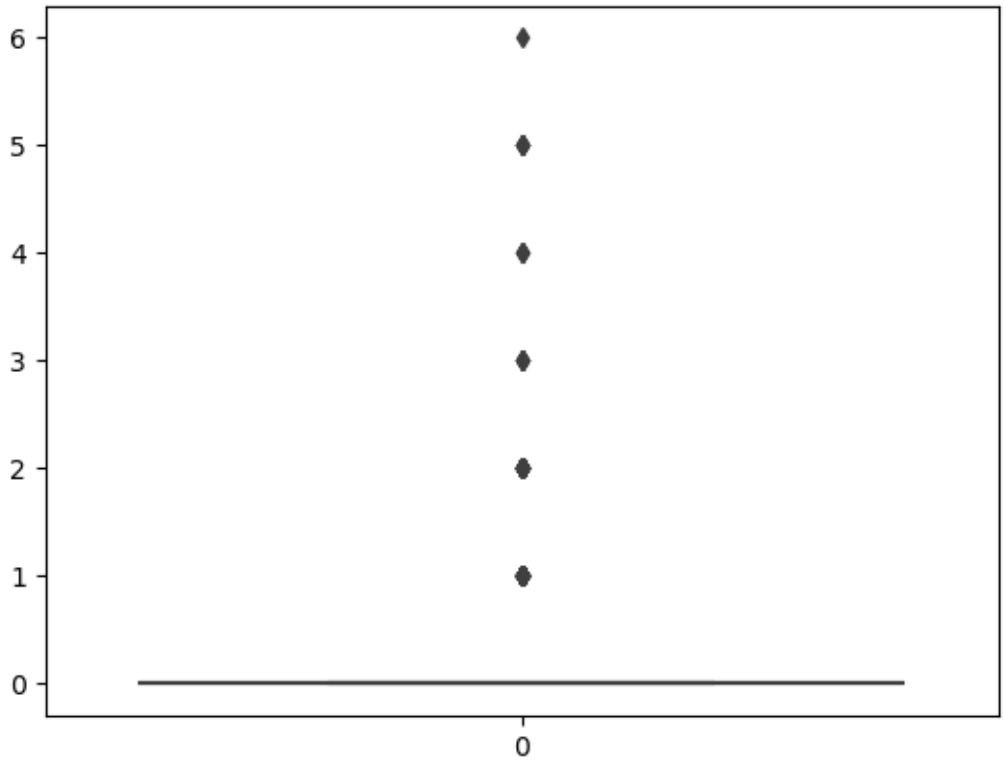
[35]:
```
sns.boxplot(df.SibSp)
```

[35]: <Axes: >



[36]:
```
sns.boxplot(df.Parch)
```

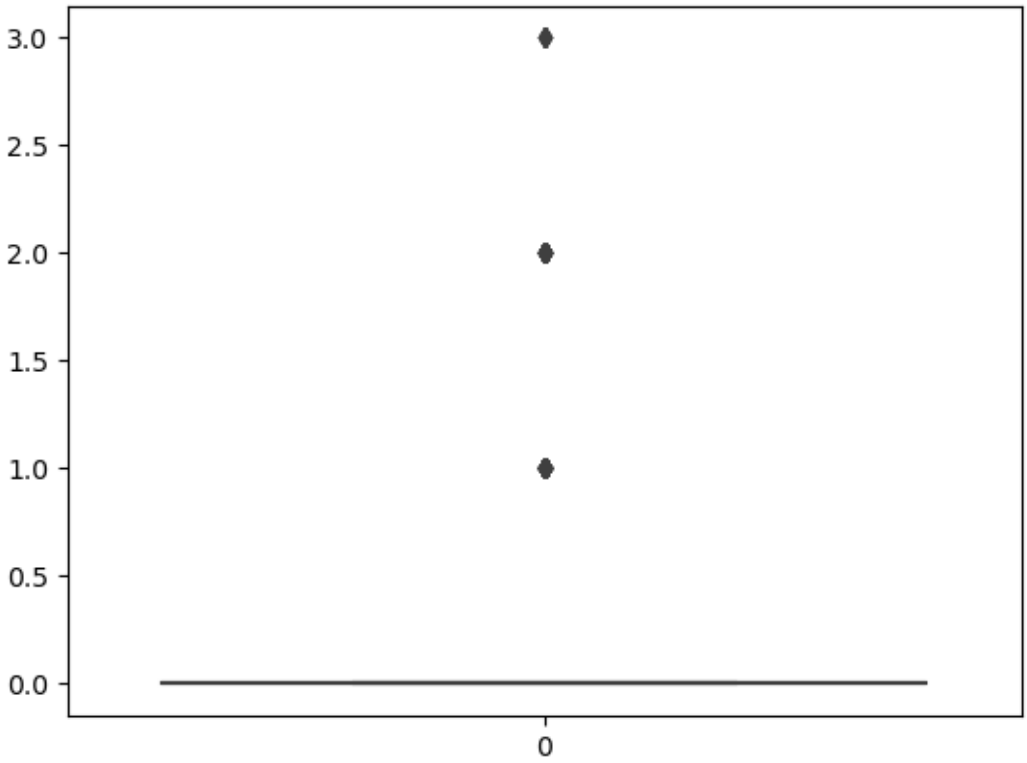
[36]: <Axes: >



[37]:
```
p99 = df.Parch.quantile(0.99)
df = df[df.Parch < p99]
```

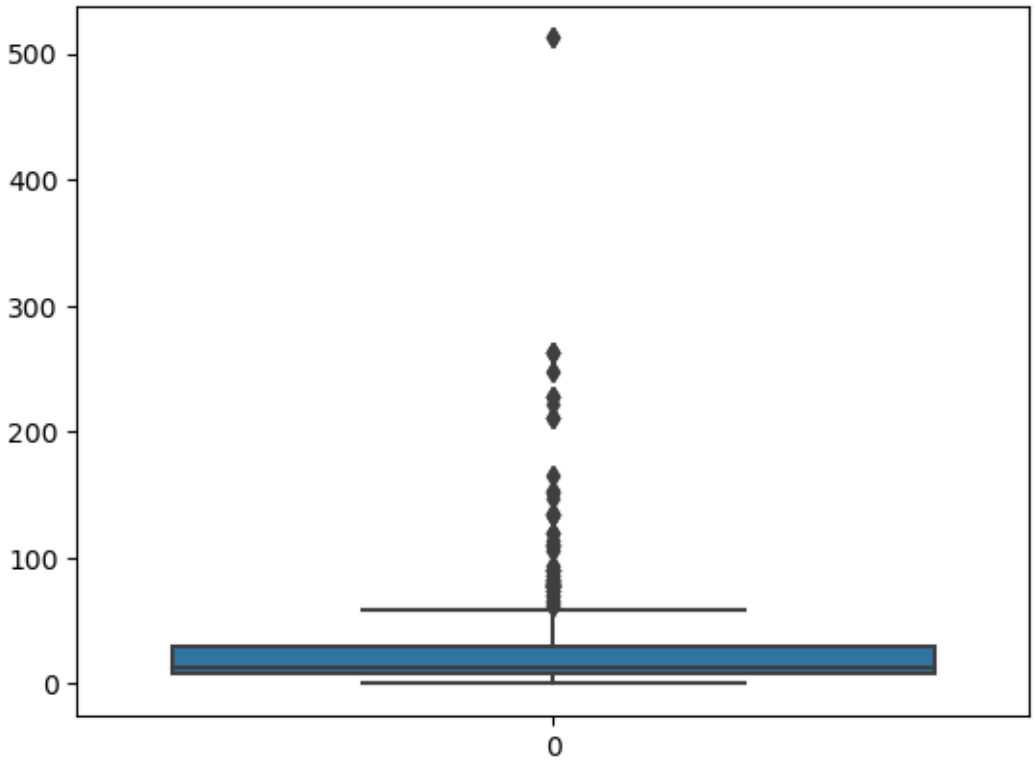
[38]:
```
sns.boxplot(df["Parch"])
```

[38]: <Axes: >



[39]:
