```
Import the necessary Libraries
```

In [1]:

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

```
Reading the Data Set
```

In [2]:

```
ds=pd.read_csv("Titanic_Dataset.csv")
```

In [3]:

```
ds
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN
...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN

891 rows × 12 columns

In [4]:

```
ds.head()
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Er
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	

In [5]:

```
ds.describe()
```

Out[5]:

	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

In [6]:

```
ds.shape
```

Out[6]:

```
(891, 12)
```

In [7]:

```
ds.info()
```

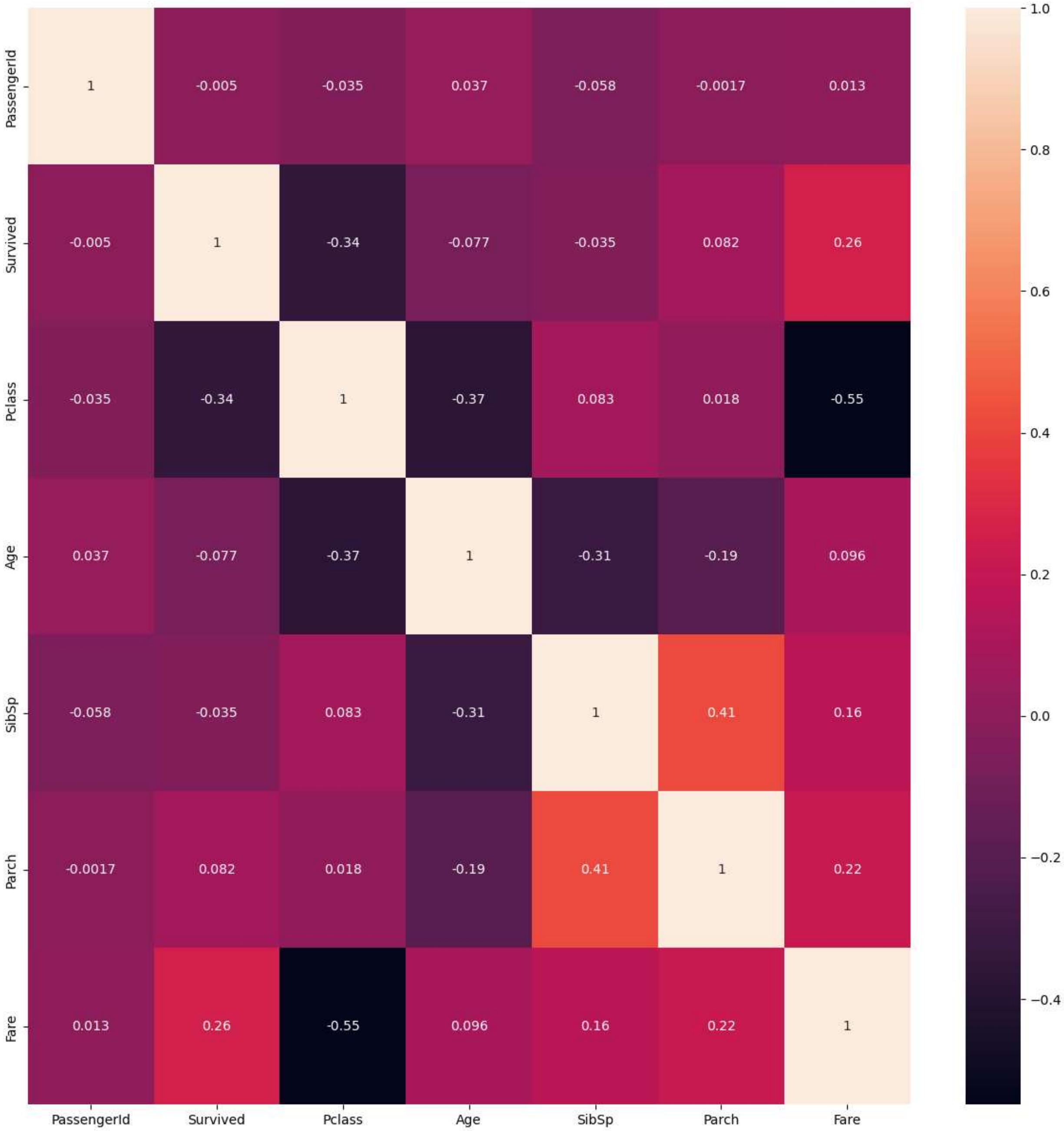
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [8]:

```
corr=ds.corr()
```

