assignment3

September 18, 2023

#1.Data Collection – From kaggle website we are downloading the dataset.

1 #2.Data Preprocessing

#Import the Libraries.

[77]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

#Importing the dataset.

[78]:
```
df = pd.read_csv("/content/Titanic-Dataset.csv")
df
```

[78]:
```
     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]
```

[79]:
```
df.head()
```

[79]:
```
   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
```

[80]:
```
df.describe()
```

[80]:
```
       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
```

[81]:
```
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
```

[82]:
```
df.shape
```

[82]:
```
(891, 12)
```

[83]:
```
df.corr()
```

```
<ipython-input-83-2f6f6606aa2c>:1: FutureWarning: 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 specify the value of numeric_only
```

```
to silence this warning.
  df.corr()
```

[83]:
```
             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
```

[84]:
```
df.corr().Fare.sort_values(ascending=False)
```

```
<ipython-input-84-f51f352aac84>:1: FutureWarning: 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 specify the value of numeric_only
to silence this warning.
  df.corr().Fare.sort_values(ascending=False)
```

[84]:
```
Fare           1.000000
Survived       0.257307
Parch          0.216225
SibSp          0.159651
Age            0.096067
PassengerId    0.012658
Pclass        -0.549500
Name: Fare, dtype: float64
```

#Checking for Null Values.

[85]:
```
df.isnull().any()
```

[85]:
```
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
```

[86]:
```
df.isnull().sum()
```

[86]:
```
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
```

[87]:
```
df1 = df.fillna(value=df['Age'].mean())
df1.isnull().sum()
```

[87]:
```
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
```

[88]:
```
df1
```

[88]:
```
     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  \
0                                Braund, Mr. Owen Harris    male  22.000000
1    Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.000000
2                                Heikkinen, Miss. Laina  female  26.000000
3          Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.000000
4                              Allen, Mr. William Henry    male  35.000000
..                                                  …     …        …
886                              Montvila, Rev. Juozas    male  27.000000
887                       Graham, Miss. Margaret Edith  female  19.000000
888            Johnston, Miss. Catherine Helen "Carrie"  female  29.699118
889                              Behr, Mr. Karl Howell    male  26.000000
890                                Dooley, Mr. Patrick    male  32.000000

     SibSp  Parch            Ticket     Fare      Cabin Embarked
0        1      0         A/5 21171   7.2500  29.699118        S
1        1      0          PC 17599  71.2833        C85        C
2        0      0  STON/O2. 3101282   7.9250  29.699118        S
3        1      0            113803  53.1000       C123        S
4        0      0            373450   8.0500  29.699118        S
..     …    …                …      …        …      …
886      0      0            211536  13.0000  29.699118        S
887      0      0            112053  30.0000        B42        S
888      1      2        W./C. 6607  23.4500  29.699118        S
889      0      0            111369  30.0000       C148        C
890      0      0            370376   7.7500  29.699118        Q

[891 rows x 12 columns]
```

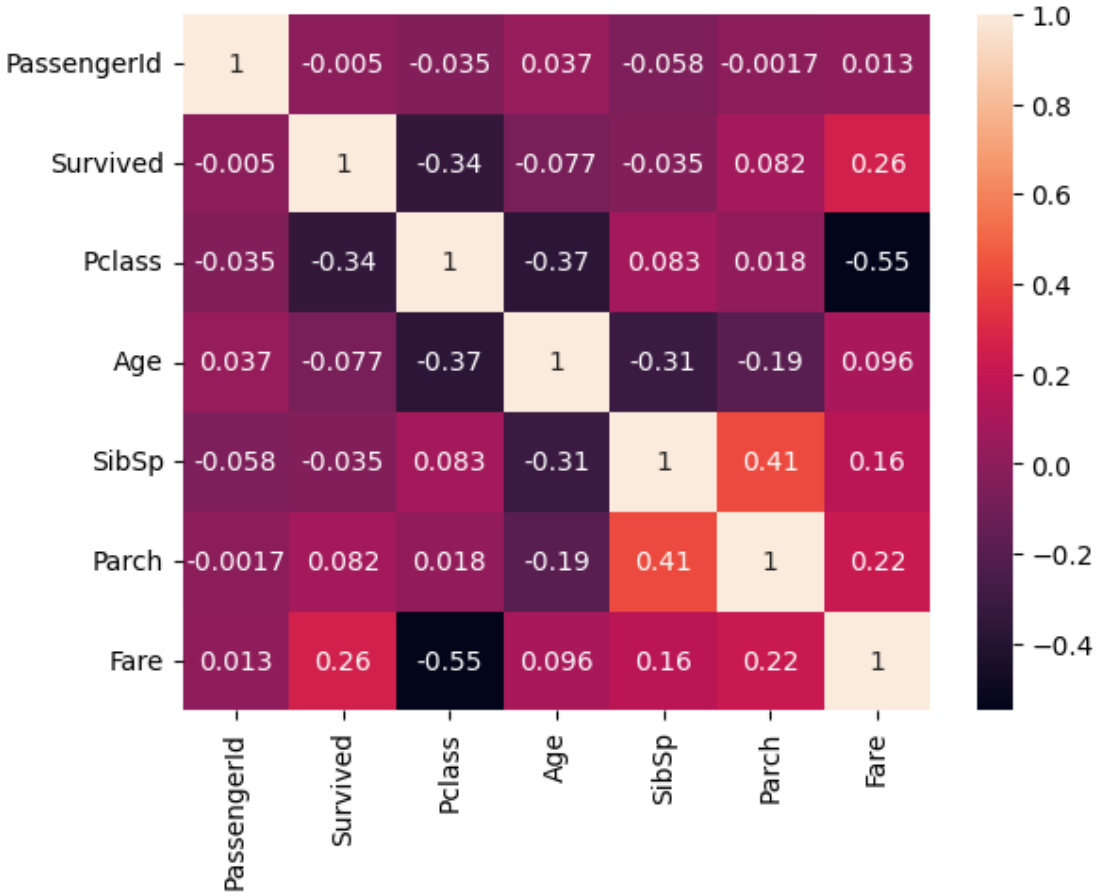
[89]:
```
df1.isnull().sum().sum()
```

[89]:
```
0
```

#Data Visualization.

