Import Necessary Libraries

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

Import DataSet

In [2]:
```
data=pd.read_csv("Titanic-Dataset.csv")
```

In [3]:
```
data.head()
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
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

In [4]:
```
data.tail()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	Na
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B4
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	Na
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C14
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	Na

In [5]:
```
data.columns
```

Out[5]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [6]:

```
data.info
```

Out[6]:

```
<bound method DataFrame.info of      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]>
```

In [7]:

```
data.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
```

In [8]:

```
data.describe()
```

Out[8]:

	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

Handling Null Values

In [9]:

```
data.isnull().any()
```

Out[9]:

```
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 [10]:

```
data.isnull().sum()
```

Out[10]:

```
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
```

Filling the null values in Age column with mean

In [11]:

```
mean_age=data["Age"].mean()
```

In [12]:

```
data["Age"].fillna(mean_age,inplace=True)
```

In [13]:
```
data.tail()
```

Out[13]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.00
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.00
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.45
889	890	1	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.00
890	891	0	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.75

In [14]:
```
data["Age"].isnull().sum()
```

Out[14]:
```
0
```

Filling the null values in Cabin with mode

In [15]:
```
mode_cabin=data["Cabin"].mode()
```

In [16]:
```
mode_cabin
```

Out[16]:
```
0        B96 B98
1    C23 C25 C27
2             G6
Name: Cabin, dtype: object
```

In [17]:
```
data["Cabin"].fillna(mode_cabin[2],inplace=True)
```

In [18]:
```
data["Cabin"].isnull().sum()
```

Out[18]:
```
0
```

In [19]:
```
data["Cabin"]
```

Out[19]:
```
0        G6
1       C85
2        G6
3      C123
4        G6
       ... 
886      G6
887     B42
888      G6
889    C148
890      G6
Name: Cabin, Length: 891, dtype: object
```

Filling the Null values in Embarked with mode

In [20]:
```
mode_emb=data["Embarked"].mode()
mode_emb
```

Out[20]:
```
0    S
Name: Embarked, dtype: object
```

In [21]:
```
data["Embarked"].fillna(mode_emb[0],inplace=True)
```

In [22]:
```
data["Embarked"].isnull().sum()
```

Out[22]:
```
0
```

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

Out[23]:
```
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
```

Data Visualisation

In [24]:

```
corr=data.corr()
```

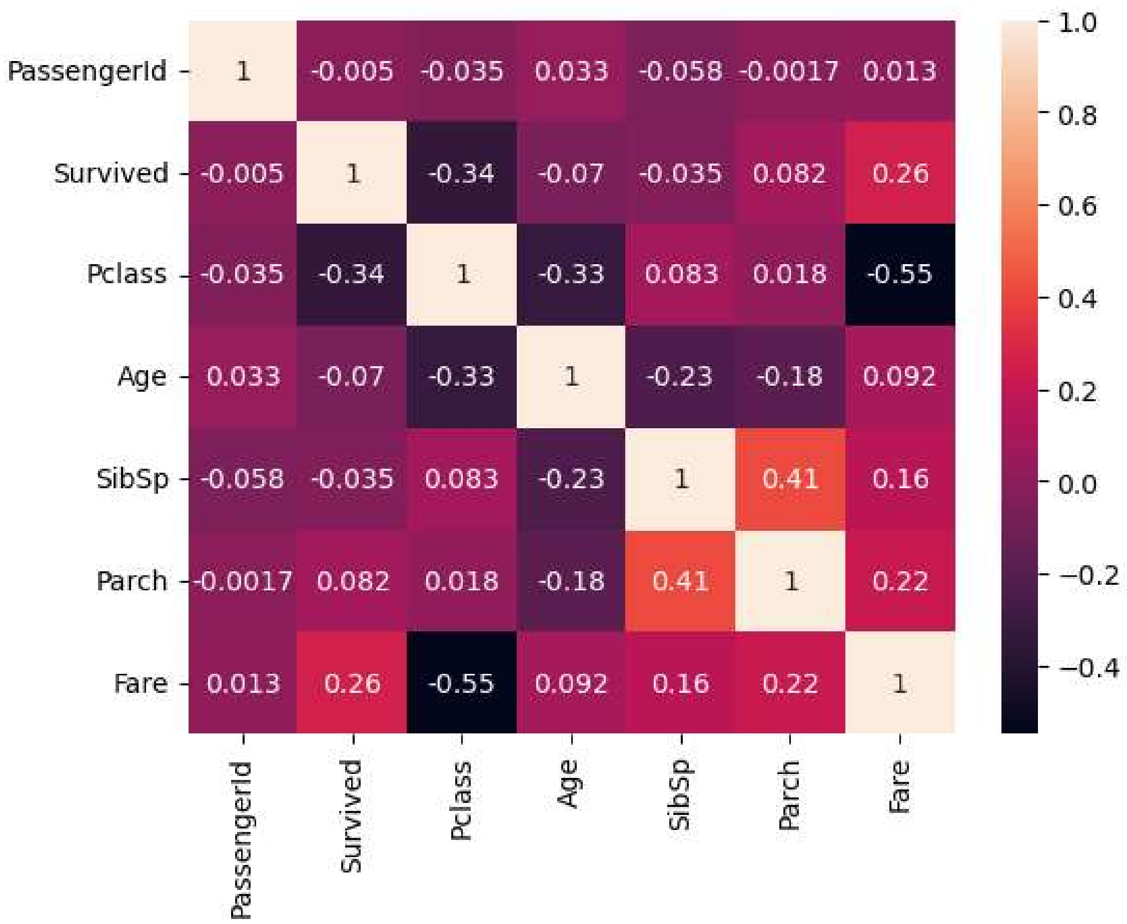
```
C:\Users\dines\AppData\Local\Temp\ipykernel_8724\2057684327.py:1: FutureWar
ning: The default value of numeric_only in DataFrame.corr is deprecated. In
a future version, it will default to False. Select only valid columns or sp
