In [1]:

```
print("Name: RAMISETTY PAVANI")
print("Registration Number: 21BCE9521")
print("Morning Batch")
```

```
Name: RAMISETTY PAVANI
Registration Number: 21BCE9521
Morning Batch
```

Importing the necessary libraries

In [2]:

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

Importing the DataSet

In [3]:

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

```
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
```

In [5]:

```
df.shape
```

Out[5]:

```
(891, 12)
```

In [6]:

```
df.describe()
```

Out[6]:

	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

Checking for null values

In [7]:

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

Out[7]:

```
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 [8]:

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

Out[8]:

```
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 [9]:

```
df.corr()
```

Out[9]:

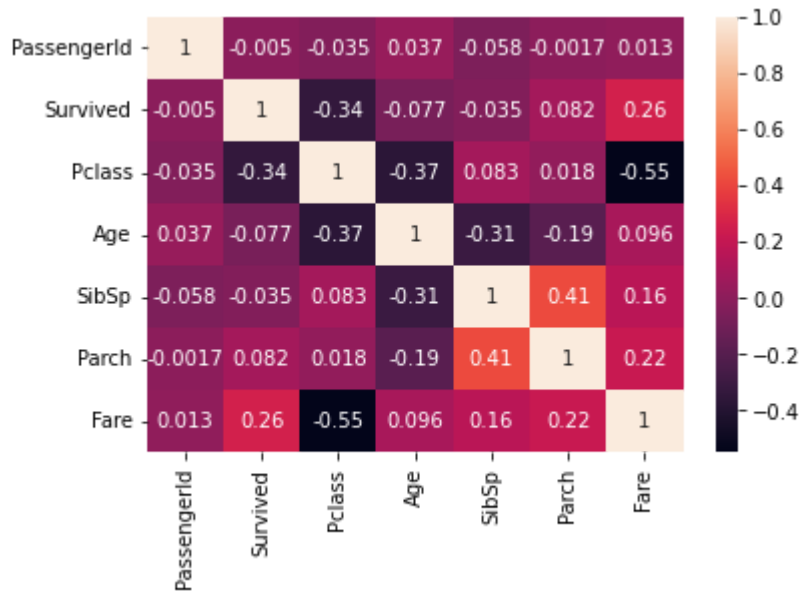
	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

In [10]:

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

Out[10]:

<AxesSubplot:>



Handling null values

In [11]:

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

In [12]:

```
df.head()
```

Out[12]:

	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 [13]:

```
print(df["Embarked"].mode())
```

```
0    S
dtype: object
```

In [14]:

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

In [15]:

```
df.drop(columns="Cabin", axis=1, inplace=True)
```

In [16]:

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

Out[16]:

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

In [17]:

```
df.head()
```

Out[17]:

	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

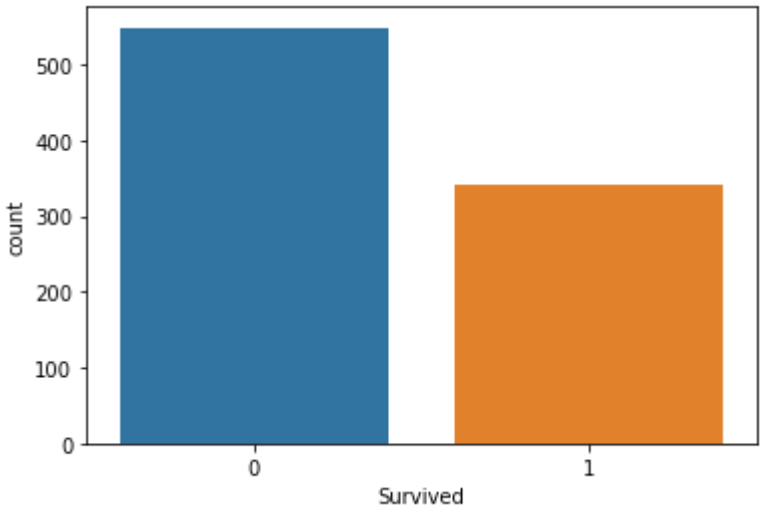
Data Visualization

In [18]:

```
sns.countplot(x="Survived", data=df)
```

Out[18]:

```
<AxesSubplot:xlabel='Survived', ylabel='count'>
```



In [19]:

```
df["Survived"].value_counts()
```

Out[19]:

```
0    549
1    342
Name: Survived, dtype: int64
```

In [20]:

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

Out[20]:

```
male      577
female    314
Name: Sex, dtype: int64
```

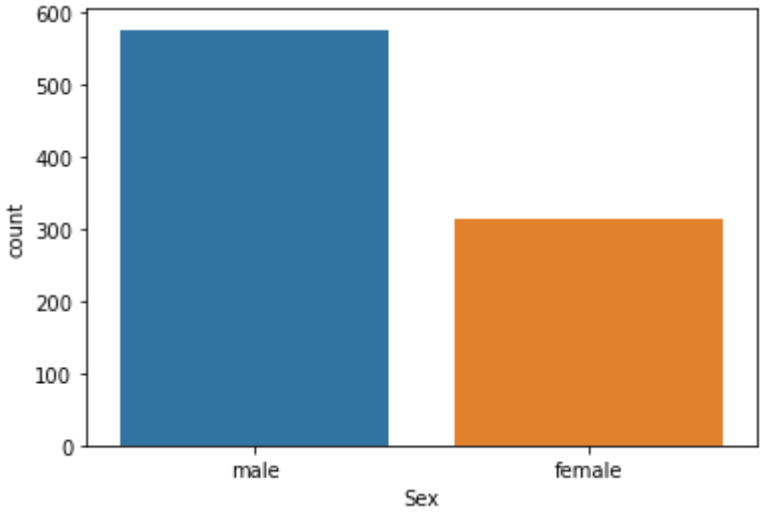
In [21]:

```
sns.countplot(df["Sex"], data=df)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[21]:

```
<AxesSubplot:xlabel='Sex', ylabel='count'>
```



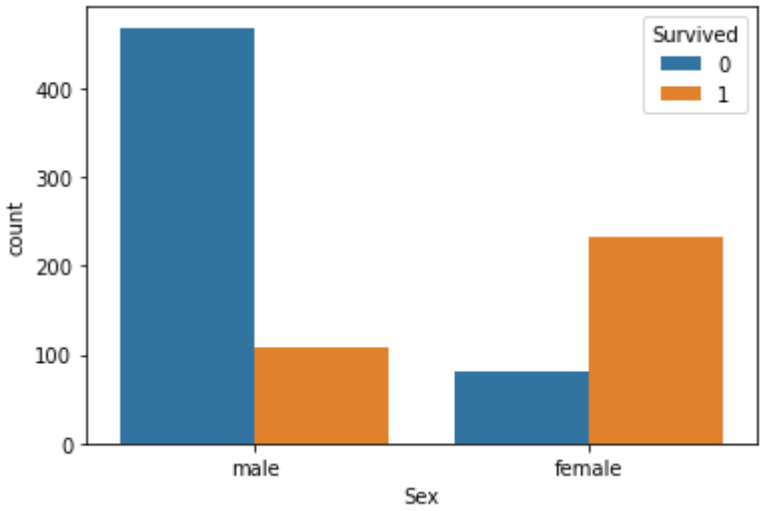
In [22]:

```
sns.countplot(df["Sex"], hue="Survived", data=df)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[22]:

```
<AxesSubplot:xlabel='Sex', ylabel='count'>
```



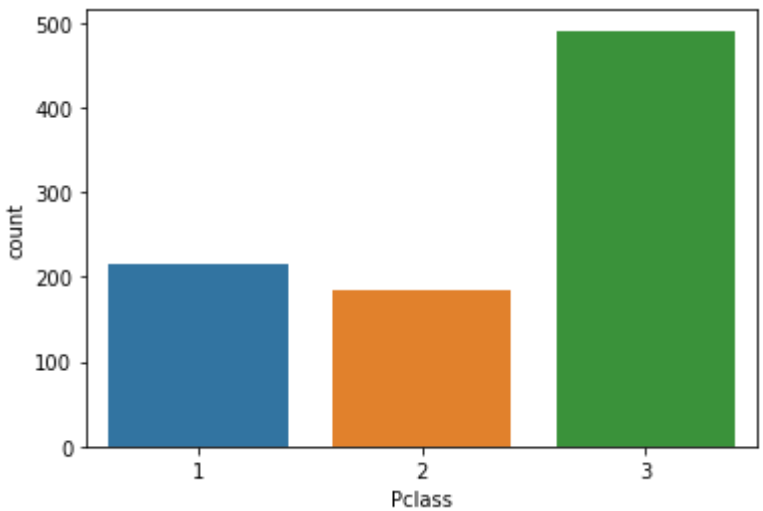
In [23]:

```
sns.countplot(df["Pclass"], data=df)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[23]:

```
<AxesSubplot:xlabel='Pclass', ylabel='count'>
```



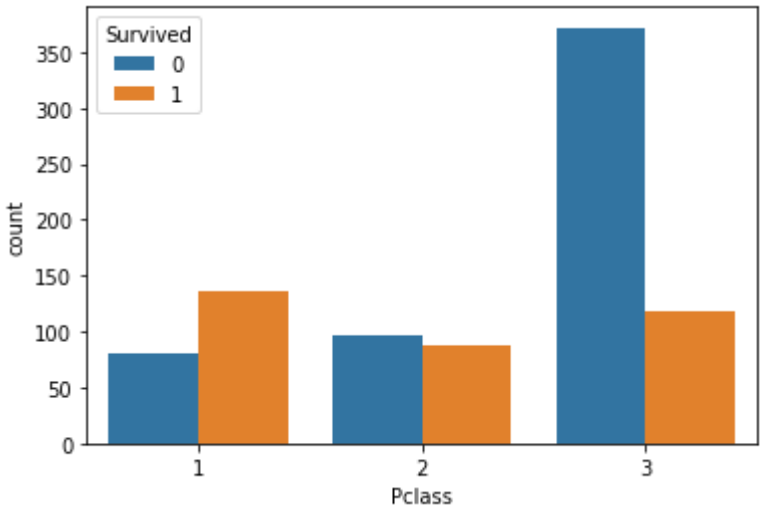
In [24]:

```
sns.countplot(df["Pclass"], hue="Survived", data=df)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[24]:

```
<AxesSubplot:xlabel='Pclass', ylabel='count'>
```

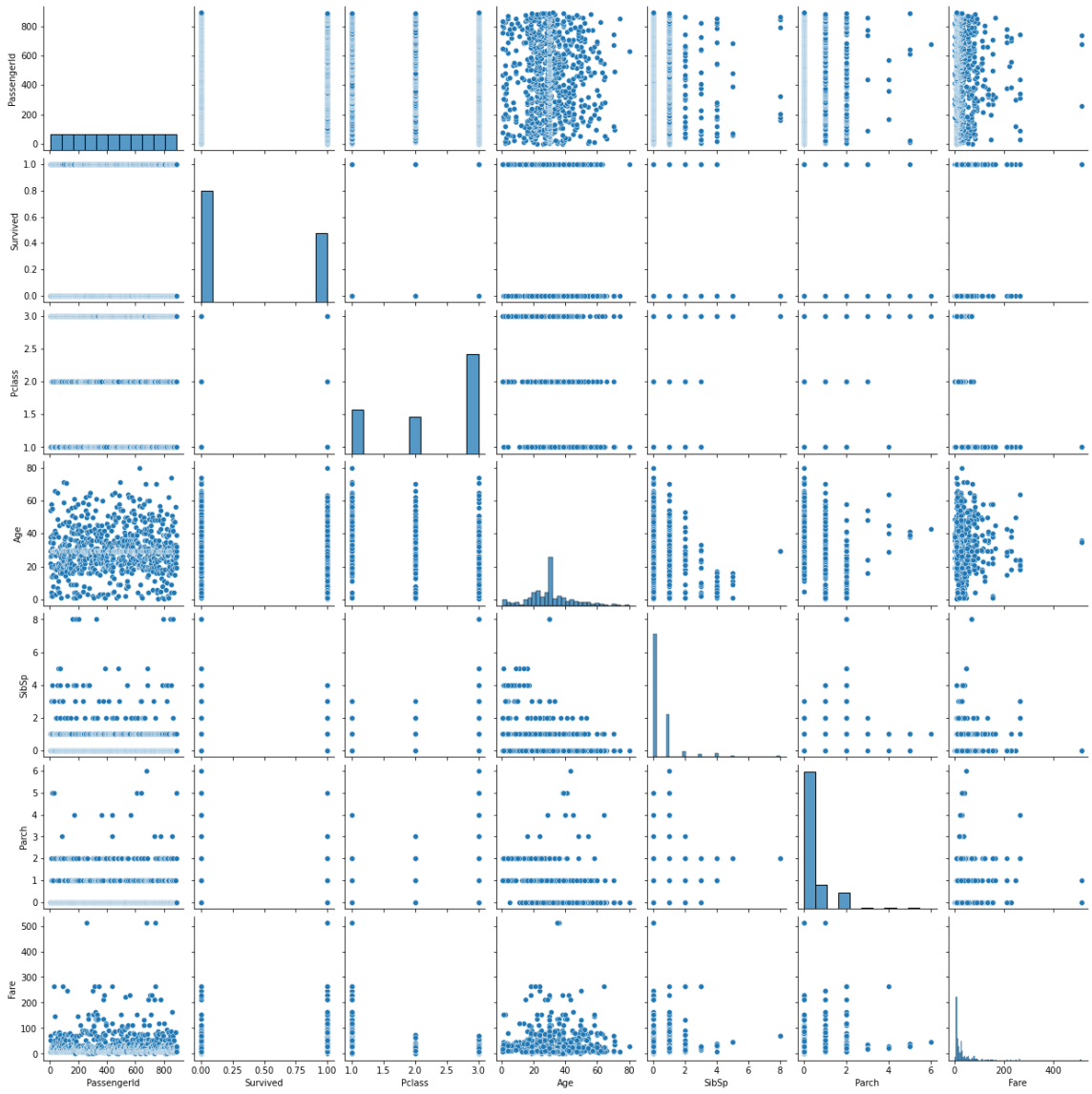


In [25]:

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

Out[25]:

```
<seaborn.axisgrid.PairGrid at 0x25f1dfdcc40>
```



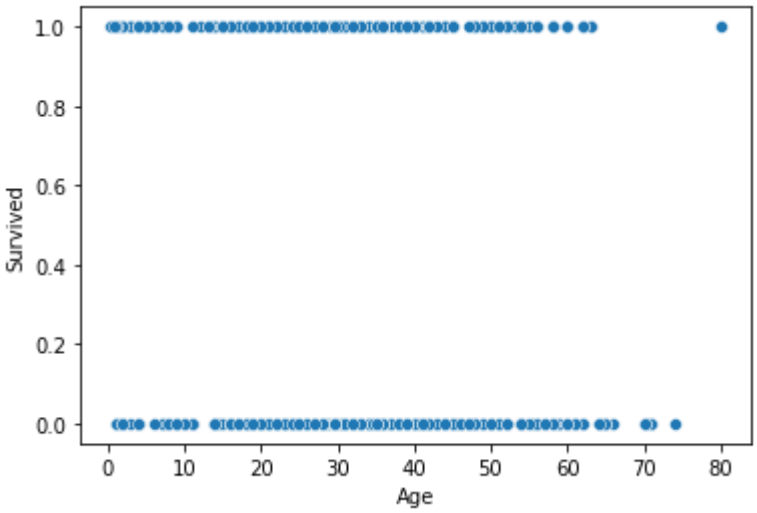
In [26]:

```
sns.scatterplot(df["Age"], df["Survived"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing oth
er arguments without an explicit keyword will result in an error or misint
erpretation.
  warnings.warn(
```

Out[26]:

```
<AxesSubplot:xlabel='Age', ylabel='Survived'>
```



Outlier Detection

In [27]:

```
df.head()
```

Out[27]:

	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

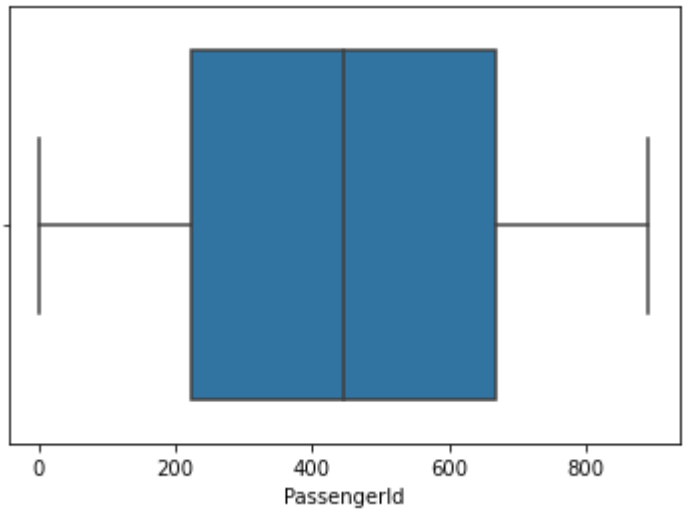
In [28]:

```
sns.boxplot(df.PassengerId)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[28]:

```
<AxesSubplot:xlabel='PassengerId'>
```



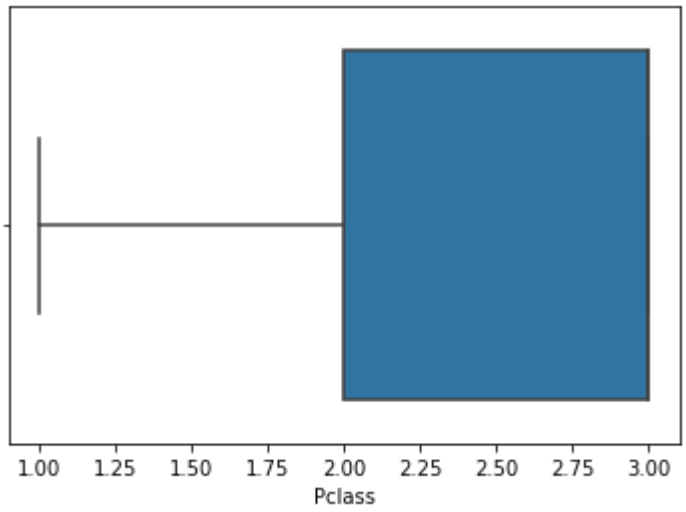
In [29]:

```
sns.boxplot(df.Pclass)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[29]:

```
<AxesSubplot:xlabel='Pclass'>
```



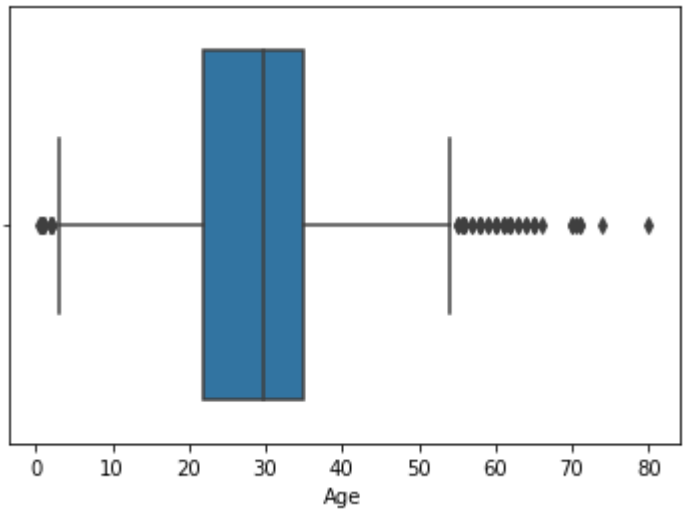
In [30]:

```
sns.boxplot(df.Age)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[30]:

```
<AxesSubplot:xlabel='Age'>
```



In [31]:

```
q1=df.Age.quantile(0.25)
q3=df.Age.quantile(0.75)
```

In [32]:

```
IQR=q3-q1
```

In [33]:

```
upper_limit=q3+1.5*IQR
lower_limit=q1-1.5*IQR
```

In [34]:

```
df["Age"]=np.where(df["Age"]>upper_limit,upper_limit, np.where(df["Age"]<lower_limit,low
```

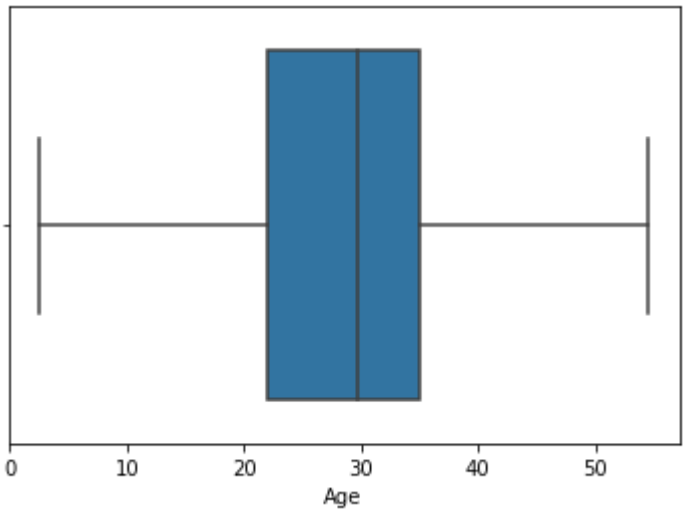
In [35]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[35]:

```
<AxesSubplot:xlabel='Age'>
```



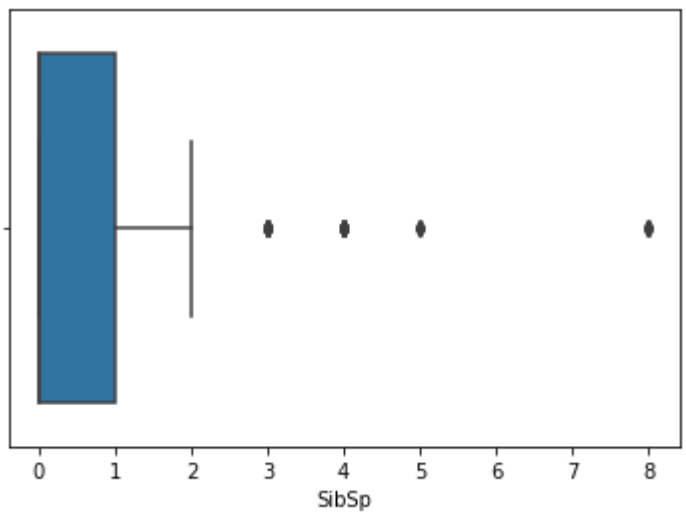
In [36]:

```
sns.boxplot(df.SibSp)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[36]:

```
<AxesSubplot:xlabel='SibSp'>
```



In [37]:

```
q1=df.SibSp.quantile(0.25)
q3=df.SibSp.quantile(0.75)
```

In [38]:

```
IQR=q3-q1
```

In [39]:

```
upper_limit=q3+1.5*IQR
```

In [40]:

```
df["SibSp"]=np.where(df["SibSp"]>upper_limit,upper_limit,df["SibSp"])
```

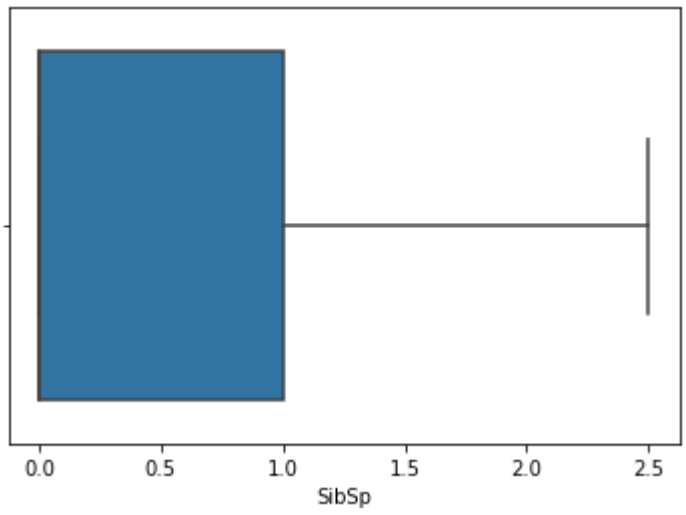
In [41]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[41]:

```
<AxesSubplot:xlabel='SibSp'>
```



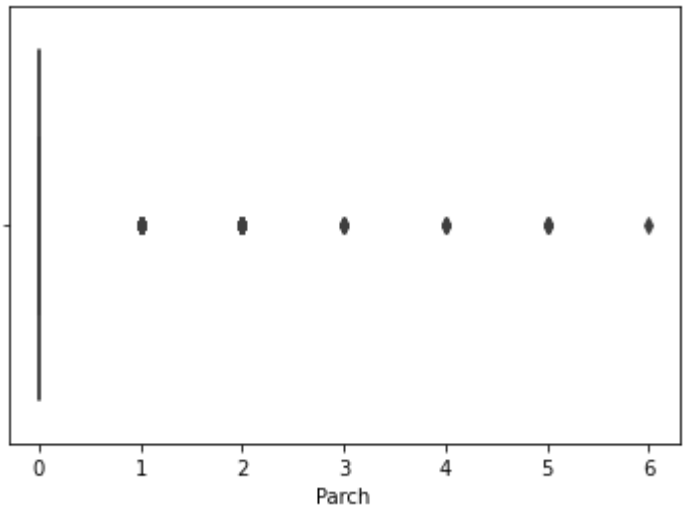
In [42]:

```
sns.boxplot(df.Parch)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[42]:

```
<AxesSubplot:xlabel='Parch'>
```



In [43]:

```
q1=df.Parch.quantile(0.25)
q3=df.Parch.quantile(0.75)
```

In [44]:

```
IQR=q3-q1
```

In [45]:

```
upper_limit=q3+1.5*IQR
```

In [46]:

```
df["Parch"]=np.where(df["Parch"]>upper_limit,upper_limit,df["Parch"])
```

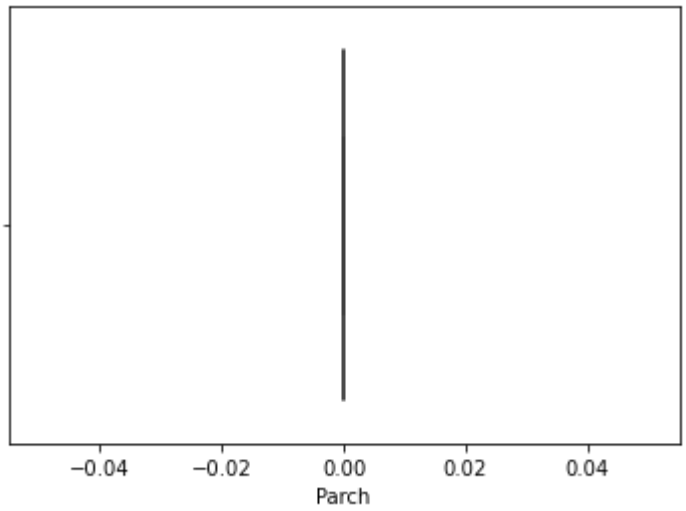
In [47]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[47]:

```
<AxesSubplot:xlabel='Parch'>
```



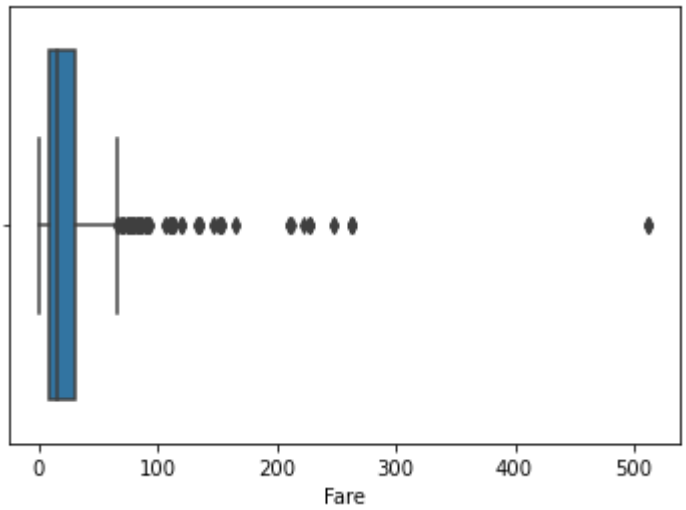
In [48]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[48]:

```
<AxesSubplot:xlabel='Fare'>
```



In [49]:

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

In [50]:

```
print(q1)
print(q3)
```

```
7.9104
31.0
```

In [51]:

```
IQR=q3-q1
IQR
```

Out[51]:

```
23.0896
```

In [52]:

```
upper_limit=q3+1.5*IQR
upper_limit
```

Out[52]:

```
65.6344
```

In [53]:

```
df["Fare"]=np.where(df["Fare"]>upper_limit,upper_limit,df["Fare"])
```

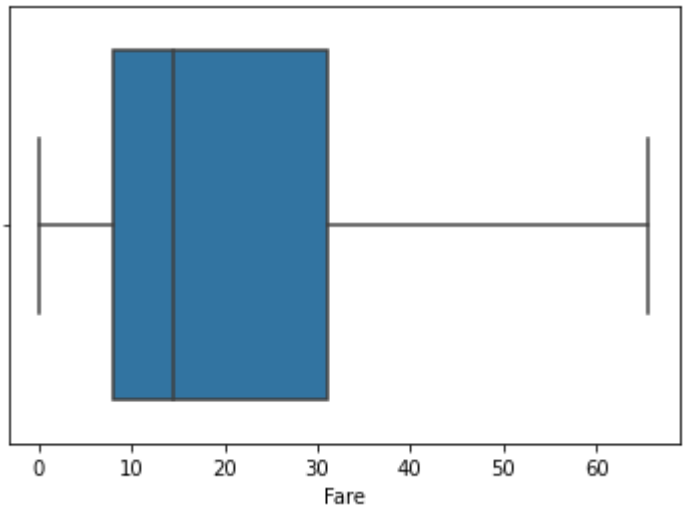
In [54]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[54]:

```
<AxesSubplot:xlabel='Fare'>
```



Splitting Dependent and Independent variables

In [55]:

```
df.head()
```

Out[55]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1.0	0.0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1.0	0.0	PC 17599	65.6344
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0.0	0.0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1.0	0.0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0.0	0.0	373450	8.0500

In [56]:

```
x=df.drop(columns=["PassengerId", "Name", "Ticket", "Survived"])
y=df["Survived"]
```

In [57]:

```
x.head()
```

Out[57]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	male	22.0	1.0	0.0	7.2500	S
1	1	female	38.0	1.0	0.0	65.6344	C
2	3	female	26.0	0.0	0.0	7.9250	S
3	1	female	35.0	1.0	0.0	53.1000	S
4	3	male	35.0	0.0	0.0	8.0500	S

In [58]:

```
y.head()
```

Out[58]:

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

In [59]:

```
print(x.shape)
print(y.shape)
```

```
(891, 7)
(891,)
```

In [60]:

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

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

Perform Encoding

In [61]:

```
x.head()
```

Out[61]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	male	22.0	1.0	0.0	7.2500	S
1	1	female	38.0	1.0	0.0	65.6344	C
2	3	female	26.0	0.0	0.0	7.9250	S
3	1	female	35.0	1.0	0.0	53.1000	S
4	3	male	35.0	0.0	0.0	8.0500	S

In [62]:

```
Sex=pd.get_dummies(x["Sex"], drop_first=True)
Embarked=pd.get_dummies(x["Embarked"], drop_first=True)
```

In [63]:

```
x.drop(["Sex", "Embarked"], axis=1, inplace=True)
x.head()
```

Out[63]:

	Pclass	Age	SibSp	Parch	Fare
0	3	22.0	1.0	0.0	7.2500
1	1	38.0	1.0	0.0	65.6344
2	3	26.0	0.0	0.0	7.9250
3	1	35.0	1.0	0.0	53.1000
4	3	35.0	0.0	0.0	8.0500

In [64]:

```
x=pd.concat([x,Sex,Embarked], axis=1)
x.head()
```

Out[64]:

	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	3	22.0	1.0	0.0	7.2500	1	0	1
1	1	38.0	1.0	0.0	65.6344	0	0	0
2	3	26.0	0.0	0.0	7.9250	0	0	1
3	1	35.0	1.0	0.0	53.1000	0	0	1
4	3	35.0	0.0	0.0	8.0500	1	0	1

In [65]:

```
x.shape
```

Out[65]:

```
(891, 8)
```

Feature Scaling

In [66]:

```
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
```

In [67]:

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

Out[67]:

	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	1.0	0.375000	0.4	0.0	0.110460	1.0	0.0	1.0
1	0.0	0.682692	0.4	0.0	1.000000	0.0	0.0	0.0
2	1.0	0.451923	0.0	0.0	0.120745	0.0	0.0	1.0
3	0.0	0.625000	0.4	0.0	0.809027	0.0	0.0	1.0
4	1.0	0.625000	0.0	0.0	0.122649	1.0	0.0	1.0

Splitting Data into Train and Test

In [68]:

```
from sklearn.model_selection import train_test_split
```

In [69]:

```
x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.2,random_state=0)
```

In [70]:

```
x_train.head()
```

Out[70]:

	Pclass	Age	SibSp	Parch	Fare	male	Q	S
140	1.0	0.523060	0.0	0.0	0.232284	0.0	0.0	0.0
439	0.5	0.548077	0.0	0.0	0.159977	1.0	0.0	1.0
817	0.5	0.548077	0.4	0.0	0.563793	1.0	0.0	0.0
378	1.0	0.336538	0.0	0.0	0.061134	1.0	0.0	0.0
491	1.0	0.355769	0.0	0.0	0.110460	1.0	0.0	1.0

In [71]:

```
x_test.head()
```

Out[71]:

	Pclass	Age	SibSp	Parch	Fare	male	Q	S
495	1.0	0.523060	0.0	0.0	0.220285	1.0	0.0	0.0
648	1.0	0.523060	0.0	0.0	0.115031	1.0	0.0	1.0
278	1.0	0.086538	1.0	0.0	0.443746	1.0	1.0	0.0
31	0.0	0.523060	0.4	0.0	1.000000	0.0	0.0	0.0
255	1.0	0.509615	0.0	0.0	0.232284	0.0	0.0	0.0

In [72]:

```
y_train.head()
```

Out[72]:

```
140    0
439    0
817    0
378    0
491    0
Name: Survived, dtype: int64
```

In [73]:

```
y_test.head()
```

Out[73]:

```
495    0
648    0
278    0
31     1
255    1
Name: Survived, dtype: int64
```

In [74]:

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

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