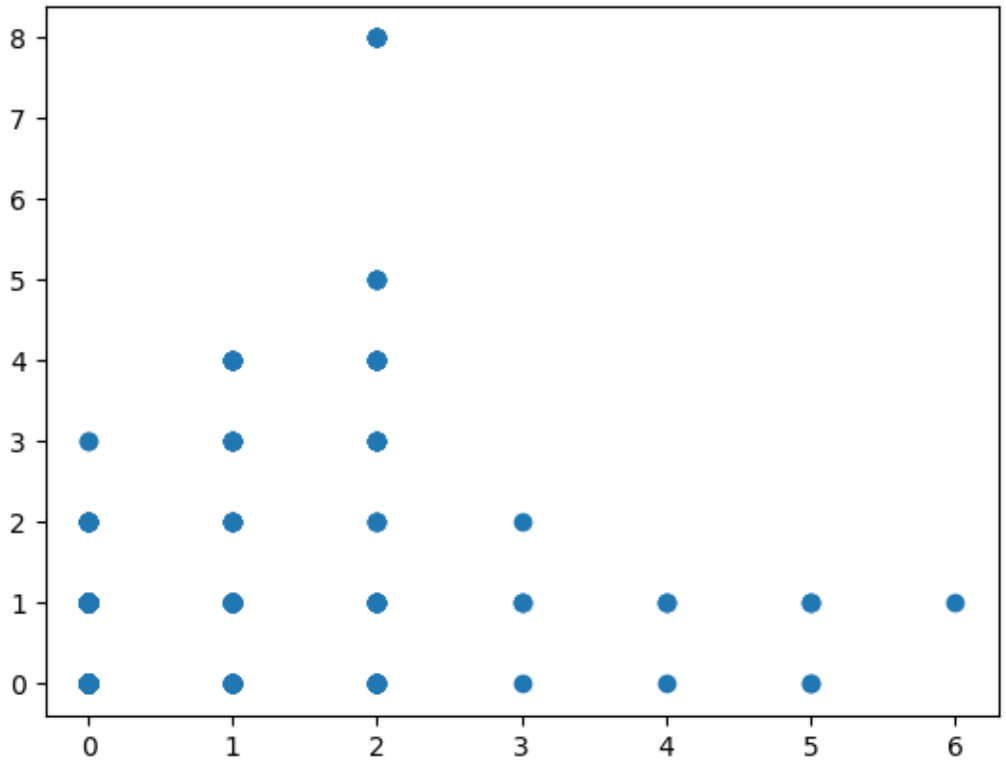
[90]:
```
sns.heatmap(df.corr(numeric_only = True),annot = True)
```

[90]:
```
<Axes: >
```



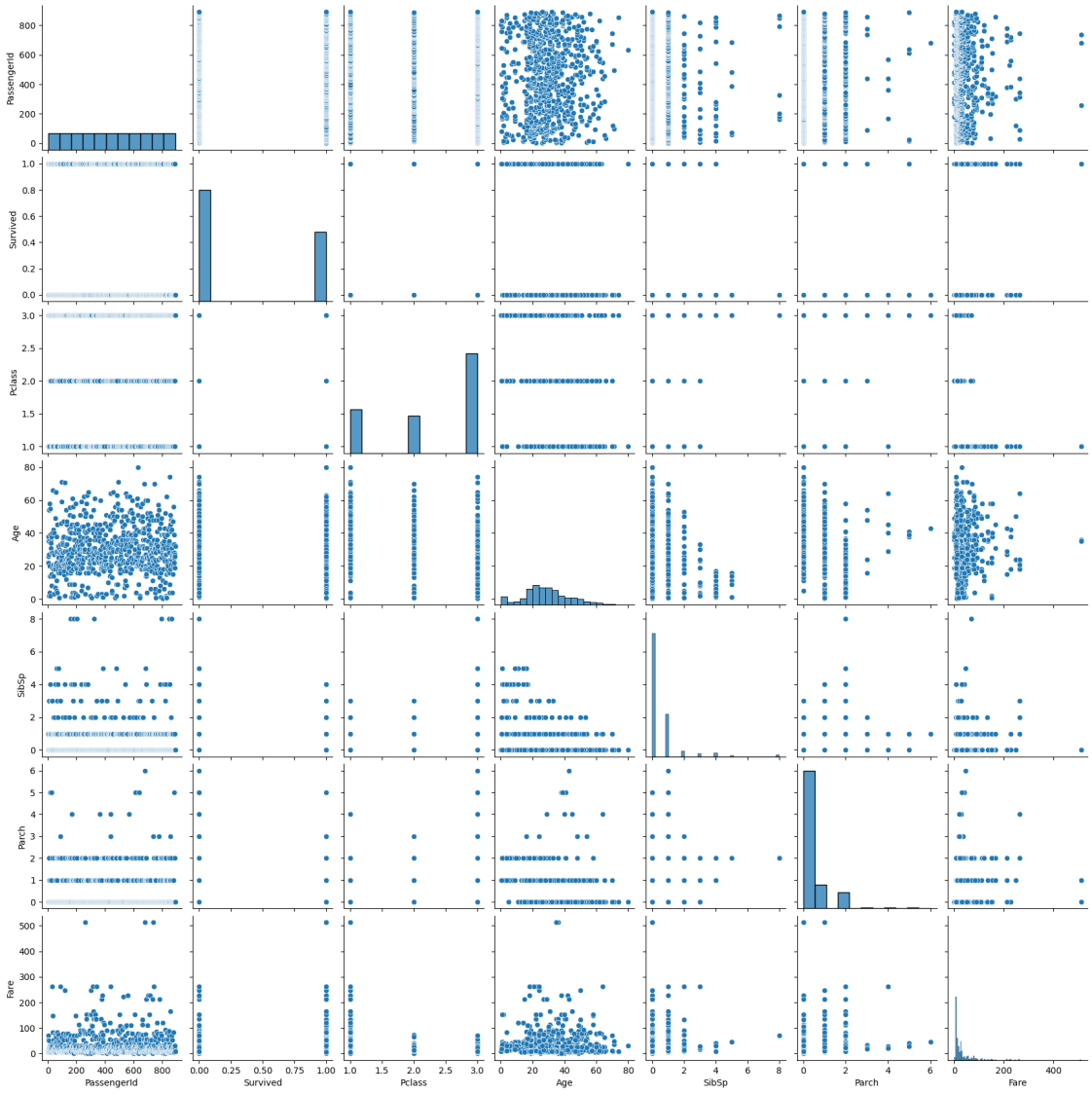
[91]:
```
plt.scatter(df["Parch"],df["SibSp"])
```

[91]:
```
<matplotlib.collections.PathCollection at 0x7ba255e0bdf0>
```



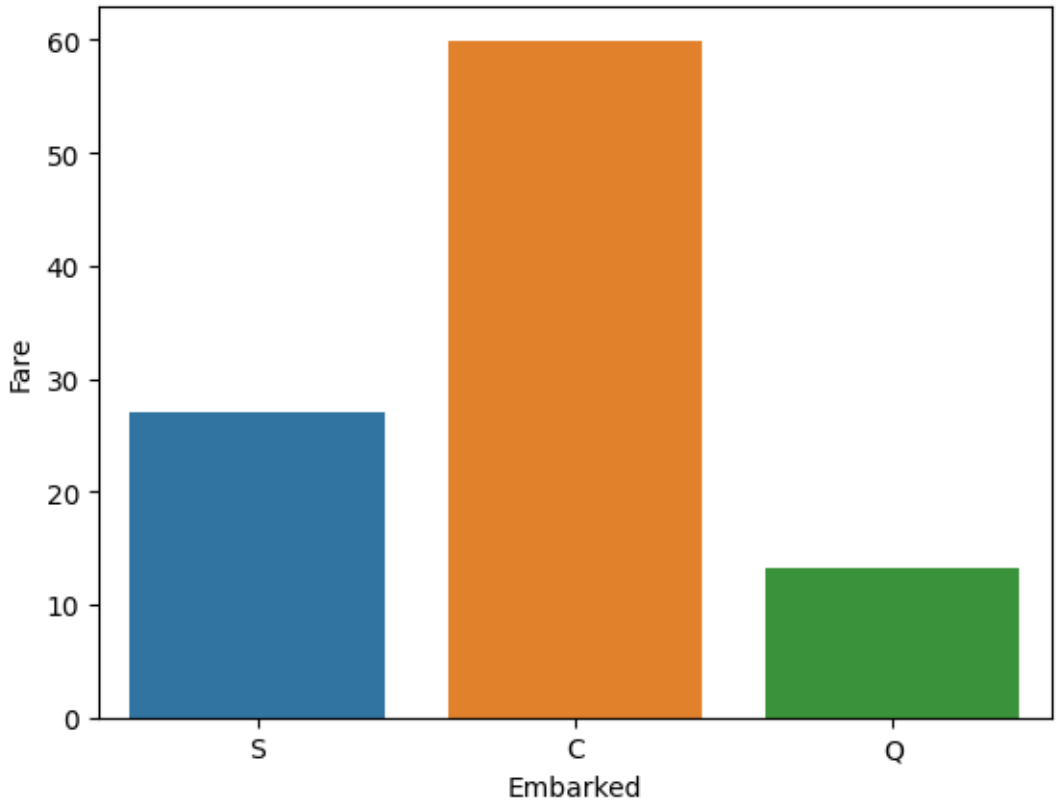
[92]:
```
sns.pairplot(df)
```

[92]: <seaborn.axisgrid.PairGrid at 0x7ba255eedcc0>



[93]:
```
sns.barplot(x=df["Embarked"],y=df["Fare"],errorbar=('ci',0))
```

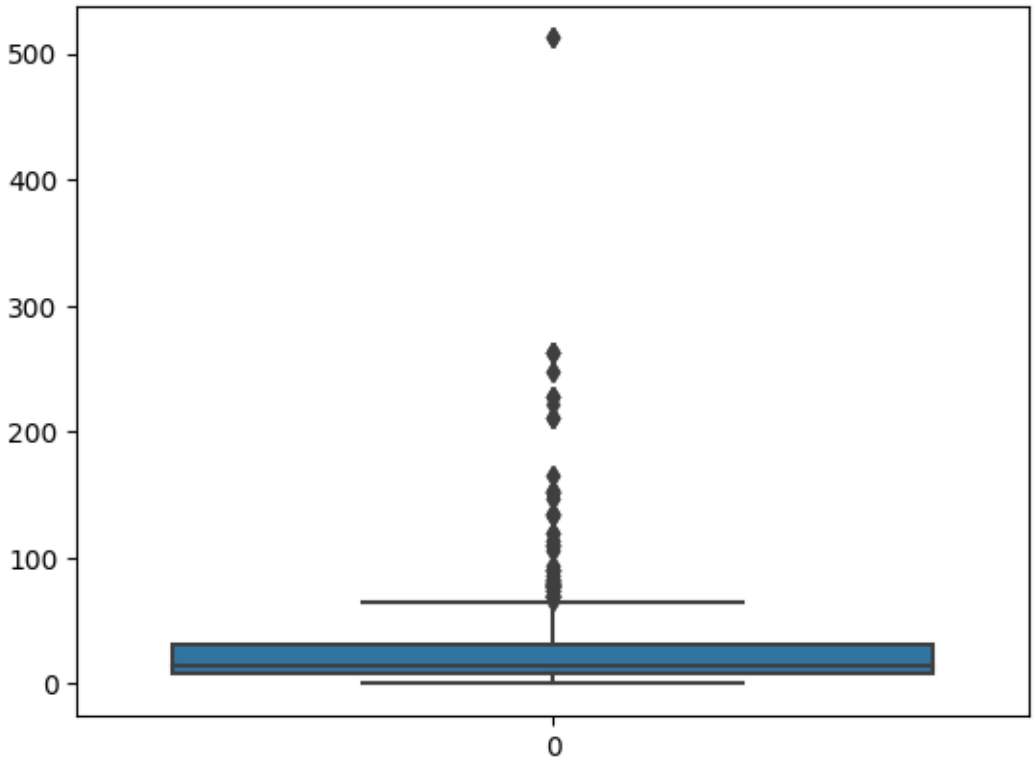
[93]: <Axes: xlabel='Embarked', ylabel='Fare'>



