21BIT0433_Assignment-3 (Data Preprocessing) YASODA RUSHITHA

September 21, 2023

REPAKULA YASODA RUSHITHA - 21BIT0433

0.1 Data Preprocessing

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

0.2 Perform Data preprocessing on Titanic dataset

0.2.1 Import the Libraries.

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

0.2.2 Importing the dataset.

[7]:
```
df =pd.read_csv("Titanic-Dataset.csv")
```

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

[8]:
```
   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
```

[9]:
```
df.tail()
```

[9]:
```
     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
```

[10]:
```
df.shape
```

[10]:
```
(891, 12)
```

[11]:
```
df.ndim
```

[11]:
```
2
```

[12]:
```
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
```

[13]:
```
df.describe()
```

[13]:

	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

[14]:
```
corr=df.corr()
corr
```

```
C:\Users\RUSHITHA REPAKULA\AppData\Local\Temp\ipykernel_13584\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()
```

[14]:

	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
```

[15]:
```
ports=pd.get_dummies(df.Embarked,prefix='Embarked')
ports.head()
```

[15]:
```
   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
```

[16]:
```
df=df.join(ports)
df.drop(['Embarked'],axis=1,inplace=True)
```

[17]:
```
df.head()
```

[17]:
```
   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

[18]:
```
df.isnull().any()
```

[18]:
```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin           True
Embarked_C     False
Embarked_Q     False
Embarked_S     False
dtype: bool
```

[19]:
```
df.isnull().sum()
```

[19]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked_C       0
Embarked_Q       0
Embarked_S       0
dtype: int64
```

[20]:
```
df['Age'].fillna(df['Age'].mean(),inplace=True)
```

[21]:
```
df.isnull().sum()
```

[21]:
```
PassengerId      0
Survived         0
Pclass           0
```