In [25]:

```
plt.subplots(figsize=(15,15))
sns.heatmap(corr,annot=True);
```



```
Checking the null values
```

In [10]:

```
ds.isnull().any()
```

Out[10]:

```
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 [11]:

```
ds.isna().sum()
```

Out[11]:

```
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 [12]:

```
ds.Embarked.value_counts()
```

Out[12]:

```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [13]:

```
ds.Age.mean()
```

Out[13]:

```
29.69911764705882
```

In [14]:

```
ds.Age.median()
```

Out[14]:

```
28.0
```

In [15]:

```
ds.Age.mode()
```

Out[15]:

```
0    24.0
Name: Age, dtype: float64
```

```
Filling the null values
```

In [16]:

```
ds.Age.fillna(ds.Age.median(),inplace=True)
```

In [17]:

```
ds.Age.isna().sum()
```

Out[17]:

```
0
```

In [18]:

```
ds.Embarked.mode()
```

Out[18]:

```
0    S
Name: Embarked, dtype: object
```

In [19]:

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

In [20]:

```
ds.Embarked.isna().sum()
```

Out[20]:

```
0
```

In [23]:

```
ds.drop(['Cabin'],axis=1,inplace=True)
```

In [24]:

```
ds.isnull().any()
```

Out[24]:

```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age            False
SibSp          False
Parch          False
Ticket         False
Fare           False
Embarked       False
dtype: bool
```

In [ ]:

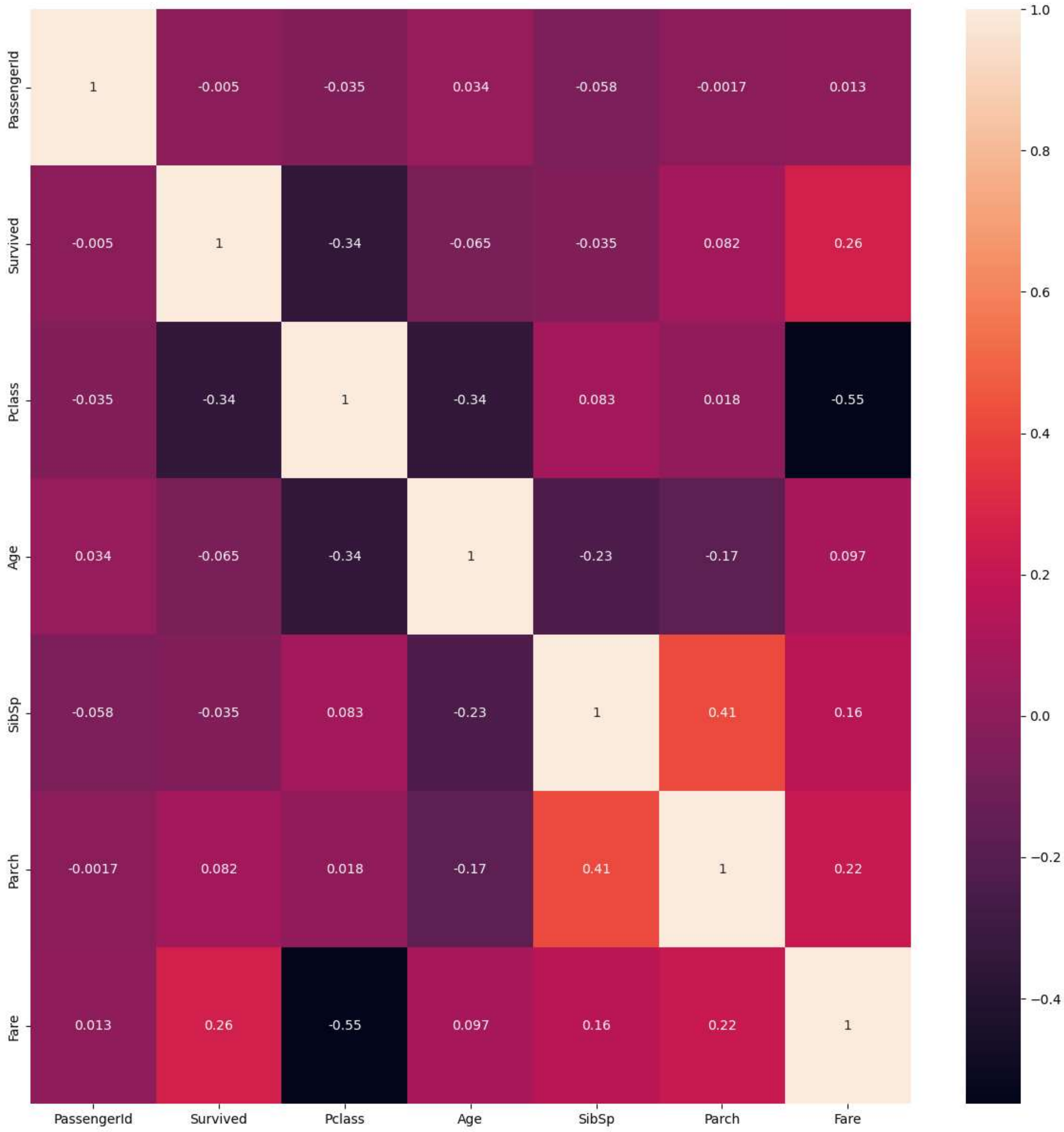
```
Data Visualization
```

In [26]:

```
corr=ds.corr()
```

In [27]:

```
plt.subplots(figsize=(15,15))
sns.heatmap(corr,annot=True);
```



```
Encoding
```

In [28]:

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

In [29]:

```
l=LabelEncoder()
```

In [30]:

```
ds["Sex"]=l.fit_transform(ds["Sex"])
```

In [33]:

```
ds.Sex
```

Out[33]:

```
0      1
1      0
2      0
3      0
4      1
      ..
886    1
887    0
888    0
889    1
890    1
Name: Sex, Length: 891, dtype: int32
```

In [34]:

```
ds.Sex.value_counts()
```

Out[34]:

```
1    577
0    314
Name: Sex, dtype: int64
```

In [37]:

```
ds.shape
```

Out[37]:

```
(891, 11)
```

In [39]:

```
emb=pd.get_dummies(ds["Embarked"],drop_first=True)
emb
```

Out[39]:

	Q	S
0	0	1
1	0	0
2	0	1
3	0	1
4	0	1
...	...	...
886	0	1
887	0	1
888	0	1
889	0	0
890	1	0

891 rows × 2 columns

In [40]:

```
ds=pd.concat([ds,emb],axis=1)
ds.head()
```

Out[40]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.1000	S
4	5	0	3	Allen, Mr. William Henry	1	35.0	0	0	373450	8.0500	S

In [41]:

```
ds.drop(['Embarked'],axis=1,inplace=True)
```

In [42]:

```
ds.head()
```

Out[42]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Q	S
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	0	0
2	3	1	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250	0	1
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.1000	0	1
4	5	0	3	Allen, Mr. William Henry	1	35.0	0	0	373450	8.0500	0	1

In [44]:

```
ds.shape
```

Out[44]:

```
(891, 12)
```

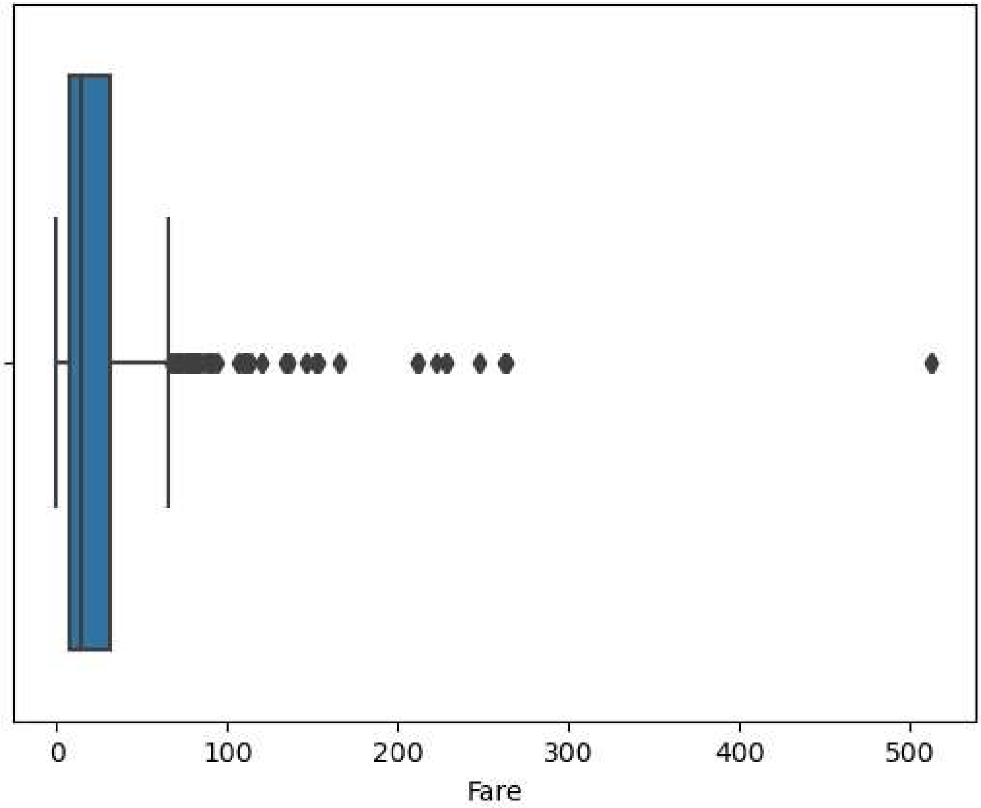
```
Detecting and removing the outliers
```

In [47]:

```
sns.boxplot(ds['Fare'],data=ds);

```

```
C:\Users\M DIVYA\DataScience\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: 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 ex
plicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [54]:

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

In [55]:

```
q1
```

Out[55]:

```
7.9104
```

In [56]:

```
q3
```

Out[56]:

```
31.0
```

In [58]:

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

Out[58]:

```
23.0896
```

In [59]:
```
upper_limit =q3+1.5*IQR
upper_limit
```

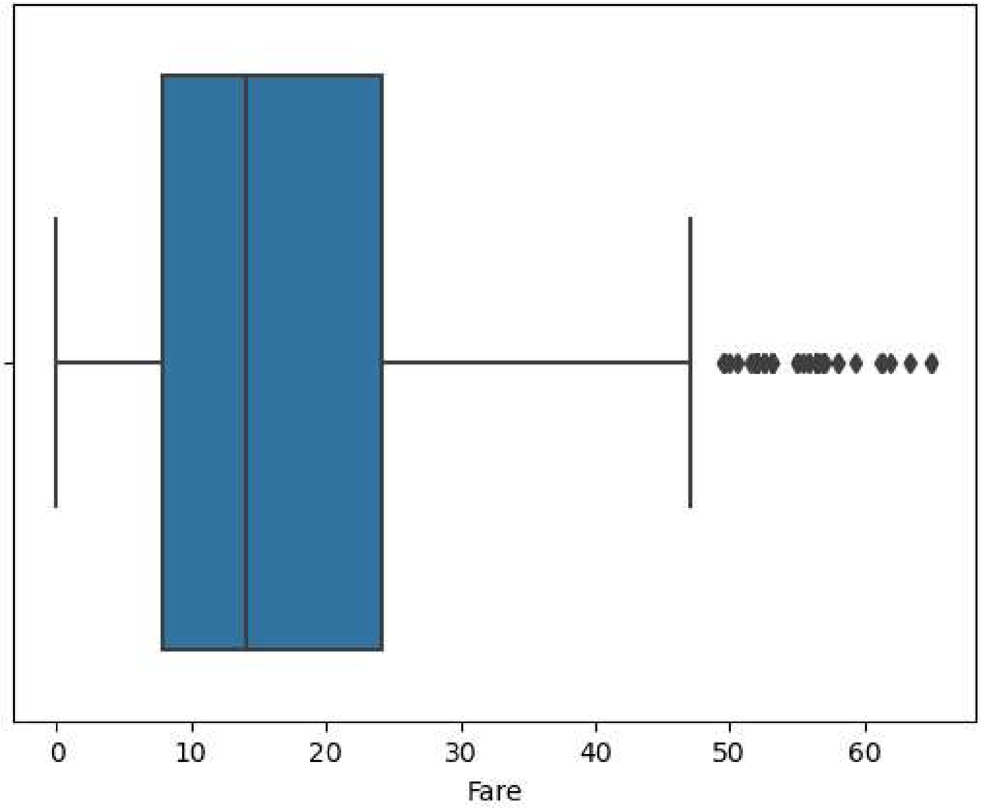
Out[59]:
```
65.6344
```

In [66]:
```
ds = ds[ds.Fare<upper_limit]
```

In [67]:
```
sns.boxplot(ds['Fare'],data=ds);

