P J N D M PRAKASH

durga.21bce8942@vitapstudent.ac.in

VIT-AP MORNING SLOT

ASSIGNMENT - 3

DATA PREPROCESSING ON TITANIC DATASET

```
# Data Preprocessing.
# Import the Libraries.
# Import the dataset
# Checking for Null Values.

# Data Visualization.
# Outlier Detection
# Splitting Dependent and Independent variables
# Encoding
# Feature Scaling.
# Splitting Data into Train and Test.
```

Import libraries and dataset

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

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

```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Tick
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	3309
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	3632
2	894	0	2	Myles, Mr. Thomas	male	62.0	0	0	2402

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

```
df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch
count	418.000000	418.000000	418.000000	332.000000	418.000000	418.00000

```
df.corr()
```

```
<ipython-input-6-2f6f6606aa2c>:1: FutureWarning: The default value of numeri
  df.corr()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch
PassengerId	1.000000	-0.023245	-0.026751	-0.034102	0.003818	0.043080
Survived	-0.023245	1.000000	-0.108615	-0.000013	0.099943	0.159120
Pclass	-0.026751	-0.108615	1.000000	-0.492143	0.001087	0.018721
Age	-0.034102	-0.000013	-0.492143	1.000000	-0.091587	-0.061249
SibSp	0.003818	0.099943	0.001087	-0.091587	1.000000	0.306895
Parch	0.043080	0.159120	0.018721	-0.061249	0.306895	1.000000
Fare	0.008211	0.191514	-0.577147	0.337932	0.171539	0.230046

```
df.corr().Survived.sort_values(ascending=False)
```

```
<ipython-input-8-fe51b8bb09d5>:1: FutureWarning: The default value of numeri
  df.corr().Survived.sort_values(ascending=False)
Survived       1.000000
Fare           0.191514
Parch          0.159120
SibSp          0.099943
Age           -0.000013
PassengerId   -0.023245
Pclass        -0.108615
Name: Survived, dtype: float64
```

Handling missing values

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

```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
```

```
Parch          False
Ticket         False
Fare            True
Cabin           True
Embarked       False
dtype: bool
```

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

```
86
```

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

```
1
```

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

```
327
```

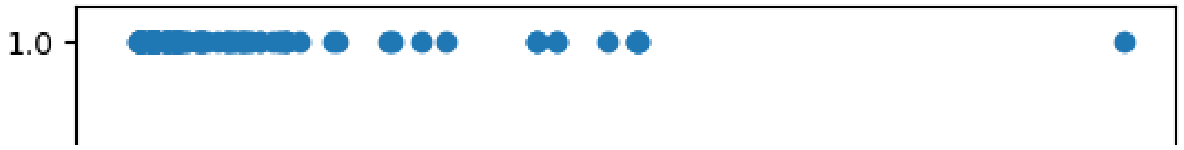
```
df["Age"].fillna(df["Age"].mean(),inplace=True)
```

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

Data - Visualzation

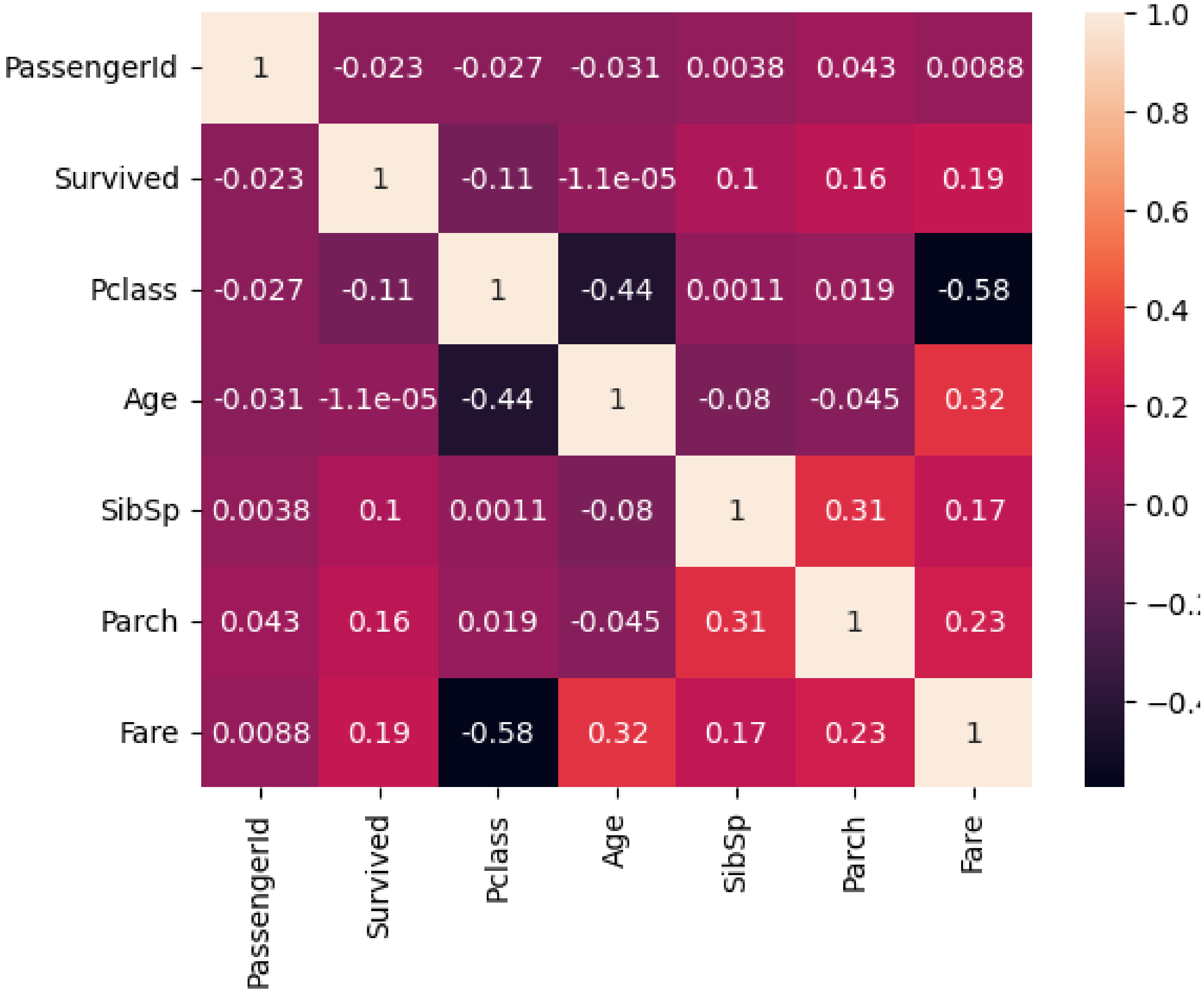
```
plt.scatter(df['Fare'],df['Survived'])
```

```
<matplotlib.collections.PathCollection at 0x7827f0e064d0>
```



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

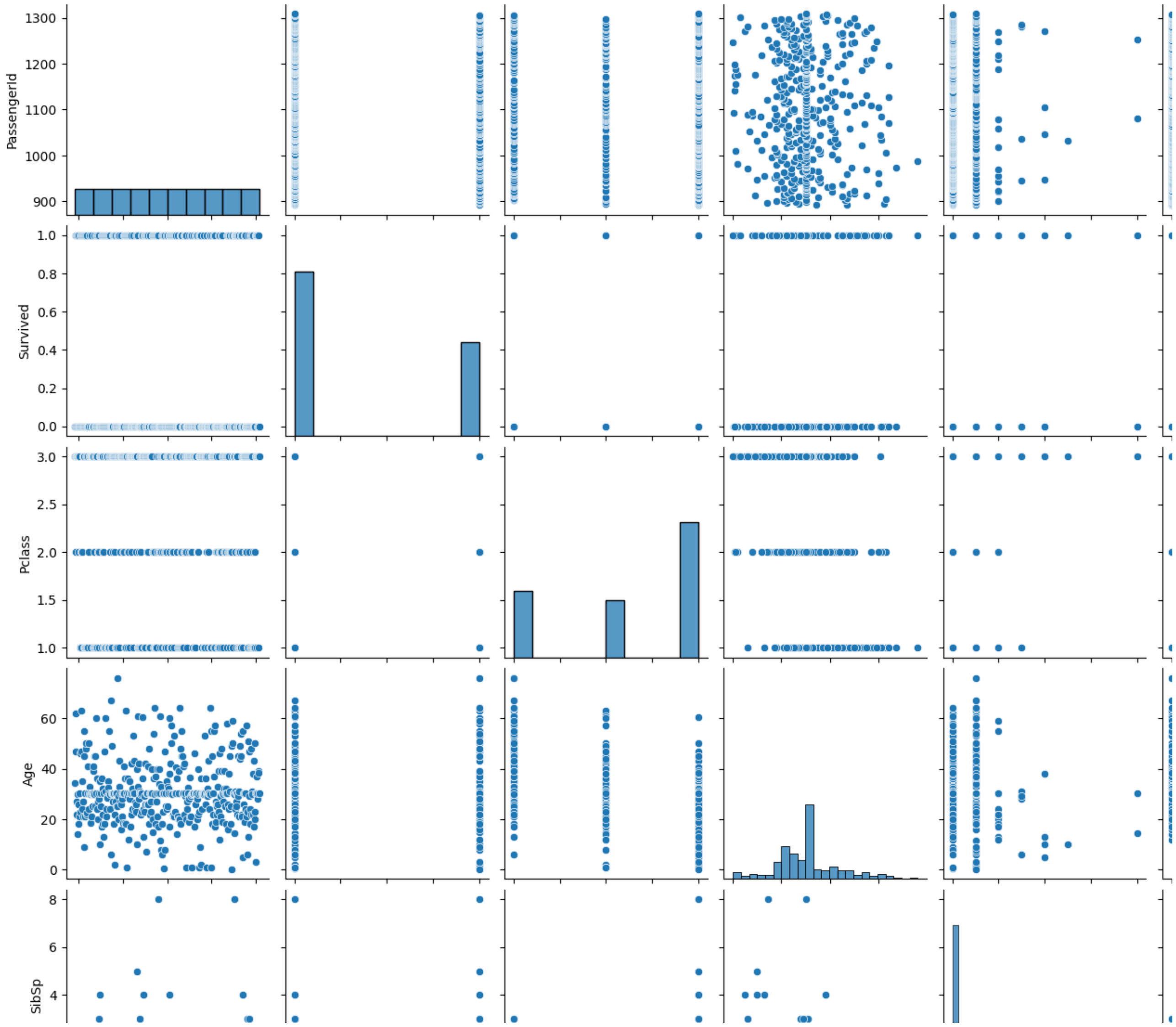
```
<ipython-input-21-8df7bcac526d>:1: FutureWarning: The default value of numer
  sns.heatmap(df.corr(),annot=True)
<Axes: >
```



	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1	-0.023	-0.027	-0.031	0.0038	0.043	0.0088
Survived	-0.023	1	-0.11	-1.1e-05	0.1	0.16	0.19
Pclass	-0.027	-0.11	1	-0.44	0.0011	0.019	-0.58
Age	-0.031	-1.1e-05	-0.44	1	-0.08	-0.045	0.32
SibSp	0.0038	0.1	0.0011	-0.08	1	0.31	0.17
Parch	0.043	0.16	0.019	-0.045	0.31	1	0.23
Fare	0.0088	0.19	-0.58	0.32	0.17	0.23	1

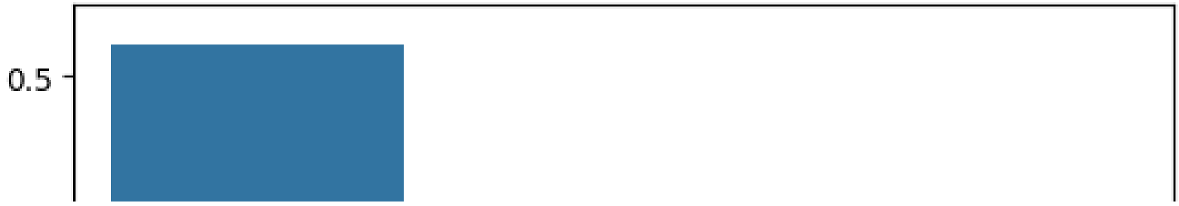
```
plt.figure(figsize=(20,15))
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7827e6b69720>
<Figure size 2000x1500 with 0 Axes>
```



```
sns.barplot(x='Embarked',y='Survived',data=df,ci=0)
```

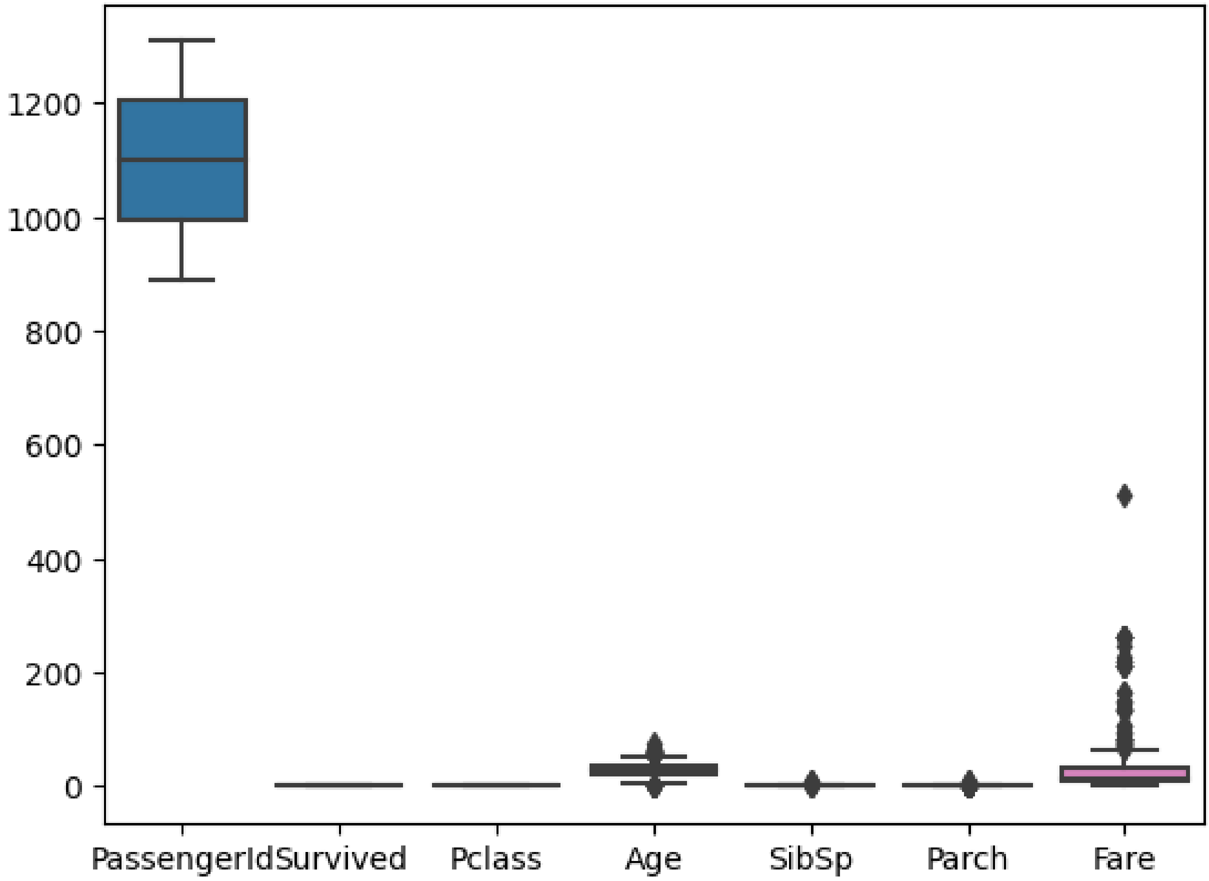
```
<ipython-input-33-b5d9aff878fc>:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=('ci', 0)` for the same effe

  sns.barplot(x='Embarked',y='Survived',data=df,ci=0)
<Axes: xlabel='Embarked', ylabel='Survived'>
```



Outlier Detection

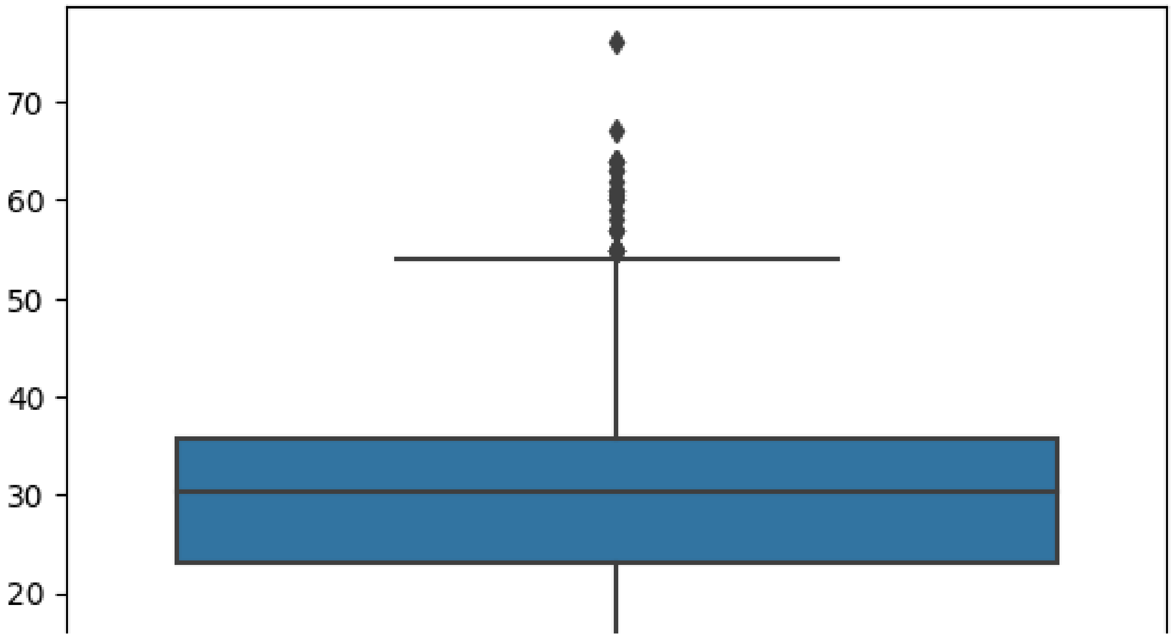
```
sns.boxplot(df)
```

```
<Axes: >
```



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

```
<Axes: >
```



```
df.median()
```

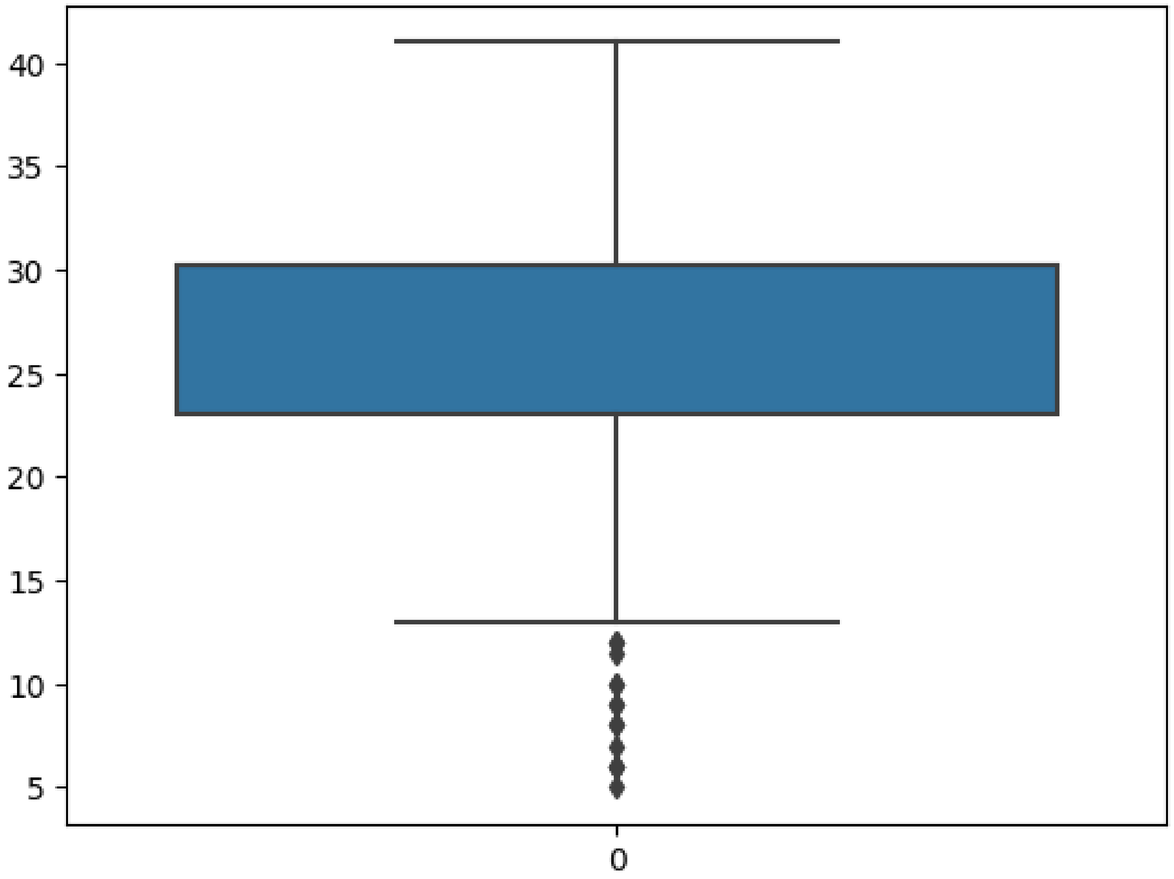
```
<ipython-input-39-6d467abf240d>:1: FutureWarning: The default value of numer
  df.median()
PassengerId    1098.50000
Survived          0.00000
Pclass            3.00000
Age              30.27259
SibSp             0.00000
Parch             0.00000
Fare             13.50000
dtype: float64
```

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

iqr = q3-q1
upperlimit = q3 + 1.5*iqr
lowerlimit = q1 - 1.5*iqr

df["Age"]=np.where(df["Age"] > upperlimit,30.27,df["Age"])  # Replace outlier val
```

```
<ipython-input-45-2f1cc8a9a168>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
  df["Age"]=np.where(df["Age"] > upperlimit,30.27,df["Age"])  # Replace outl
```

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

```
<Axes: >
```



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

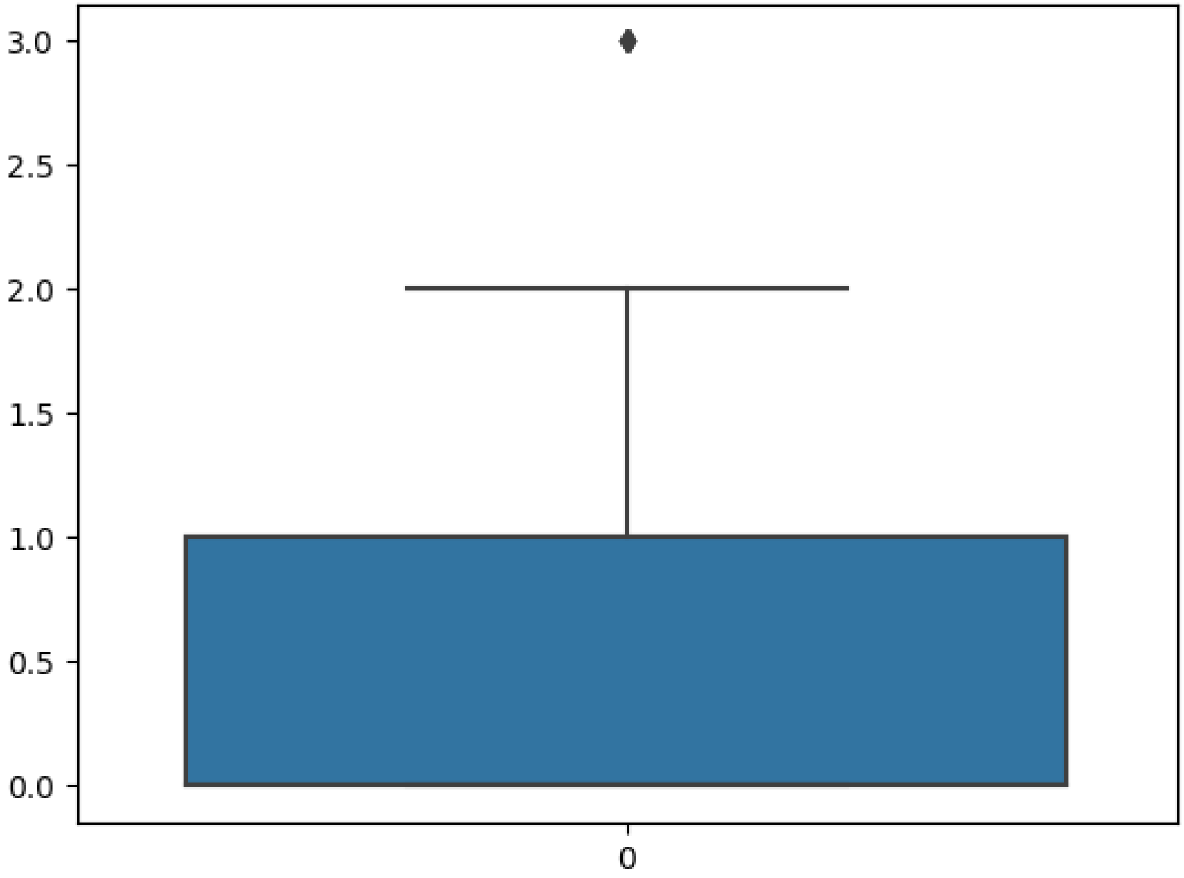
```
<Axes: >
```

```
p99 = df.SibSp.quantile(0.99)
```

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

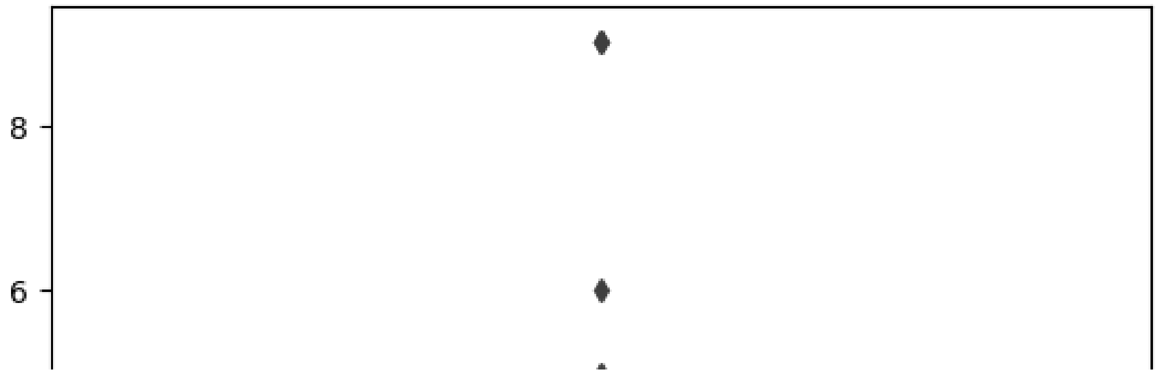
```
sns.boxplot(df.SibSp)
```

```
<Axes: >
```



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

```
<Axes: >
```

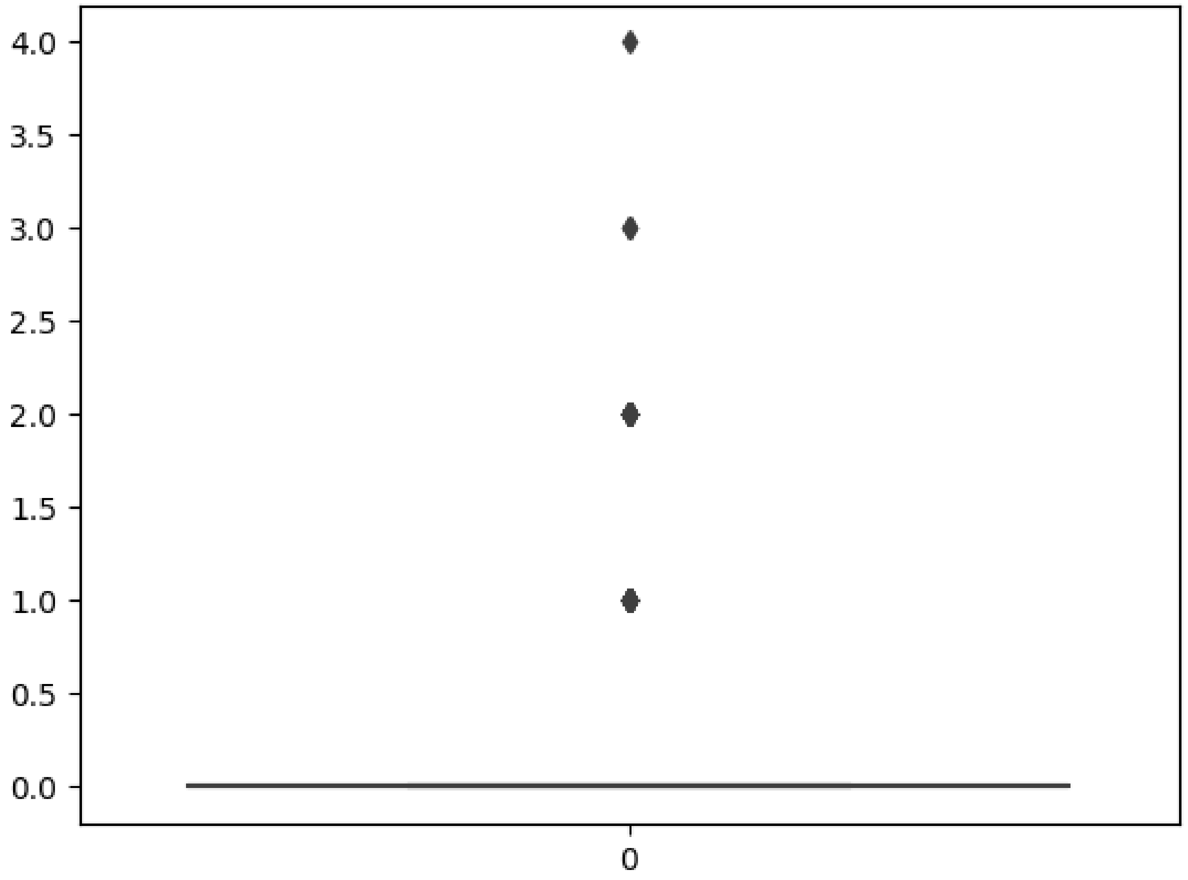


```
p99 = df.Parch.quantile(0.99)
```

```
df=df[df.Parch<p99]
```

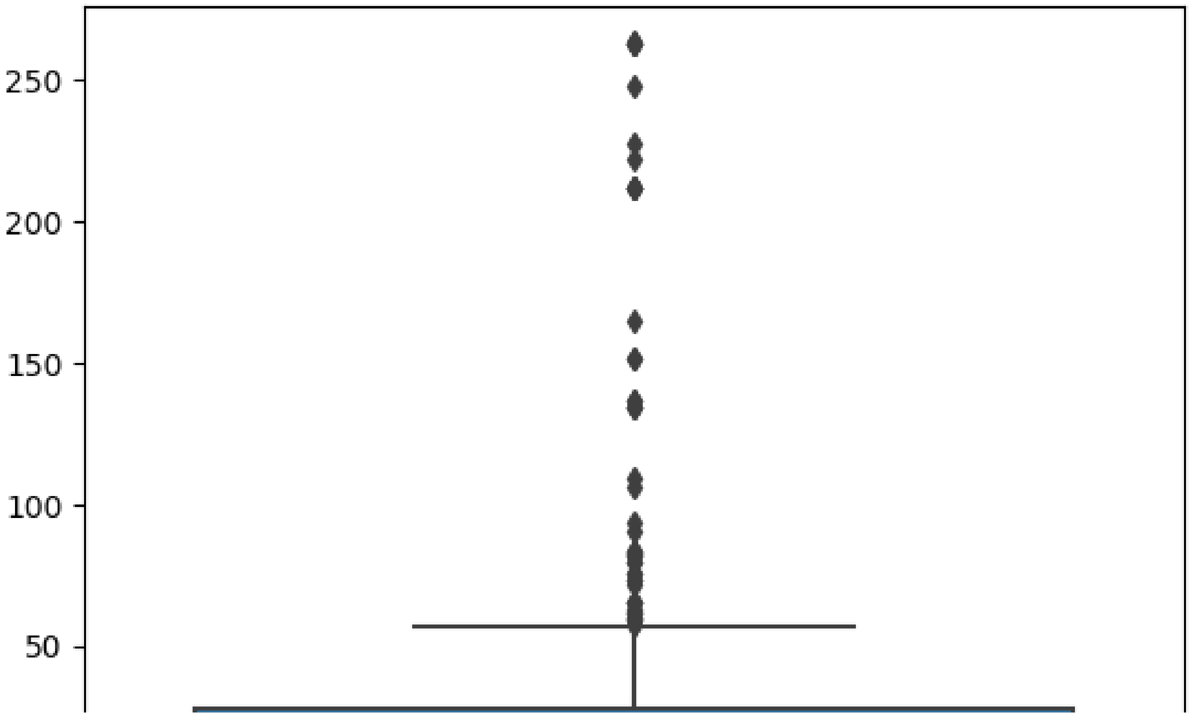
```
sns.boxplot(df['Parch'])
```

```
<Axes: >
```



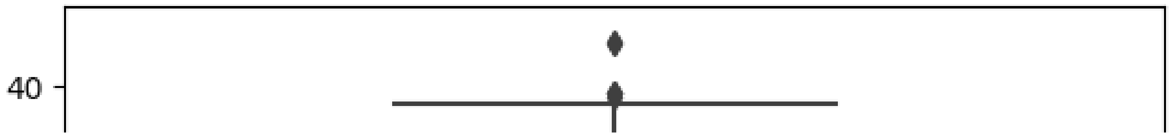
```
sns.boxplot(df['Fare'])
```

```
<Axes: >
```



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

iqr = q3-q1
upperlimit = q3 + 1.5*iqr
lowerlimit = q1 - 1.5*iqr

df['Fare']=np.where(df["Fare"] > upperlimit,13.50,df["Fare"])


sns.boxplot(df.Fare)
```

```
<Axes: >
```



Spliting Dependent and Independent Variables

```
x = df.drop(columns=["Survived","PassengerId","Name","Ticket","Cabin"],axis=1)
```

```
x.head()
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	male	34.50	0	0	7.8292	Q
1	3	female	30.27	1	0	7.0000	S
3	3	male	27.00	0	0	8.6625	S
4	3	female	22.00	1	1	12.2875	S
5	3	male	14.00	0	0	9.2250	S

```
y = pd.Series(df["Survived"])
```

```
y.head()
```

```
0    0
1    1
3    0
4    1
5    0
Name: Survived, dtype: int64
```

Encoding

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

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

```
x.head()
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1	34.50	0	0	7.8292	Q
1	3	0	30.27	1	0	7.0000	S
3	3	1	27.00	0	0	8.6625	S
4	3	0	22.00	1	1	12.2875	S

```
print(le.classes_)
```

```
['female' 'male']
```

```
mapping=dict(zip(le.classes_,range(len(le.classes_))))
```

```
mapping
```

```
{'female': 0, 'male': 1}
```

```
le1 = LabelEncoder()
```

```
x["Embarked"] = le1.fit_transform(x["Embarked"])
x.head()
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1	34.50	0	0	7.8292	1
1	3	0	30.27	1	0	7.0000	2
3	3	1	27.00	0	0	8.6625	2
4	3	0	22.00	1	1	12.2875	2
5	3	1	14.00	0	0	9.2250	2

```
print(le1.classes_)
```

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

```
mapping1=dict(zip(le1.classes_,range(len(le1.classes_))))
mapping1
```

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

Feature - Scaling

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

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

```
x_Scaled.head()
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1.0	1.0	0.814286	0.000000	0.00	0.184216	0.5
1	1.0	0.0	0.693429	0.333333	0.00	0.164706	1.0
2	1.0	1.0	0.600000	0.000000	0.00	0.203824	1.0
3	1.0	0.0	0.457143	0.333333	0.25	0.289118	1.0
4	1.0	1.0	0.228571	0.000000	0.00	0.217059	1.0

Splitting , Training and Testing Data

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

```
x_train,x_test,y_train,y_test = train_test_split(x_Scaled,y,test_size = 0.2,rando
```

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

    (256, 7) (64, 7) (256,) (64,)
```

```
**THE  END**
```

✓ 0s completed at 12:16 PM ● ✕