```
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked_C       0
Embarked_Q       0
Embarked_S       0
dtype: int64
```

[22]:
```
df.drop(['Cabin'],axis=1,inplace=True)
```

[23]:
```
df.drop(['Embarked_C'],axis=1,inplace=True)
df.drop(['Embarked_Q'],axis=1,inplace=True)
df.drop(['Embarked_S'],axis=1,inplace=True)
```

[24]:
```
df.head()
```

[24]:
```
   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
```

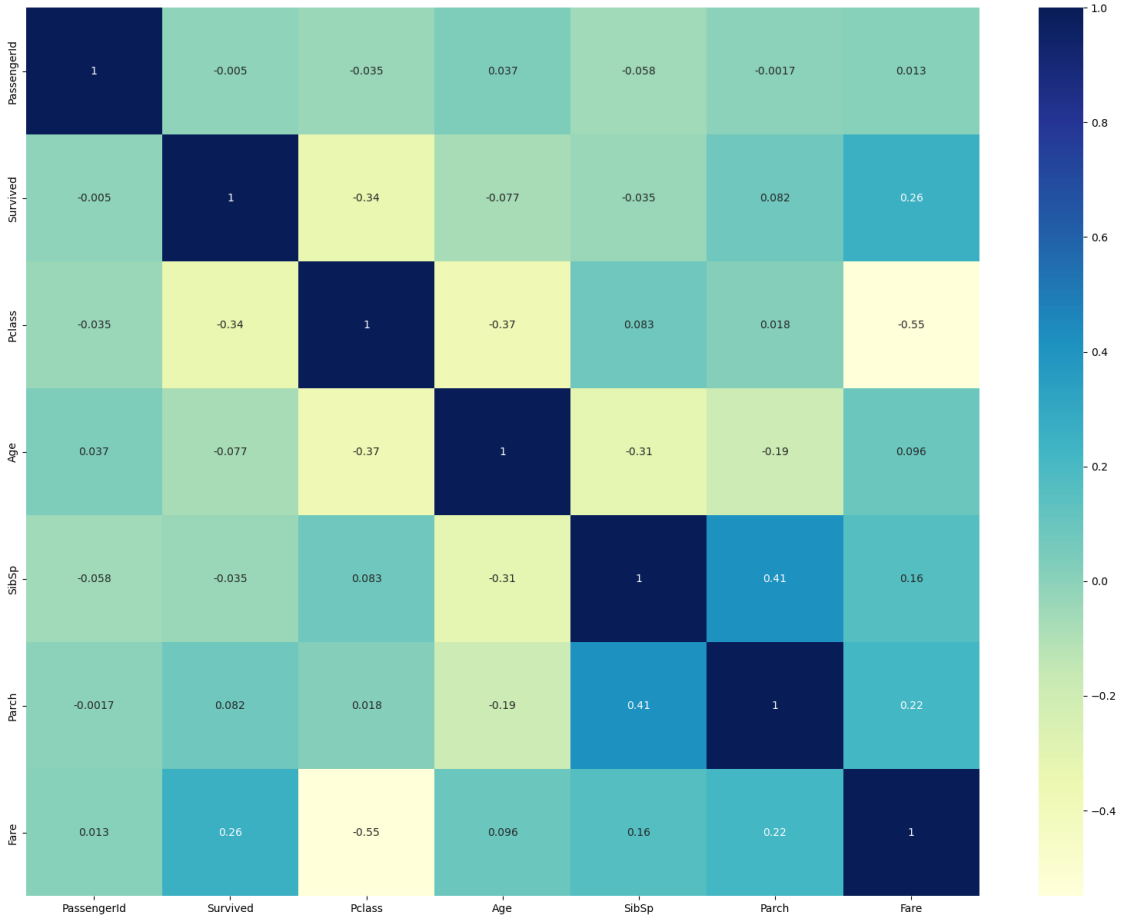
[25]:
```
df.shape
```

[25]:
```
(891, 10)
```

1 Data Visualization

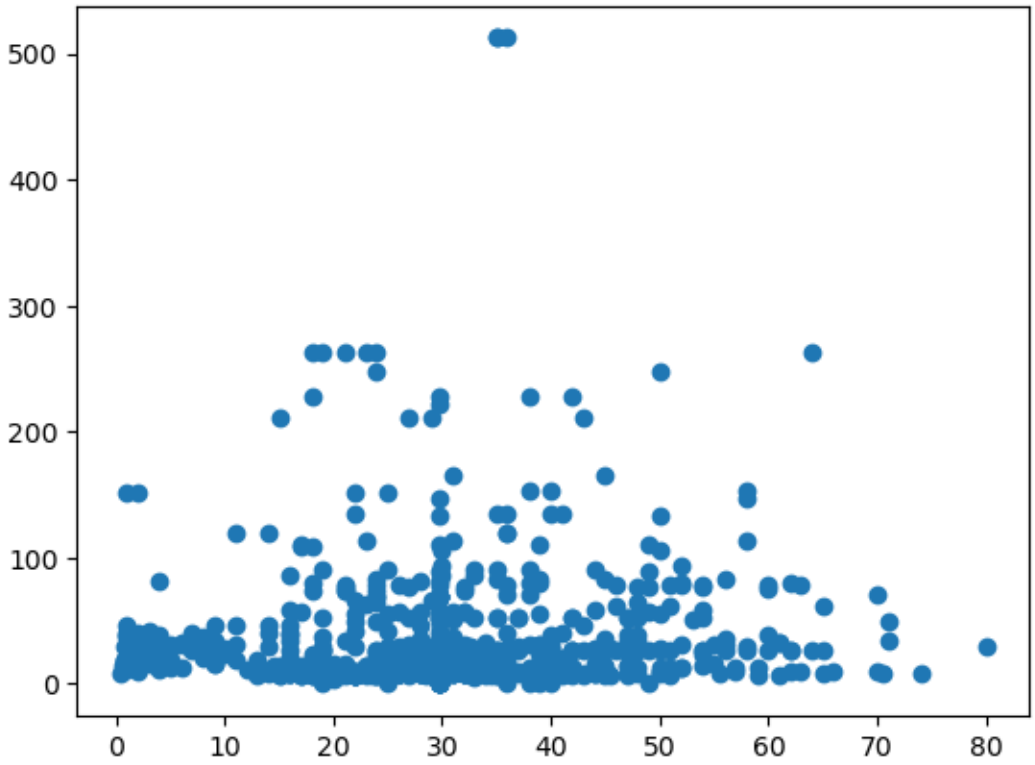
[26]:
```
plt.subplots(figsize=(20,15))
sns.heatmap(corr,annot=True,cmap='YlGnBu')
```

[26]: <Axes: >

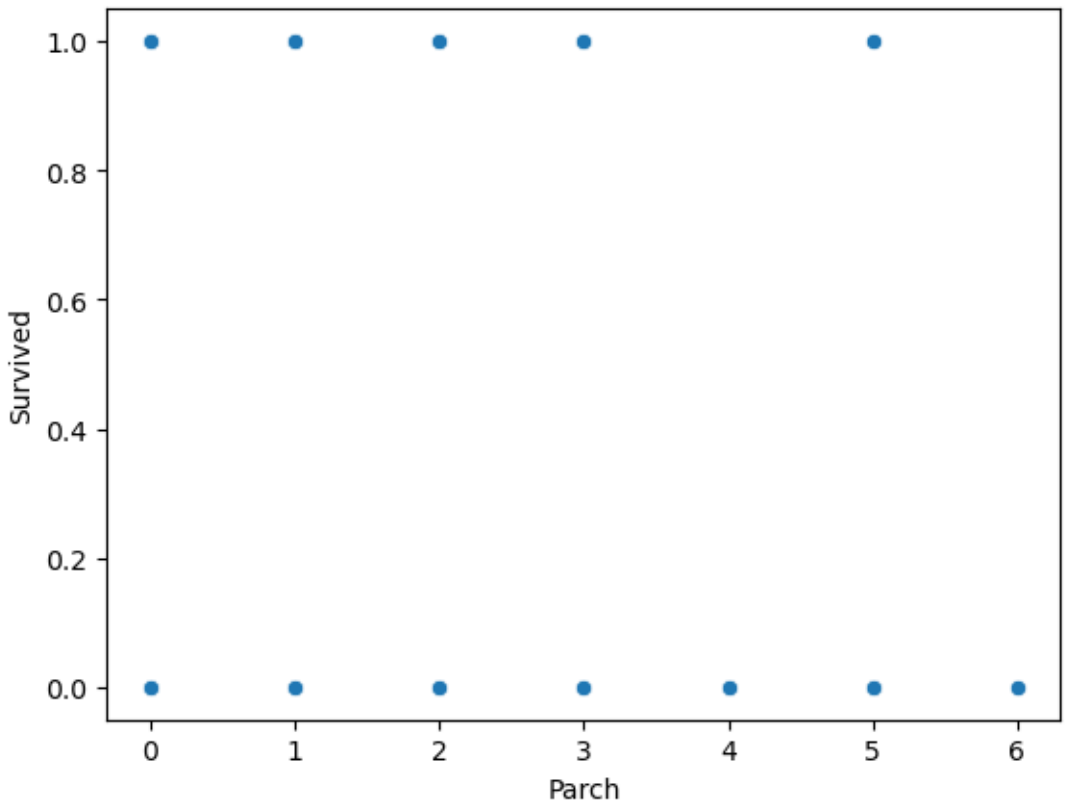


[27]:
```
plt.scatter(df["Age"],df["Fare"])
```

[27]: <matplotlib.collections.PathCollection at 0x179e5a4f050>

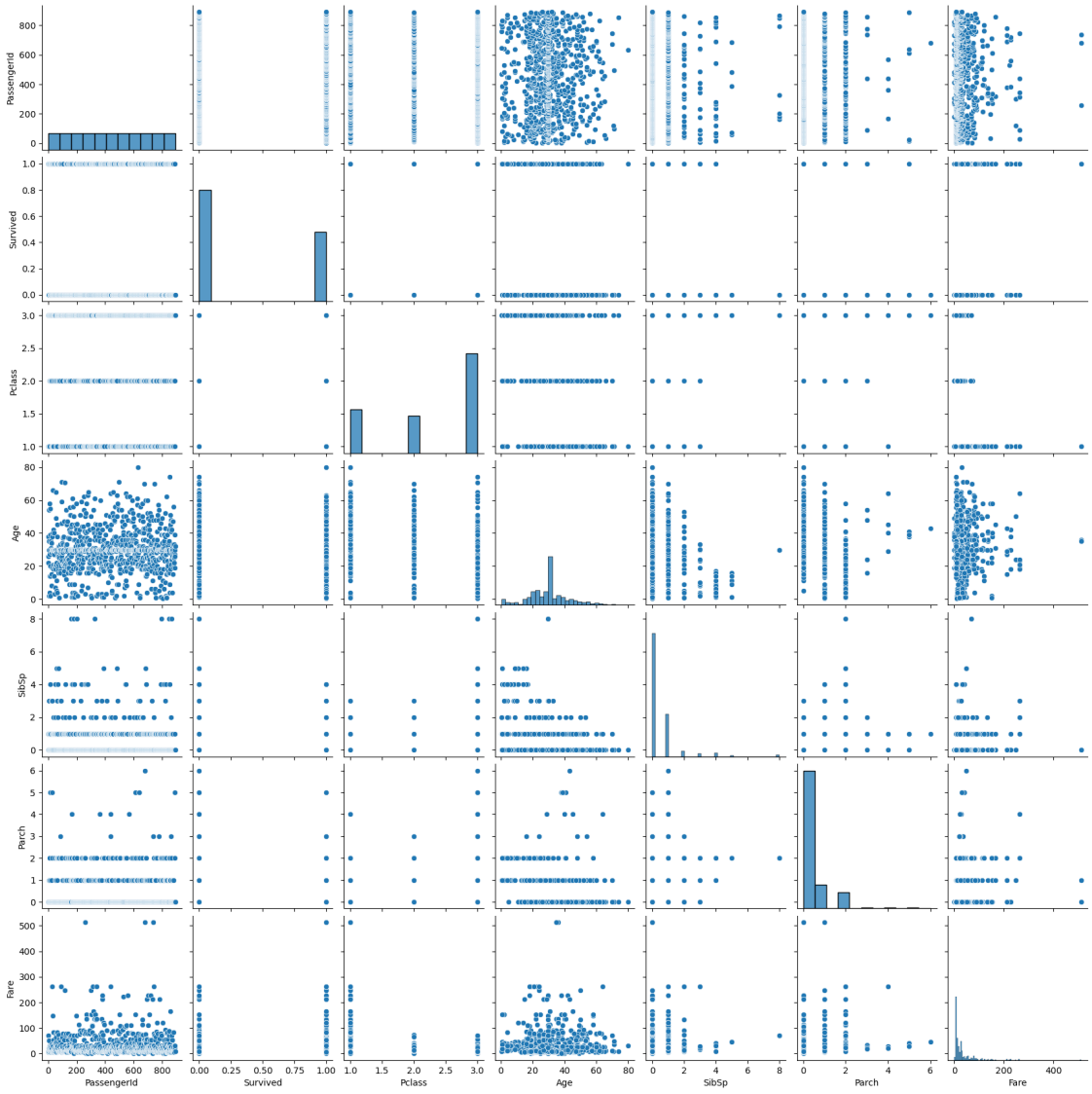


[28]:
```
