

NAME:B CHAITANYA KUMAR

REG NO:21BCE7581

import libraries

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

Importing the dataset

```
In [2]: df=pd.read_csv("Titanic-Dataset.csv")
```

```
In [3]: df.head()
```

```
Out[3]:
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
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 [4]: df.describe()
```

Out[4]:

	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 [5]:

`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 [6]:

`df.corr()`

```
C:\Users\chait\AppData\Local\Temp\ipykernel_19424\1134722465.py:1: FutureWarning: The
default value of numeric_only in DataFrame.corr is deprecated. In a future version, i
t will default to False. Select only valid columns or specify the value of numeric_on
ly to silence this warning.
 df.corr()
```

Out[6]:

	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 [7]: `df.corr().Survived.sort_values(ascending=False)`

```
C:\Users\chait\AppData\Local\Temp\ipykernel_19424\1717937034.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.
```

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

Out[7]:

Survived	1.000000
Fare	0.257307
Parch	0.081629
PassengerId	-0.005007
SibSp	-0.035322
Age	-0.077221
Pclass	-0.338481
Name: Survived, dtype: float64	

null values

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	

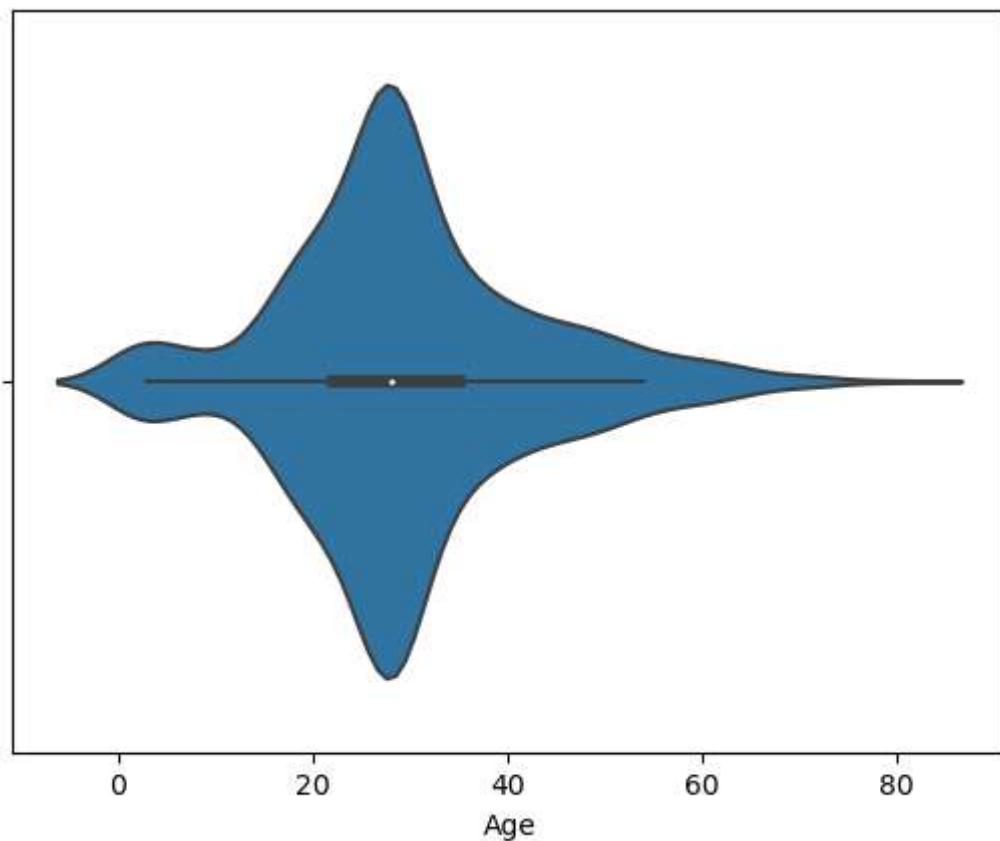
In [9]: `df['Cabin'].fillna(df['Cabin'].mode()[0], inplace=True)`In [10]: `df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)`In [11]: `median_age = df['Age'].median()
df['Age'].fillna(median_age, inplace=True)`In [12]: `df.isnull().sum()`

```
Out[12]: PassengerId      0
          Survived        0
          Pclass          0
          Name           0
          Sex            0
          Age            0
          SibSp          0
          Parch          0
          Ticket         0
          Fare           0
          Cabin          0
          Embarked       0
          dtype: int64
```