ecify the value of numeric_only to silence this warning.
  corr=data.corr()
```

In [25]:

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

Out[25]:

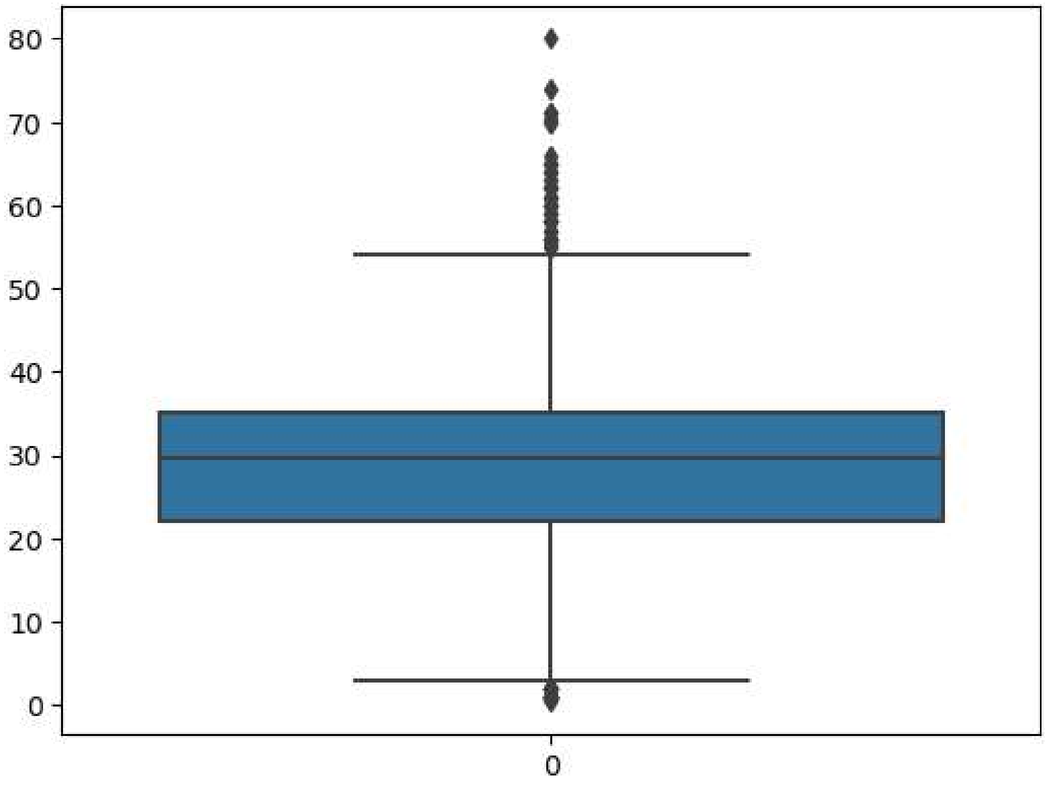
```
<Axes: >
```



Handling the outliers

```
In [26]:  sns.boxplot(data["Age"])
```

```
Out[26]: <Axes: >
```



Inference: There are outliers in Age column

```
In [27]:  age_q1=data.Age.quantile(0.25)
          age_q2=data.Age.quantile(0.5)
          age_q3=data.Age.quantile(0.75)
          print(age_q1)
          print(age_q2)
          print(age_q3)
```

```
22.0
29.69911764705882
35.0
```

```
In [28]:  IQR_AGE=age_q3-age_q1
          IQR_AGE
```

```
Out[28]: 13.0
```

In [29]:
```
upperlimit_age=age_q3+1.5*IQR_AGE
upperlimit_age
```

Out[29]:
```
54.5
```

In [30]:
```
lowerlimit_age=age_q1-1.5*IQR_AGE
lowerlimit_age
```

Out[30]:
```
2.5
```

In [31]:
```
median_age=data["Age"].median()
median_age
```

Out[31]:
```
29.69911764705882
```

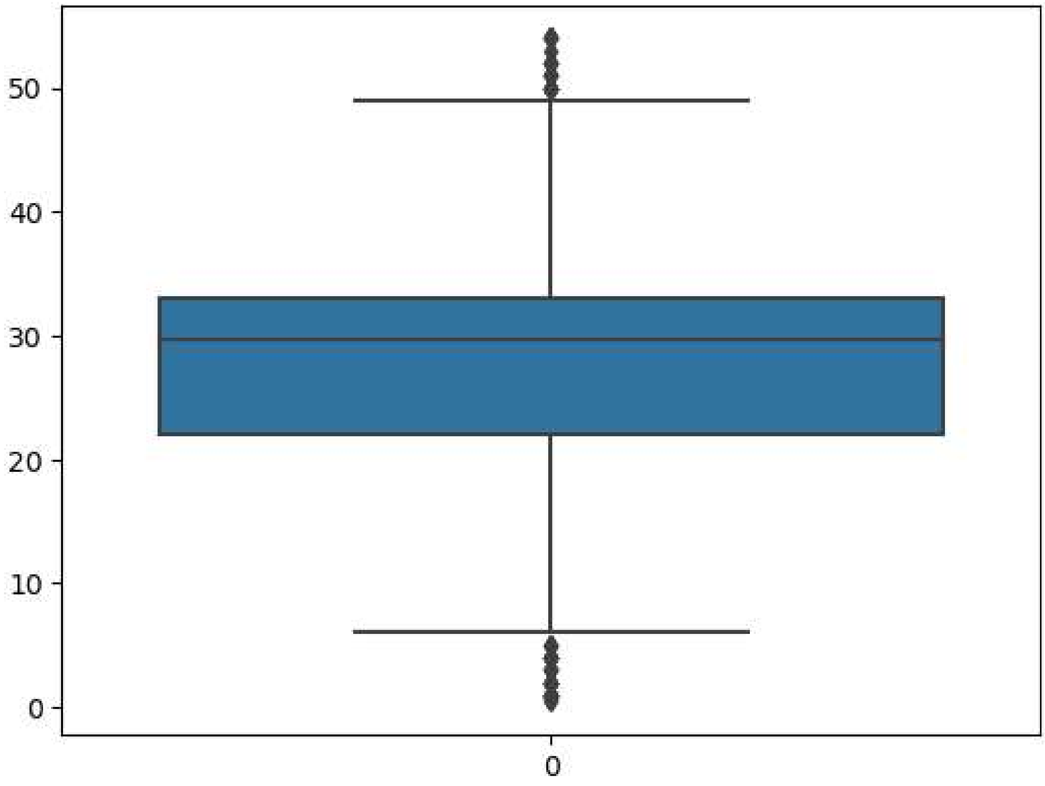
In [32]:
```
data["Age"]=np.where(data["Age"]>upperlimit_age,median_age,data["Age"])
```

In [33]:
```
(data["Age"]>upperlimit_age).sum()
```

Out[33]:
```
0
```

In [34]:
```
sns.boxplot(data["Age"])
```

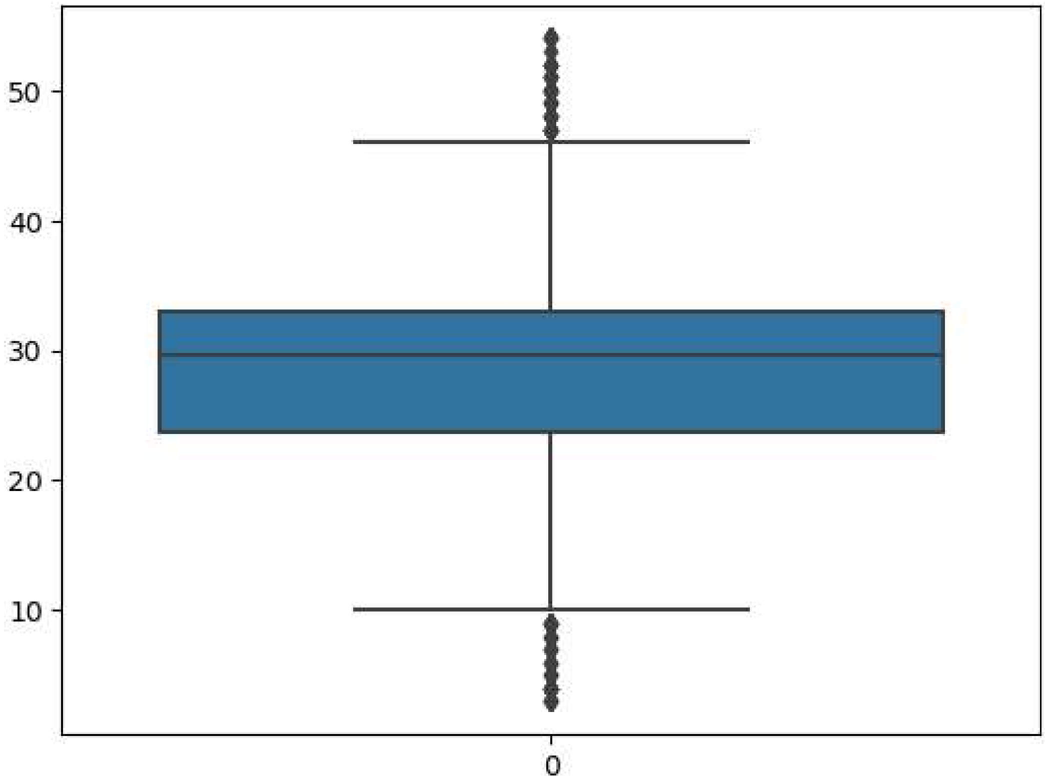
Out[34]:
```
<Axes: >
```



In [35]:
```
data["Age"]=np.where(data["Age"]<lowerlimit_age,median_age,data["Age"])
```

In [36]:
```
sns.boxplot(data["Age"])
```

Out[36]:
```
<Axes: >
```

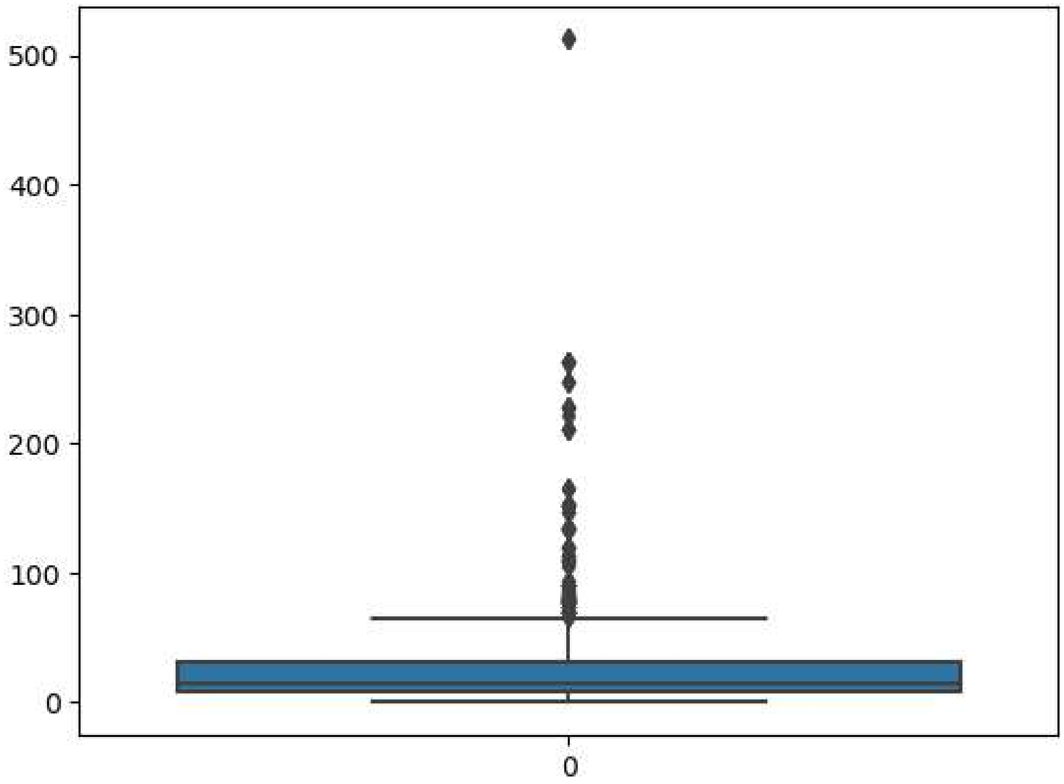


In [37]:

```
sns.boxplot(data["Fare"])
```

Out[37]:

```
<Axes: >
```



In [38]:

```
fare_q1=data.Fare.quantile(0.25)
fare_q2=data.Fare.quantile(0.5)
fare_q3=data.Fare.quantile(0.75)
print(fare_q1)
print(fare_q2)
print(fare_q3)
```

```
7.9104
14.4542
31.0
```

In [39]:

```
IQR_FARE=fare_q3-fare_q1
IQR_FARE
```

Out[39]:

```
23.0896
```

In [40]:

```
upperlimit_fare=fare_q3+1.5*IQR_FARE
upperlimit_fare
```

Out[40]:

```
65.6344
```

In [41]:

```
lowerlimit_fare=fare_q1-1.5*IQR_FARE
lowerlimit_fare
```

Out[41]:

```
-26.724
```

In [42]:

```
median_fare=data["Fare"].median()
median_fare
```

Out[42]:

```
14.4542
```

In [43]:

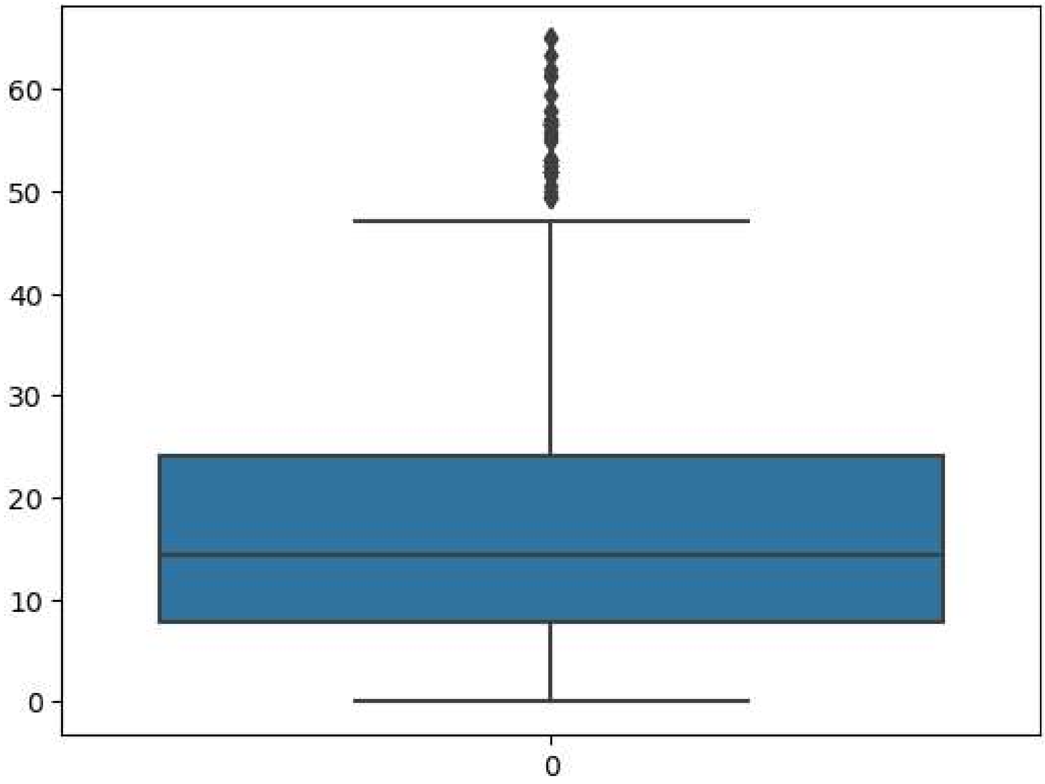
```
data["Fare"]=np.where(data["Fare"]>upperlimit_fare,median_fare,data["Fare"])
```

In [44]:

```
sns.boxplot(data["Fare"])
```

Out[44]:

```
<Axes: >
```



In [45]:

```
(data["Fare"]>upperlimit_fare).sum()
```

Out[45]:

```
0
```

Dropping Variables

In [46]:

```
data.head()
```

Out[46]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
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	14.4542
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

In [48]:

```
data.drop("Name",axis=1,inplace=True)
```

In [49]:

```
data.head()
```

Out[49]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emb
0	1	0	3	male	22.0	1	0	A/5 21171	7.2500	G6	
1	2	1	1	female	38.0	1	0	PC 17599	14.4542	C85	
2	3	1	3	female	26.0	0	0	STON/O2. 3101282	7.9250	G6	
3	4	1	1	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	male	35.0	0	0	373450	8.0500	G6	

In [50]:

```
data.drop("Ticket",axis=1,inplace=True)
```

In [51]:
```
data.head()
```

Out[51]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	1	0	3	male	22.0	1	0	7.2500	G6	S
1	2	1	1	female	38.0	1	0	14.4542	C85	C
2	3	1	3	female	26.0	0	0	7.9250	G6	S
3	4	1	1	female	35.0	1	0	53.1000	C123	S
4	5	0	3	male	35.0	0	0	8.0500	G6	S

In [52]:
```
data.drop("PassengerId",axis=1,inplace=True)
```

In [53]:
```
data.head()
```

Out[53]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	3	male	22.0	1	0	7.2500	G6	S
1	1	1	female	38.0	1	0	14.4542	C85	C
2	1	3	female	26.0	0	0	7.9250	G6	S
3	1	1	female	35.0	1	0	53.1000	C123	S
4	0	3	male	35.0	0	0	8.0500	G6	S

In [54]:
```
data.drop("Cabin",axis=1,inplace=True)
```

In [55]:
```
data.head()
```

Out[55]:

	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	14.4542	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

Splitting Dependent and Independent variables

In [56]:
```
#independent varible
x=data.drop("Survived",axis=1)
```

In [57]:

```
x.head()
```

Out[57]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	male	22.0	1	0	7.2500	S
1	1	female	38.0	1	0	14.4542	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

In [58]:

```
#dependent variable
y=data["Survived"]
```

In [59]:

```
y.head()
```

Out[59]:

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

Encoding

In [60]:

```
from sklearn.preprocessing import LabelEncoder
```

In [61]:

```
le=LabelEncoder()
```

In [62]:

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

In [63]:

```
x["Sex"]
```

Out[63]:

```
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: int32
```

In [64]:
```
x.head()
```

Out[64]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1	22.0	1	0	7.2500	S
1	1	0	38.0	1	0	14.4542	C
2	3	0	26.0	0	0	7.9250	S
3	1	0	35.0	1	0	53.1000	S
4	3	1	35.0	0	0	8.0500	S

In [65]:
```
x["Embarked"]=le.fit_transform(data["Embarked"])
```

In [66]:
```
x.head()
```

Out[66]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1	22.0	1	0	7.2500	2
1	1	0	38.0	1	0	14.4542	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

In [67]:
```
x["Pclass"].unique()
```

Out[67]:
```
array([3, 1, 2], dtype=int64)
```

In [68]:
```
x["Pclass"].nunique()
```

Out[68]:
```
3
```

In [69]:
```
x["Sex"].unique()
```

Out[69]:
```
array([1, 0])
```

In [70]:
```
x["Embarked"].unique()
```

Out[70]:
```
array([2, 0, 1])
```

Splitting Data into Training data and Testing data

In [71]:
```
from sklearn.model_selection import train_test_split
```

```
In [72]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state
```

```
In [73]: x_train.shape,x_test.shape,y_train.shape,y_test.shape

Out[73]: ((712, 7), (179, 7), (712,), (179,))
```

Feature Scaling

```
In [74]: from sklearn.preprocessing import StandardScaler
```

```
In [75]: sc=StandardScaler()
```

```
In [76]: x_train=sc.fit_transform(x_train)
```

```
In [77]: x_train

Out[77]: array([[ 0.81925059, -1.37207547,  0.07840244, ...,  1.95926403,
                 -0.17726299, -1.98156574],
                [-0.38096838,  0.72882288,  0.21232456, ..., -0.47741019,
                 -0.54667438,  0.5790056 ],
                [-0.38096838,  0.72882288,  0.21232456, ...,  0.74092692,
                  1.51640316, -1.98156574],
                ...,
                [ 0.81925059,  0.72882288,  0.07840244, ..., -0.47741019,
                 -0.76203333, -0.70128007],
                [ 0.81925059, -1.37207547,  0.72706025, ..., -0.47741019,
                 -0.00958083,  0.5790056 ],
                [-0.38096838,  0.72882288,  0.07840244, ...,  0.74092692,
                  1.67175552,  0.5790056 ]])
```

```
In [78]: x_test=sc.fit_transform(x_test)
```

```
In [79]: x_test

Out[79]: array([[ 0.86022947,  0.77344314,  0.0739225 , ..., -0.46006628,
                 -0.19571051, -1.80134224],
                [ 0.86022947,  0.77344314,  0.0739225 , ..., -0.46006628,
                 -0.76604362,  0.61394061],
                [ 0.86022947,  0.77344314, -2.17117908, ...,  0.88996427,
                  1.01513799, -0.59370081],
                ...,
                [-1.50871015, -1.29291987,  0.20258887, ..., -0.46006628,
                 -0.19604899, -1.80134224],
                [ 0.86022947,  0.77344314, -0.58866712, ..., -0.46006628,
                 -0.74092958,  0.61394061],
                [ 0.86022947,  0.77344314, -0.98429511, ..., -0.46006628,
                 -0.72476479,  0.61394061]])
```

In [ ]: