Vajjah Srinivasa Taaran Assignment - 3

In [1]:

```
import numpy as np
```

In [2]:

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

In [3]:

```
df = pd.read_csv("Titanic-Dataset.csv")
df
```

Out[3]:

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

891 rows × 12 columns

In [4]:

```
df.head()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cal
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	N
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	N
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	N

In [5]:

```
df.isnull().any()
```

Out[5]:

```
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
```

In [6]:

```
df.isnull().sum()
```

Out[6]:

```
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 [7]:

```
df.describe()
```

Out[7]:

	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 [8]:

```
df.shape
```

Out[8]:

```
(891, 12)
```

In [9]:

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

In [10]:

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

In [11]:

```
df.isnull().any()
```

Out[11]:

```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age            False
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin          False
Embarked       False
dtype: bool
```

In [12]:

```
df.isnull().sum()
```

Out[12]:

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

In [13]:

```
corr = df.corr()
corr
```

```
/var/folders/0g/xqmh0yz92jx_s8ljsv3x08wr0000gn/T/ipykernel_92535/32818
36264.py:1: FutureWarning: The default value of numeric_only in DataFr
ame.corr is deprecated. In a future version, it will default to False.
Select only valid columns or specify the value of numeric_only to sile
nce this warning.
  corr = df.corr()
```

Out[13]:

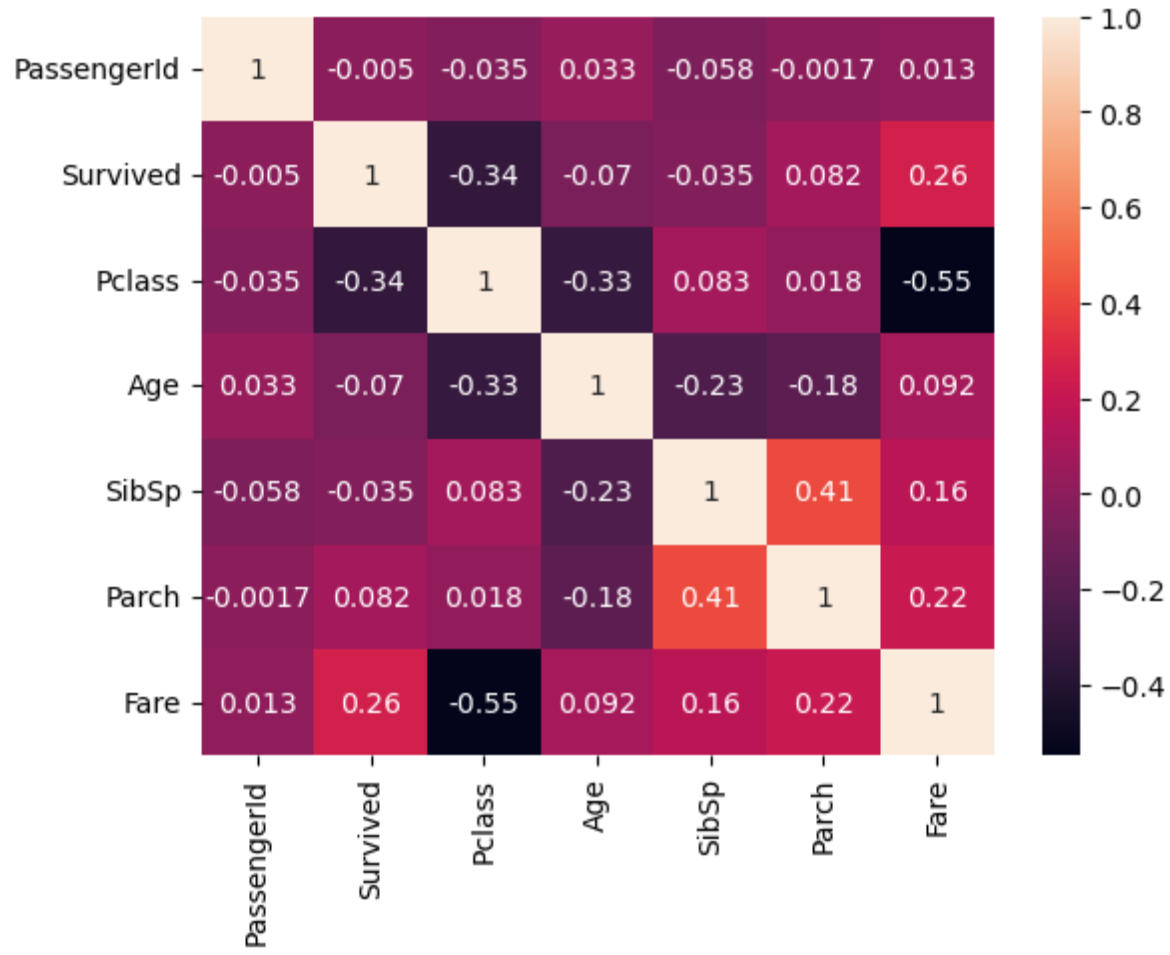
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.033207	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.069809	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.331339	0.083081	0.018443	-0.549500
Age	0.033207	-0.069809	-0.331339	1.000000	-0.232625	-0.179191	0.091566
SibSp	-0.057527	-0.035322	0.083081	-0.232625	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.179191	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.091566	0.159651	0.216225	1.000000

