

alexiaprincecheenath-assignment3

September 22, 2023

#Import the libraries

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

#Import the Dataset

```
[77]: from google.colab import files
df=files.upload()
```

<IPython.core.display.HTML object>

Saving Titanic-Dataset.csv to Titanic-Dataset (1).csv

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

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

```
[79]:
```

| | 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 |

```
[80]: 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
```

```
[81]: df.corr()
```

```
<ipython-input-81-2f6f6606aa2c>: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.

```
df.corr()
```

```
[81]:
```

| | 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 |

#Handling the Null Values

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

```
[82]: 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
```

```
[83]: #fill in values for Age,Cabin,Embarked
df["Age"].fillna(df["Age"].mean(),inplace=True)
df["Cabin"].fillna(df["Cabin"].mode()[0],inplace=True)
df["Embarked"].fillna(df["Embarked"].mode()[0],inplace=True)
```

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

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

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

```
[85]:   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      Cabin Embarked
0         1     0    A/5 21171   7.2500   B96 B98        S
1         1     0    PC 17599  71.2833     C85        C
2         0     0  STON/O2. 3101282   7.9250   B96 B98        S
3         1     0    113803   53.1000     C123        S
4         0     0    373450   8.0500   B96 B98        S
5         0     0    330877   8.4583   B96 B98        Q
```

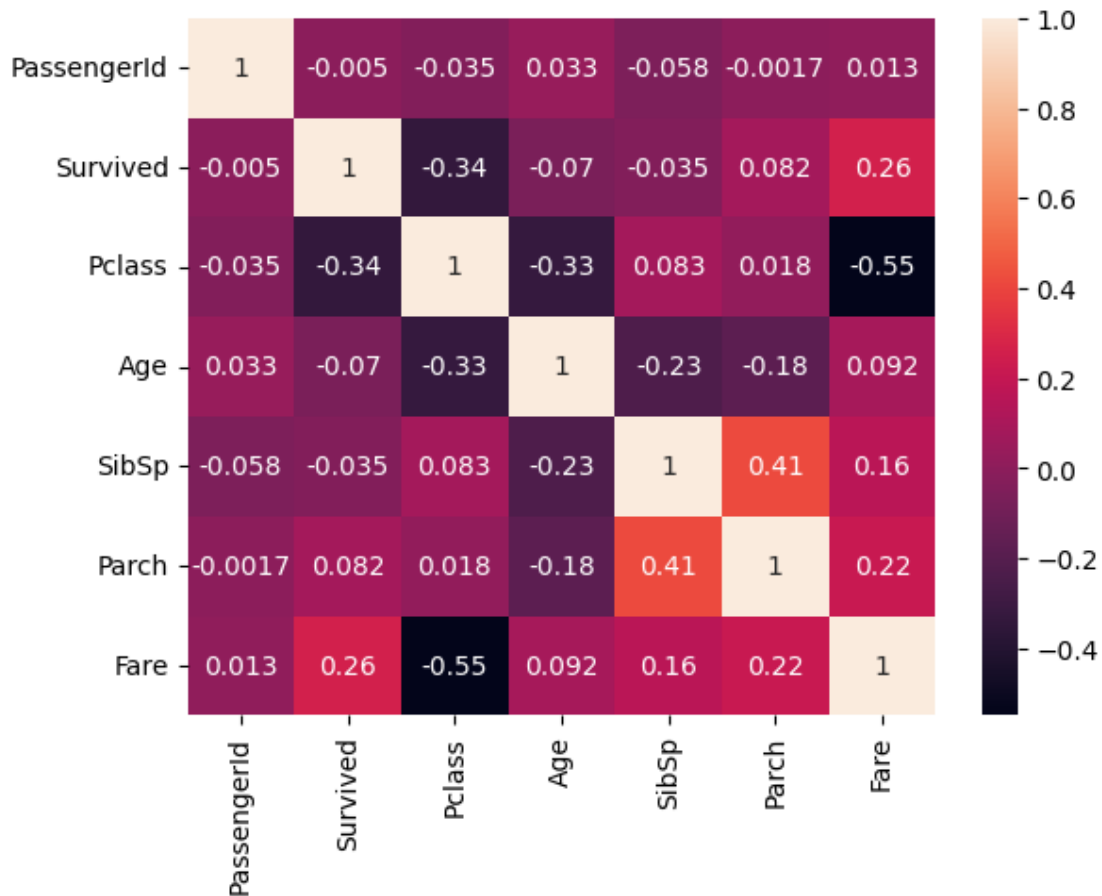
| | | | | | | |
|---|---|---|--------|---------|---------|---|
| 6 | 0 | 0 | 17463 | 51.8625 | E46 | S |
| 7 | 3 | 1 | 349909 | 21.0750 | B96 B98 | S |
| 8 | 0 | 2 | 347742 | 11.1333 | B96 B98 | S |
| 9 | 1 | 0 | 237736 | 30.0708 | B96 B98 | C |

#Data Visualisation

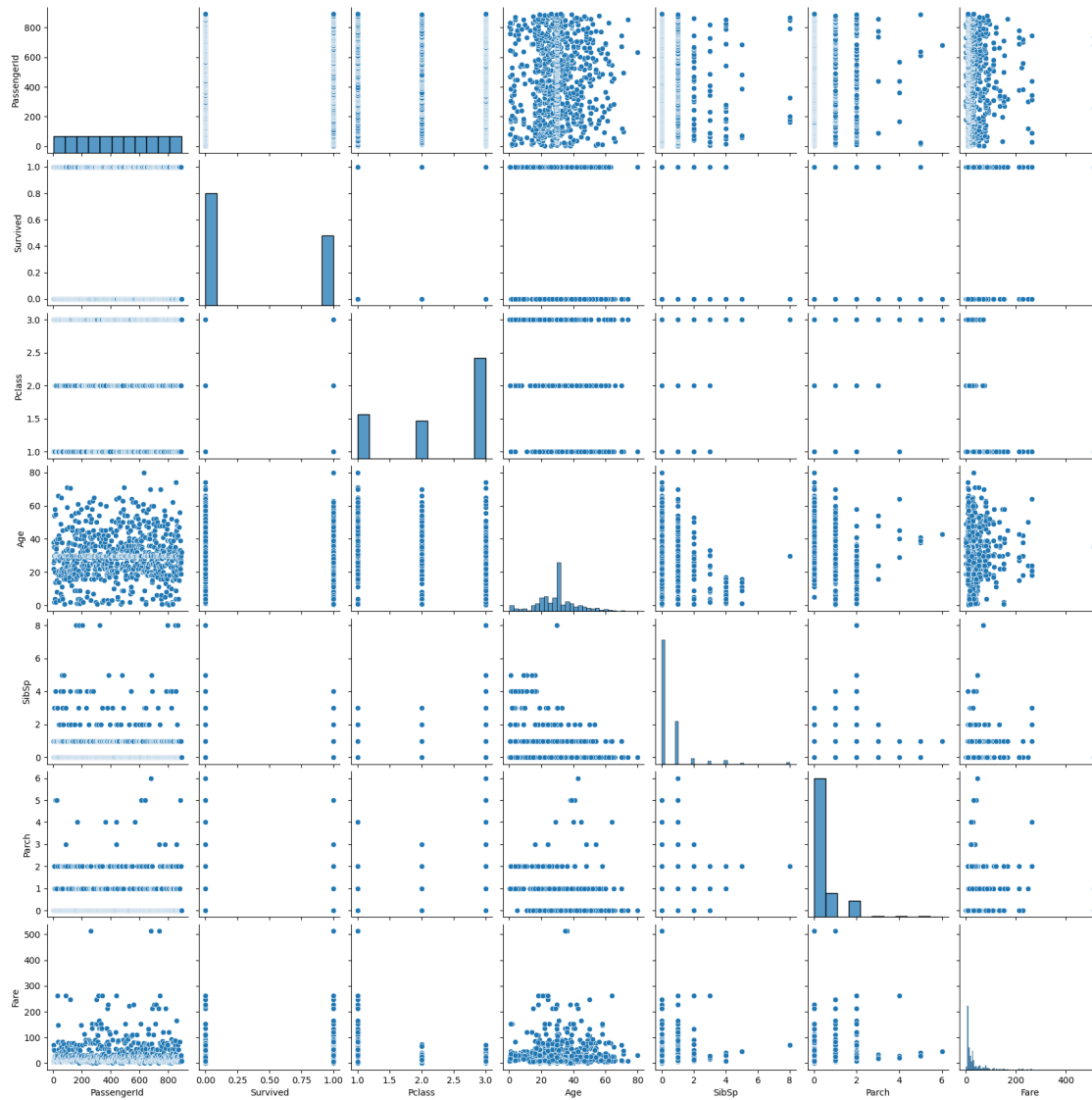
```
[86]: sns.heatmap(df.corr(),annot=True)
plt.show()
```

<ipython-input-86-f6412ee67fb3>: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.

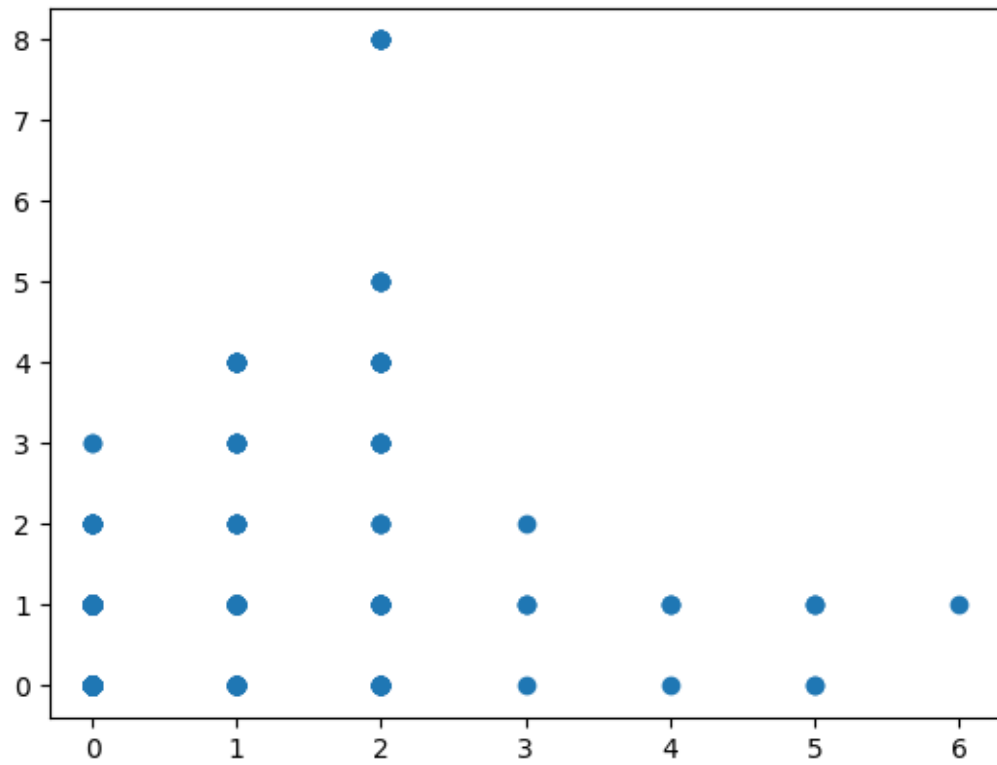
```
sns.heatmap(df.corr(),annot=True)
```



```
[87]: sns.pairplot(df)
plt.show()
```

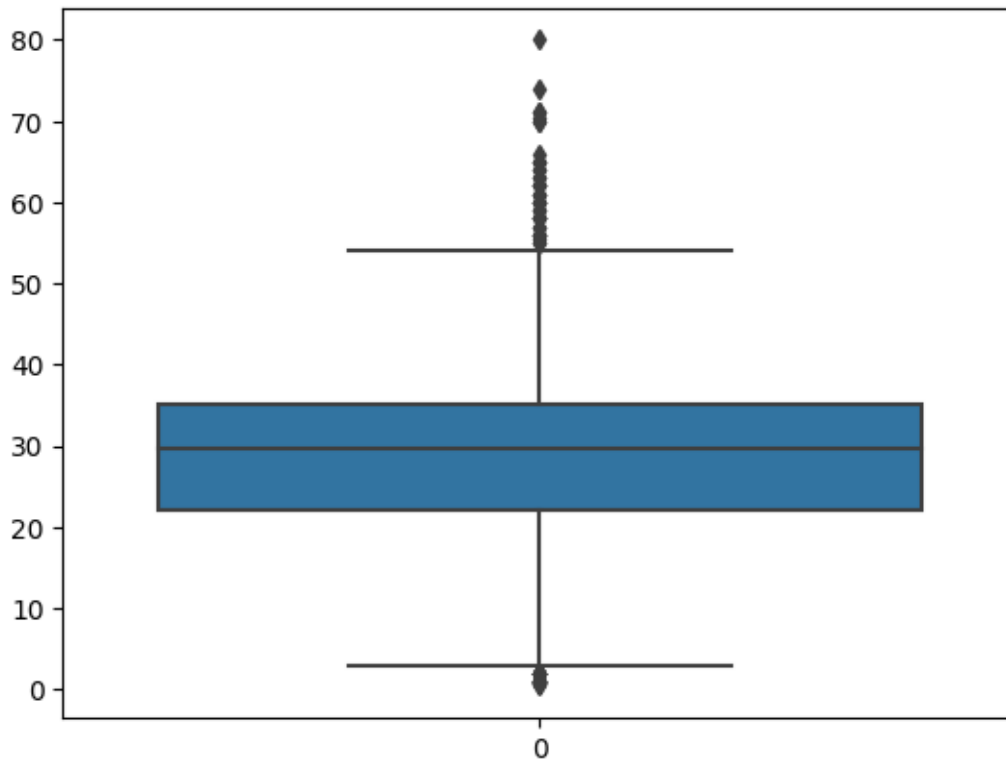


```
[88]: plt.scatter(df["Parch"],df["SibSp"])
plt.show()
```



#Outlier Detection

```
[89]: sns.boxplot(df["Age"])  
plt.show()
```



```
[90]: q1=df.Age.quantile(0.25)
      q3=df.Age.quantile(0.75)
      print(q1)
      print(q3)
```

```
22.0
35.0
```

```
[91]: iqr=q3-q1
      iqr
```

```
[91]: 13.0
```

```
[92]: upper_limit=q3+1.5*iqr
      upper_limit
```

```
[92]: 54.5
```

```
[93]: lower_limit=q1-1.5*iqr
      lower_limit
```

```
[93]: 2.5
```



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

```
<ipython-input-94-6d467abf240d>: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.
```

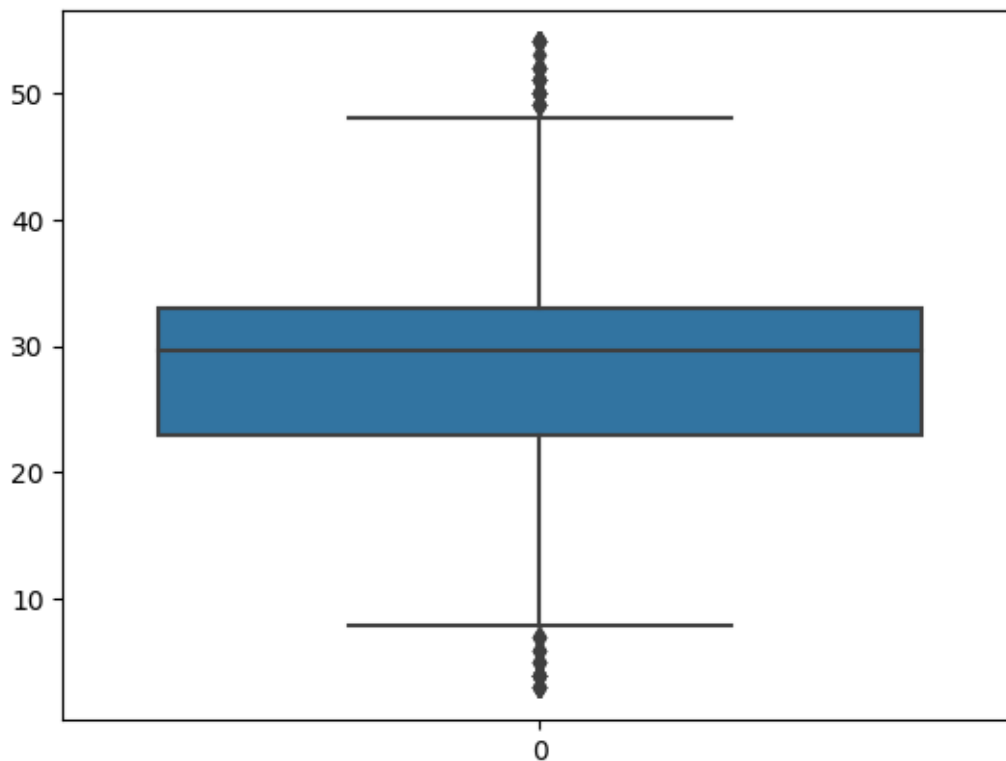
```
df.median()
```

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

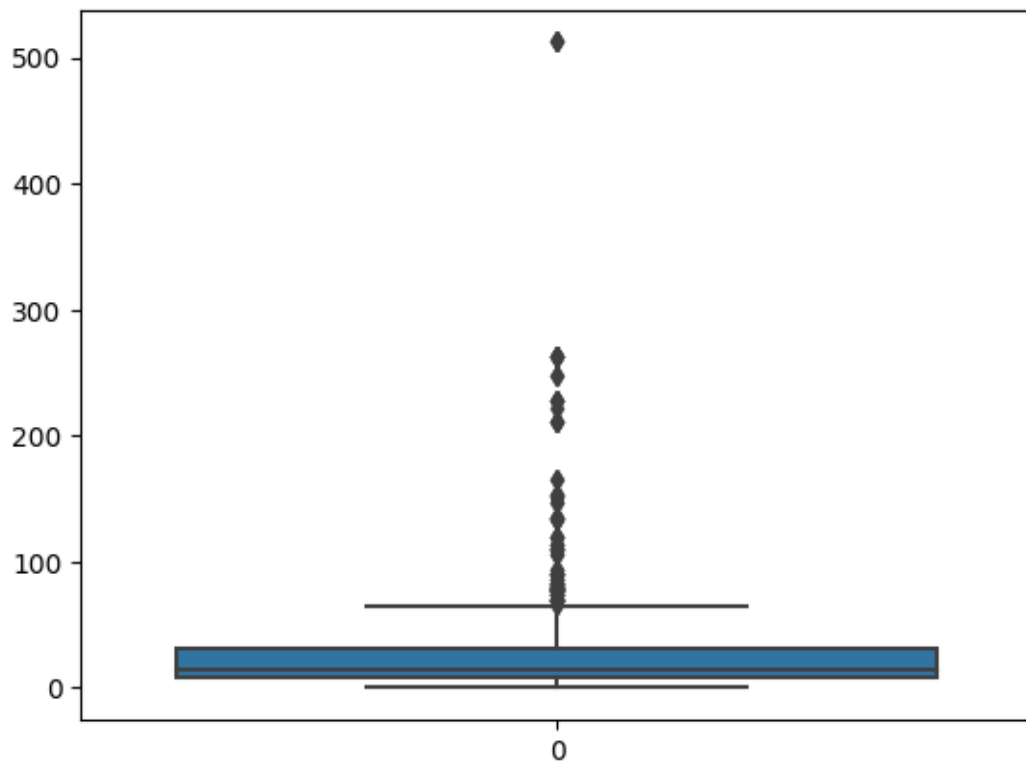
```
[95]: df['Age']=np.where(df['Age']>upper_limit,29.699118,df['Age'])
```

```
[96]: df=df[df.Age>lower_limit]
```

```
[97]: sns.boxplot(df["Age"])  
plt.show()
```



```
[98]: sns.boxplot(df["Fare"])  
plt.show()
```



```
[99]: q1=df.Fare.quantile(0.25)  
q3=df.Fare.quantile(0.75)  
print(q1)  
print(q3)
```

```
7.8958  
30.8479
```

```
[100]: iqr=q3-q1  
iqr
```

```
[100]: 22.952099999999998
```

```
[101]: upper_limit=q3+1.5*iqr  
upper_limit
```

```
[101]: 65.27605
```

```
[102]: lower_limit=q1-1.5*iqr  
lower_limit
```

```
[102]: -26.532349999999994
```

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

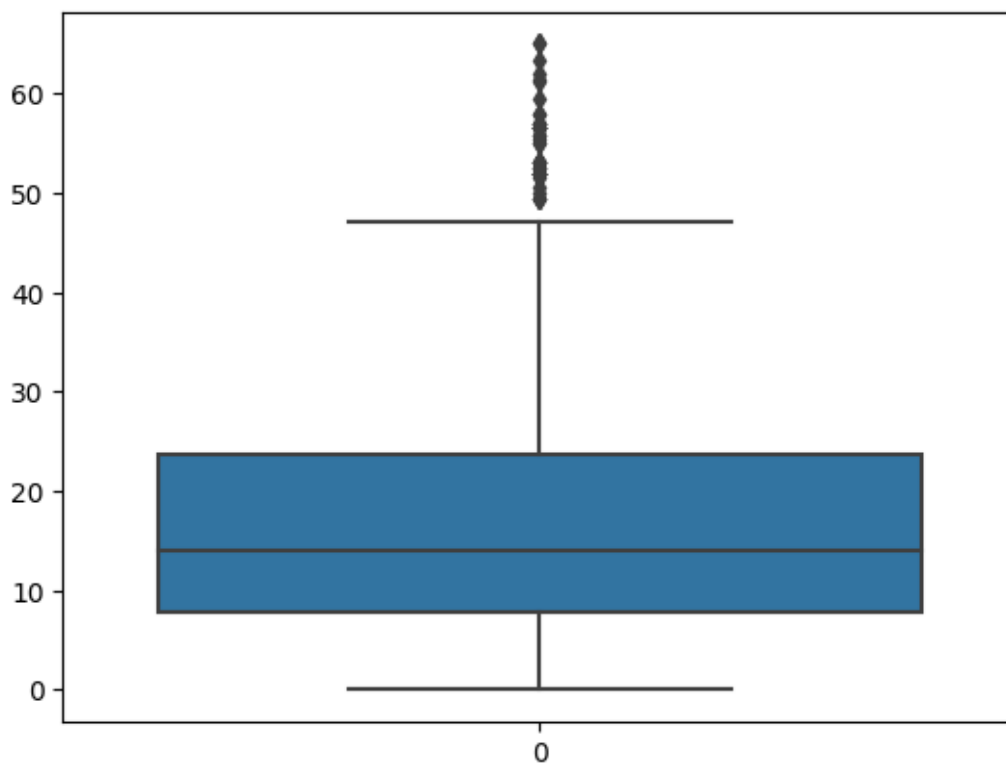
<ipython-input-103-6d467abf240d>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.

```
df.median()
```

```
[103]: PassengerId    447.000000  
Survived           0.000000  
Pclass             3.000000  
Age                29.699118  
SibSp              0.000000  
Parch              0.000000  
Fare               14.108300  
dtype: float64
```

```
[104]: df["Fare"]=np.where(df["Fare"]>upper_limit,14.4542,df["Fare"])
```

```
[105]: sns.boxplot(df["Fare"])  
plt.show()
```



#Splitting Independent and Dependent Variables

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

```
[106]: 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  B96 B98      S
1      0    PC 17599   14.4542   C85      C
2      0  STON/O2. 3101282   7.9250  B96 B98      S
```

| | | | | | |
|---|---|--------|---------|---------|---|
| 3 | 0 | 113803 | 53.1000 | C123 | S |
| 4 | 0 | 373450 | 8.0500 | B96 B98 | S |

```
[107]: df.shape
```

```
[107]: (867, 12)
```

```
[108]: df.Ticket.nunique()
```

```
[108]: 679
```

```
[109]: df.Name.nunique()
```

```
[109]: 867
```

```
[110]: df.drop(columns=["Name"], inplace=True)
```

```
[111]: df
```

```
[111]:
```

| | PassengerId | Survived | Pclass | Sex | Age | SibSp | Parch | \ |
|-----|-------------|----------|--------|--------|-----------|-------|-------|---|
| 0 | 1 | 0 | 3 | male | 22.000000 | 1 | 0 | |
| 1 | 2 | 1 | 1 | female | 38.000000 | 1 | 0 | |
| 2 | 3 | 1 | 3 | female | 26.000000 | 0 | 0 | |
| 3 | 4 | 1 | 1 | female | 35.000000 | 1 | 0 | |
| 4 | 5 | 0 | 3 | male | 35.000000 | 0 | 0 | |
| .. | ... | ... | ... | ... | ... | ... | ... | |
| 886 | 887 | 0 | 2 | male | 27.000000 | 0 | 0 | |
| 887 | 888 | 1 | 1 | female | 19.000000 | 0 | 0 | |
| 888 | 889 | 0 | 3 | female | 29.699118 | 1 | 2 | |
| 889 | 890 | 1 | 1 | male | 26.000000 | 0 | 0 | |
| 890 | 891 | 0 | 3 | male | 32.000000 | 0 | 0 | |

| | Ticket | Fare | Cabin | Embarked |
|-----|------------------|---------|---------|----------|
| 0 | A/5 21171 | 7.2500 | B96 B98 | S |
| 1 | PC 17599 | 14.4542 | C85 | C |
| 2 | STON/O2. 3101282 | 7.9250 | B96 B98 | S |
| 3 | 113803 | 53.1000 | C123 | S |
| 4 | 373450 | 8.0500 | B96 B98 | S |
| .. | ... | ... | ... | ... |
| 886 | 211536 | 13.0000 | B96 B98 | S |
| 887 | 112053 | 30.0000 | B42 | S |
| 888 | W./C. 6607 | 23.4500 | B96 B98 | S |
| 889 | 111369 | 30.0000 | C148 | C |
| 890 | 370376 | 7.7500 | B96 B98 | Q |

```
[867 rows x 11 columns]
```

```
[112]: x=df.drop(columns=["Survived"],axis=1)
x.head()
```

```
[112]:
```

| | PassengerId | Pclass | Sex | Age | SibSp | Parch | Ticket | Fare | \ |
|---|-------------|--------|--------|------|-------|-------|------------------|---------|---|
| 0 | 1 | 3 | male | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | |
| 1 | 2 | 1 | female | 38.0 | 1 | 0 | PC 17599 | 14.4542 | |
| 2 | 3 | 3 | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | |
| 3 | 4 | 1 | female | 35.0 | 1 | 0 | 113803 | 53.1000 | |
| 4 | 5 | 3 | male | 35.0 | 0 | 0 | 373450 | 8.0500 | |

| | Cabin | Embarked |
|---|---------|----------|
| 0 | B96 B98 | S |
| 1 | C85 | C |
| 2 | B96 B98 | S |
| 3 | C123 | S |
| 4 | B96 B98 | S |

```
[113]: x.shape
```

```
[113]: (867, 10)
```

```
[114]: type(x)
```

```
[114]: pandas.core.frame.DataFrame
```

```
[115]: y=df["Survived"]
y.head()
```

```
[115]:
```

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 0 |

Name: Survived, dtype: int64

```
[116]: y.shape
```

```
[116]: (867,)
```

```
[117]: type(y)
```

```
[117]: pandas.core.series.Series
```

#Encoding

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

```
[118]:
```

| | PassengerId | Pclass | Sex | Age | SibSp | Parch | Ticket | Fare | \ |
|---|-------------|--------|--------|------|-------|-------|------------------|---------|---|
| 0 | 1 | 3 | male | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | |
| 1 | 2 | 1 | female | 38.0 | 1 | 0 | PC 17599 | 14.4542 | |
| 2 | 3 | 3 | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | |
| 3 | 4 | 1 | female | 35.0 | 1 | 0 | 113803 | 53.1000 | |
| 4 | 5 | 3 | male | 35.0 | 0 | 0 | 373450 | 8.0500 | |

```

Cabin Embarked
0 B96 B98      S
1      C85      C
2 B96 B98      S
3      C123      S
4 B96 B98      S

```

```
[119]: from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

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

```
[120]:
```

| | PassengerId | Pclass | Sex | Age | SibSp | Parch | Ticket | Fare | \ |
|---|-------------|--------|-----|------|-------|-------|------------------|---------|---|
| 0 | 1 | 3 | 1 | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | |
| 1 | 2 | 1 | 0 | 38.0 | 1 | 0 | PC 17599 | 14.4542 | |
| 2 | 3 | 3 | 0 | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | |
| 3 | 4 | 1 | 0 | 35.0 | 1 | 0 | 113803 | 53.1000 | |
| 4 | 5 | 3 | 1 | 35.0 | 0 | 0 | 373450 | 8.0500 | |

```

Cabin Embarked
0 B96 B98      S
1      C85      C
2 B96 B98      S
3      C123      S
4 B96 B98      S

```

```
[121]: x["Cabin"]=le.fit_transform(x["Cabin"])
x.head()
```

```
[121]:
```

| | PassengerId | Pclass | Sex | Age | SibSp | Parch | Ticket | Fare | \ |
|---|-------------|--------|-----|------|-------|-------|------------------|---------|---|
| 0 | 1 | 3 | 1 | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | |
| 1 | 2 | 1 | 0 | 38.0 | 1 | 0 | PC 17599 | 14.4542 | |
| 2 | 3 | 3 | 0 | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | |
| 3 | 4 | 1 | 0 | 35.0 | 1 | 0 | 113803 | 53.1000 | |
| 4 | 5 | 3 | 1 | 35.0 | 0 | 0 | 373450 | 8.0500 | |

```

Cabin Embarked
0 47      S
1 81      C

```

```

2      47      S
3      55      S
4      47      S

```

```
[122]: x["Embarked"]=le.fit_transform(x["Embarked"])
x.head()
```

```
[122]:
```

| | PassengerId | Pclass | Sex | Age | SibSp | Parch | Ticket | Fare | \ |
|---|-------------|--------|-----|------|-------|-------|------------------|---------|---|
| 0 | 1 | 3 | 1 | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | |
| 1 | 2 | 1 | 0 | 38.0 | 1 | 0 | PC 17599 | 14.4542 | |
| 2 | 3 | 3 | 0 | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | |
| 3 | 4 | 1 | 0 | 35.0 | 1 | 0 | 113803 | 53.1000 | |
| 4 | 5 | 3 | 1 | 35.0 | 0 | 0 | 373450 | 8.0500 | |

| | Cabin | Embarked |
|---|-------|----------|
| 0 | 47 | 2 |
| 1 | 81 | 0 |
| 2 | 47 | 2 |
| 3 | 55 | 2 |
| 4 | 47 | 2 |

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

```
[123]:
```

| | PassengerId | Pclass | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | \ |
|---|-------------|--------|-----|------|-------|-------|--------|---------|-------|---|
| 0 | 1 | 3 | 1 | 22.0 | 1 | 0 | 521 | 7.2500 | 47 | |
| 1 | 2 | 1 | 0 | 38.0 | 1 | 0 | 594 | 14.4542 | 81 | |
| 2 | 3 | 3 | 0 | 26.0 | 0 | 0 | 667 | 7.9250 | 47 | |
| 3 | 4 | 1 | 0 | 35.0 | 1 | 0 | 49 | 53.1000 | 55 | |
| 4 | 5 | 3 | 1 | 35.0 | 0 | 0 | 470 | 8.0500 | 47 | |

| | Embarked |
|---|----------|
| 0 | 2 |
| 1 | 0 |
| 2 | 2 |
| 3 | 2 |
| 4 | 2 |

#Feature Scaling

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

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

```
[127]: x_scaled
```



```
[127]: array([[ -1.73564798,  0.83135617,  0.73482275, ..., -0.783075 ,
           -0.27749855,  0.58664444],
          [ -1.73175204, -1.55140681, -1.3608724 , ..., -0.21638374,
            1.18401696, -1.94380694],
          [ -1.7278561 ,  0.83135617, -1.3608724 , ..., -0.72997867,
            -0.27749855,  0.58664444],
          ...,
          [  1.72394668,  0.83135617, -1.3608724 , ...,  0.491237 ,
            -0.27749855,  0.58664444],
          [  1.72784262, -1.55140681,  0.73482275, ...,  1.00646808,
            0.2813162 , -1.94380694],
          [  1.73173856,  0.83135617,  0.73482275, ..., -0.74374438,
            -0.27749855, -0.67858125]])
```

#Train Test Data Split

```
[128]: 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,random_state=0)
```

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

```
(693, 10) (174, 10) (693,) (174,)
```

#Predictions

```
[130]: from sklearn.linear_model import LinearRegression
```

```
[131]: lr=LinearRegression()
```

```
[132]: lr.fit(x_train,y_train)
```

```
[132]: LinearRegression()
```

```
[133]: y_pred=lr.predict(x_test)
       y_pred
```

```
[133]: array([ 0.65158663,  0.90072513,  0.13714898,  0.17267957,  0.98894389,
           0.52540306,  0.0962403 ,  0.82258251,  0.15278946,  0.10434956,
          -0.01157224,  0.08850255,  0.07038507,  0.34539804,  0.23346271,
           0.51448921,  0.03600387, -0.06013896,  0.64554218,  0.61846683,
           0.77988713, -0.03266761,  0.20125924,  1.09522316,  0.37406881,
           0.05994004,  0.12337424,  0.1940884 ,  0.30116449,  0.1476358 ,
           0.23537614,  0.13745178,  0.69427707,  0.64984075,  0.93443983,
           0.11386889,  0.27069061,  0.19438355,  0.93253712,  0.84723055,
           0.62202344,  0.70895822,  0.36218061,  0.21092761,  0.01529542,
           0.02893766,  0.76895231,  0.66723078,  0.12615712,  0.40653281,
           0.71997174,  0.28867855,  0.08457081,  0.06969991,  0.1126615 ,
```

```

0.21834596, 0.23561211, 0.04501408, 0.08447272, 0.10765501,
0.27226727, 0.00994394, 0.19410614, 0.67163225, 0.27216031,
0.82047952, 0.72016384, 0.624188 , 0.07237395, 0.12073832,
0.17125711, 0.17217271, 0.82192634, -0.02957249, 0.51302043,
0.58397554, 0.58146812, 0.09926588, 0.07037332, 0.09648288,
1.03117184, 0.86628559, 0.52192508, 0.21952361, 0.39920966,
0.2991001 , 0.99498822, 0.75555912, 0.14966159, 0.26448953,
0.52558879, 0.89437143, 0.72686699, 0.3545353 , 0.75695945,
0.33158584, 0.16103141, 0.80338904, 0.12974075, 0.65437483,
0.67708661, 0.10620137, 0.08769622, 0.79542594, 0.5323638 ,
-0.04568792, 0.81296876, 0.10944351, 0.14217753, 0.66560231,
-0.02060749, 0.17816917, 0.18006544, 0.10772514, 0.86827098,
0.25713005, 1.1048813 , 0.73057606, 0.07503933, 0.02705824,
0.72935106, 0.15305605, 0.1067475 , 0.77404734, 0.77656633,
1.01982356, 0.13172862, 0.04571019, 0.44089952, 0.66681869,
0.83938323, 0.20118698, 0.89587575, 0.1062632 , 0.28874488,
0.08948849, -0.0243854 , 0.06343322, 0.10938967, 0.76124064,
0.80452712, 0.07413591, 0.09288351, 0.16794007, 0.1124509 ,
0.62641129, 0.97936367, 0.88545028, 0.91500611, 0.20039097,
0.55112955, 0.50627899, 0.11044259, 0.05121753, 0.30678393,
0.10403542, 0.29983521, 0.46175864, 0.10477112, 0.61720283,
0.11290175, 1.11764685, 0.42743347, 0.68585748, 0.07836586,
0.69112287, 0.59377434, 0.61996372, 0.76980611, 0.68645524,
0.66991204, 0.2921748 , 0.05082041, 0.14783109])

```

```
[134]: survival=pd.DataFrame({"Actually_Survived":y_test,"Predicted_Survival":y_pred})
survival
```

```
[134]:
```

| | Actually_Survived | Predicted_Survival |
|-----|-------------------|--------------------|
| 767 | 0 | 0.651587 |
| 303 | 1 | 0.900725 |
| 80 | 0 | 0.137149 |
| 500 | 0 | 0.172680 |
| 291 | 1 | 0.988944 |
| .. | ... | ... |
| 47 | 1 | 0.686455 |
| 502 | 0 | 0.669912 |
| 181 | 0 | 0.292175 |
| 471 | 0 | 0.050820 |
| 408 | 0 | 0.147831 |

```
[174 rows x 2 columns]
```

```
[135]: from sklearn import metrics
```

```
[138]: #evaluating testing accuracy
metrics.r2_score(y_test,y_pred)
```

[138]: 0.46493081773565037