```

```
C:\Users\M DIVYA\DataScience\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: 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 ex
plicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [68]:
```
ds.shape
```

Out[68]:
```
(891, 12)
```

In [69]:
```
ds.head()
```

Out[69]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Q	S
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.250	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	14.000	0	0
2	3	1	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.925	0	1
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.100	0	1
4	5	0	3	Allen, Mr. William Henry	1	35.0	0	0	373450	8.050	0	1

```
Seperating Dependent and Independent Variables
```

In [71]:
```
x=ds.loc[:,['Age','SibSp','Parch','Fare']]
y=ds.iloc[:,1:2]
```

In [72]:
```
x.head()
```

Out[72]:

	Age	SibSp	Parch	Fare
0	22.0	1	0	7.250
1	38.0	1	0	14.000
2	26.0	0	0	7.925
3	35.0	1	0	53.100
4	35.0	0	0	8.050

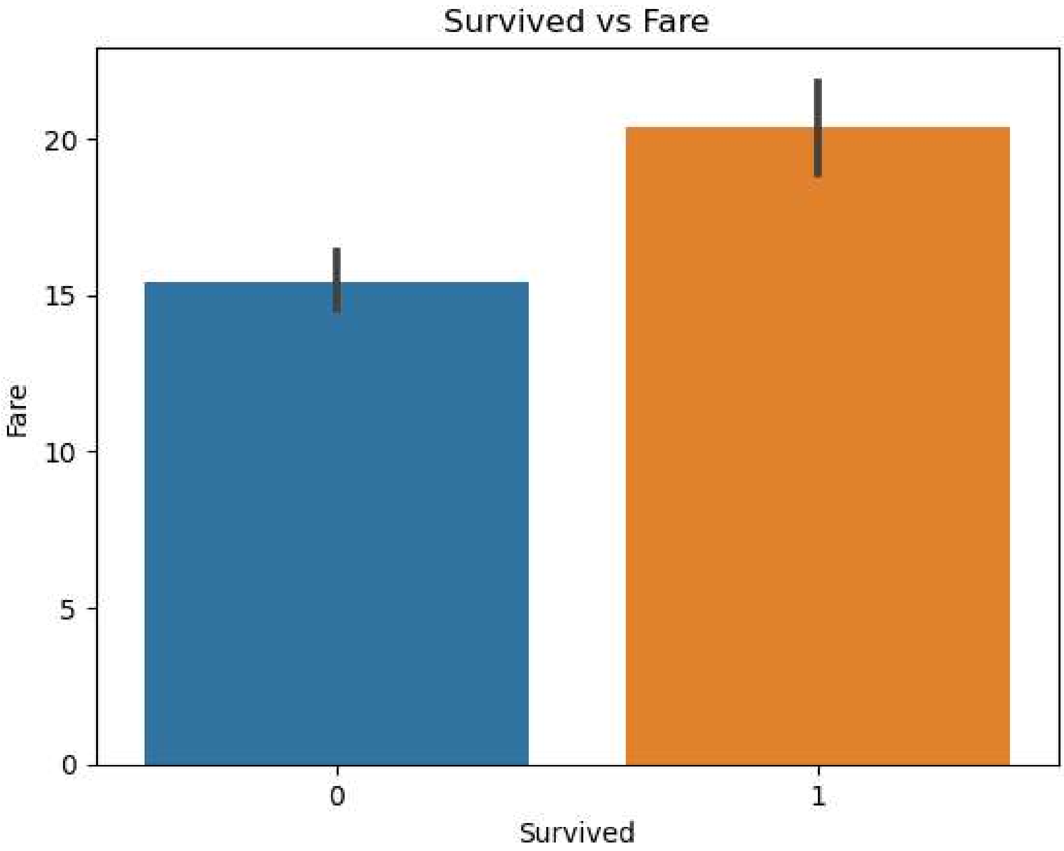
In [73]:
```
y.head()
```

Out[73]:

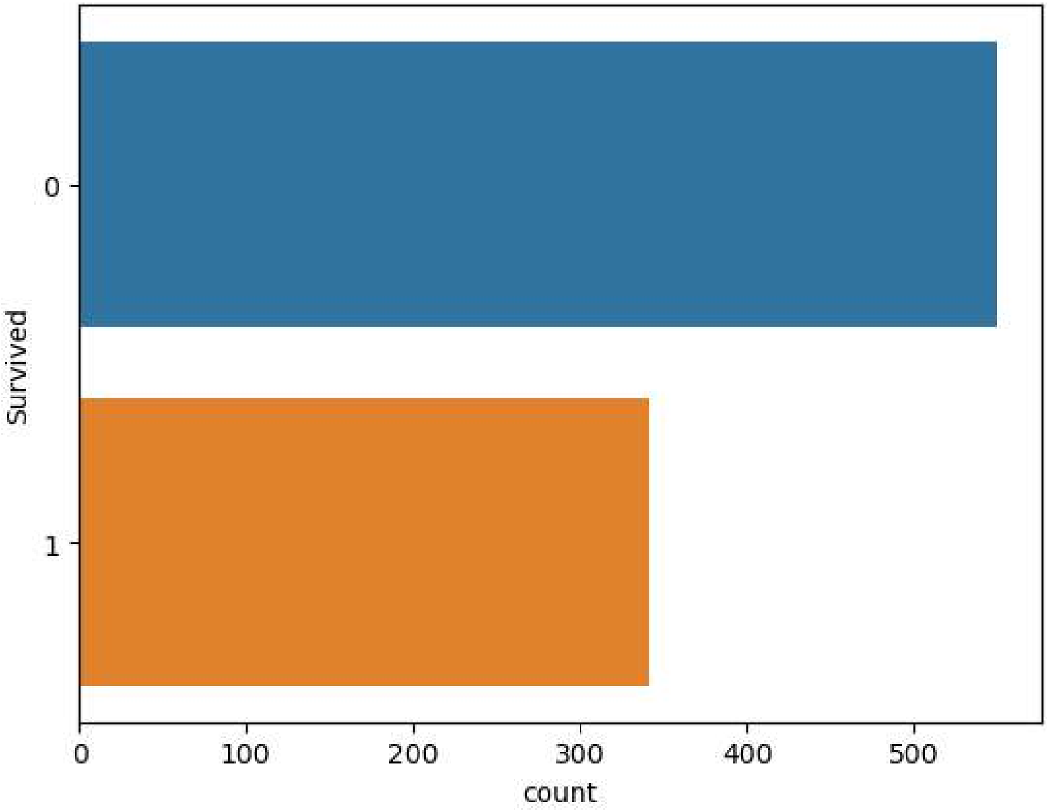
	Survived
0	0
1	1
2	1
3	1
4	0