In [14]:

```
sns.heatmap(corr,annot = True)
```

Out[14]:

<Axes: >

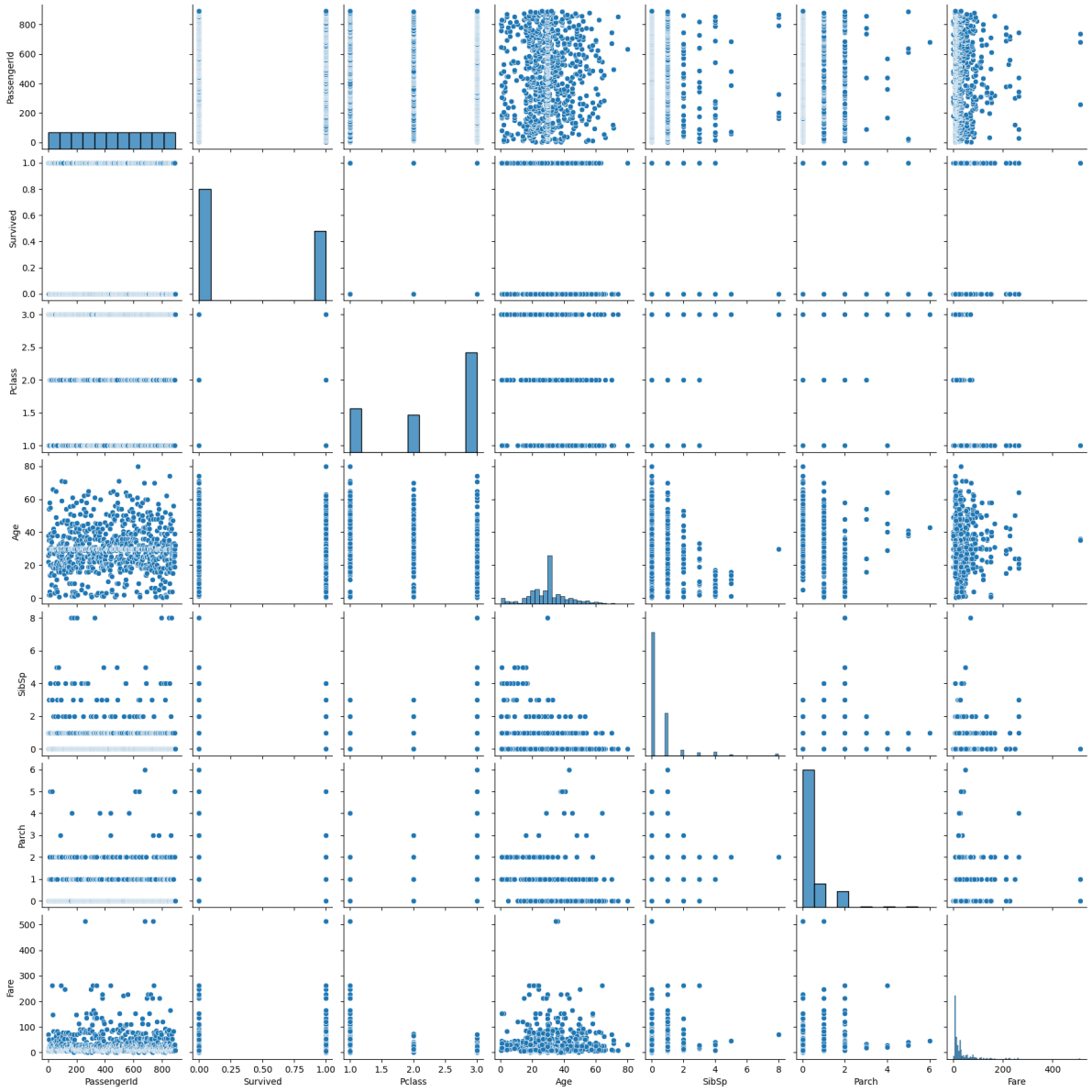


In [15]:

```
sns.pairplot(df)
```

Out[15]:

```
<seaborn.axisgrid.PairGrid at 0x15adda410>
```

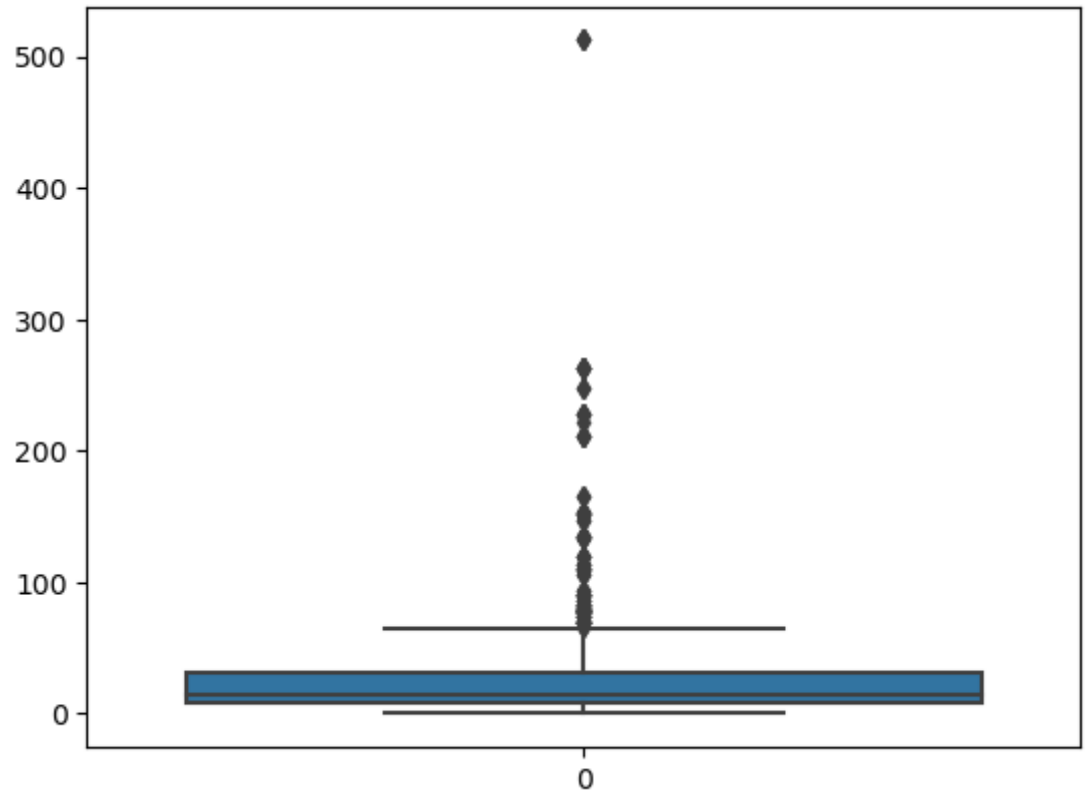


In [16]:

```
sns.boxplot(df.Fare)
```

Out[16]:

```
<Axes: >
```



In [17]:

```
q1 = df.Fare.quantile(0.25)
q3 = df.Fare.quantile(0.75)
print(q1)
print(q3)
```

```
7.9104
31.0
```

In [18]:

```
q3-q1
```

Out[18]:

```
23.0896
```

In [19]:

```
upperlimit = q3+1.5*(q3-q1)
upperlimit
```

Out[19]:

```
65.6344
```

In [20]:

```
lowerlimit = q1-1.5*(q3-q1)
lowerlimit
```

Out[20]:

```
-26.724
```

In [21]:

```
df.median()
```

```
/var/folders/0g/xqmh0yz92jx_s8ljsv3x08wr0000gn/T/ipykernel_92535/53005
1474.py:1: FutureWarning: The default value of numeric_only in DataFra
me.median is deprecated. In a future version, it will default to Fals
e. In addition, specifying 'numeric_only=None' is deprecated. Select o
nly valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

Out[21]:

