Assignment-3 (Data_Preprocessing) sreya

September 21, 2023

0.1 Data Preprocessing

```
o   Import the Libraries.
o   Importing the dataset.
o   Checking for Null Values.
o   Data Visualization.
o   Outlier Detection
o   Splitting Dependent and Independent variables
o   Perform Encoding
o   Feature Scaling.
o   Splitting Data into Train and Test
```

0.2 Perform Data preprocessing on Titanic dataset

0.2.1 Import the Libraries.

[67]:

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

0.2.2 Importing the dataset.

[68]:

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

[69]:

```
df.head()
```

[69]:

```
   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
```

```
[70]: df.tail()
```

```
[70]:      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
```

```
[71]: df.shape
```

```
[71]: (891, 12)
```

```
[72]: df.ndim
```

```
[72]: 2
```

```
[73]: 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
```

[74]:

```
df.describe()
```

[74]:

```
       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
```

[75]:

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

```
C:\Users\vishnu vardhan\AppData\Local\Temp\ipykernel_193160\3182140910.py: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.
  corr=df.corr()
```

[75]:

```
             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
```

```
[76]: ports=pd.get_dummies(df.Embarked,prefix='Embarked')
      ports.head()
```

```
[76]:    Embarked_C  Embarked_Q  Embarked_S
      0           0           0           1
      1           1           0           0
      2           0           0           1
      3           0           0           1
      4           0           0           1
```

```
[77]: df=df.join(ports)
      df.drop(['Embarked'],axis=1,inplace=True)
```

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

```
[78]:    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_C  Embarked_Q  Embarked_S
      0      0         A/5 21171   7.2500   NaN           0           0           1
      1      0          PC 17599  71.2833   C85           1           0           0
      2      0  STON/O2. 3101282   7.9250   NaN           0           0           1
      3      0            113803  53.1000  C123           0           0           1
      4      0            373450   8.0500   NaN           0           0           1
```

```
[ ]:
```

0.2.3 Checking for Null Values

[80]:

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

[80]:

```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age            False
SibSp          False
Parch          False
Ticket         False
Fare           False
Embarked_C     False
Embarked_Q     False
Embarked_S     False
dtype: bool
```

[81]:

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

[81]:

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked_C     0
Embarked_Q     0
Embarked_S     0
dtype: int64
```

[82]:

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

[83]:

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

[83]:

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
```

```
Ticket         0
Fare           0
Embarked_C     0
Embarked_Q     0
Embarked_S     0
dtype: int64
```

[88]:
```
df.drop(['Cabin'],axis=1,inplace=True)
```

[86]:
```
df.drop(['Embarked_C'],axis=1,inplace=True)
df.drop(['Embarked_Q'],axis=1,inplace=True)
df.drop(['Embarked_S'],axis=1,inplace=True)
```

[87]:
```
df.head()
```

[87]:
```
   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
0      0         A/5 21171   7.2500
1      0          PC 17599  71.2833
2      0  STON/O2. 3101282   7.9250
3      0            113803  53.1000
4      0            373450   8.0500
```

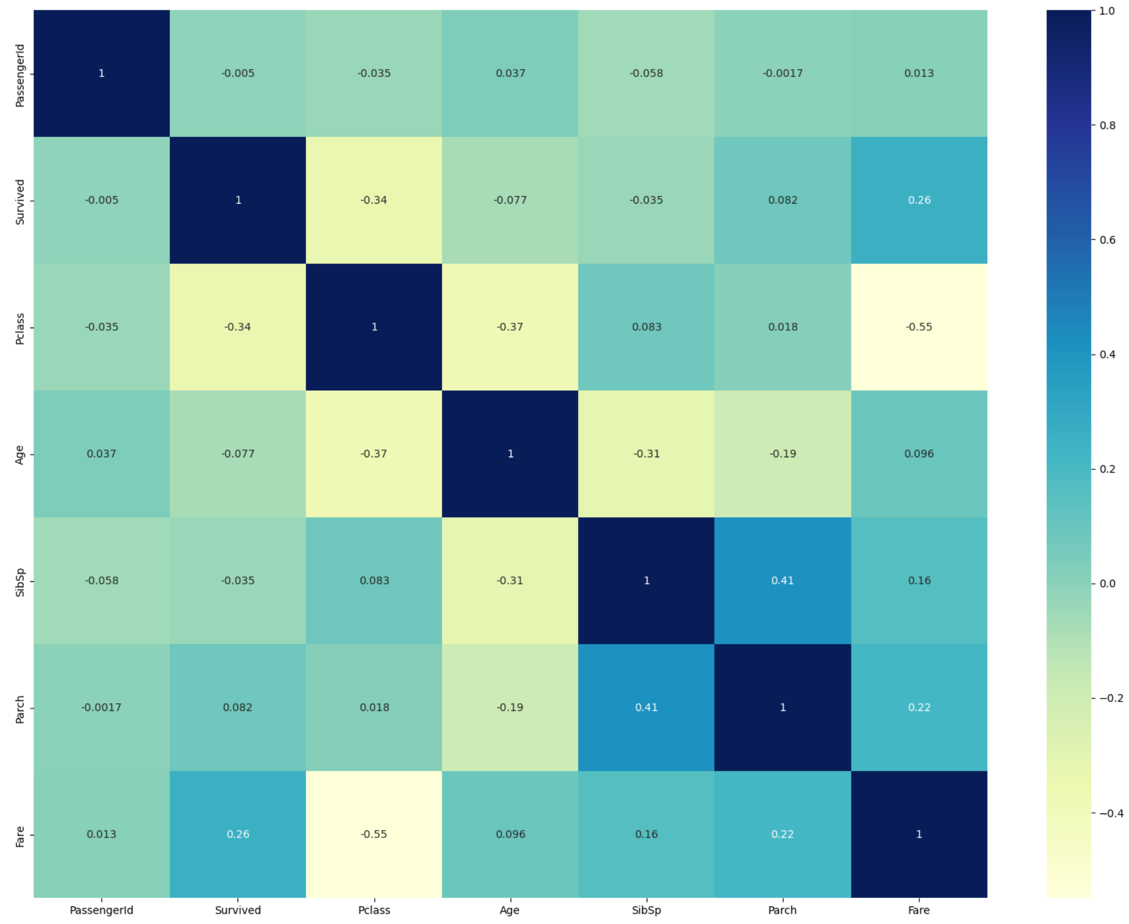
[21]:
```
df.shape
```

[21]:
```
(891, 10)
```

0.2.4 Data Visualization

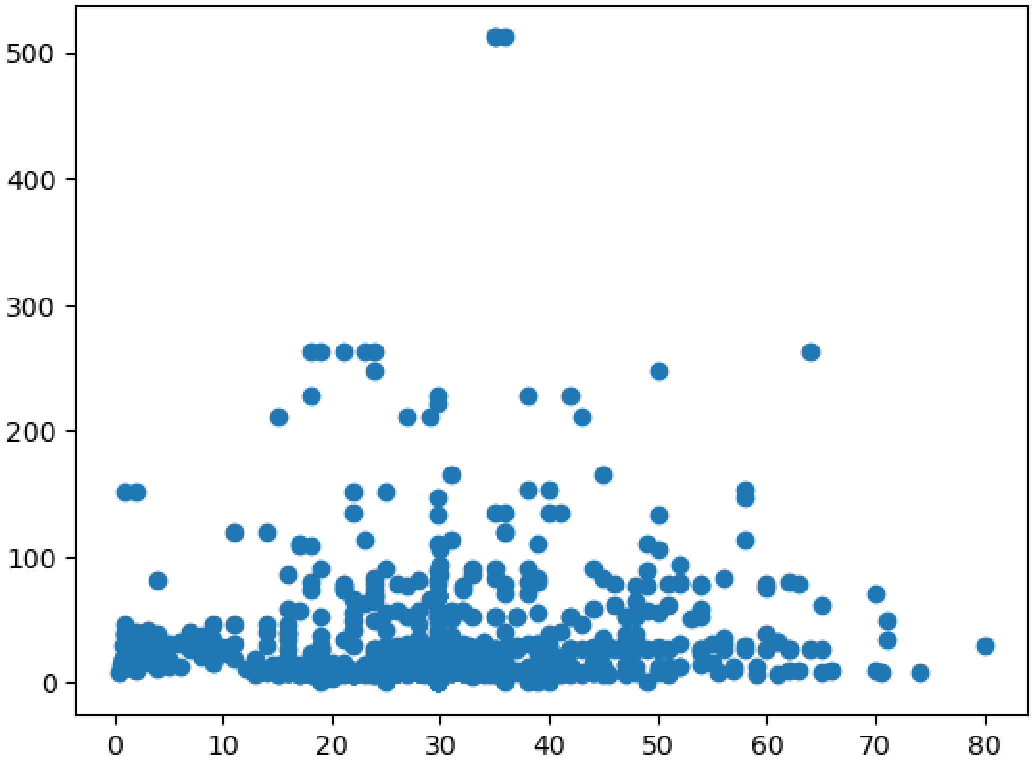
[89]:
```
plt.subplots(figsize=(20,15))
sns.heatmap(corr,annot=True,cmap='YlGnBu')
```

[89]:
```
<Axes: >
```

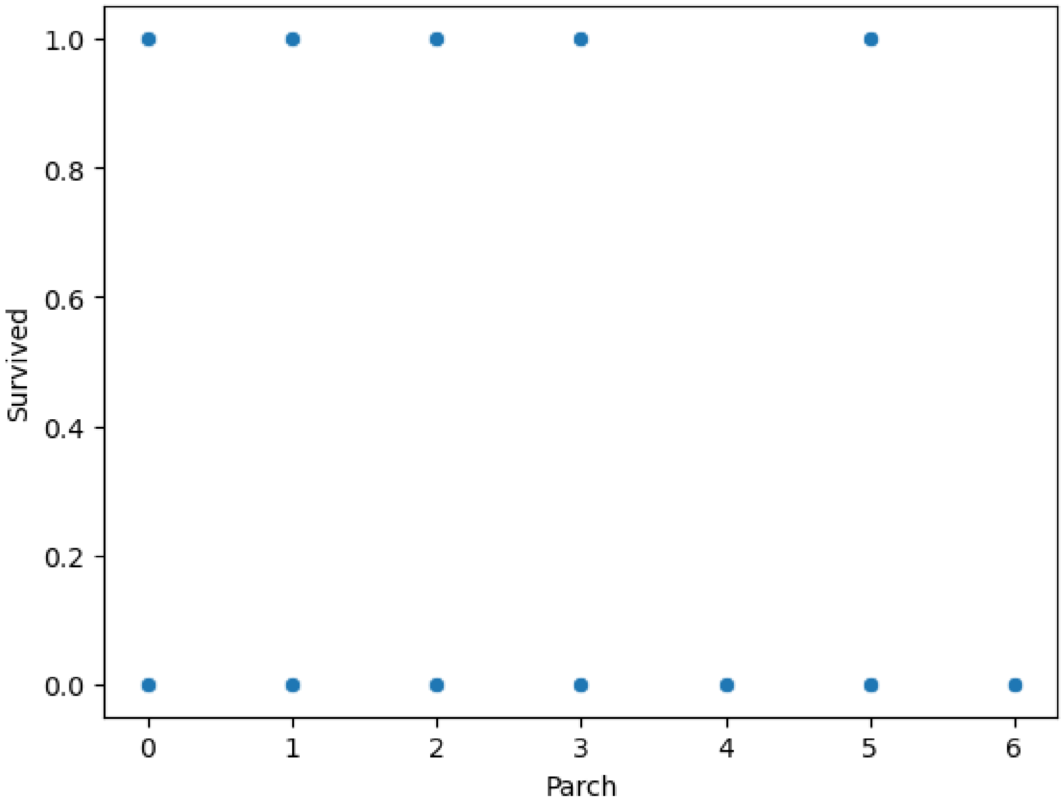


[90]:
```
plt.scatter(df["Age"],df["Fare"])
```

[90]:
```
<matplotlib.collections.PathCollection at 0x298f4e4f850>
```

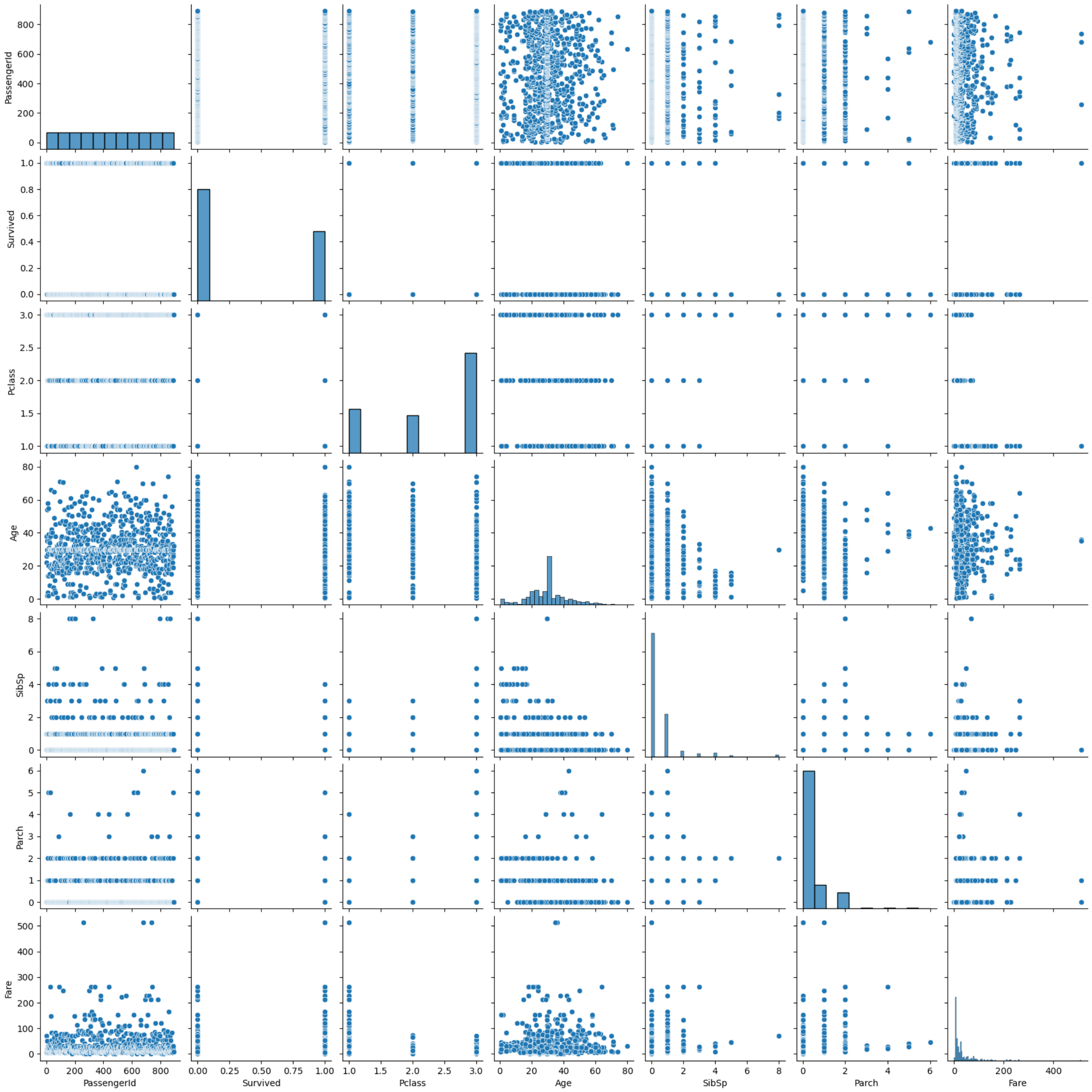


[91]:
```
sns.scatterplot(x="Parch",y="Survived",data=df)
plt.show()
```



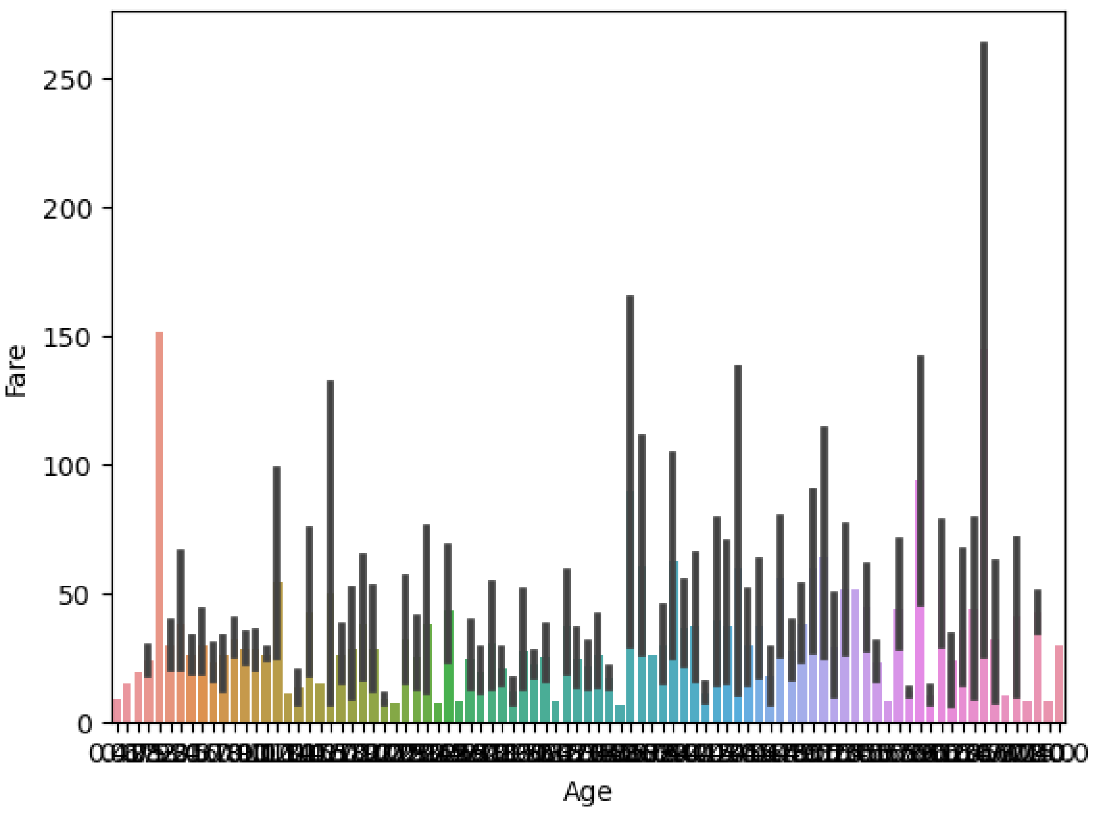
[92]:
```
sns.pairplot(df)
```

[92]:
```
<seaborn.axisgrid.PairGrid at 0x298f4cd2c10>
```



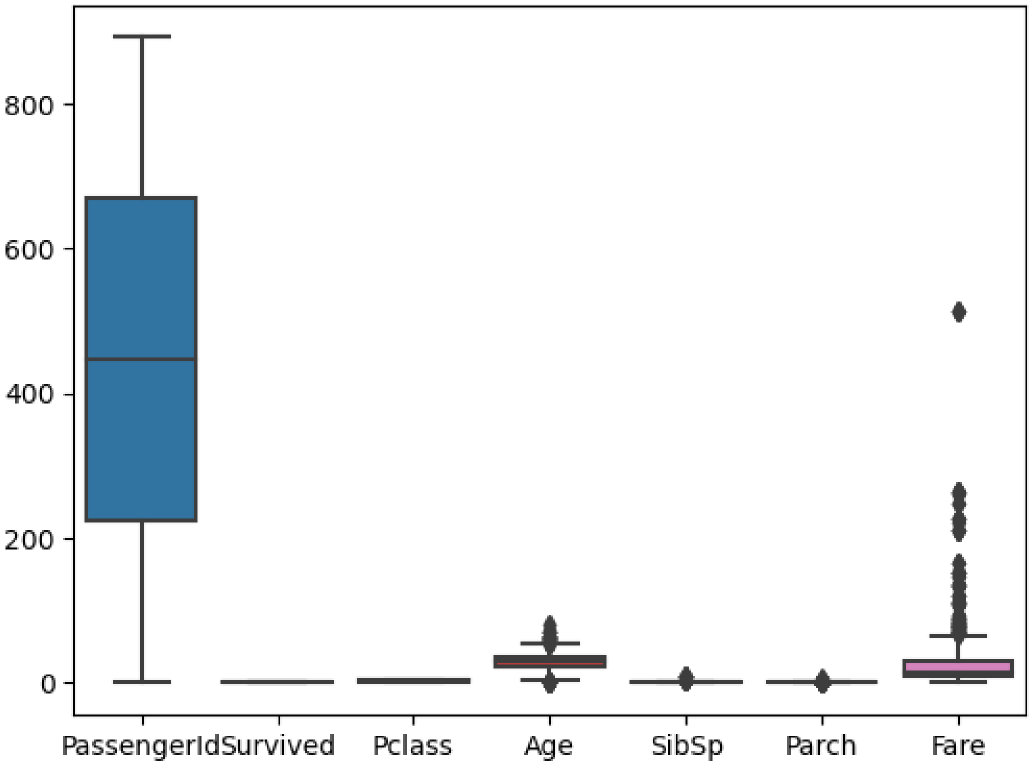
[93]:
```
sns.barplot(x=df["Age"],y=df["Fare"])
```

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



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

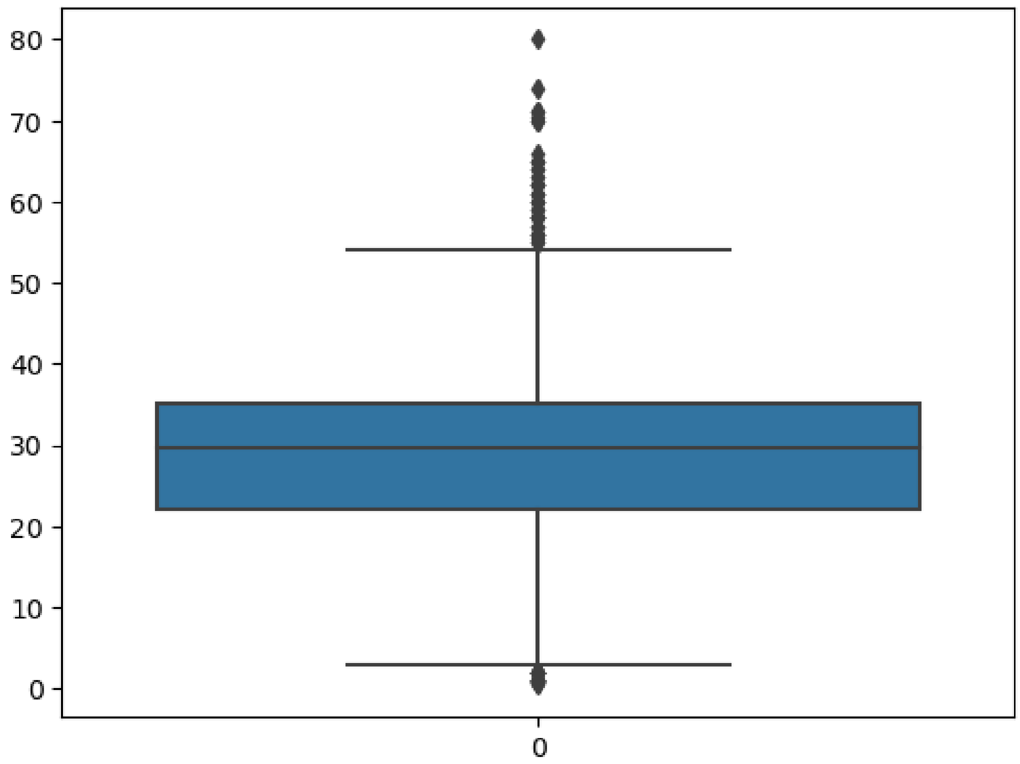
[32]:
```
<Axes: >
```



0.2.5 Outlier Detection

[94]:
```
sns.boxplot(df.Age)
```

[94]:
```
<Axes: >
```



```
[248]: q1=df.Age.quantile(0.25)
       q3=df.Age.quantile(0.75)
       q2=df.Age.quantile(0.50)
```

```
[249]: q1
```

```
[249]: 30.0
```

```
[250]: q2
```

```
[250]: 30.0
```

```
[251]: q3
```

```
[251]: 30.0
```

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

```
[252]: 0.0
```

[253]:
```
upper_limit=q3+1.5*IQR
lower_limit=q1-1.5*IQR
```

[254]:
```
upper_limit
```

[254]:
```
30.0
```

[255]:
```
lower_limit
```

[255]:
```
30.0
```

[256]:
```
df.median()
```

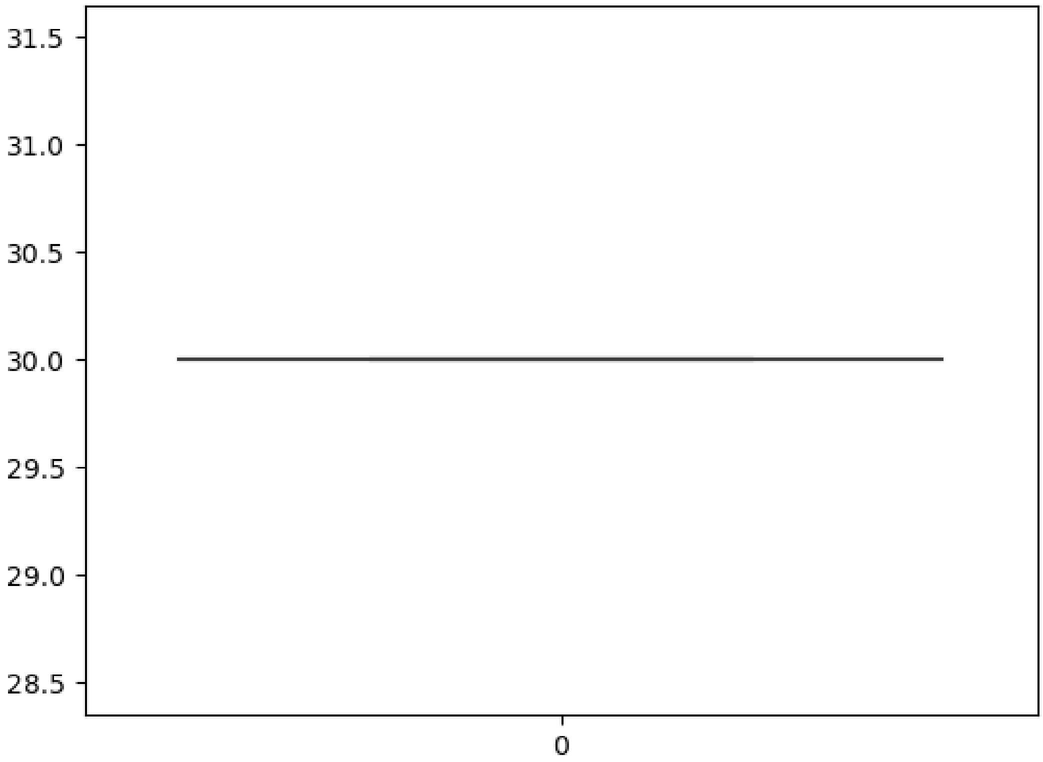
```
C:\Users\vishnu vardhan\AppData\Local\Temp\ipykernel_193160\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  df.median()
```

[256]:
```
PassengerId    446.00
Survived         0.00
Pclass           3.00
Age             30.00
SibSp            0.00
Parch            0.00
Fare            14.45
dtype: float64
```

[257]:
```
df['Age']=np.where(df['Age']>upper_limit,30,df['Age'])
df['Age']=np.where(df['Age']<lower_limit,30,df['Age'])
#df=df[(df.Age<lower_limit)&(df.Age>upper_limit)]
```

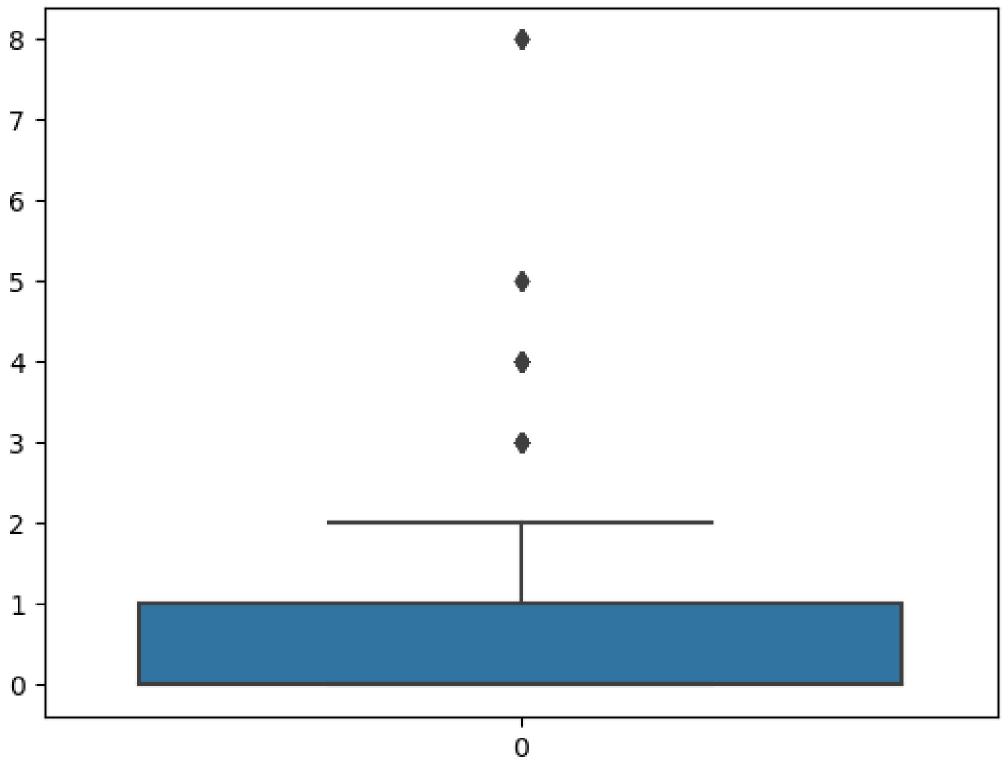
[258]:
```
sns.boxplot(df.Age)
```

[258]:
```
<Axes: >
```



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

```
[106]: <Axes: >
```



```
[107]: q1=df.SibSp.quantile(0.25)
       q3=df.SibSp.quantile(0.75)
       q2=df.SibSp.quantile(0.50)
```

```
[108]: q1
```

```
[108]: 0.0
```

```
[109]: q2
```

```
[109]: 0.0
```

```
[110]: q3
```

```
[110]: 1.0
```

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

```
[111]: 1.0
```

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

```
[112]: 2.5
```

```
[113]: lower_limit=q1-1.5*IQR
       lower_limit
```

```
[113]: -1.5
```

```
[114]: df.median()
```

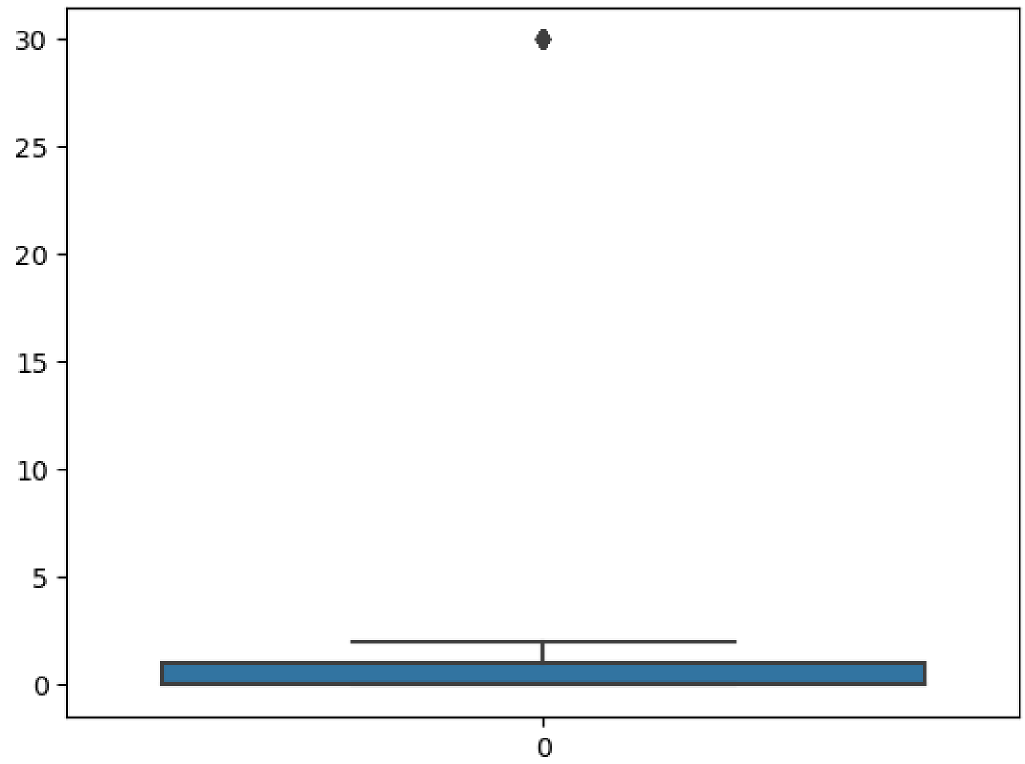
```
C:\Users\vishnu vardhan\AppData\Local\Temp\ipykernel_193160\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  df.median()
```

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

```
[115]: df['SibSp']=np.where(df['SibSp']>upper_limit,30,df['SibSp'])
```

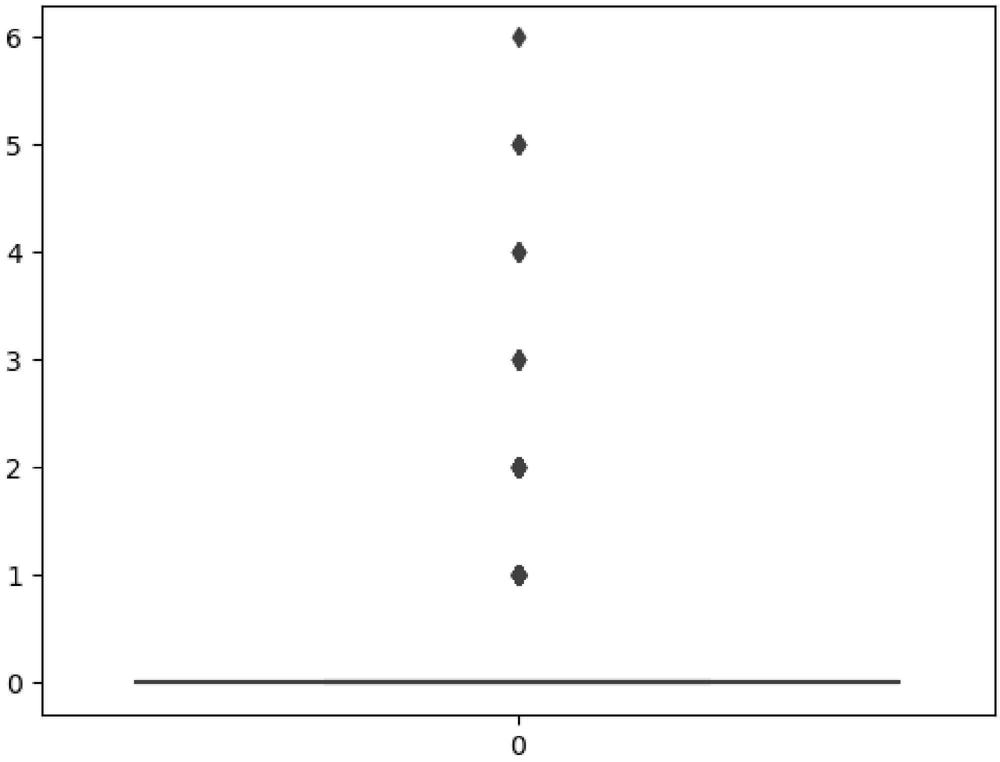
```
[116]: sns.boxplot(df.SibSp)
```

```
[116]: <Axes: >
```



[117]:
```
sns.boxplot(df.Parch)
```

[117]:
```
<Axes: >
```



```
[118]: q1=df.Parch.quantile(0.25)
       q3=df.Parch.quantile(0.75)
       q2=df.Parch.quantile(0.50)
```

```
[119]: q1
```

```
[119]: 0.0
```

```
[120]: q2
```

```
[120]: 0.0
```

```
[121]: q3
```

```
[121]: 0.0
```

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

```
[122]: 0.0
```

[123]:
```
upper_limit=q3+1.5*IQR
upper_limit
```

[123]:
```
0.0
```

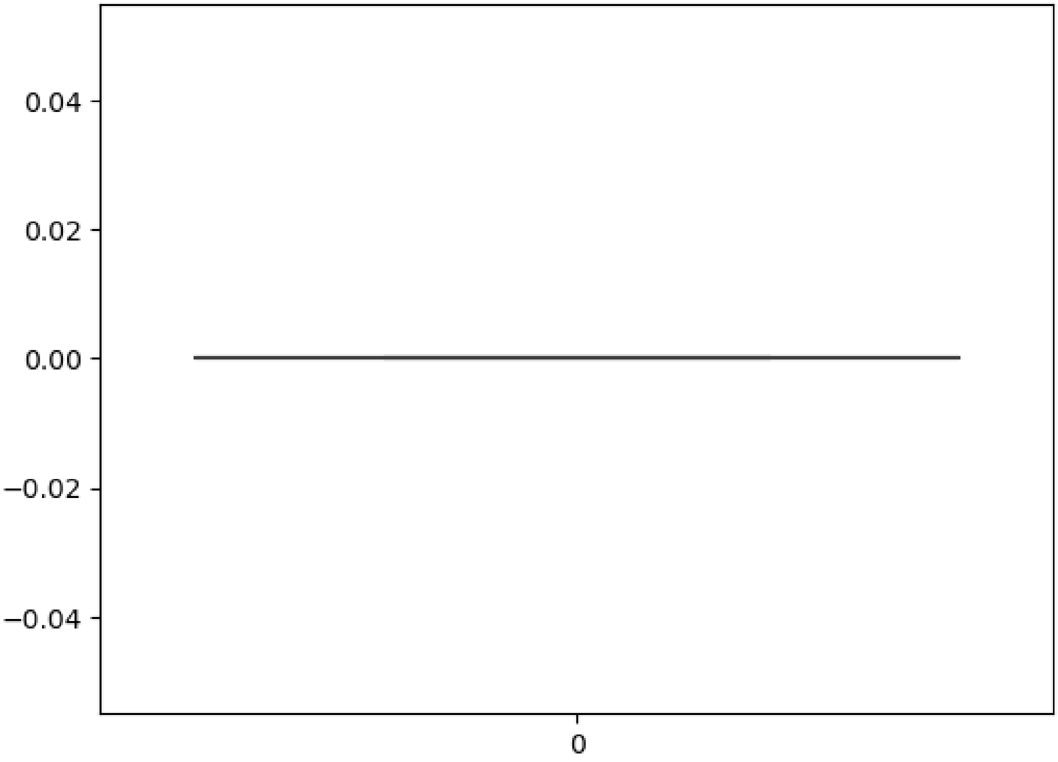
[124]:
```
lower_limit=q1-1.5*IQR
lower_limit
```

[124]:
```
0.0
```

[125]:
```
df['Parch']=np.where(df['Parch']>upper_limit,0,df['Parch'])
```

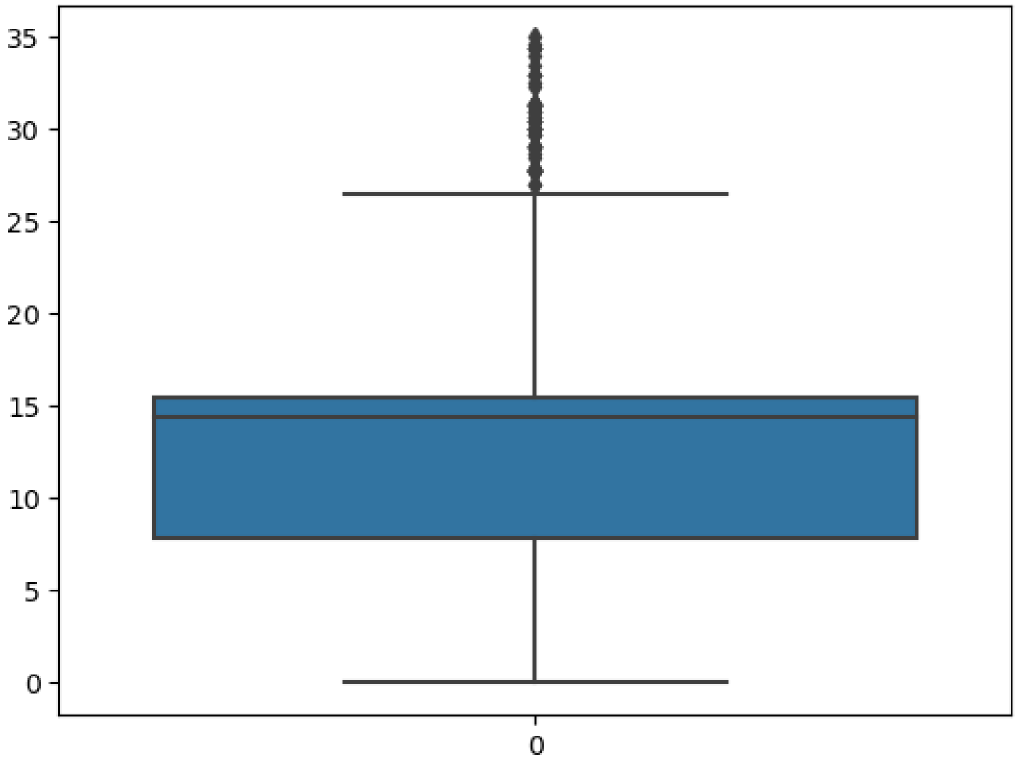
[164]:
```
sns.boxplot(df.Parch)
```

[164]:
```
<Axes: >
```



[165]:
```
sns.boxplot(df.Fare)
```

[165]:
```
<Axes: >
```



```
[259]: q1=df.Fare.quantile(0.25)
       q3=df.Fare.quantile(0.75)
       q2=df.Fare.quantile(0.50)

[260]: q1

[260]: 7.9104

[261]: q2

[261]: 14.45

[262]: q3

[262]: 14.45

[263]: IQR=q3-q1
       IQR

[263]: 6.539599999999999
```

[264]:
```
upper_limit=q3+1.5*IQR
upper_limit
```

[264]:
```
24.2594
```

[265]:
```
lower_limit=q1-1.5*IQR
lower_limit
```

[265]:
```
-1.8989999999999982
```

[266]:
```
df.median()
```

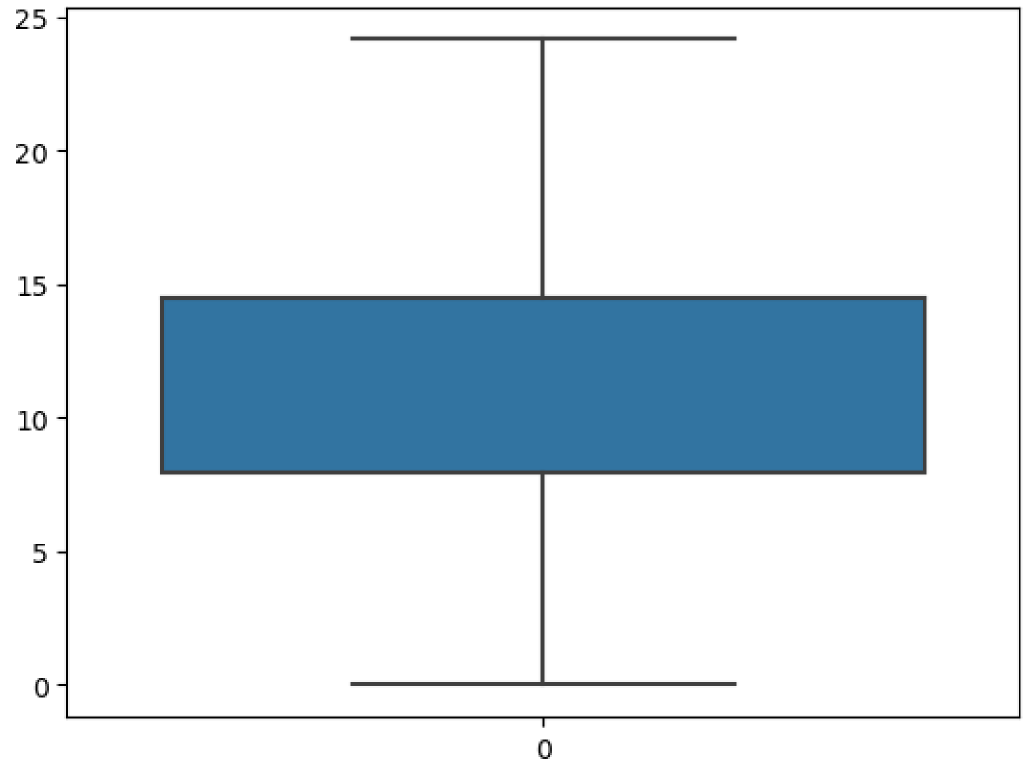
```
C:\Users\vishnu vardhan\AppData\Local\Temp\ipykernel_193160\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  df.median()
```

[266]:
```
PassengerId    446.00
Survived         0.00
Pclass           3.00
Age             30.00
SibSp            0.00
Parch            0.00
Fare            14.45
dtype: float64
```

[267]:
```
df['Fare']=np.where(df['Fare']>upper_limit,14.45,df['Fare'])
```

[268]:
```
sns.boxplot(df.Fare)
```

[268]:
```
<Axes: >
```



0.2.6 Splitting Dependent and Independent variables

[177]:
```
df.head(10)
```

[177]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3
5            6         0       3
6            7         0       1
7            8         0       3
8            9         1       3
9           10         1       2

                                                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
```

```
5                                   Moran, Mr. James    male  29.699118
6                            McCarthy, Mr. Timothy J    male  54.000000
7                     Palsson, Master. Gosta Leonard    male   2.000000
8  Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)  female  27.000000
9                Nasser, Mrs. Nicholas (Adele Achem)  female  14.000000

   SibSp  Parch            Ticket     Fare
0      1      0         A/5 21171   7.2500
1      1      0          PC 17599  14.4500
2      0      0  STON/O2. 3101282   7.9250
3      1      0            113803  14.4500
4      0      0            373450   8.0500
5      0      0            330877   8.4583
6      0      0             17463  14.4500
7     30      0            349909  21.0750
8      0      0            347742  11.1333
9      1      0            237736  14.4500
```

[178]:
```
x=df.iloc[:,2:]
y=df.iloc[:,1:2]
```

[179]:
```
x
```

[179]:
```
     Pclass                                               Name     Sex  \
0         3                            Braund, Mr. Owen Harris    male
1         1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female
2         3                             Heikkinen, Miss. Laina  female
3         1       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female
4         3                           Allen, Mr. William Henry    male
..       …                                                …     …
886       2                              Montvila, Rev. Juozas    male
887       1                       Graham, Miss. Margaret Edith  female
888       3           Johnston, Miss. Catherine Helen "Carrie"  female
889       1                              Behr, Mr. Karl Howell    male
890       3                                Dooley, Mr. Patrick    male

           Age  SibSp  Parch            Ticket    Fare
0    22.000000      1      0         A/5 21171   7.250
1    38.000000      1      0          PC 17599  14.450
2    26.000000      0      0  STON/O2. 3101282   7.925
3    35.000000      1      0            113803  14.450
4    35.000000      0      0            373450   8.050
..         …    …    …               …     …
886  27.000000      0      0            211536  13.000
887  19.000000      0      0            112053  14.450
888  29.699118      1      0        W./C. 6607  23.450
889  26.000000      0      0            111369  14.450
```

```
890  32.000000      0      0             370376   7.750

[891 rows x 8 columns]
```

[180]:
```
y
```

[180]:
```
     Survived
0           0
1           1
2           1
3           1
4           0
..        ...
886         0
887         1
888         0
889         1
890         0

[891 rows x 1 columns]
```

[181]:
```
x.shape
```

[181]:
```
(891, 8)
```

0.2.7 Perform Encoding

[182]:
```
from sklearn.preprocessing import LabelEncoder
```

[185]:
```
le=LabelEncoder()
```

[186]:
```
x["Name"]=le.fit_transform(x["Name"])
```

[187]:
```
x.head()
```

[187]:
```
   Pclass  Name     Sex   Age  SibSp  Parch            Ticket    Fare
0       3   108    male  22.0      1      0         A/5 21171   7.250
1       1   190  female  38.0      1      0          PC 17599  14.450
2       3   353  female  26.0      0      0  STON/O2. 3101282   7.925
3       1   272  female  35.0      1      0            113803  14.450
4       3    15    male  35.0      0      0            373450   8.050
```

[196]:
```
x["Name"].value_counts()
```

[196]:
```
108    1
98     1
267    1
284    1
```

```
566    1
      ..
431    1
518    1
411    1
428    1
220    1
Name: Name, Length: 891, dtype: int64
```

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

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

```
[189]:    Pclass  Name  Sex   Age  SibSp  Parch            Ticket    Fare
       0       3   108    1  22.0      1      0         A/5 21171   7.250
       1       1   190    0  38.0      1      0          PC 17599  14.450
       2       3   353    0  26.0      0      0  STON/O2. 3101282   7.925
       3       1   272    0  35.0      1      0            113803  14.450
       4       3    15    1  35.0      0      0            373450   8.050
```

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

```
[194]: 1    577
       0    314
       Name: Sex, dtype: int64
```

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

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

```
[191]:    Pclass  Name  Sex   Age  SibSp  Parch  Ticket    Fare
       0       3   108    1  22.0      1      0     523   7.250
       1       1   190    0  38.0      1      0     596  14.450
       2       3   353    0  26.0      0      0     669   7.925
       3       1   272    0  35.0      1      0      49  14.450
       4       3    15    1  35.0      0      0     472   8.050
```

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

```
[195]: 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
```

0.2.8 Feature Scaling.

[197]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
```

[198]:
```
x_scaled=sc.fit_transform(x)
x_scaled
```

[198]:
```
array([[ 0.82737724, -1.31021659,  0.73769513, …,  0.        ,
         0.91896631, -1.0191909 ],
       [-1.56610693, -0.99141018, -1.35557354, …,  0.        ,
         1.28262456,  0.27123506],
       [ 0.82737724, -0.35768524, -1.35557354, …,  0.        ,
         1.64628282, -0.89821347],
       …,
       [ 0.82737724, -0.12441226, -1.35557354, …,  0.        ,
         1.67617254,  1.88426751],
       [-1.56610693, -1.41518943,  0.73769513, …,  0.        ,
        -1.64656796,  0.27123506],
       [ 0.82737724, -0.87477369,  0.73769513, …,  0.        ,
         0.63501397, -0.92957799]])
```

0.2.9 Splitting Data into Train and Test

[201]:
```
from sklearn.model_selection import train_test_split
```

[203]:
```
tts=train_test_split
```

[204]:
```
x_train,x_test,y_train,y_test=tts(x_scaled,y,test_size=0.2,random_state=0)
```

[206]:
```
print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

```
(712, 8) (179, 8) (712, 1) (179, 1)
```

[270]:
```
x_train
```

[270]:
```
array([[ 0.82737724, -1.34520754, -1.35557354, …,  0.        ,
        -0.67515207,  0.41386297],
       [-0.36936484,  0.00777577,  0.73769513, …,  0.        ,
         1.03852519, -0.43670696],
       [-0.36936484,  0.2293851 ,  0.73769513, …,  0.        ,
         1.3922202 ,  0.27123506],
       …,
       [ 0.82737724,  0.61039764,  0.73769513, …,  0.        ,
        -0.26167762, -0.93257106],
       [ 0.82737724,  1.71066854, -1.35557354, …,  0.        ,
        -0.19193494,  0.79995125],
       [-0.36936484, -1.29466506,  0.73769513, …,  0.        ,
        -0.49083214,  0.27123506]])
```

```
[271]: x_test
```

```
[271]: array([[ 0.82737724,  1.69122913,  0.73769513, …,  0.        ,
        -0.80965581,  0.27272263],
       [ 0.82737724,  1.63291088,  0.73769513, …,  0.        ,
         1.40218344, -0.96542316],
       [ 0.82737724,  0.9175404 ,  0.73769513, …,  0.        ,
         0.70475665,  0.27123506],
       …,
       [-1.56610693,  0.53263998, -1.35557354, …,  0.        ,
         0.38593297,  0.27123506],
       [ 0.82737724, -1.53960169,  0.73769513, …,  0.        ,
         0.0172931 , -0.91090266],
       [ 0.82737724, -1.43851673,  0.73769513, …,  0.        ,
        -0.32643868, -0.87581024]])
```

```
[207]: y_train
```

```
[207]:      Survived
140         0
439         0
817         0
378         0
491         0
..        …
835         1
192         1
629         0
559         1
684         0

[712 rows x 1 columns]
```

```
[269]: y_test
```

```
[269]:      Survived
495         0
648         0
278         0
31          1
255         1
```

```
..        …
780         1
837         0
215         1
833         0
372         0

[179 rows x 1 columns]
```

[ ]:

```
```