4) data visualization

```
In [13]: sns.violinplot(data=df, x="Age")
```

```
Out[13]: <Axes: xlabel='Age'>
```

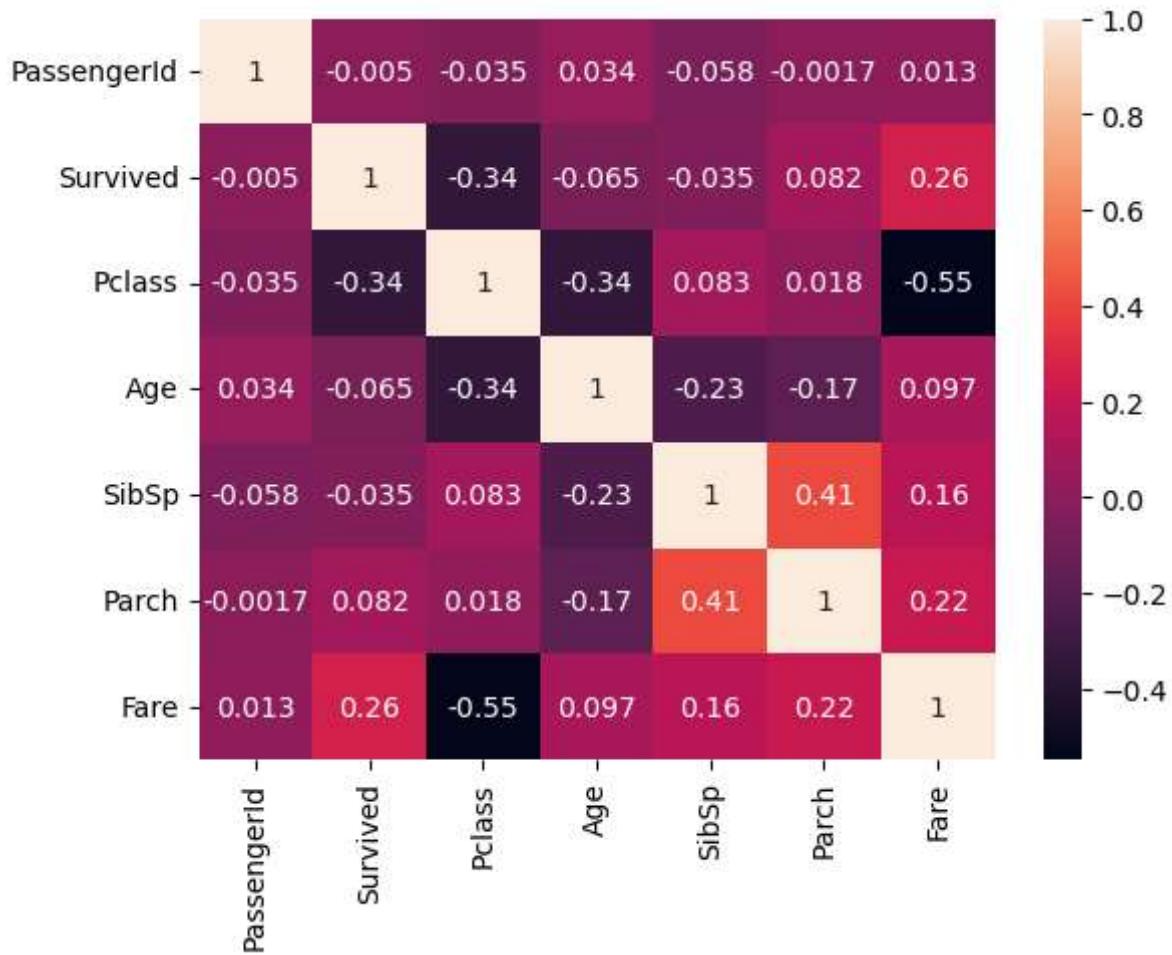


```
In [14]: sns.heatmap(df.corr(), annot=True)
```

```
C:\Users\chait\AppData\Local\Temp\ipykernel_19424\4277794465.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.
```