```
sns.boxplot(df["Fare"])
```

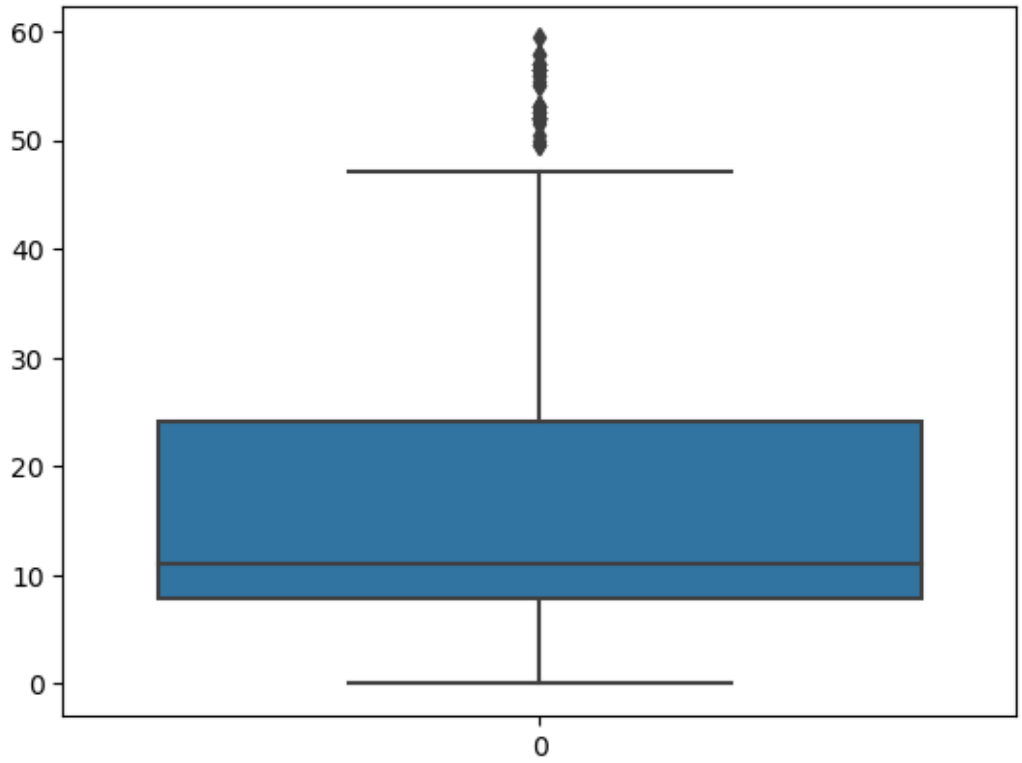
[39]: <Axes: >



[40]:
```
Q1 = df['Fare'].quantile(0.25)
Q3 = df['Fare'].quantile(0.75)

IQR = Q3 - Q1

threshold = 1.5 * IQR

df = df[(df['Fare'] >= Q1 - threshold) & (df['Fare'] <= Q3 + threshold)]
```

[41]:
```
sns.boxplot(df.Fare)
```

[41]: <Axes: >



1.1.6 6. Splitting Dependent and Independent Columns

[44]:
```
x = df.drop(columns=["Survived","PassengerId","Name","Ticket","Cabin"],axis=1)
```

[45]:
```
x.head()
```

[45]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	male	22.000000	1	0	7.2500	S
2	3	female	26.000000	0	0	7.9250	S
3	1	female	35.000000	1	0	53.1000	S
4	3	male	35.000000	0	0	8.0500	S
5	3	male	29.699118	0	0	8.4583	Q