sns.scatterplot(x="Parch",y="Survived",data=df)
plt.show()
```



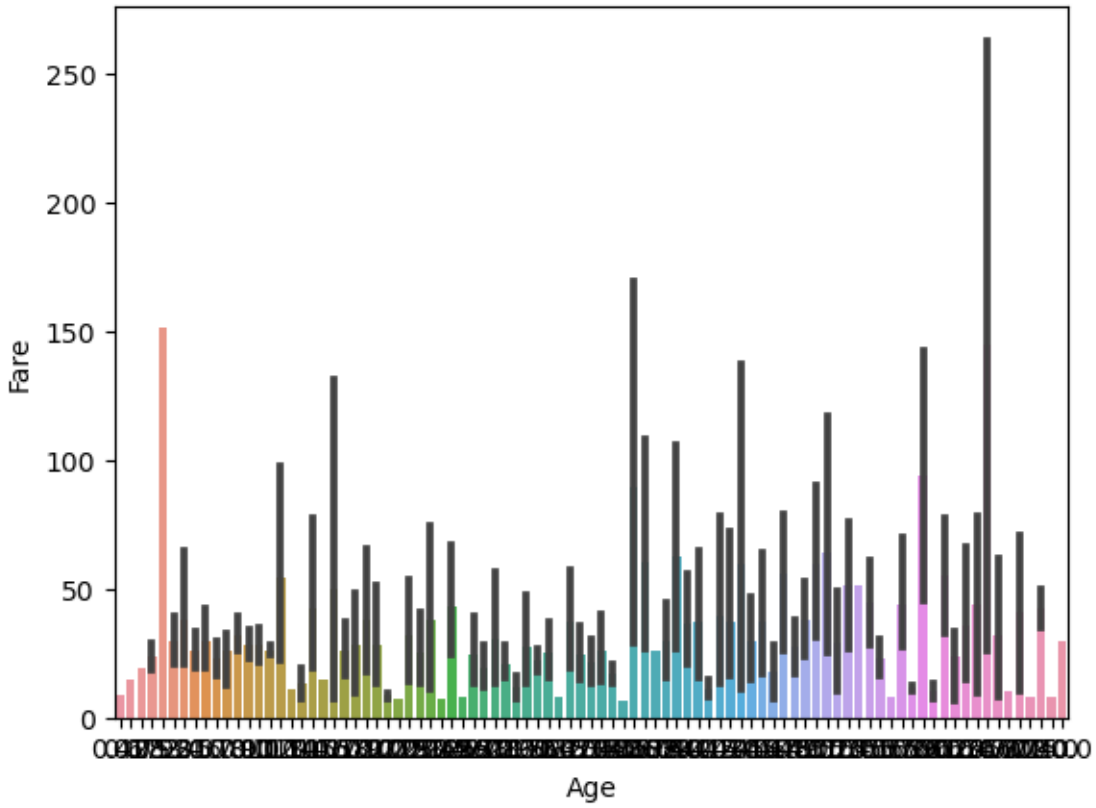
[29]:
```
sns.pairplot(df)
```

[29]:
```
<seaborn.axisgrid.PairGrid at 0x179e5a76f10>
```



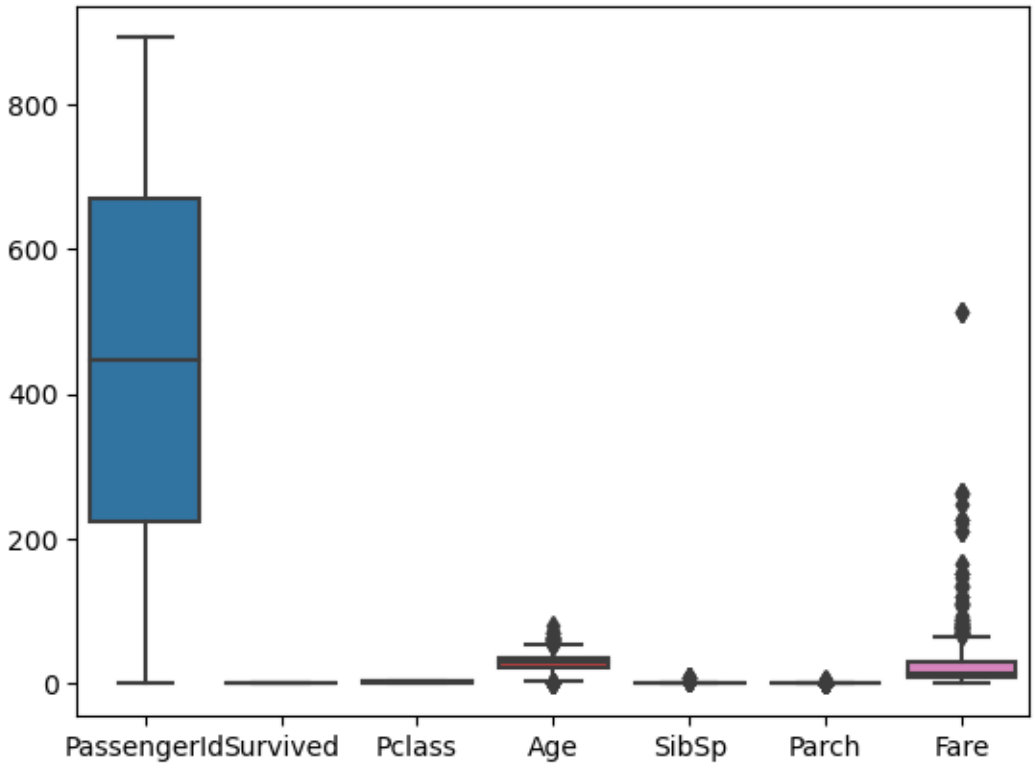
[30]:
```
sns.barplot(x=df["Age"],y=df["Fare"])
```

[30]:
```
<Axes: xlabel='Age', ylabel='Fare'>
```



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

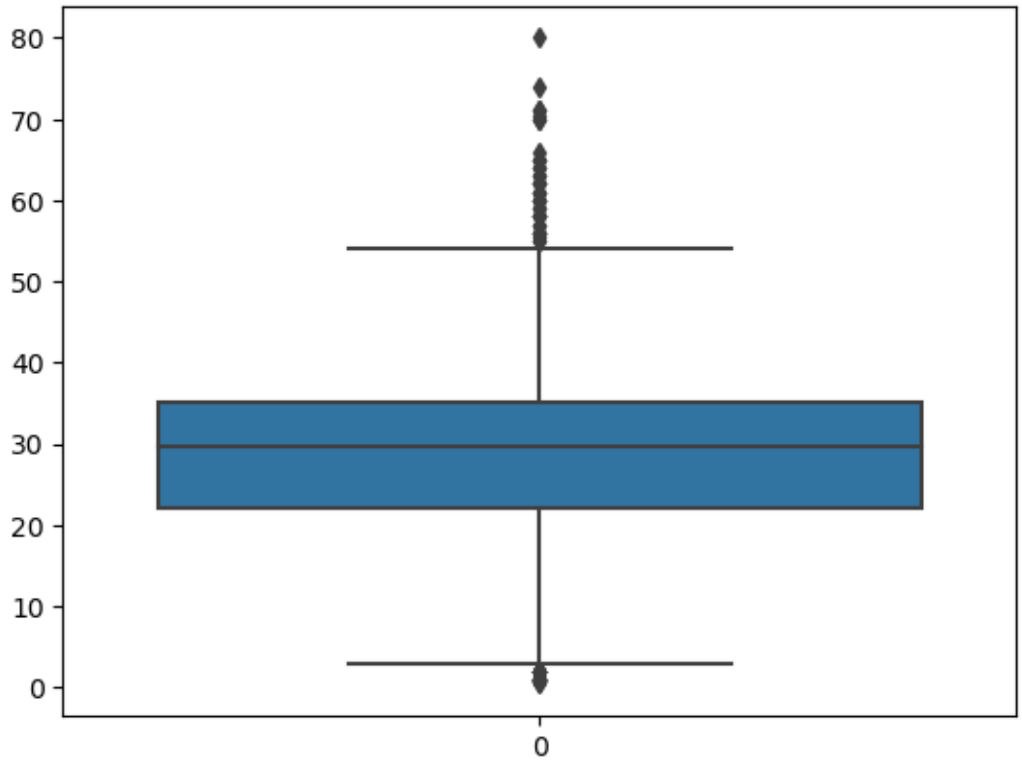
[31]: <Axes: >



1.0.1 Outlier Detection

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

[32]: <Axes: >



[33]:
```
q1=df.Age.quantile(0.25)
q3=df.Age.quantile(0.75)
q2=df.Age.quantile(0.50)
```

[34]:
```
q1
```

[34]: 22.0

[35]:
```
q2
```

[35]: 29.69911764705882

[36]:
```
q3
```

[36]: 35.0

[37]:
```
IQR=q3-q1
IQR
```

[37]: 13.0

[38]:
```
upper_limit=q3+1.5*IQR
lower_limit=q1-1.5*IQR
```

[39]:
```
upper_limit
```

[39]:
```
54.5
```

[40]:
```
lower_limit
```

[40]:
```
2.5
```

[41]:
```
df.median()
```

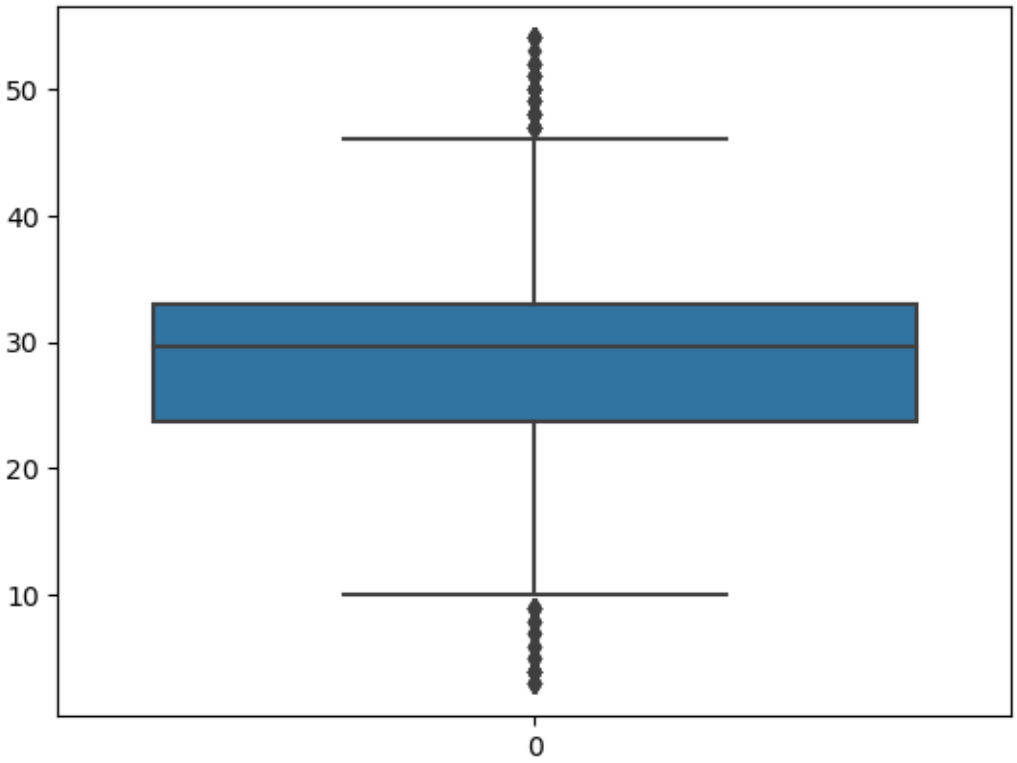
```
C:\Users\RUSHITHA REPAKULA\AppData\Local\Temp\ipykernel_13584\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  df.median()
```

[41]:
```
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.454200
dtype: float64
```

[42]:
```
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)]
```

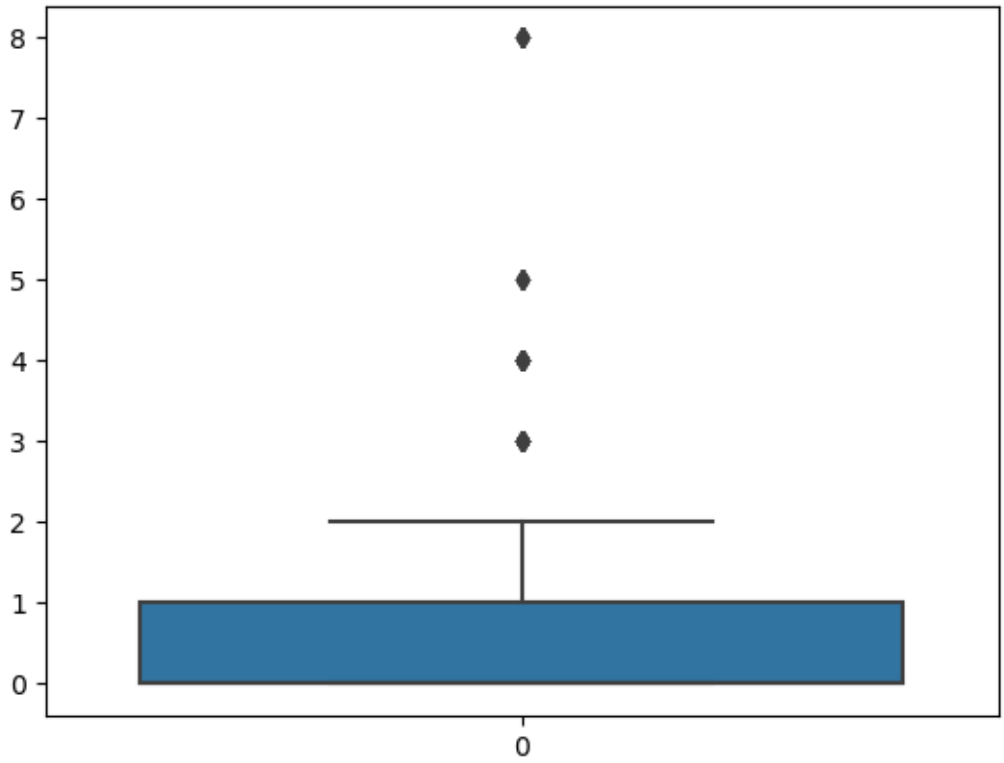
[43]:
```
sns.boxplot(df.Age)
```

[43]:
```
<Axes: >
```



[44]:
```
sns.boxplot(df.SibSp)
```

[44]: <Axes: >



[45]:
```
q1=df.SibSp.quantile(0.25)
q3=df.SibSp.quantile(0.75)
q2=df.SibSp.quantile(0.50)
```

[46]:
```
q1
```

[46]: 0.0

[47]:
```
q2
```

[47]: 0.0

[48]:
```
q3
```

[48]: 1.0

[49]:
```
IQR=q3-q1
IQR
```

[49]: 1.0

[50]:
```
upper_limit=q3+1.5*IQR
upper_limit
```

[50]:
```
2.5
```

[51]:
```
lower_limit=q1-1.5*IQR