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

```
Out[14]: <Axes: >
```



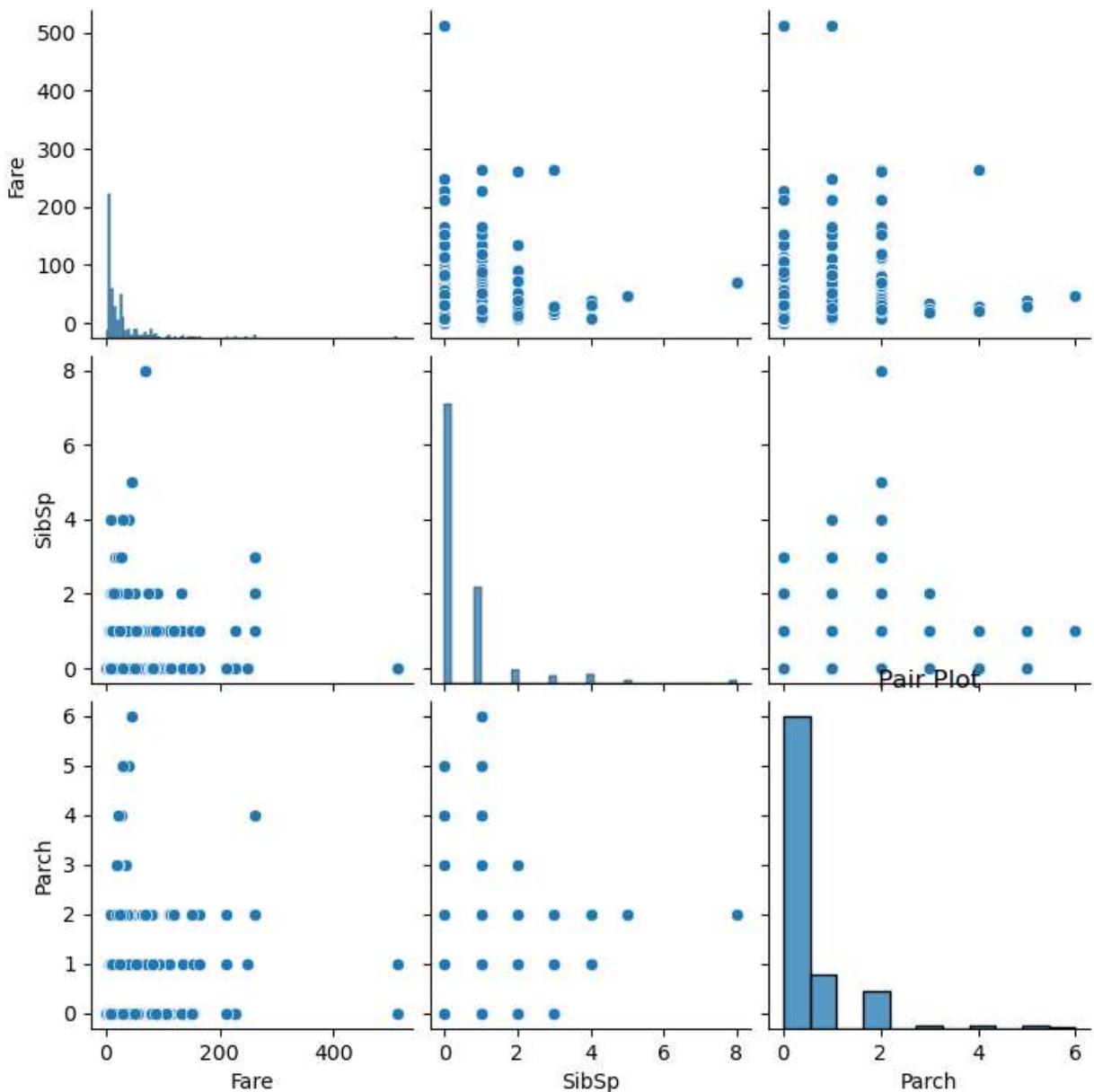
In [15]: `sns.pairplot(df)`

Out[15]: `<seaborn.axisgrid.PairGrid at 0x2c3756e4950>`

Assignment 3



```
In [16]: sns.pairplot(data=df[['Fare', 'SibSp', 'Parch']])
plt.title('Pair Plot')
plt.show()
```



```
In [17]: sns.barplot(x=df["Survived"],y=df["Age"],ci=0)
```

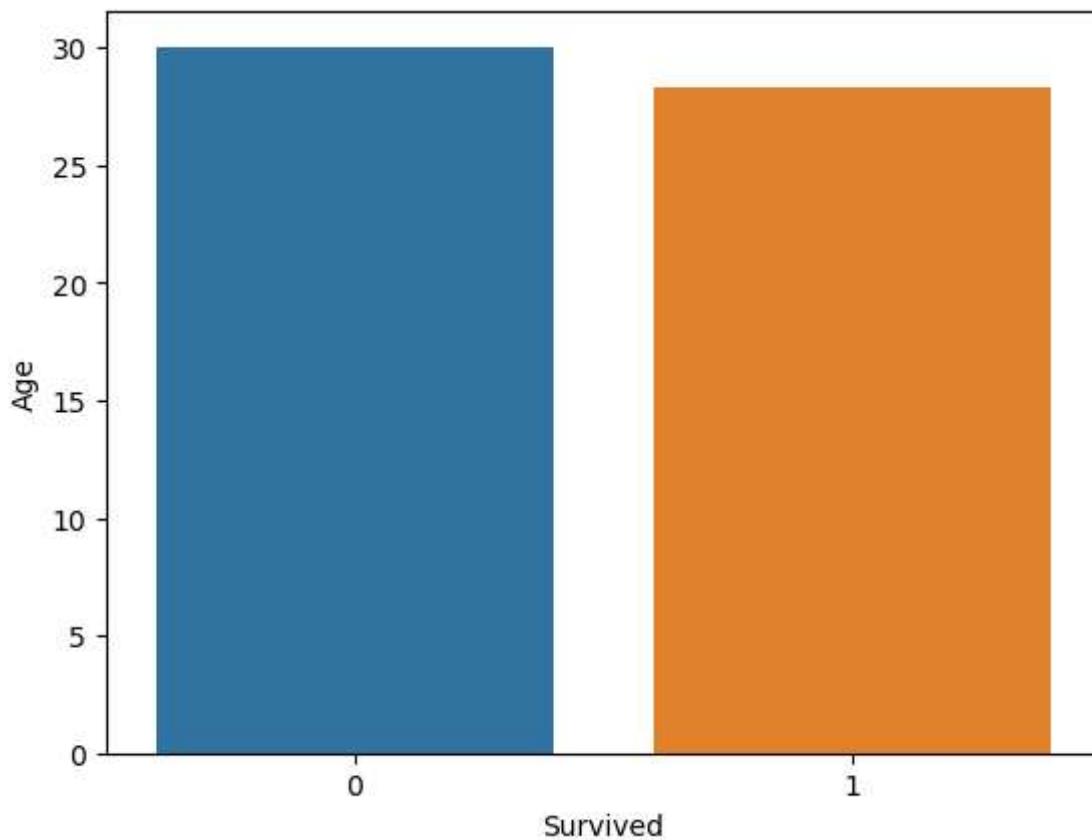
```
C:\Users\chait\AppData\Local\Temp\ipykernel_19424\332883349.py:1: FutureWarning:
```

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

```
    sns.barplot(x=df["Survived"],y=df["Age"],ci=0)
```

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

```
Out[17]:
```



5)outlier detection

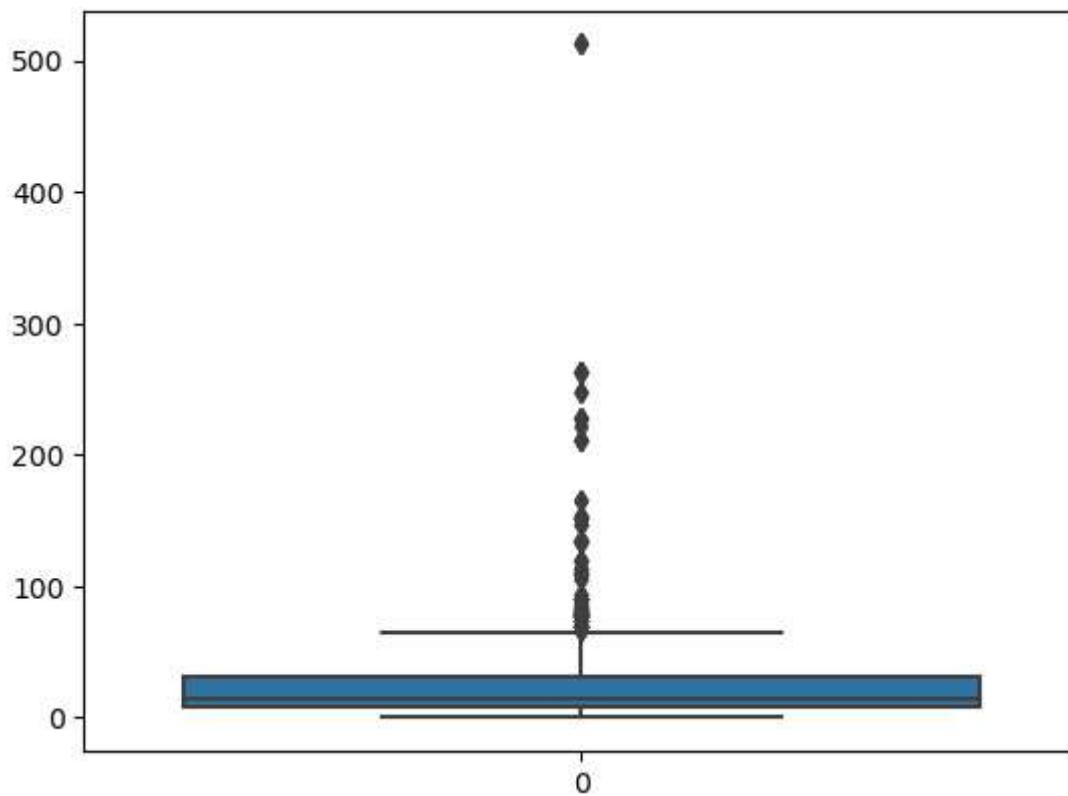
In [18]: `df.head()`

Out[18]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96 B98	
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	B96 B98	
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	B96 B98	

In [19]: `sns.boxplot(df["Fare"])`

Out[19]: <Axes: >



```
In [20]: df.shape
```

```
Out[20]: (891, 12)
```

```
In [21]: q1 = df.Fare.quantile(0.25)  
q3 = df.Fare.quantile(0.75)
```

```
In [22]: q1
```

```
Out[22]: 7.9104
```

```
In [23]: q3
```

```
Out[23]: 31.0
```

```
In [24]: IQR=q3-q1
```

```
In [25]: IQR
```

```
Out[25]: 23.0896
```

```
In [26]: upper_limit = q3+1.5*IQR
```

```
In [27]: upper_limit
```

```
Out[27]: 65.6344
```

```
In [28]: lower_limit=q1-1.5*IQR
```

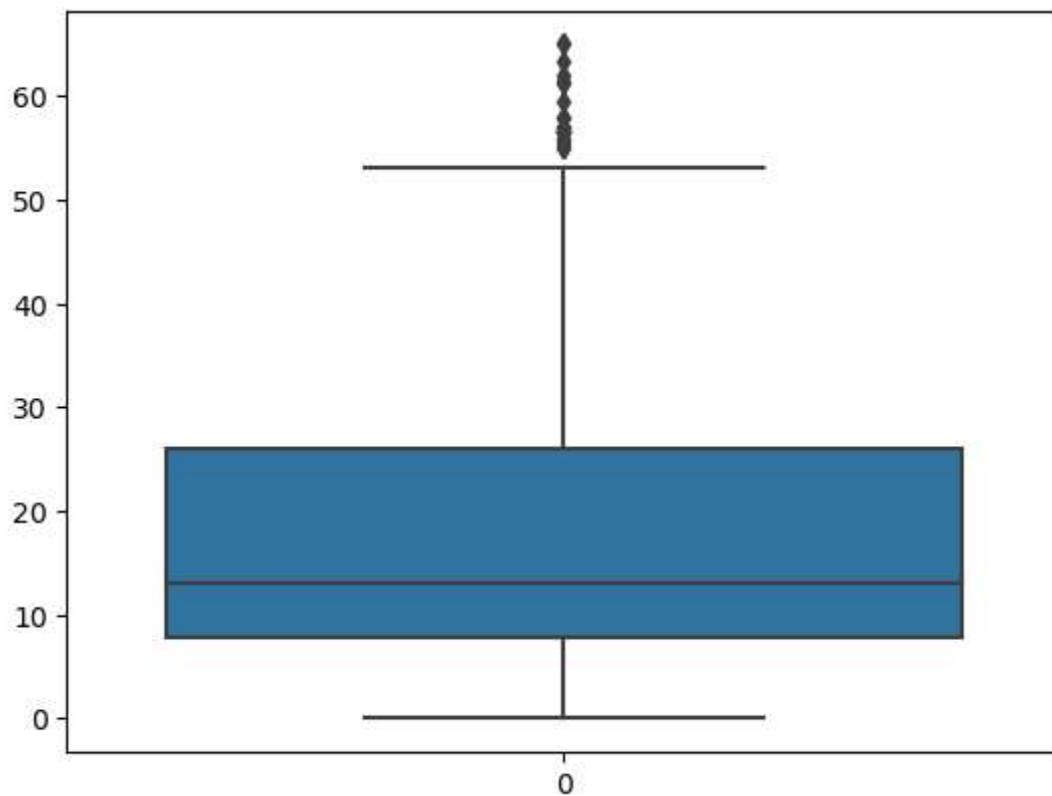
```
In [29]: lower_limit
```

```
Out[29]: -26.724
```

```
In [30]: df = df[df.Fare<upper_limit]
```

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

```
Out[31]: <Axes: >
```

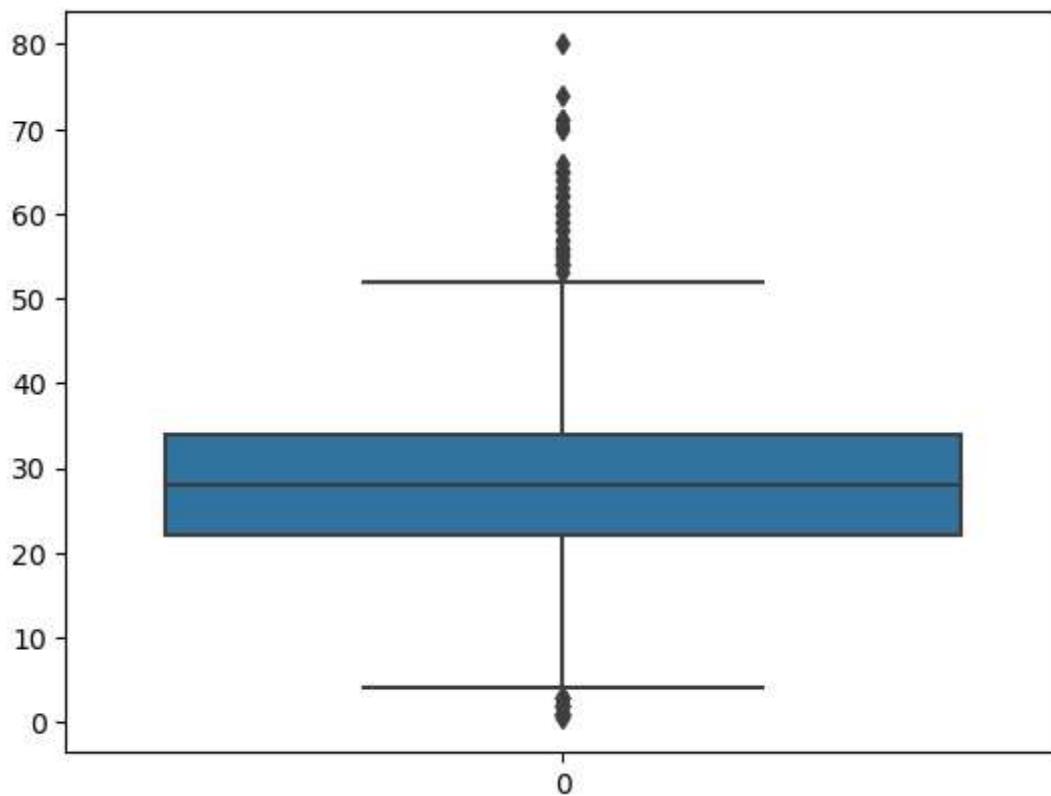


```
In [32]: df.shape
```

```
Out[32]: (775, 12)
```

```
In [33]: sns.boxplot(df["Age"])
```

```
Out[33]: <Axes: >
```



```
In [34]: df.shape
```

```
Out[34]: (775, 12)
```

```
In [35]: q1 = df.Age.quantile(0.25)  
q3 = df.Age.quantile(0.75)
```

```
In [36]: q1
```

```
Out[36]: 22.0
```

```
In [37]: q3
```

```
Out[37]: 34.0
```

```
In [38]: IQR=q3-q1
```

```
In [39]: IQR
```

```
Out[39]: 12.0
```

```
In [40]: upper_limit = q3+1.5*IQR
```

```
In [41]: upper_limit
```

```
Out[41]: 52.0
```

```
In [42]: lower_limit=q1-1.5*IQR
```

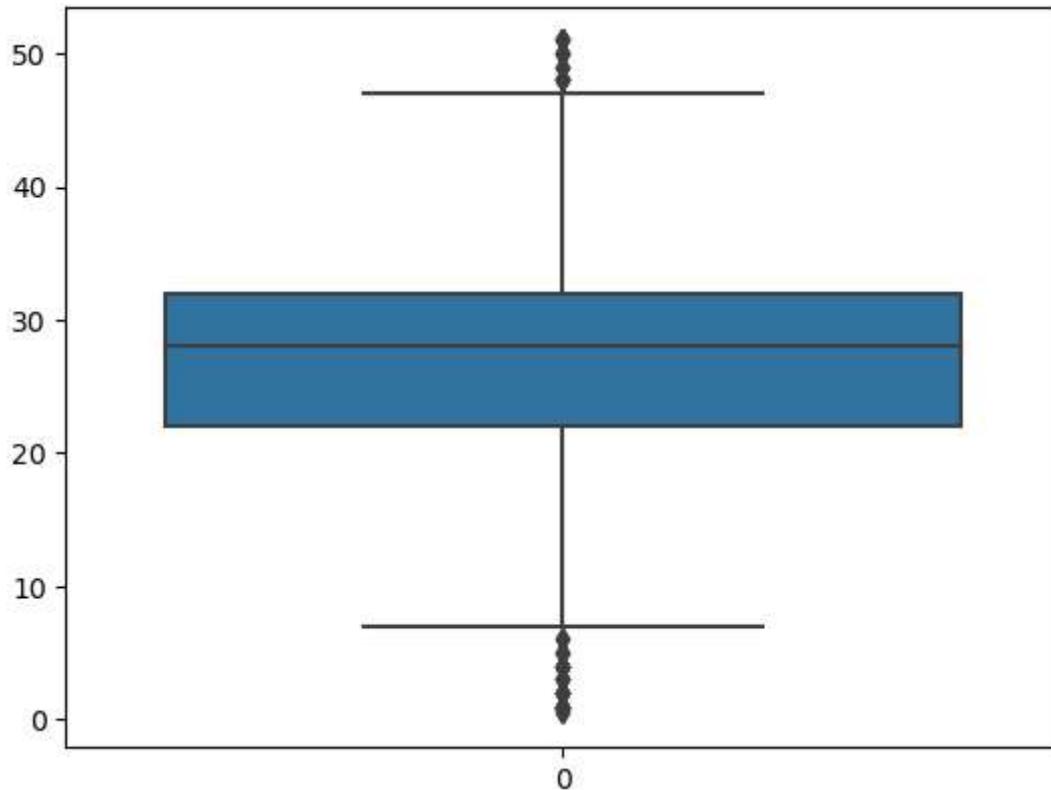
```
In [43]: lower_limit
```

```
Out[43]: 4.0
```

```
In [44]: df = df[df.Age < upper_limit]
```

```
In [45]: sns.boxplot(df.Age)
```

```
Out[45]: <Axes: >
```



```
In [46]: df.shape
```

```
Out[46]: (733, 12)
```

```
In [47]: df.head()
```

Out[47]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96 B98	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	B96 B98	
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	B96 B98	
5	6	0	3	Moran, Mr. James	male	28.0	0	0	330877	8.4583	B96 B98	

In [48]: X=df.drop(columns=["Survived"],axis=1)
X.head()

Out[48]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96 B98	S
2	3	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	B96 B98	S
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	B96 B98	S
5	6	3	Moran, Mr. James	male	28.0	0	0	330877	8.4583	B96 B98	Q

In [49]: X.shape

Out[49]: (733, 11)

In [50]: type(X)

Out[50]: pandas.core.frame.DataFrame

```
In [51]: y=df["Survived"]
y.head()
```

```
Out[51]: 0    0
2    1
3    1
4    0
5    0
Name: Survived, dtype: int64
```

encoding

In [52]: X.head()

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96 B98	S
2	3	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	B96 B98	S
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	B96 B98	S
5	6	3	Moran, Mr. James	male	28.0	0	0	330877	8.4583	B96 B98	Q

In [53]: `from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()`

In [54]: `X["Cabin"]=le.fit_transform(X["Cabin"])`

In [55]: X.head()

Out[55]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96 B98	S
2	3	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	B96 B98	S
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	B96 B98	S
5	6	3	Moran, Mr. James	male	28.0	0	0	330877	8.4583	B96 B98	Q

In [56]: `print(le.classes_)`

```
['A10' 'A14' 'A16' 'A19' 'A20' 'A24' 'A31' 'A32' 'A36' 'A6' 'B102' 'B18'
 'B20' 'B38' 'B39' 'B4' 'B42' 'B50' 'B51 B53 B55' 'B71' 'B94' 'B96 B98'
 'C106' 'C110' 'C111' 'C118' 'C123' 'C124' 'C126' 'C128' 'C148' 'C47'
 'C49' 'C52' 'C90' 'D' 'D10 D12' 'D17' 'D19' 'D21' 'D28' 'D30' 'D35' 'D45'
 'D46' 'D47' 'D56' 'D6' 'E10' 'E101' 'E12' 'E121' 'E17' 'E24' 'E25' 'E31'
 'E33' 'E36' 'E44' 'E50' 'E58' 'E63' 'E8' 'F E69' 'F G63' 'F G73' 'F2'
 'F33' 'F38' 'F4' 'G6' 'T']
```

In [57]: `X[["Embarke"]]=le.fit_transform(X[["Embarked"]])`

In [58]: `X.head()`

Out[58]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96 B98	S
2	3	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	B96 B98	S
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	B96 B98	S
5	6	3	Moran, Mr. James	male	28.0	0	0	330877	8.4583	B96 B98	Q

In [59]: `print(le.classes_)`

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

In [60]: `mapping=dict(zip(le.classes_, range(len(le.classes_))))`
`mapping`

Out[60]: `{'C': 0, 'Q': 1, 'S': 2}`

scaling

In [61]: `from sklearn.preprocessing import StandardScaler`

In [62]: `numerical_columns = ['Age', 'SibSp', 'Parch', 'Fare']`

In [63]: `scaler = StandardScaler()`

In [64]: `df[numerical_columns] = scaler.fit_transform(df[numerical_columns])`

In [65]: `df`

Out[65]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	
0	1	0	3	Braund, Mr. Owen Harris	male	-0.471318	0.595411	-0.443400	A/5 21171	-0.7
2	3	1	3	Heikkinen, Miss. Laina	female	-0.090234	-0.495680	-0.443400	STON/O2. 3101282	-0.7
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	0.767204	0.595411	-0.443400	113803	2.6
4	5	0	3	Allen, Mr. William Henry	male	0.767204	-0.495680	-0.443400	373450	-0.6
5	6	0	3	Moran, Mr. James	male	0.100307	-0.495680	-0.443400	330877	-0.6
...
886	887	0	2	Montvila, Rev. Juozas	male	0.005036	-0.495680	-0.443400	211536	-0.3
887	888	1	1	Graham, Miss. Margaret Edith	female	-0.757131	-0.495680	-0.443400	112053	0.9
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	0.100307	0.595411	2.066345	W./C. 6607	0.4
889	890	1	1	Behr, Mr. Karl Howell	male	-0.090234	-0.495680	-0.443400	111369	0.9
890	891	0	3	Dooley, Mr. Patrick	male	0.481391	-0.495680	-0.443400	370376	-0.7

733 rows × 12 columns

data train and test

In [66]: `from sklearn.model_selection import train_test_split`In [67]: `X = df[['Pclass', 'Age', 'SibSp', 'Parch']] # Use relevant columns as features`
`y = df['Survived'] # The column you want to predict`

```
In [68]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [69]: print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(586, 4)
(147, 4)
(586,)
(147,)
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```