[46]:
```
y = pd.Series(df["Survived"])
```

[47]:
```
y.head()
```

[47]:
```
0    0
2    1
3    1
4    0
5    0
```

```
Name: Survived, dtype: int64
```

1.1.7 7.Encoding

[48]:
```
from sklearn.preprocessing import LabelEncoder
```

[49]:
```
le = LabelEncoder()
```

[50]:
```
x["Sex"] = le.fit_transform(x["Sex"])
```

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

[51]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1	22.000000	1	0	7.2500	S
2	3	0	26.000000	0	0	7.9250	S
3	1	0	35.000000	1	0	53.1000	S
4	3	1	35.000000	0	0	8.0500	S
5	3	1	29.699118	0	0	8.4583	Q

[52]:
```
print(le.classes_)
```

```
['female' 'male']
```

[53]:
```
mapping=dict(zip(le.classes_,range(len(le.classes_))))
```

[54]:
```
mapping
```

[54]:
```
{'female': 0, 'male': 1}
```

[55]:
```
le1 = LabelEncoder()
```

[56]:
```
x["Embarked"] = le1.fit_transform(x["Embarked"])
```

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

[57]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1	22.000000	1	0	7.2500	2
2	3	0	26.000000	0	0	7.9250	2
3	1	0	35.000000	1	0	53.1000	2
4	3	1	35.000000	0	0	8.0500	2
5	3	1	29.699118	0	0	8.4583	1

[58]:
```
print(le1.classes_)
```

```
['C' 'Q' 'S']
```

[59]:
```
mapping1=dict(zip(le1.classes_,range(len(le1.classes_))))
```

[60]:
```
mapping
```

[60]:
```
{'female': 0, 'male': 1}
```

1.1.8 8. Feature Scaling

[61]:
```
from sklearn.preprocessing import MinMaxScaler
ms = MinMaxScaler()
```

[63]:
```
x_Scaled=pd.DataFrame(ms.fit_transform(x),columns = x.columns)
```

[64]:
```
x_Scaled.head()
```

[64]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1.0	1.0	0.372549	0.25	0.0	0.122054	1.0
1	1.0	0.0	0.450980	0.00	0.0	0.133418	1.0
2	0.0	0.0	0.627451	0.25	0.0	0.893939	1.0
3	1.0	1.0	0.627451	0.00	0.0	0.135522	1.0
4	1.0	1.0	0.523512	0.00	0.0	0.142396	0.5

1.1.9 9. Splitting Training and Testing Data

[65]:
```
from sklearn.model_selection import train_test_split
```

[66]:
```
x_train,x_test,y_train,y_test = train_test_split(x_Scaled,y,test_size = 0.
↪2,random_state =0)
```

[67]:
```
print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

```
(562, 7) (141, 7) (562,) (141,)
```

[68]:
```
x_train.head()
```

[68]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
251	0.5	0.0	0.529412	0.00	0.000000	0.207912	0.5
310	1.0	0.0	0.352941	0.25	0.000000	0.165404	1.0
582	1.0	1.0	0.549020	0.00	0.000000	0.133418	1.0
45	0.5	0.0	0.039216	0.25	0.666667	0.467172	1.0
478	0.0	1.0	0.470588	0.00	0.000000	0.513468	1.0

[69]:
```
x_test.head()
```

[69]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
539	1.0	1.0	0.523512	0.00	0.0	0.951108	1.0
477	1.0	1.0	0.529412	0.00	0.0	0.132926	1.0
548	1.0	1.0	0.450980	0.25	0.0	0.132226	1.0
605	0.5	0.0	1.000000	0.25	1.0	0.387205	1.0
518	1.0	1.0	0.647059	0.00	0.0	0.126192	1.0

[70]:
```
y_train.head()
```

[70]:
```
322    1
402    0
744    1
58     1
607    1
Name: Survived, dtype: int64
```

[71]:
```
y_test.head()
```

[71]:
```
692    1
606    0
704    0
774    1
663    0
Name: Survived, dtype: int64
```