```
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.454200
dtype: float64
```

In [22]:

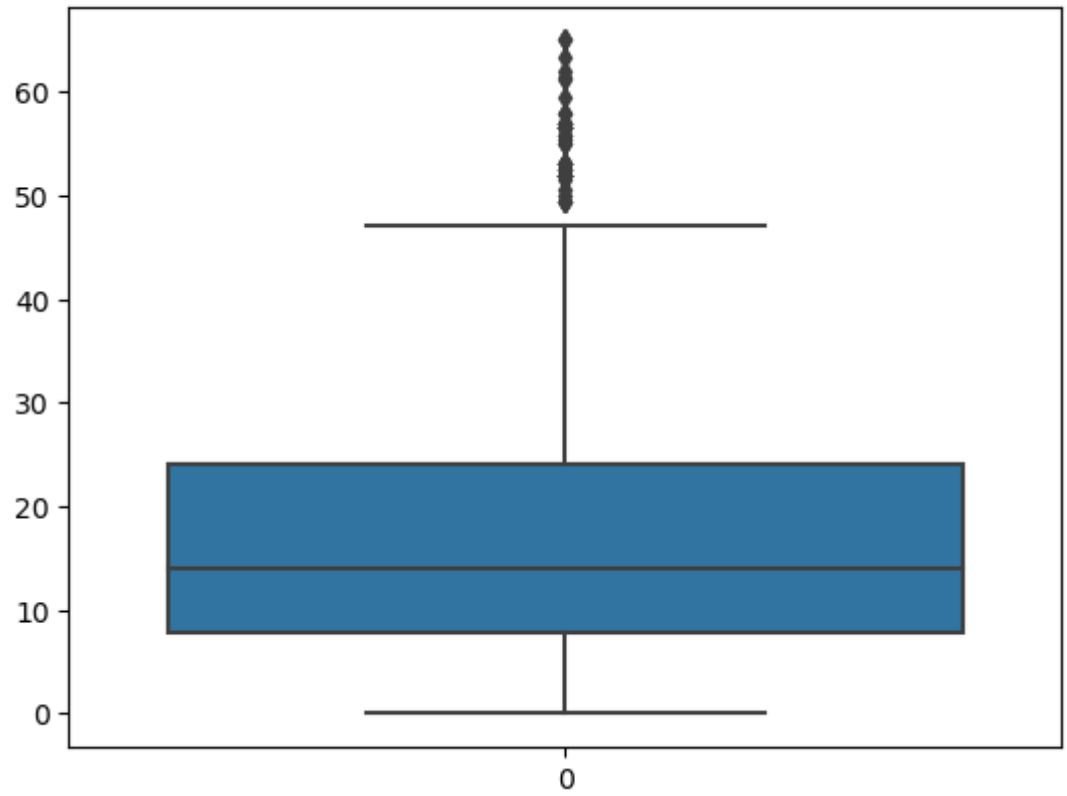
```
df["Fare"]=np.where(df["Fare"]>upperlimit,14,df['Fare'])
```

In [23]:

```
sns.boxplot(df.Fare)
```

Out[23]:

```
<Axes: >
```



In [24]:

```
q1 = df.Fare.quantile(0.25)
q3 = df.Fare.quantile(0.75)
print(q1)
print(q3)
```

```
7.9104
24.15
```

In [25]:

```
q3-q1
upperlimit = q3+1.5*(q3-q1)
upperlimit
lowerlimit = q1-1.5*(q3-q1)
lowerlimit
df.median()
```

```
/var/folders/0g/xqmh0yz92jx_s8ljsv3x08wr0000gn/T/ipykernel_92535/38262
84025.py:6: FutureWarning: The default value of numeric_only in DataFr
ame.median is deprecated. In a future version, it will default to Fals
e. In addition, specifying 'numeric_only=None' is deprecated. Select o
nly valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

Out[25]:

```
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.000000
dtype: float64
```

In [26]:

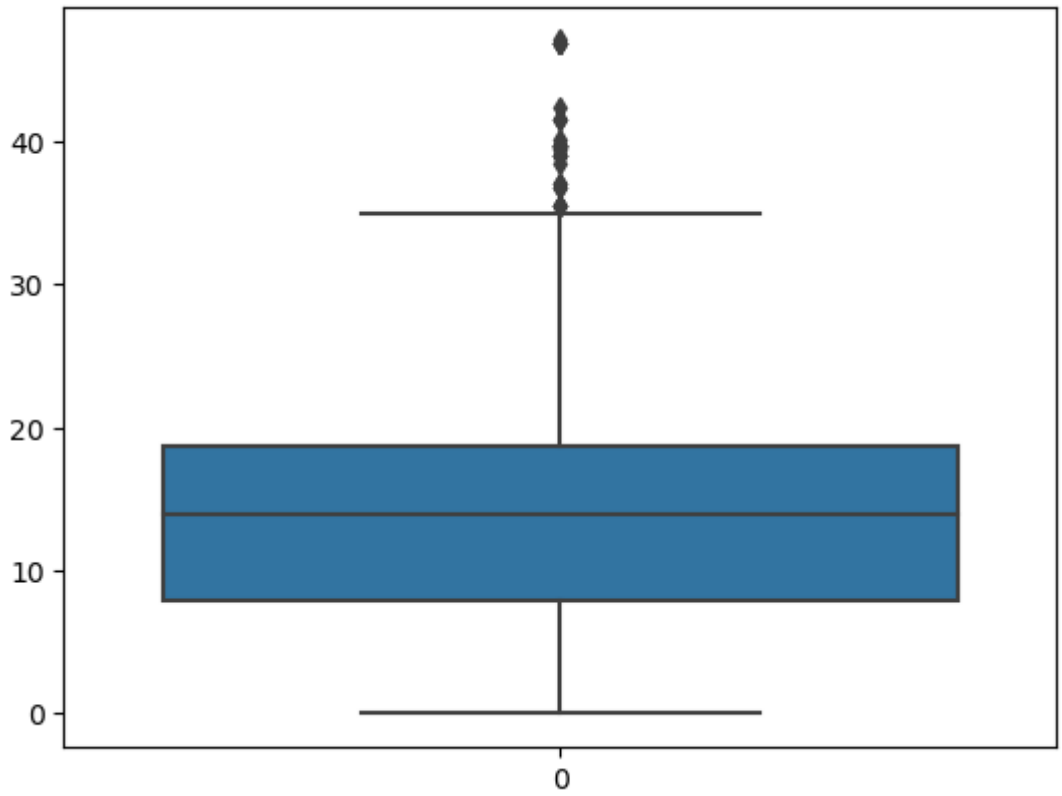
```
df["Fare"]=np.where(df["Fare"]>upperlimit,14,df['Fare'])
```

In [27]:

```
sns.boxplot(df.Fare)
```

Out[27]:

```
<Axes: >
```



In [28]:

```
q1 = df.Fare.quantile(0.25)
q3 = df.Fare.quantile(0.75)
print(q1)
print(q3)
```

```
7.9104
18.75
```

In [29]:

```
q3-q1
upperlimit = q3+1.5*(q3-q1)
upperlimit
lowerlimit = q1-1.5*(q3-q1)
lowerlimit
df.median()
```

```
/var/folders/0g/xqmh0yz92jx_s8ljsv3x08wr0000gn/T/ipykernel_92535/38262
84025.py:6: FutureWarning: The default value of numeric_only in DataFr
ame.median is deprecated. In a future version, it will default to Fals
e. In addition, specifying 'numeric_only=None' is deprecated. Select o
nly valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

Out[29]:

```
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.000000
dtype: float64
```

In [30]:

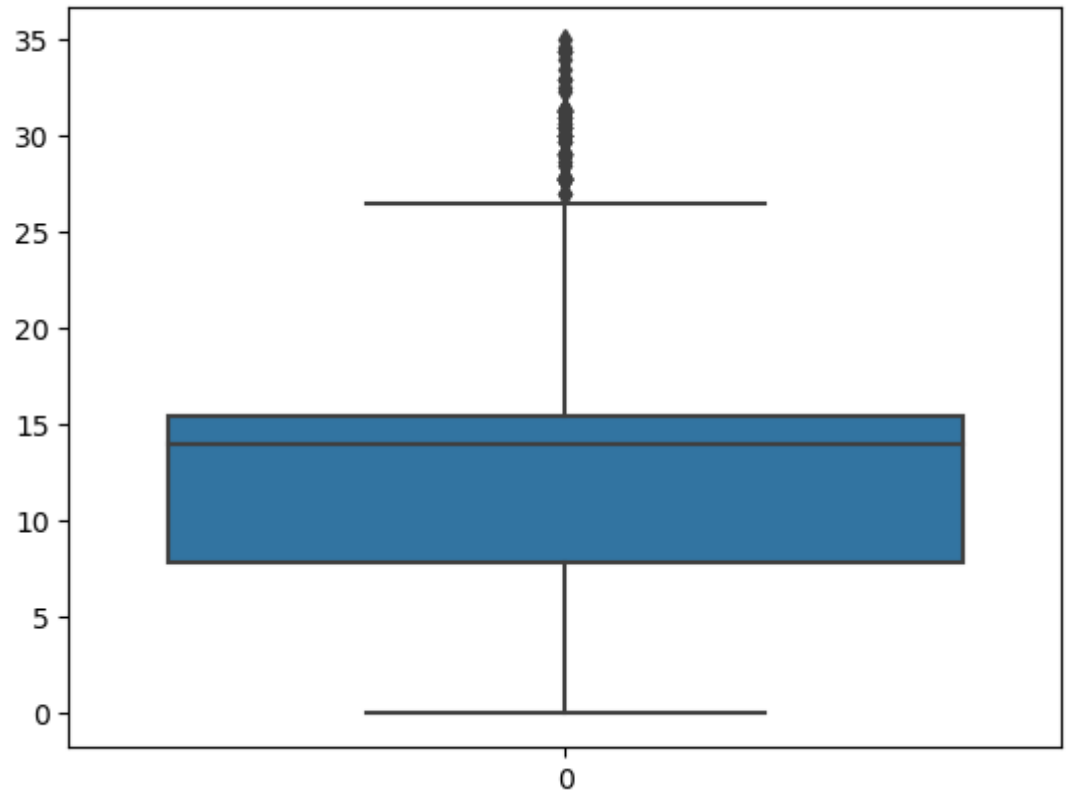
```
df["Fare"]=np.where(df["Fare"]>upperlimit,14,df['Fare'])
```

In [31]:

```
sns.boxplot(df.Fare)
```

Out[31]:

```
<Axes: >
```



In [32]:

```
q1 = df.Fare.quantile(0.25)
q3 = df.Fare.quantile(0.75)
print(q1)
print(q3)
```

```
7.9104
15.5
```

In [33]:

```
q3-q1
upperlimit = q3+1.5*(q3-q1)
upperlimit
lowerlimit = q1-1.5*(q3-q1)
lowerlimit
df.median()
```

```
/var/folders/0g/xqmh0yz92jx_s8ljsv3x08wr0000gn/T/ipykernel_92535/38262
84025.py:6: FutureWarning: The default value of numeric_only in DataFr
ame.median is deprecated. In a future version, it will default to Fals
e. In addition, specifying 'numeric_only=None' is deprecated. Select o
nly valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

Out[33]:

```
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.000000
dtype: float64
```

In [34]:

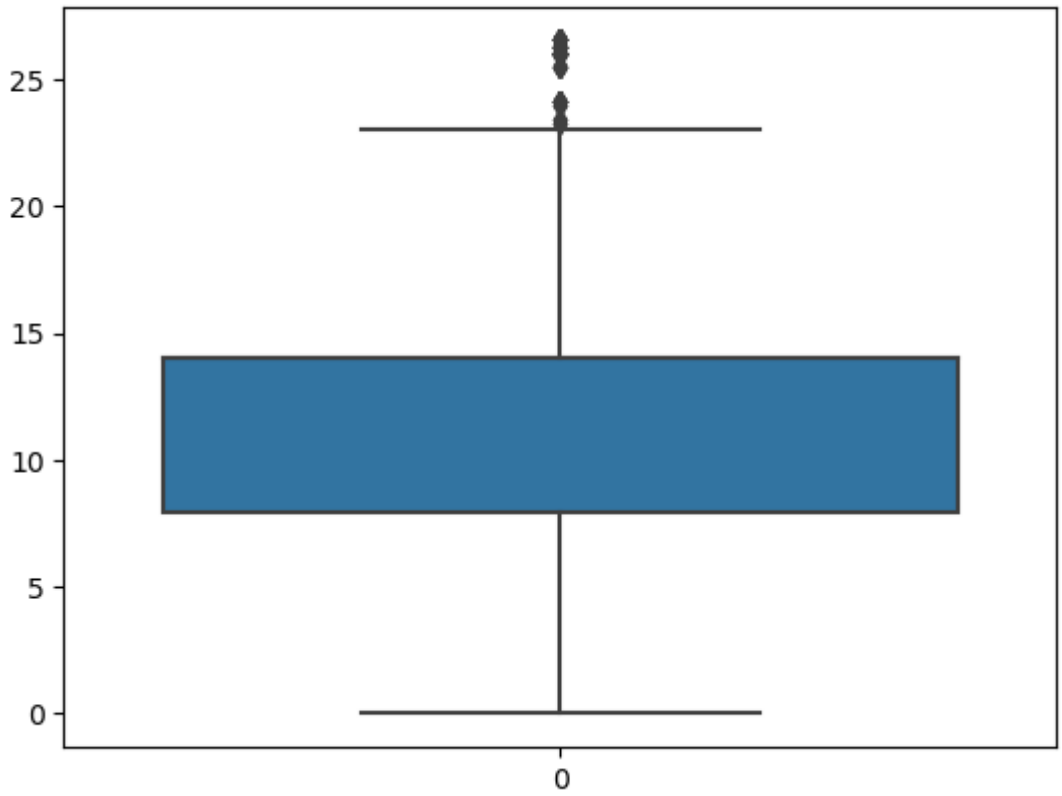
```
df["Fare"]=np.where(df["Fare"]>upperlimit,14,df['Fare'])
```