```
Data Visualization
```

In [90]:
```
sns.barplot(data=ds,x='Survived',y='Fare')
plt.xlabel("Survived")
plt.ylabel("Fare")
plt.title("Survived vs Fare");
```

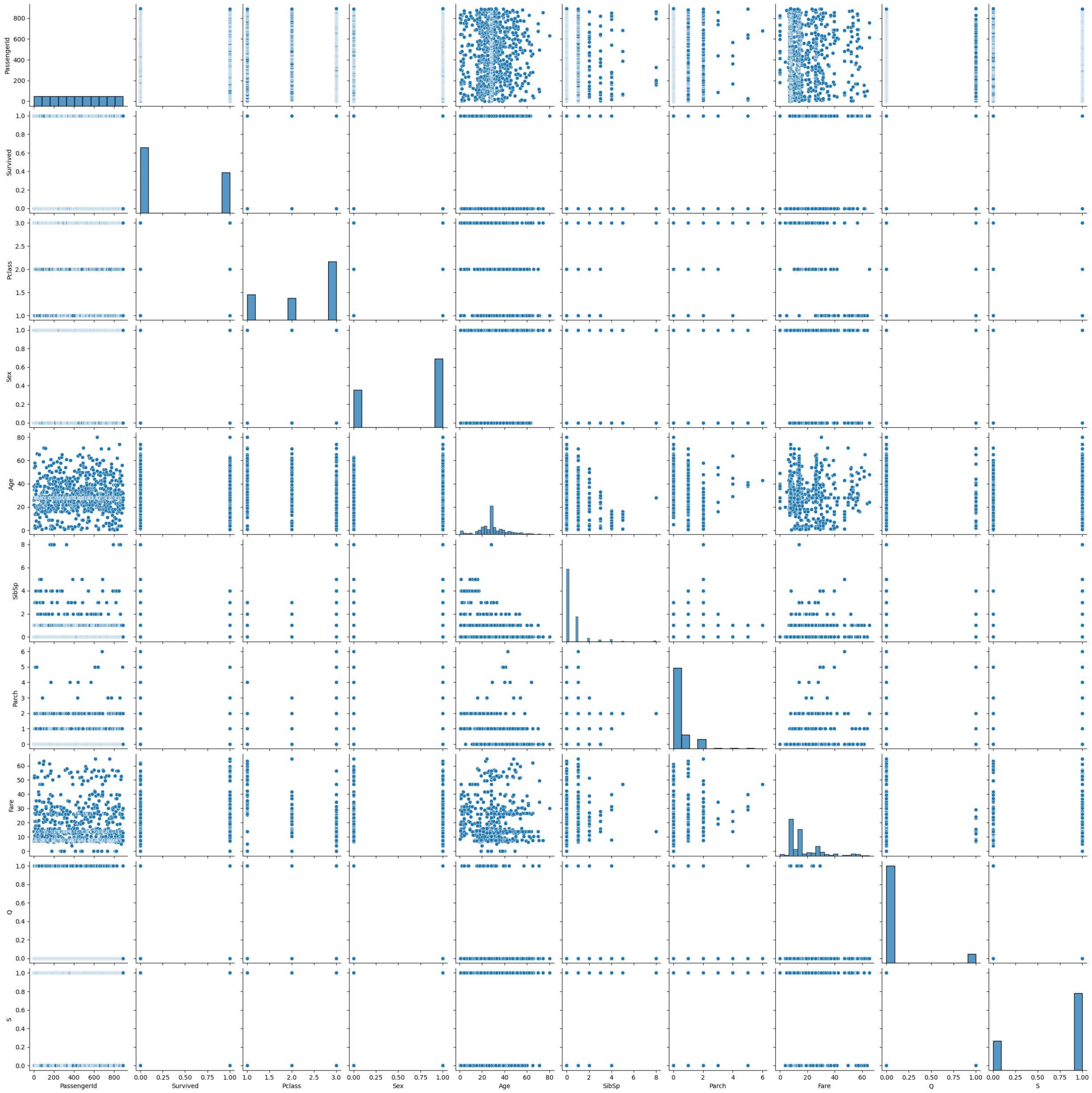


```
In [99]: sns.countplot(data=ds,y="Survived",orient='h');
```



In [98]:

```
sns.pairplot(data=ds);
```



Splitting the data into training and testing

In [74]:

```
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
```

In [75]:

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[75]:

```
((623, 4), (268, 4), (623, 1), (268, 1))
```

Feature Scaling

In [77]:
```
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
```

In [80]:
```
x_train=ss.fit_transform(x_train)
x_test=ss.transform(x_test)
```

In [82]:
```
x_train
```

Out[82]:
```
array([[ 1.64654836, -0.457246  , -0.47299765,  0.68311366],
       [ 1.4930717 ,  0.4033711 , -0.47299765, -0.29074647],
       [-2.19036814,  3.8458395 ,  1.93253327,  2.26224144],
       ...,
       [-0.11843323, -0.457246  , -0.47299765, -0.77703247],
       [ 0.49547341,  0.4033711 , -0.47299765, -0.02691185],
       [ 2.33719333,  0.4033711 ,  0.72976781,  1.64921395]])
```

In [83]:
```
x_test
```

Out[83]:
```
array([[-0.0724674 , -0.53120385, -0.47809977, -0.15359735],
       [-0.0724674 , -0.53120385, -0.47809977, -0.71667637],
       [-1.69302814,  3.68694819,  0.87064484,  1.04185031],
       ...,
       [-0.14963696,  0.52333416, -0.47809977, -0.15393153],
       [-0.84416299, -0.53120385, -0.47809977, -0.72109409],
       [-0.0724674 , -0.53120385, -0.47809977,  0.92739733]])
```

In [ ]:
```

```