lower_limit
```

[51]:
```
-1.5
```

[52]:
```
df.median()
```

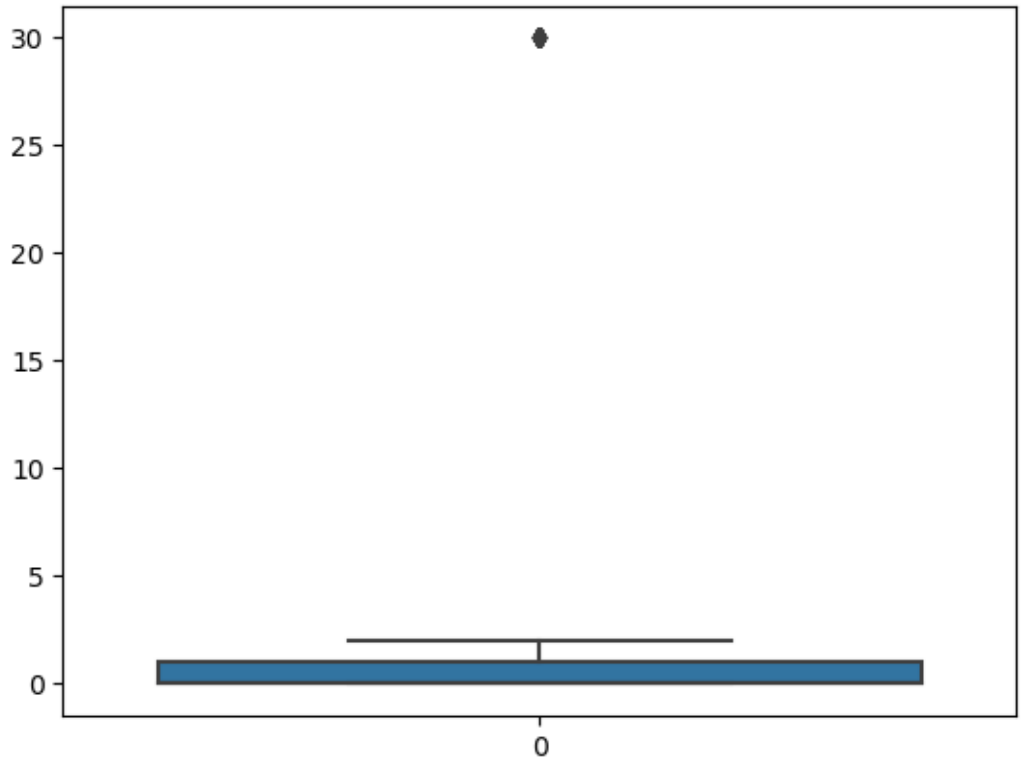
```
C:\Users\RUSHITHA REPAKULA\AppData\Local\Temp\ipykernel_13584\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  df.median()
```

[52]:
```
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.454200
dtype: float64
```

[53]:
```
df['SibSp']=np.where(df['SibSp']>upper_limit,30,df['SibSp'])
```

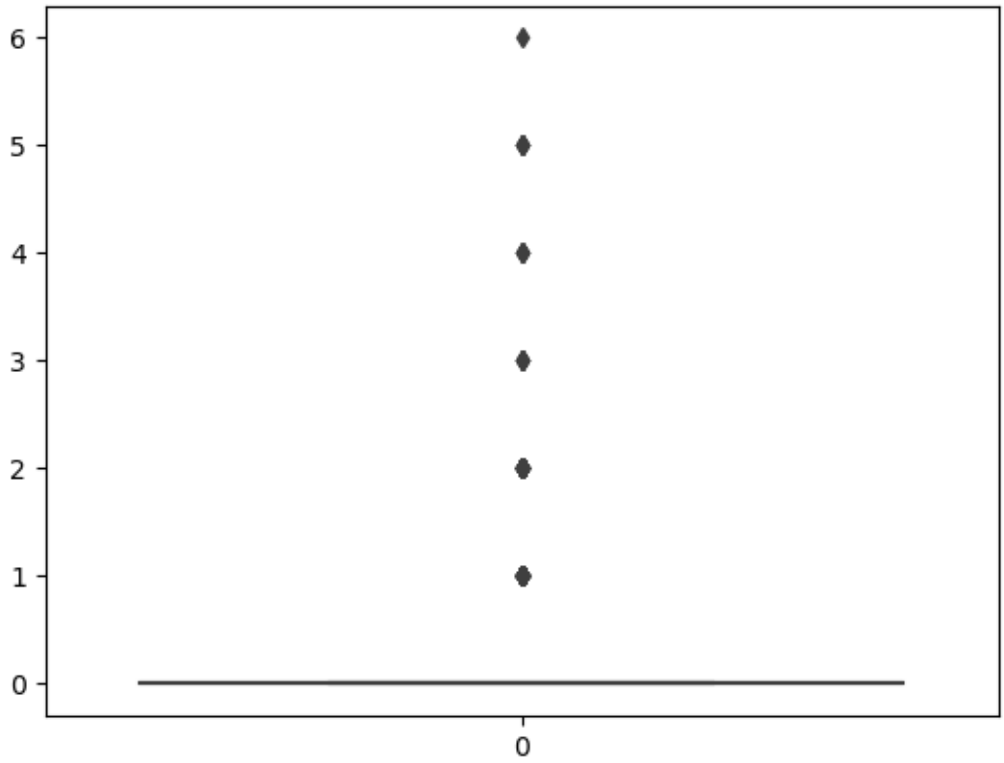
[54]:
```
sns.boxplot(df.SibSp)
```

[54]:
```
<Axes: >
```



[55]:
```
sns.boxplot(df.Parch)
```

[55]: <Axes: >



[56]:
```
q1=df.Parch.quantile(0.25)
q3=df.Parch.quantile(0.75)
q2=df.Parch.quantile(0.50)
```

[57]:
```
q1
```

[57]: 0.0

[58]:
```
q2
```

[58]: 0.0

[59]:
```
q3
```

[59]: 0.0

[60]:
```
IQR=q3-q1
IQR
```

[60]: 0.0

[61]:
```
upper_limit=q3+1.5*IQR
upper_limit
```

[61]: 0.0

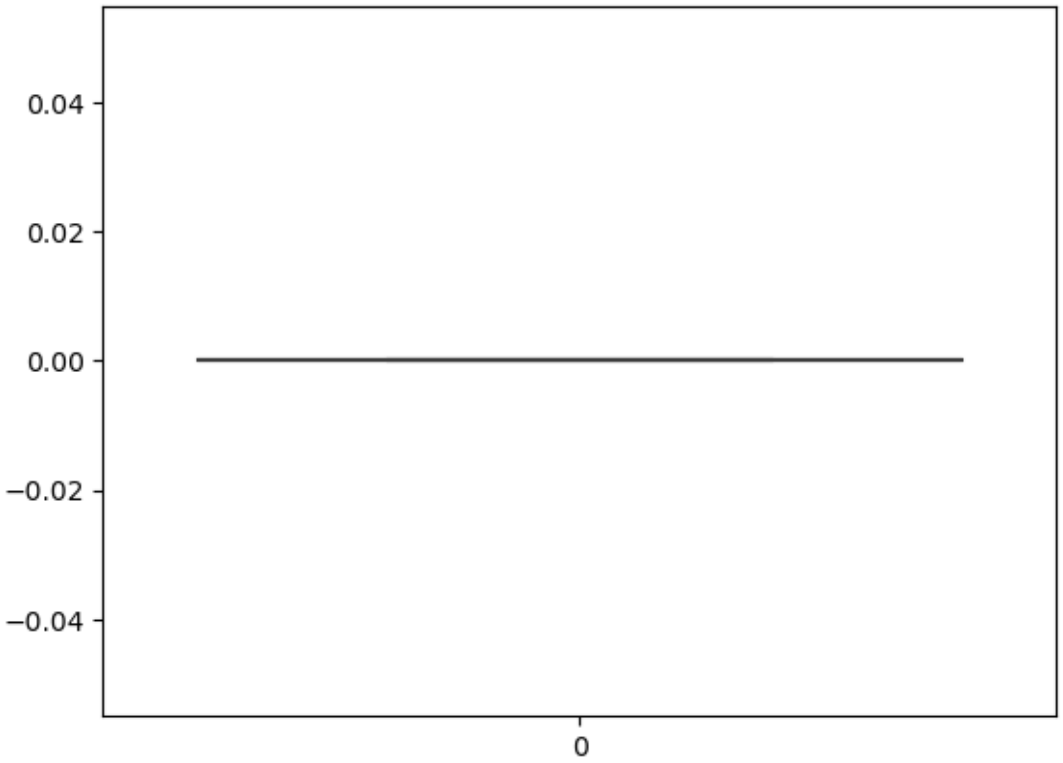