In [35]:

```
sns.boxplot(df.Fare)
```

Out[35]:

```
<Axes: >
```



In [36]:

```
q1 = df.Fare.quantile(0.25)
q3 = df.Fare.quantile(0.75)
print(q1)
print(q3)
q3-q1
upperlimit = q3+1.5*(q3-q1)
upperlimit
lowerlimit = q1-1.5*(q3-q1)
lowerlimit
df.median()
```

```
7.9104
14.0

/var/folders/0g/xqmh0yz92jx_s8ljsv3x08wr0000gn/T/ipykernel_92535/11779
66432.py:10: FutureWarning: The default value of numeric_only in DataF
rame.median is deprecated. In a future version, it will default to Fal
se. In addition, specifying 'numeric_only=None' is deprecated. Select
only valid columns or specify the value of numeric_only to silence thi
s warning.
  df.median()
```

Out[36]:

```
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.000000
dtype: float64
```

In [37]:

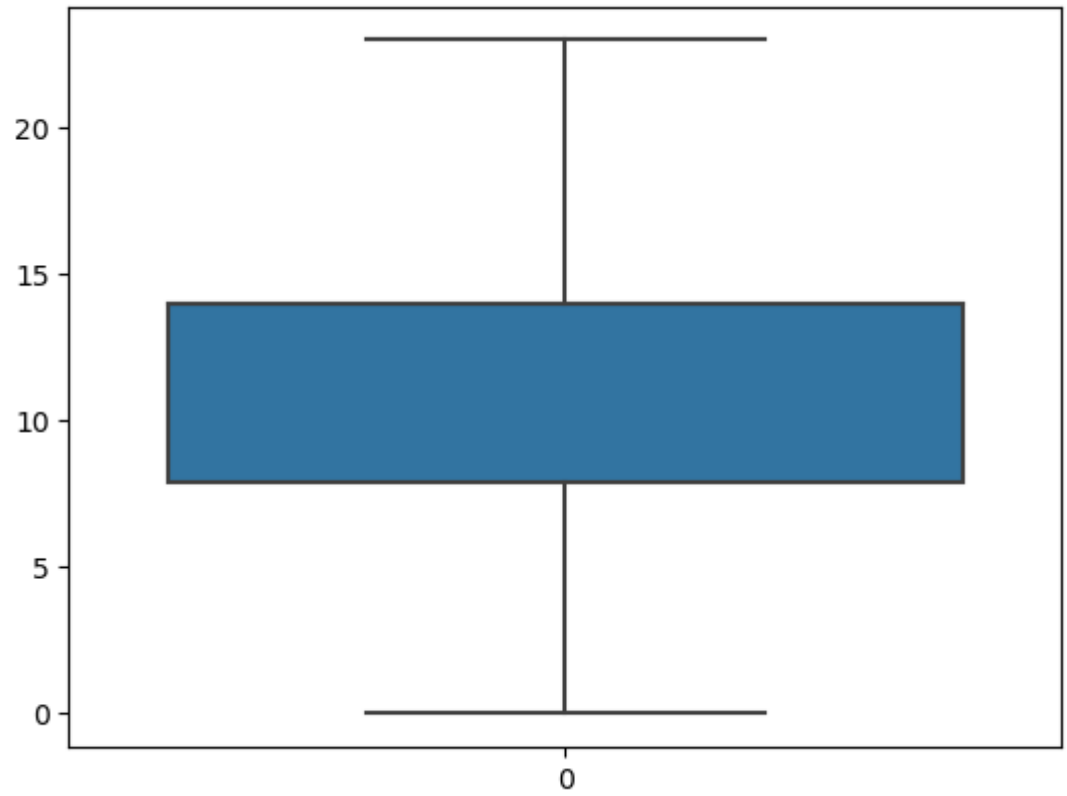
```
df["Fare"]=np.where(df["Fare"]>upperlimit,14,df['Fare'])
```

In [38]:

```
sns.boxplot(df.Fare)
```

Out[38]:

```
<Axes: >
```



In [39]:

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

In [40]:

```
x.head()
```

Out[40]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarl
0	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.250	B96 B98	
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	14.000	C85	
2	3	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925	B96 B98	
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	14.000	C123	
4	5	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.050	B96 B98	

In [41]:

```
y.head()
```

Out[41]:

```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

In [42]:

```
x = df.iloc[:,4:13]
x
```

Out[42]:

	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	male	22.000000	1	0	A/5 21171	7.250	B96 B98	S
1	female	38.000000	1	0	PC 17599	14.000	C85	C
2	female	26.000000	0	0	STON/O2. 3101282	7.925	B96 B98	S
3	female	35.000000	1	0	113803	14.000	C123	S
4	male	35.000000	0	0	373450	8.050	B96 B98	S
...	...	...	...	...	...	...	...	...
886	male	27.000000	0	0	211536	13.000	B96 B98	S
887	female	19.000000	0	0	112053	14.000	B42	S
888	female	29.699118	1	2	W./C. 6607	14.000	B96 B98	S
889	male	26.000000	0	0	111369	14.000	C148	C
890	male	32.000000	0	0	370376	7.750	B96 B98	Q

891 rows × 8 columns

In [43]:

```
print(type(x))
```

```
<class 'pandas.core.frame.DataFrame'>
```

In [44]:

```
print(type(y))
```

```
<class 'pandas.core.series.Series'>
```

In [45]:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

In [46]:

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

Out[46]:

```
0      1
1      0
2      0
3      0
4      1
      ..
886    1
887    0
888    0
889    1
890    1
Name: Sex, Length: 891, dtype: int64
```

In [47]:

```
x["Sex"].value_counts()
```

Out[47]:

```
1    577
0    314
Name: Sex, dtype: int64
```

In [48]:

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

Out[48]:

```
2
```

In [49]:

```
x["Ticket"] = le.fit_transform(x["Ticket"])
x["Ticket"]
```

Out[49]:

```
0      523
1      596
2      669
3       49
4      472
      ...
886    101
887     14
888    675
889      8
890    466
Name: Ticket, Length: 891, dtype: int64
```

In [50]:

```
x["Ticket"].value_counts()
```

Out[50]:

```
333    7
568    7
80     7
249    6
566    6
      ..
513    1
98     1
212    1
606    1
466    1
Name: Ticket, Length: 681, dtype: int64
```

In [51]:

```
x["Ticket"].nunique()
```

Out[51]:

```
681
```

In [52]:

```
x["Cabin"] = le.fit_transform(x["Cabin"])
x["Cabin"].value_counts()
```

Out[52]:

```
47     691
145      4
63       4
62       3
142      3
      ...
124      1
76       1
72       1
125      1
60       1
Name: Cabin, Length: 147, dtype: int64
```

In [53]:

```
x["Cabin"].nunique()
```

Out[53]:

```
147
```

In [54]:

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

Out[54]:

```
2    646
0    168
1     77
Name: Embarked, dtype: int64
```

In [55]:

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

Out[55]:

```
3
```

In [56]:

```
x.head()
```

Out[56]:

	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	22.0	1	0	523	7.250	47	2
1	0	38.0	1	0	596	14.000	81	0
2	0	26.0	0	0	669	7.925	47	2
3	0	35.0	1	0	49	14.000	55	2
4	1	35.0	0	0	472	8.050	47	2

In [57]:

```
from sklearn.preprocessing import MinMaxScaler
ms = MinMaxScaler()
x_scaled = ms.fit_transform(x)
```

In [58]:

```
x_scaled = pd.DataFrame(ms.fit_transform(x),columns =x.columns)
```

In [59]:

```
x_scaled.head()
```

Out[59]:

	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1.0	0.271174	0.125	0.0	0.769118	0.315217	0.321918	1.0
1	0.0	0.472229	0.125	0.0	0.876471	0.608696	0.554795	0.0
2	0.0	0.321438	0.000	0.0	0.983824	0.344565	0.321918	1.0
3	0.0	0.434531	0.125	0.0	0.072059	0.608696	0.376712	1.0
4	1.0	0.434531	0.000	0.0	0.694118	0.350000	0.321918	1.0

In [60]:

```
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)
```

In [61]:

```
print(x_train.shape,y_train.shape,x_test.shape,y_test.shape)
```

```
(623, 8) (623,) (268, 8) (268,)
```

In [ ]:

```

```