#Outlier Detection

[94]:
```
sns.boxplot(df["Fare"])
```

[94]: <Axes: >



[95]:
```
q1 = df.Fare.quantile(0.25)
q3 = df.Fare.quantile(0.75)
```

[96]:
```
IQR = q3-q1
IQR
```

[96]: 23.0896

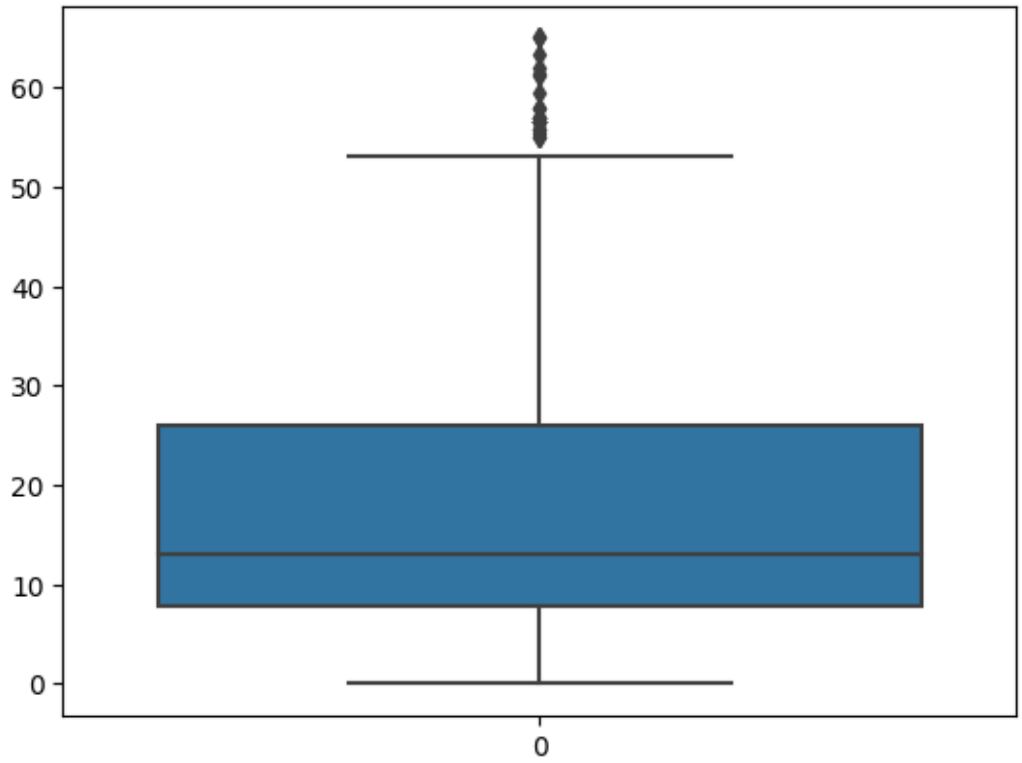
[97]:
```
upper_limit =q3+1.5*IQR
upper_limit
```

[97]: 65.6344

[98]:
```
df = df[df.Fare<upper_limit]
```

[99]:
```
sns.boxplot(df["Fare"])
```

[99]: <Axes: >



[100]:
```
q1 = df.Fare.quantile(0.25)
q3 = df.Fare.quantile(0.75)
```

[101]:
```
IQR = q3-q1
IQR
```

[101]: 18.1042

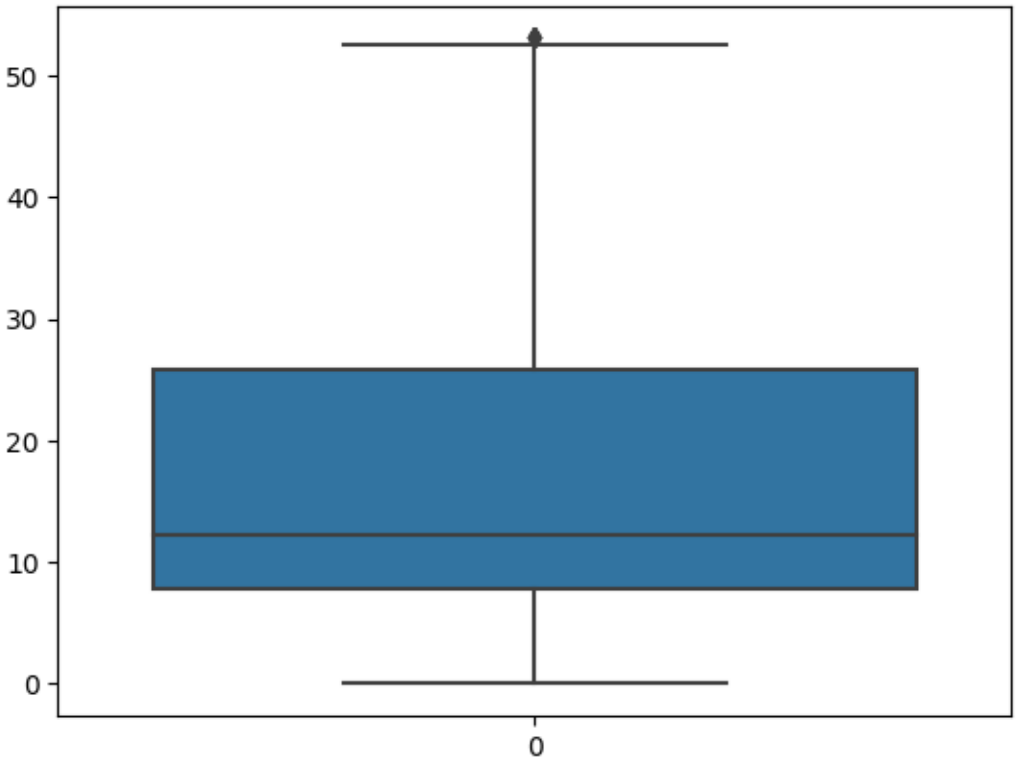
[102]:
```
upper_limit =q3+1.5*IQR
upper_limit
```

[102]: 53.1563

[103]:
```
df = df[df.Fare<upper_limit]
```

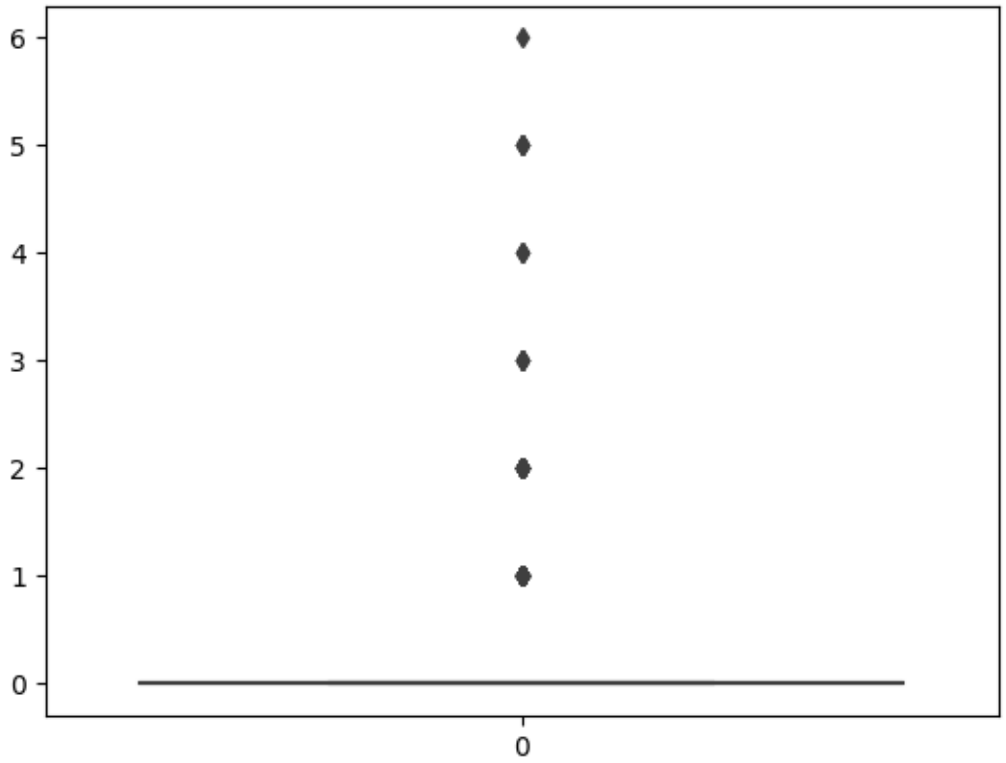
[104]:
```
sns.boxplot(df["Fare"])
```

[104]: <Axes: >



[107]:
```
sns.boxplot(df["Parch"])
```

[107]: <Axes: >



[108]:
```
from scipy import stats
```

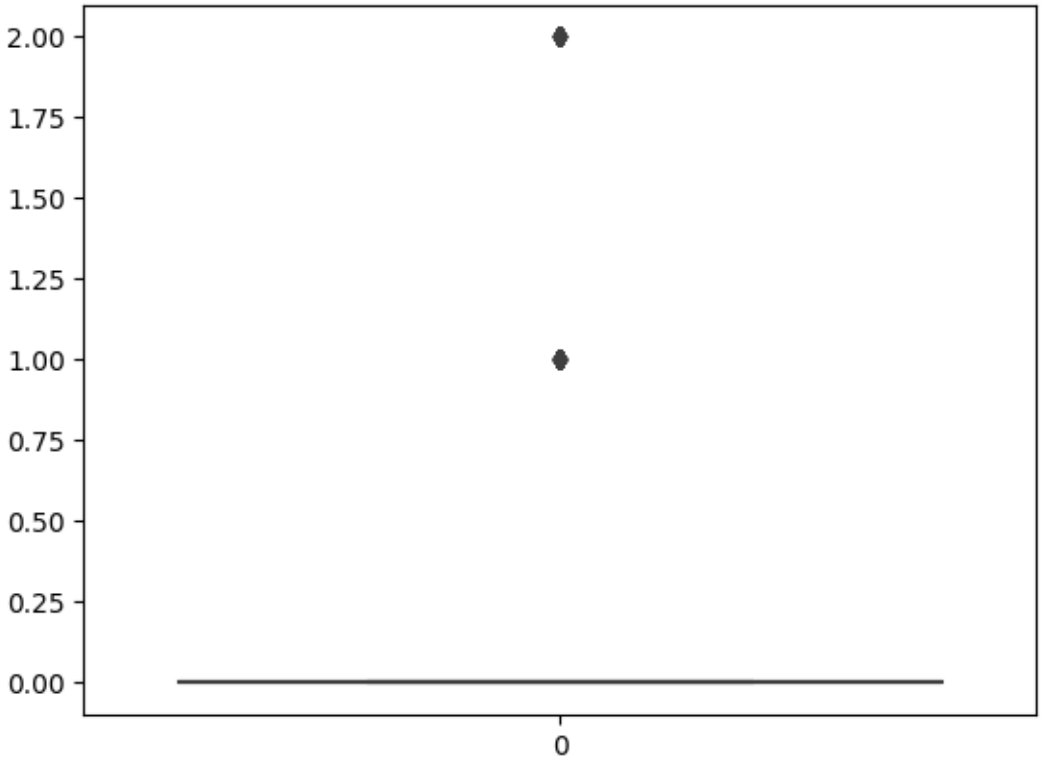
[109]:
```
parch_zscore = stats.zscore(df.Parch)
parch_zscore
```

[109]:
```
0     -0.427196
2     -0.427196
3     -0.427196
4     -0.427196
5     -0.427196
         …
886   -0.427196
887   -0.427196
888    2.105587
889   -0.427196
890   -0.427196
Name: Parch, Length: 750, dtype: float64
```

[110]:
```
df_z = df[np.abs(parch_zscore)<=3]
```

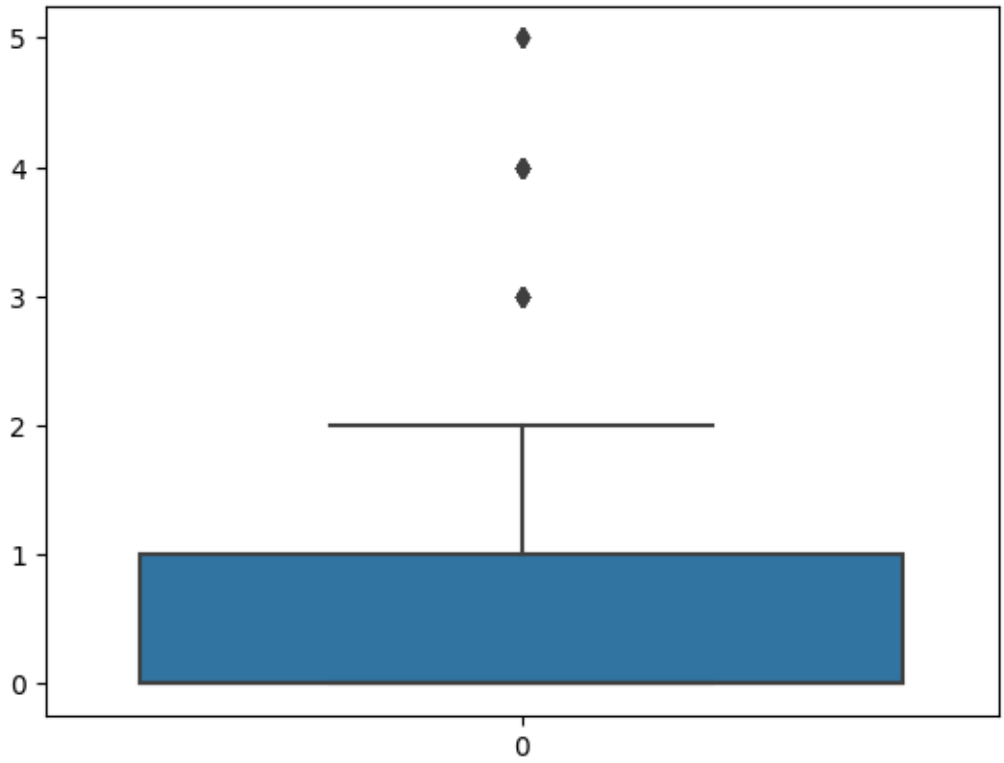
[111]:
```
sns.boxplot(df_z.Parch)
```

[111]: <Axes: >



[105]:
```
sns.boxplot(df["SibSp"])
```

[105]: <Axes: >



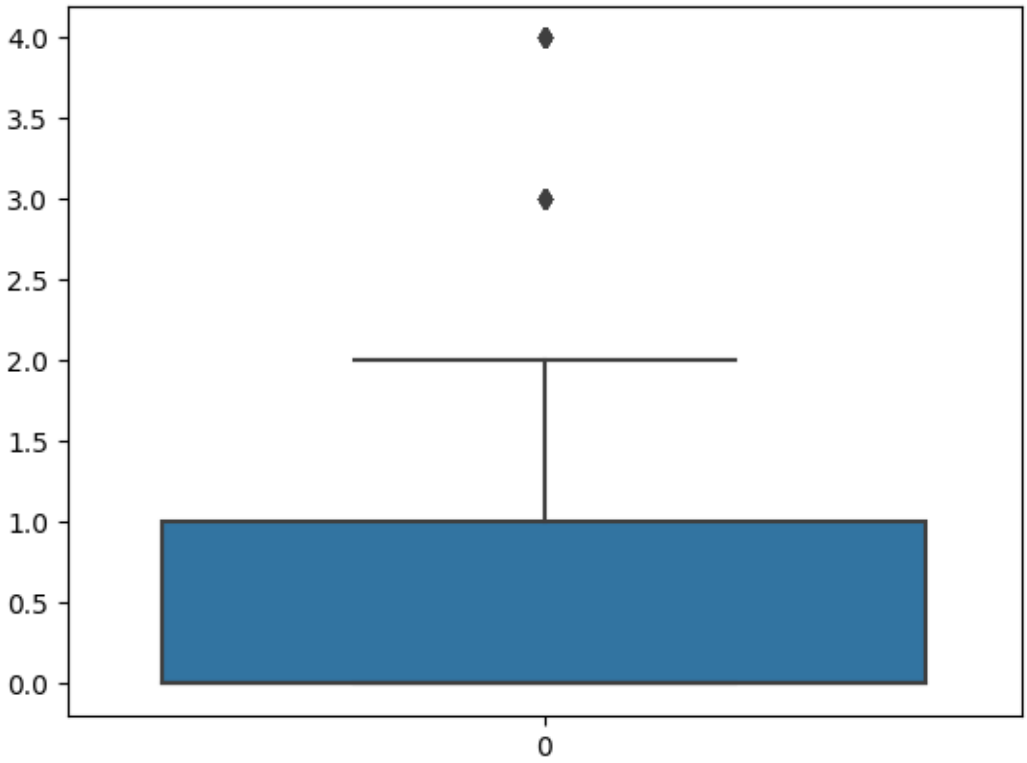
[113]:
```
p99 = df.SibSp.quantile(0.99)
p99
```

[113]: 4.0

[114]:
```
df = df[df.SibSp<=p99]
```

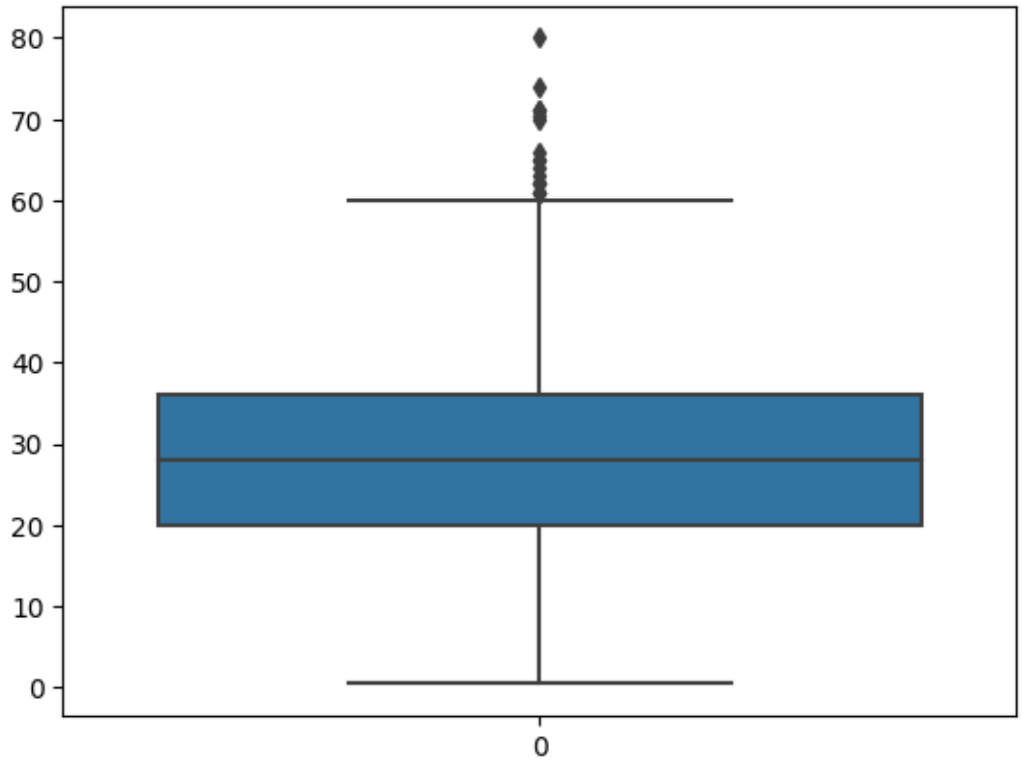
[118]:
```
sns.boxplot(df["SibSp"])
```

[118]: <Axes: >



[120]:
```
sns.boxplot(df["Age"])
```

[120]: <Axes: >



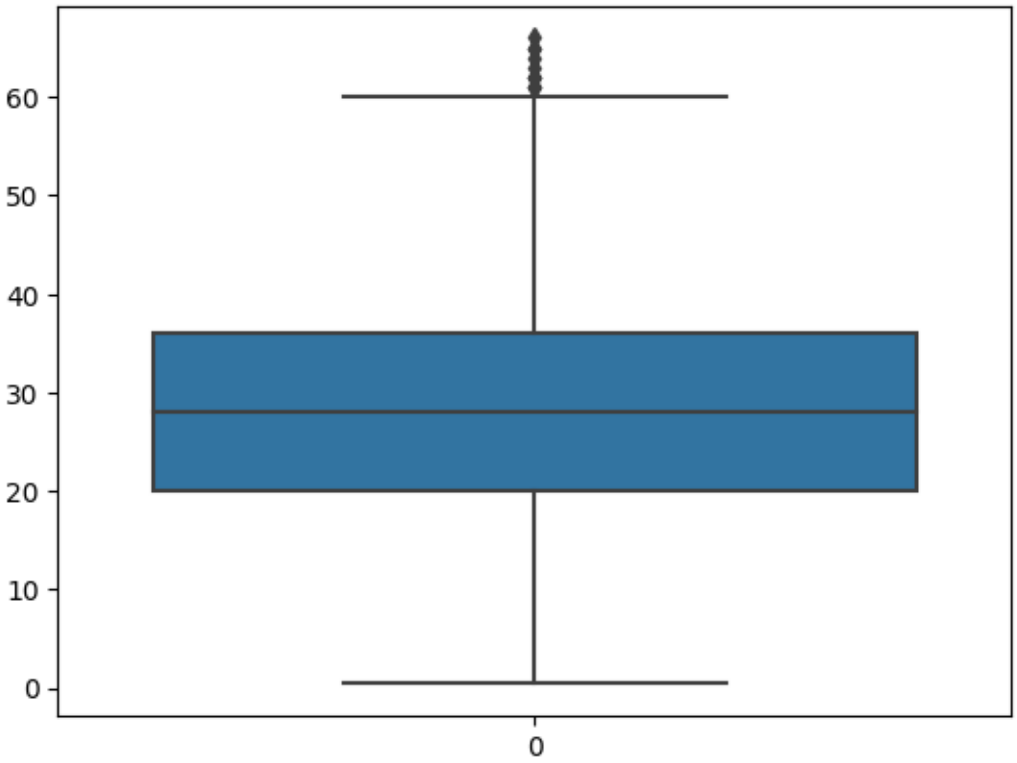
[121]:
```
p99 = df.Age.quantile(0.99)
p99
```

[121]: 66.55999999999995

[122]:
```
df = df[df.Age<=p99]
```

[123]:
```
sns.boxplot(df["Age"])
```

[123]: <Axes: >



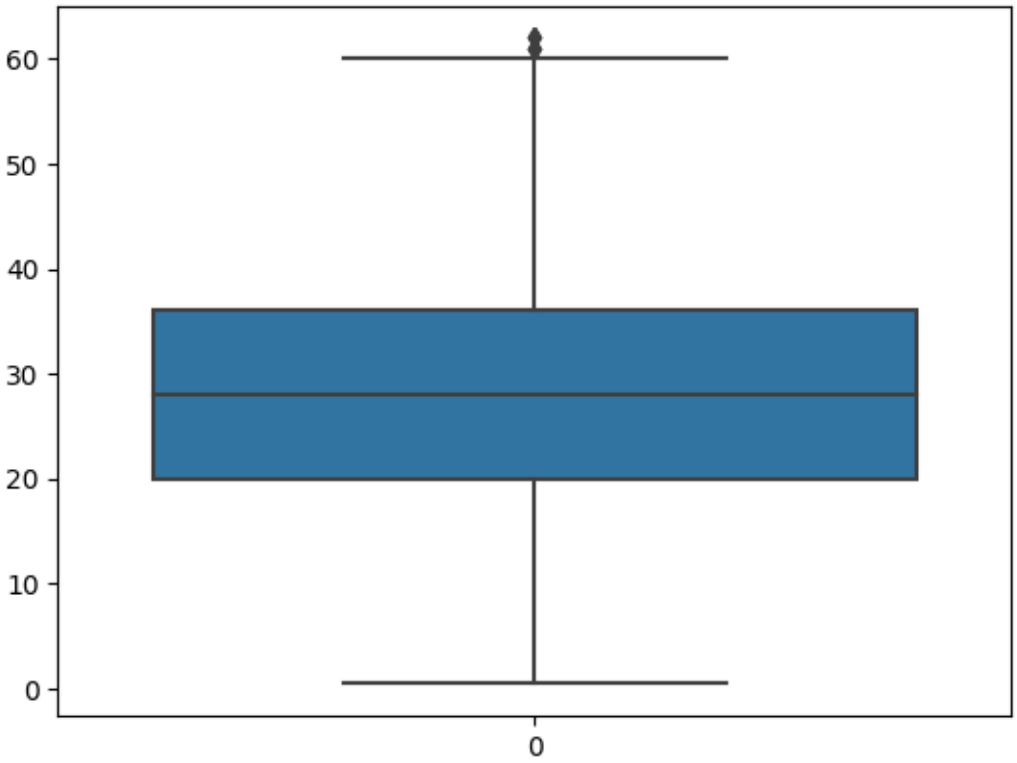
[124]:
```
p99 = df.Age.quantile(0.99)
p99
```

[124]: 62.0

[125]:
```
df = df[df.Age<=p99]
```

[126]:
```
sns.boxplot(df["Age"])
```

[126]: <Axes: >



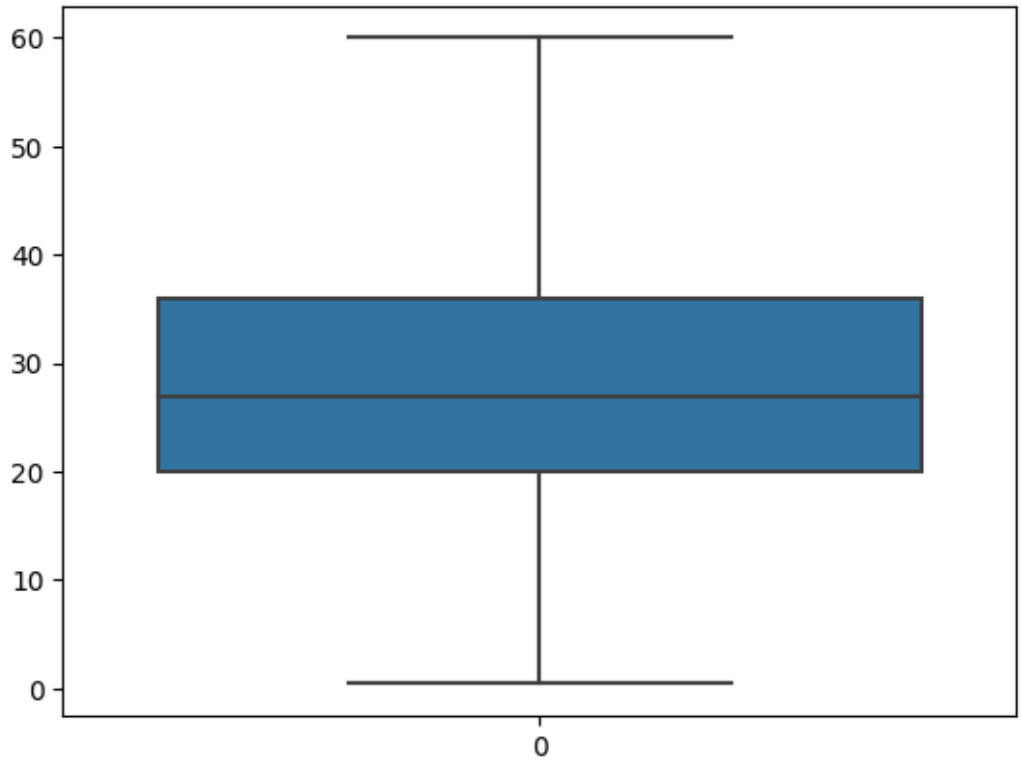
[127]:
```
p99 = df.Age.quantile(0.99)
p99
```

[127]: 60.25

[128]:
```
df = df[df.Age<=p99]
```

[129]:
```
sns.boxplot(df["Age"])
```

[129]: <Axes: >



#Splitting Dependent and Independent variables

[131]:
```
df1.head()
```

[131]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	29.699118	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th…	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	29.699118	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1					
4	5	0	3	Allen, Mr. William Henry	male	35.0	0					

```
3      0            113803  53.1000       C123        S
4      0            373450   8.0500   29.699118       S
```

[132]:
```
X = df.drop(columns=["Fare"],axis=1)
X.head()
```

[132]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
2            3         1       3
3            4         1       1
4            5         0       3
6            7         0       1

                                                Name     Sex   Age  SibSp  Parch  \
0                            Braund, Mr. Owen Harris    male  22.0      1      0
2                             Heikkinen, Miss. Laina  female  26.0      0      0
3  Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1      0
4                           Allen, Mr. William Henry    male  35.0      0      0
6                            McCarthy, Mr. Timothy J    male  54.0      0      0

             Ticket Cabin Embarked
0         A/5 21171   NaN        S
2  STON/O2. 3101282   NaN        S
3            113803  C123        S
4            373450   NaN        S
6             17463   E46        S
```

[133]:
```
X.shape
```

[133]:
```
(570, 11)
```

[134]:
```
y = df["Fare"]
y.head()
```

[134]:
```
0     7.2500
2     7.9250
3    53.1000
4     8.0500
6    51.8625
Name: Fare, dtype: float64
```

#Encoding

[135]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

[136]:
```
X["Embarked"] = le.fit_transform(X["Embarked"])
```

[138]:
```
X
```

[138]:
```
     PassengerId  Survived  Pclass  \
0              1         0       3
2              3         1       3
3              4         1       1
4              5         0       3
6              7         0       1
..           …       …     …
885          886         0       3
886          887         0       2
887          888         1       1
889          890         1       1
890          891         0       3

                                                 Name     Sex   Age  SibSp  Parch  \
0                             Braund, Mr. Owen Harris    male  22.0      1      0
2                              Heikkinen, Miss. Laina  female  26.0      0      0
3    Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1      0
4                            Allen, Mr. William Henry    male  35.0      0      0
6                             McCarthy, Mr. Timothy J    male  54.0      0      0
..                                                …     …   …    …     …
885          Rice, Mrs. William (Margaret Norton)  female  39.0      0      5
886                              Montvila, Rev. Juozas    male  27.0      0      0
887                   Graham, Miss. Margaret Edith  female  19.0      0      0
889                           Behr, Mr. Karl Howell    male  26.0      0      0
890                             Dooley, Mr. Patrick    male  32.0      0      0

                Ticket Cabin  Embarked
0            A/5 21171   NaN         2
2    STON/O2. 3101282   NaN         2
3               113803  C123         2
4               373450   NaN         2
6                17463   E46         2
..                 …   …       …
885             382652   NaN         1
886             211536   NaN         2
887             112053   B42         2
889             111369  C148         0
890             370376   NaN         1

[570 rows x 11 columns]
```

[139]:
```
print(le.classes_)
```

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

[140]:
```
mapping=dict(zip(le.classes_,range(len(le.classes_))))
mapping
```

[140]:
```
{'C': 0, 'Q': 1, 'S': 2}
```

#Feature Scaling

[155]:
```
from sklearn.preprocessing import MinMaxScaler
ms = MinMaxScaler()
```

[ ]:
```
X_Scaled = pd.DataFrame(ms.fit_transform(X),columns=X.columns)
X_Scaled
```

#Splitting Data into Train and Test

[151]:
```
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)
```

[152]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

[152]:
```
((399, 11), (171, 11), (399,), (171,))
```