[62]:
```
lower_limit=q1-1.5*IQR
lower_limit
```

[62]: 0.0

[63]:
```
df['Parch']=np.where(df['Parch']>upper_limit,0,df['Parch'])
```

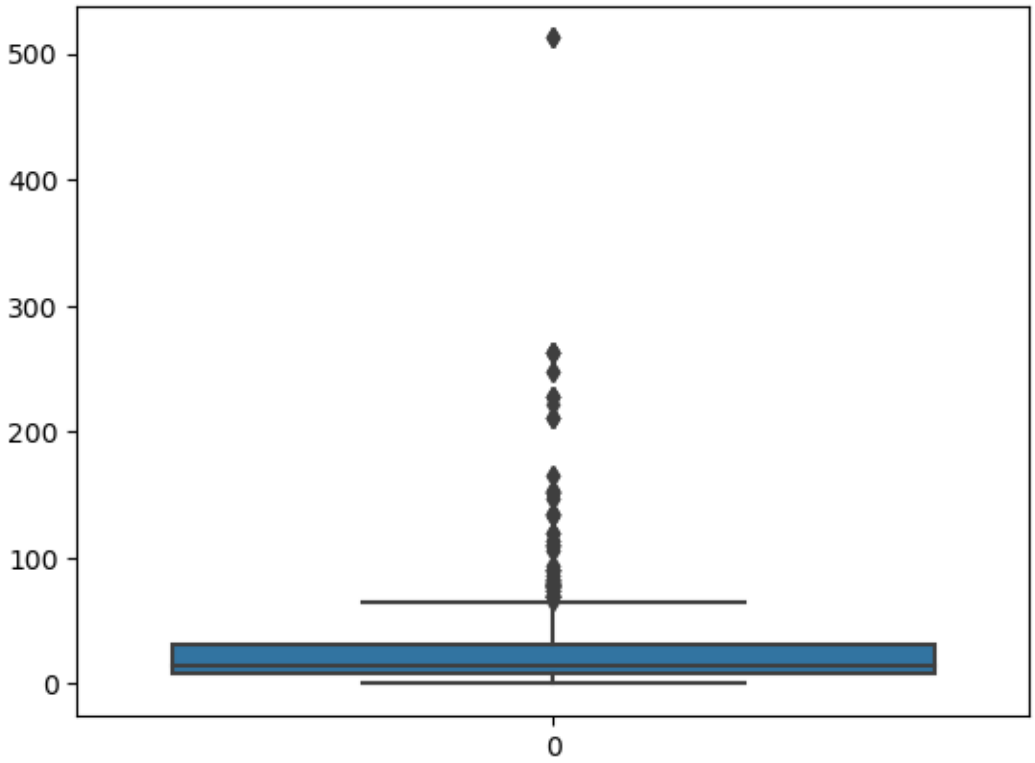
[64]:
```
sns.boxplot(df.Parch)
```

[64]: <Axes: >



[65]:
```
sns.boxplot(df.Fare)
```

[65]: <Axes: >



[66]:
```
q1=df.Fare.quantile(0.25)
q3=df.Fare.quantile(0.75)
q2=df.Fare.quantile(0.50)
```

[67]:
```
q1
```

[67]: 7.9104

[68]:
```
q2
```

[68]: 14.4542

[69]:
```
q3
```

[69]: 31.0

[70]:
```
IQR=q3-q1
IQR
```

[70]: 23.0896

[71]:
```
upper_limit=q3+1.5*IQR
upper_limit
```

[71]:
```
65.6344
```

[72]:
```
lower_limit=q1-1.5*IQR
lower_limit
```

[72]:
```
-26.724
```

[73]:
```
df.median()
```

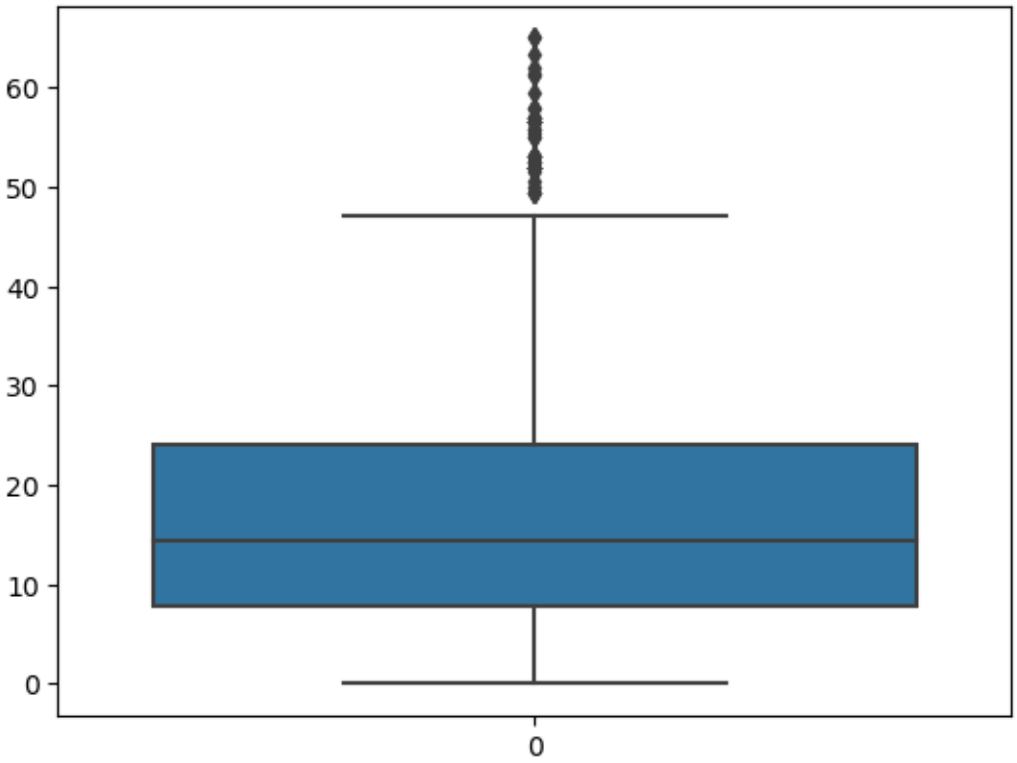
```
C:\Users\RUSHITHA REPAKULA\AppData\Local\Temp\ipykernel_13584\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median is
deprecated. In a future version, it will default to False. In addition,
specifying 'numeric_only=None' is deprecated. Select only valid columns or
specify the value of numeric_only to silence this warning.
  df.median()
```

[73]:
```
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.454200
dtype: float64
```

[74]:
```
df['Fare']=np.where(df['Fare']>upper_limit,14.45,df['Fare'])
```

[75]:
```
sns.boxplot(df.Fare)
```

[75]:
```
<Axes: >
```



1.0.2 Splitting Dependent and Independent variables

[76]:
```
df.head(10)
```

[76]:

	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  30.000000
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  53.1000
4      0      0            373450   8.0500
5      0      0            330877   8.4583
6      0      0             17463  51.8625
7     30      0            349909  21.0750
8      0      0            347742  11.1333
9      1      0            237736  30.0708
```

[77]:
```
x=df.iloc[:,2:]
y=df.iloc[:,1:2]
```

[78]:
```
x
```

[78]:
```
     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  53.100
4    35.000000      0      0            373450   8.050
..         …    …    …                 …     …
886  27.000000      0      0            211536  13.000
887  19.000000      0      0            112053  30.000
888  29.699118      1      0        W./C. 6607  23.450
889  26.000000      0      0            111369  30.000
```

```
890  32.000000      0      0            370376   7.750

[891 rows x 8 columns]
```

[79]:
```
y
```

[79]:
```
     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]
```

[80]:
```
x.shape
```

[80]:
```
(891, 8)
```

1.0.3 Perform Encoding

[81]:
```
from sklearn.preprocessing import LabelEncoder
```

[82]:
```
le=LabelEncoder()
```

[83]:
```
x["Name"]=le.fit_transform(x["Name"])
```

[84]:
```
x.head()
```

[84]:

	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	53.100
4	3	15	male	35.0	0	0	373450	8.050

[85]:
```
x["Name"].value_counts()
```

[85]:
```
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
```

[86]:
```
x["Sex"]=le.fit_transform(x["Sex"])
```

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

[87]:

	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	53.100
4	3	15	1	35.0	0	0	373450	8.050

[88]:
```
x["Sex"].value_counts()
```

[88]:
```
1    577
0    314
Name: Sex, dtype: int64
```

[89]:
```
x["Ticket"]=le.fit_transform(x["Ticket"])
```

[90]:
```
x.head()
```

[90]:

	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	53.100
4	3	15	1	35.0	0	0	472	8.050

[91]:
```
x["Ticket"].value_counts()
```

[91]:
```
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
```

1.0.4 Feature Scaling.

[92]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
```

[93]:
```
x_scaled=sc.fit_transform(x)
x_scaled
```

[93]:
```
array([[ 0.82737724, -1.31021659,  0.73769513, …,  0.        ,
         0.91896631, -0.79750279],
       [-1.56610693, -0.99141018, -1.35557354, …,  0.        ,
         1.28262456, -0.23084165],
       [ 0.82737724, -0.35768524, -1.35557354, …,  0.        ,
         1.64628282, -0.7443783 ],
       …,
       [ 0.82737724, -0.12441226, -1.35557354, …,  0.        ,
         1.67617254,  0.47748476],
       [-1.56610693, -1.41518943,  0.73769513, …,  0.        ,
        -1.64656796,  0.99298899],
       [ 0.82737724, -0.87477369,  0.73769513, …,  0.        ,
         0.63501397, -0.75815132]])
```

1.0.5 Splitting Data into Train and Test

[105]:
```
from sklearn.model_selection import train_test_split
```

[106]:
```
tts=train_test_split
```

[107]:
```
x_train,x_test,y_train,y_test=tts(x_scaled,y,test_size=0.2,random_state=0)
```

[108]:
```
print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

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

[109]:
```
x_train
```

[109]:
```
array([[ 8.27377244e-01, -1.34520754e+00, -1.35557354e+00, …,
         0.00000000e+00, -6.75152074e-01, -1.68209858e-01],
       [-3.69364841e-01,  7.77576610e-03,  7.37695132e-01, …,
         0.00000000e+00,  1.03852519e+00, -5.41718247e-01],
       [-3.69364841e-01,  2.29385100e-01,  7.37695132e-01, …,
         0.00000000e+00,  1.39222020e+00,  1.54424009e+00],
       …,
       [ 8.27377244e-01,  6.10397639e-01,  7.37695132e-01, …,
```

```
          0.00000000e+00, -2.61677619e-01, -7.59465657e-01],
        [ 8.27377244e-01,  1.71066854e+00, -1.35557354e+00, …,
          0.00000000e+00, -1.91934939e-01,  1.33200522e-03],
        [-3.69364841e-01, -1.29466506e+00,  7.37695132e-01, …,
          0.00000000e+00, -4.90832136e-01,  1.70131540e+00]])
```

[110]:
```
x_test
```

[110]:
```
array([[ 0.82737724,  1.69122913,  0.73769513, …,  0.        ,
        -0.80965581, -0.23018842],
       [ 0.82737724,  1.63291088,  0.73769513, …,  0.        ,
         1.40218344, -0.77389191],
       [ 0.82737724,  0.9175404 ,  0.73769513, …,  0.        ,
         0.70475665,  0.92412392],
       …,
       [-1.56610693,  0.53263998, -1.35557354, …,  0.        ,
         0.38593297, -0.23084165],
       [ 0.82737724, -1.53960169,  0.73769513, …,  0.        ,
         0.0172931 , -0.74995047],
       [ 0.82737724, -1.43851673,  0.73769513, …,  0.        ,
        -0.32643868, -0.73454044]])
```

[111]:
```
y_train
```

[111]:
```
     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]
```

[112]:
```
y_test
```

[112]:
```
     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]
```

[ ]: