Sourav Samaddar 21MIS1046
sourav.samaddar2021@vitstudent.ac.in

Assignment-3 (15-08-23)

Problem

```
Assignment 15 sep
Perform Data preprocessing on Titanic dataset
1.Data Collection.
   Please download the dataset from
   https://www.kaggle.com/datasets/yasserh/titanic-dataset

2.Data Preprocessing
    o   Import the Libraries.
    o   Importing the dataset.
    o   Checking for Null Values.
    o   Data Visualization.
    o   Outlier Detection
    o   Splitting Dependent and Independent variables
    o   Perform Encoding
    o   Feature Scaling.
    o   Splitting Data into Train and Test
```

1.Importing Libraries

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

2.Importing the Dataset

+ Code + Text

```
[2] dataset=pd.read_csv("Titanic-Dataset.csv")
```

```
[3] dataset
```

	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	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine	female	NaN	1	2	W./C. 6607	23.4500	NaN	S

```
[4] dataset.head()
```

	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	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
dataset.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
```

```
[6] dataset.describe()
```

	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

3.Null values

```
dataset.isnull()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False	True	False

```
[11] dataset.isnull().any()
```

```
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
```

```
[12] dataset.isnull().sum()
```

```
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
```

Handling Null Values with Mean and Median

```
[18] dataset['Age'].fillna(int(dataset['Age'].mean()), inplace=True)
```

```
[29] dataset['Cabin'] = dataset['Cabin'].fillna(dataset['Cabin'].mode()[0])
```

```
[31] dataset['Embarked'] = dataset['Embarked'].fillna(dataset['Embarked'].mode()[0])
```

```
[32] dataset.isnull().sum()
```

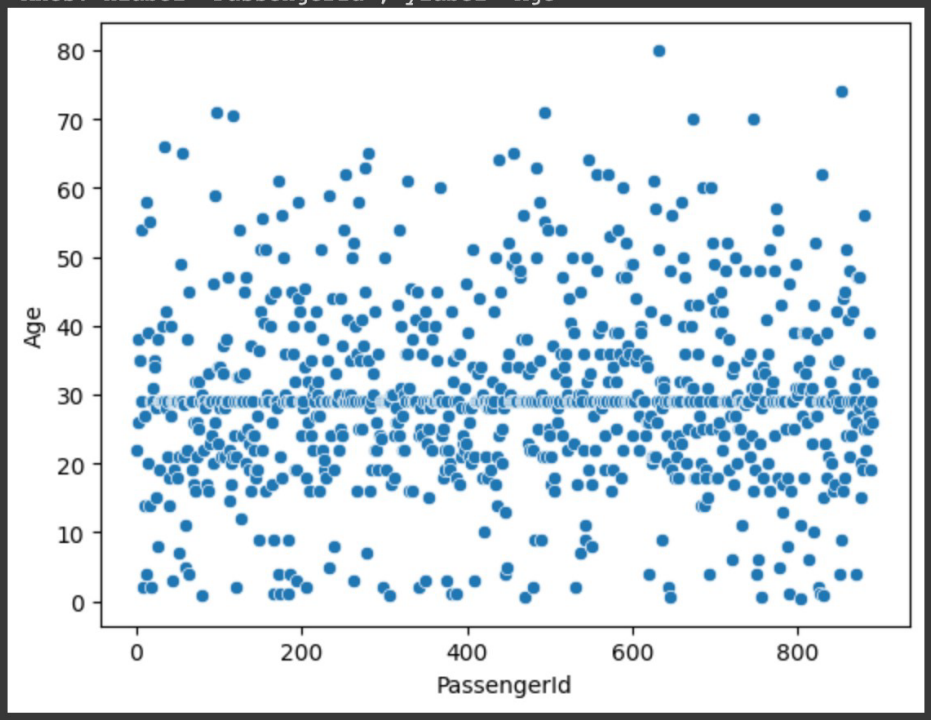
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

4.Data Visualization

```
[37] df=dataset
```

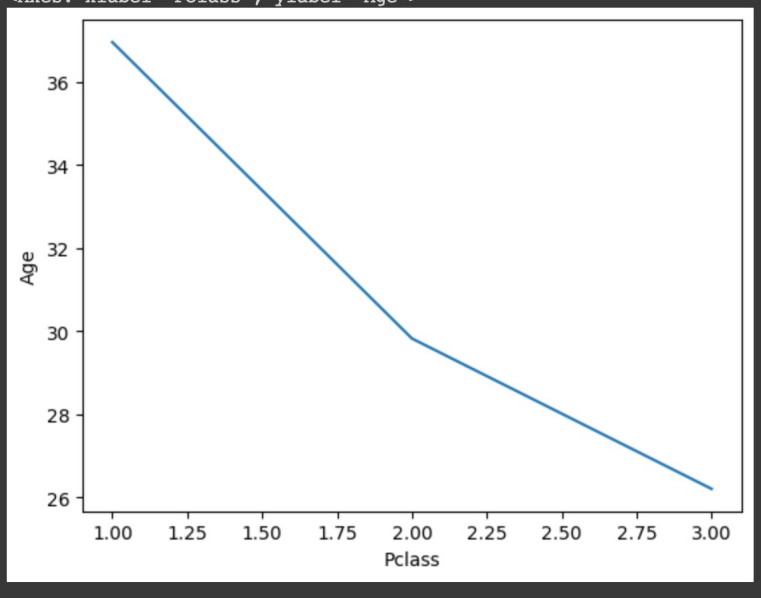
```
sns.scatterplot(x="PassengerId",y="Age",data=df)
```

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



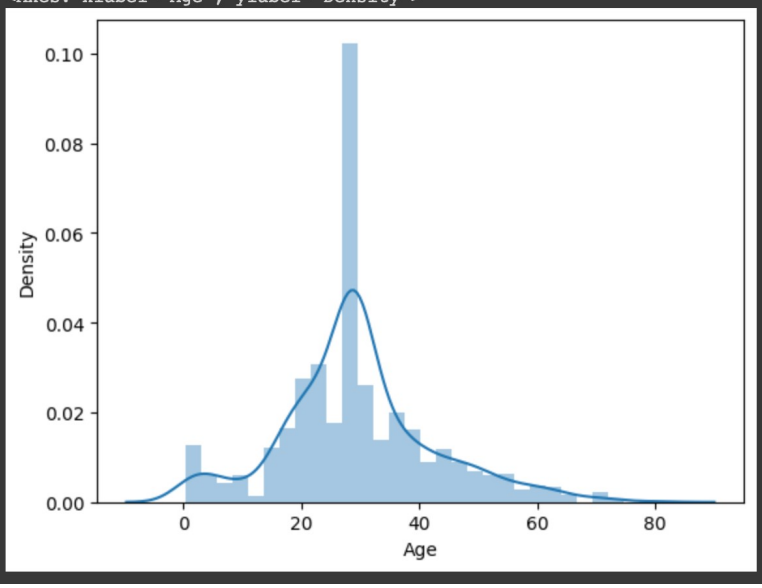
```
sns.lineplot(x="Pclass",y="Age",data=df,ci=None)
```

```
<ipython-input-41-49081a25326a>:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(x="Pclass",y="Age",data=df,ci=None)
<Axes: xlabel='Pclass', ylabel='Age'>
```



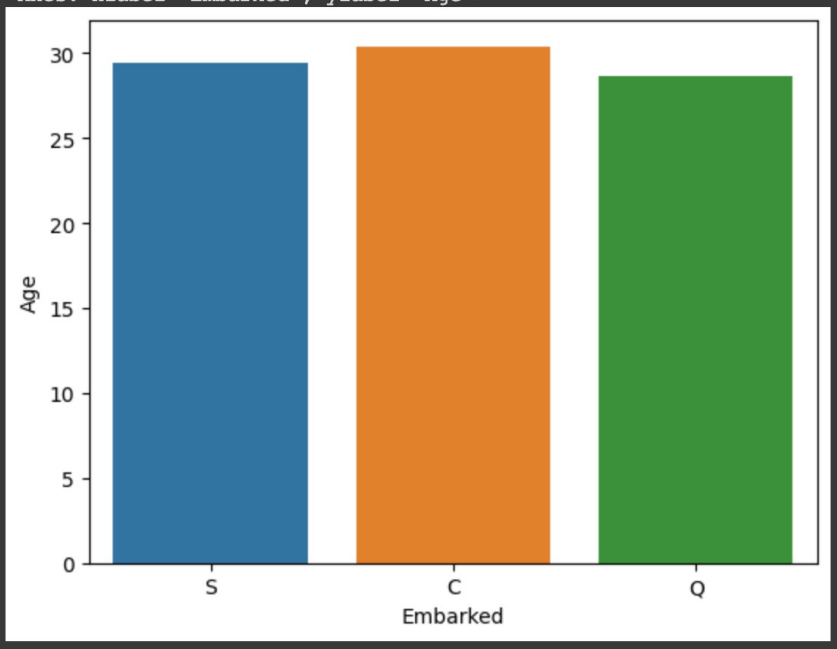

```
sns.distplot(df["Age"])
```

```
<ipython-input-42-cf0334540b62>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df["Age"])
<Axes: xlabel='Age', ylabel='Density'>
```



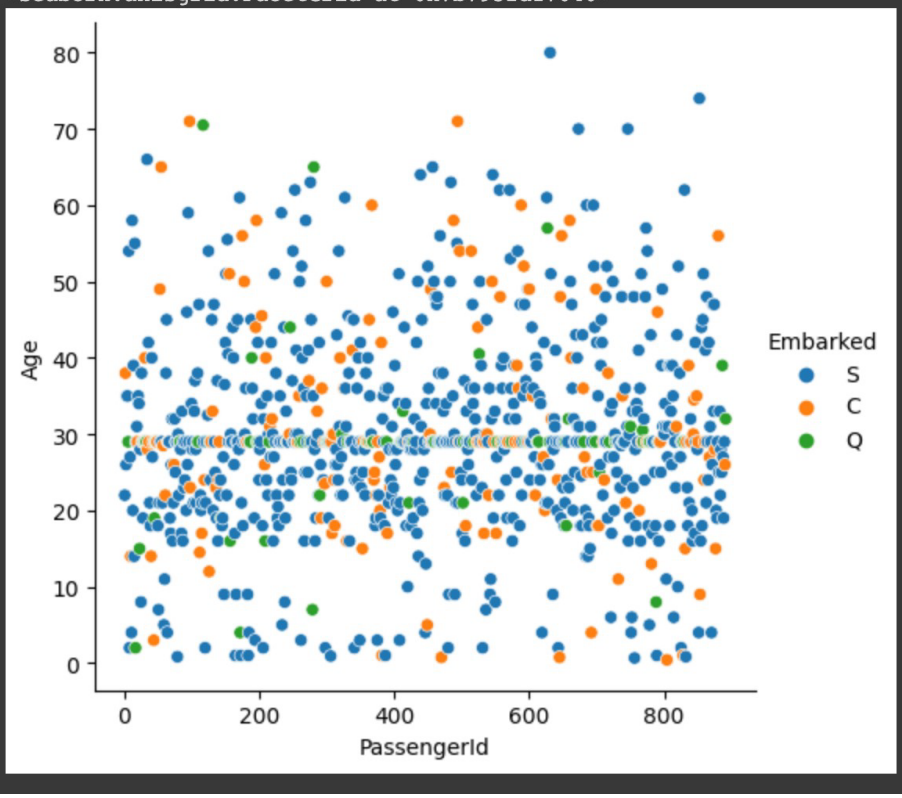
```
[45] sns.barplot(data=df,x="Embarked",y="Age",ci=None)
```

```
<ipython-input-45-d520e7c42b53>:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(data=df,x="Embarked",y="Age",ci=None)
<Axes: xlabel='Embarked', ylabel='Age'>
```



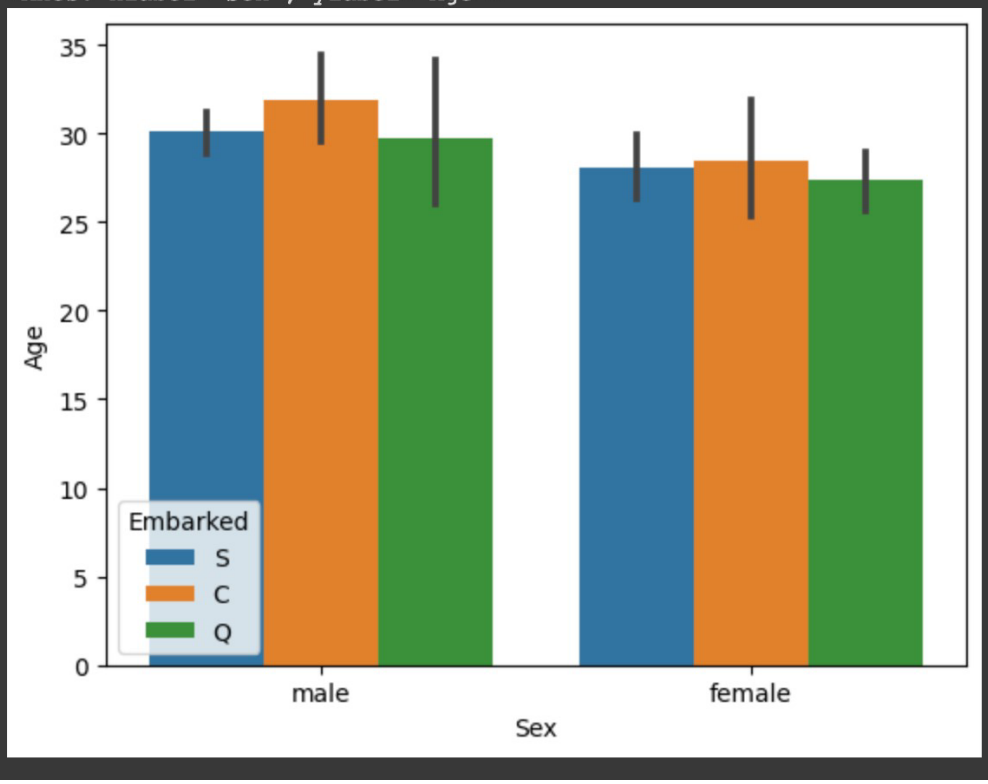

```
[43] sns.relplot(x="PassengerId",y="Age",data=df,hue="Embarked")
```

```
<seaborn.axisgrid.FacetGrid at 0x7b7952d17040>
```



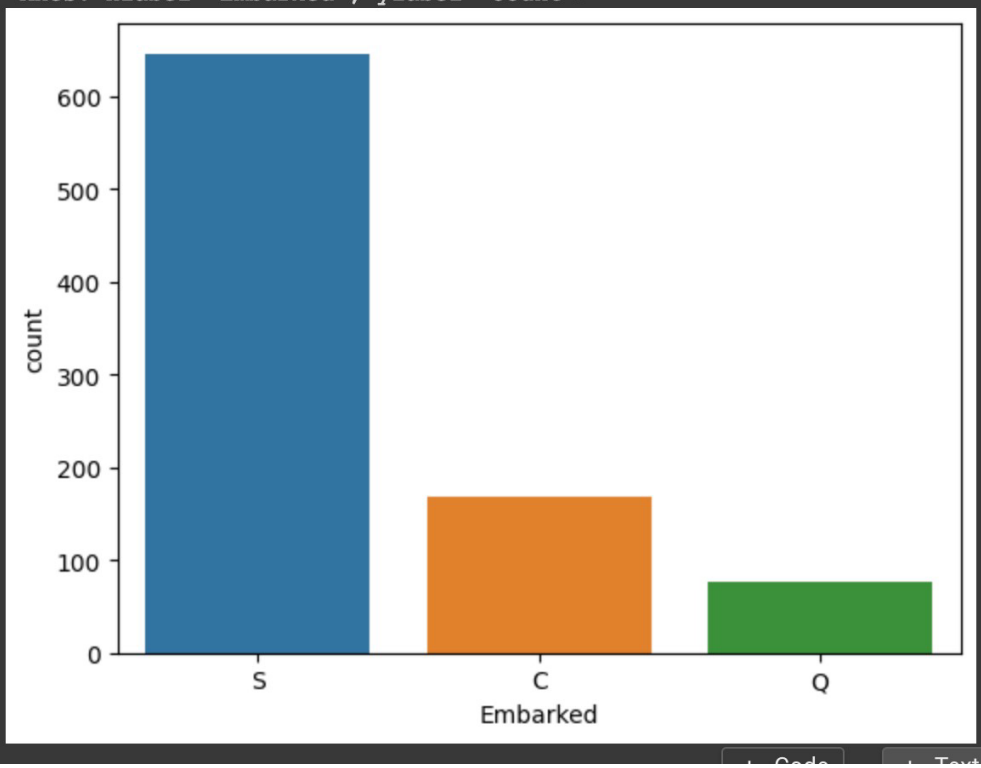
```
[46] sns.barplot(data=df,x="Sex",y="Age",hue="Embarked")
```

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



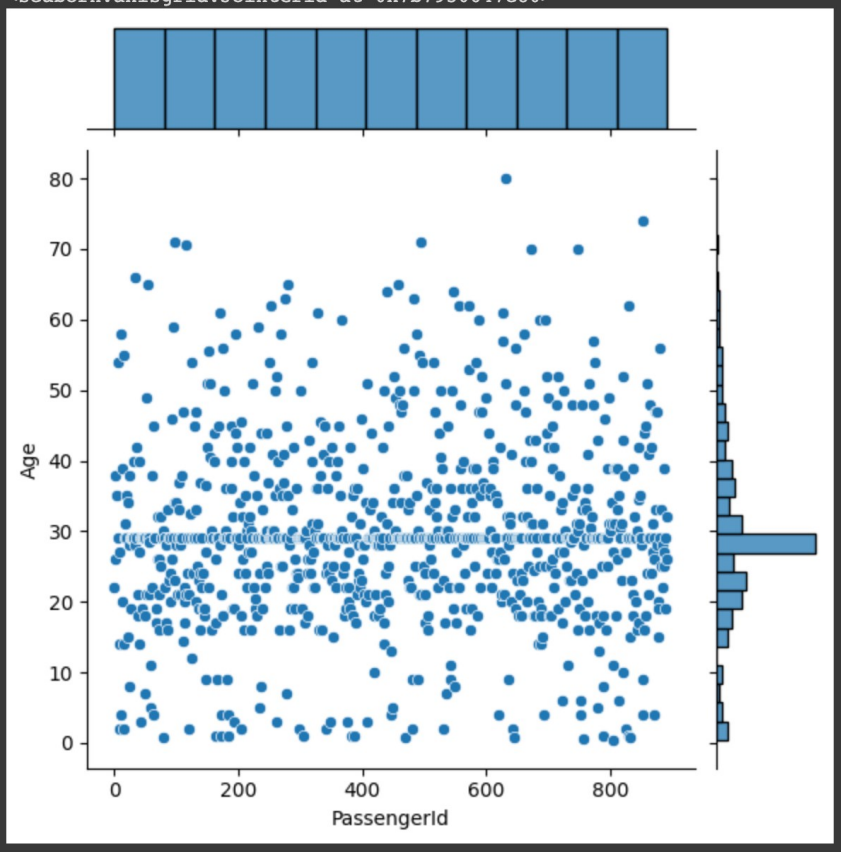

```
sns.countplot(x="Embarked",data=df)
```

```
<Axes: xlabel='Embarked', ylabel='count'>
```



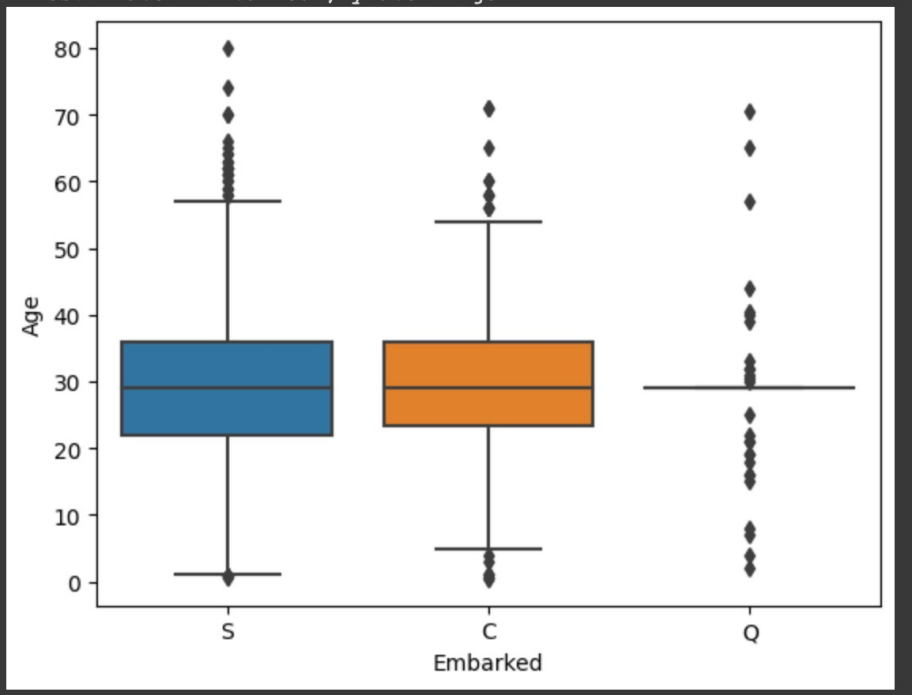
```
sns.jointplot(x="PassengerId",y="Age",data=df)
```

```
<seaborn.axisgrid.JointGrid at 0x7b7950647e80>
```




```
[53] sns.boxplot(x="Embarked",y="Age",data=df)
```

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



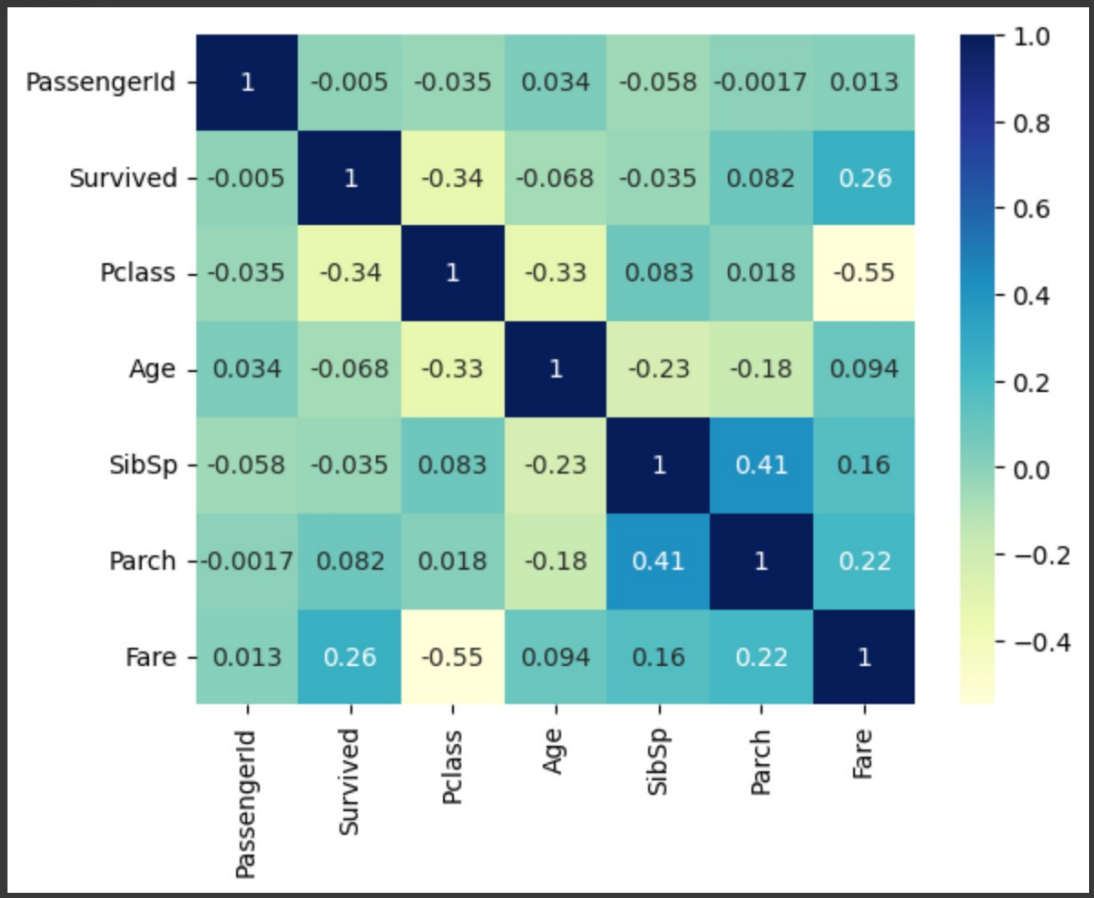
```
[55] corr=df.corr()
     corr
```

```
<ipython-input-55-7d5195e2bf4d>:1: FutureWarning: The default value of numeric_only in
  corr=df.corr()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.033632	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.067814	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.334974	0.083081	0.018443	-0.549500
Age	0.033632	-0.067814	-0.334974	1.000000	-0.232978	-0.176486	0.093706
SibSp	-0.057527	-0.035322	0.083081	-0.232978	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.176486	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.093706	0.159651	0.216225	1.000000

```
sns.heatmap(corr,annot=True,cmap="YlGnBu")
```

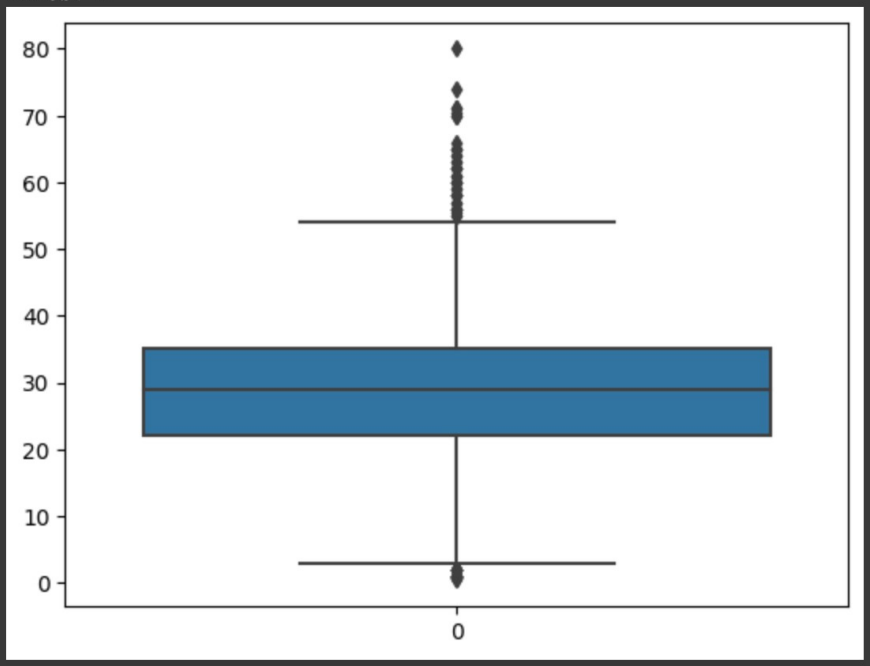
```
<Axes: >
```



5.Outlier Detection

```
[58] sns.boxplot(dataset.Age)
```

```
<Axes: >
```



6.Splitting Dependent and Independent variables.

7.Perform Encoding

```
[98] x=dataset.iloc[:,1:3]
     x1=dataset.iloc[:,4:8]
     c=pd.concat([x,x1],axis=1)
     y=dataset.iloc[:,10:12]
```

```
[99] c.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch
0	0	3	male	22.0	1	0
1	1	1	female	38.0	1	0
2	1	3	female	26.0	0	0
3	1	1	female	35.0	1	0
4	0	3	male	35.0	0	0

```
y.head()
```

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

```
[101] from sklearn.preprocessing import LabelEncoder
```

```
[102] le=LabelEncoder()
```

```
[103] c["Sex"]=le.fit_transform(c["Sex"])
```

```
[104] c["Sex"]
```

```
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: int64
```

```
[105] c["Sex"].value_counts()
```

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

```
[108] c["Sex"].nunique()
```

```
2
```

```
[107] c.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch
0	0	3	1	22.0	1	0
1	1	1	0	38.0	1	0
2	1	3	0	26.0	0	0
3	1	1	0	35.0	1	0
4	0	3	1	35.0	0	0

```
[129] c.Pclass.value_counts()
```

```
3    491
1    216
2    184
Name: Pclass, dtype: int64
```

```
[131] Pclass=pd.get_dummies(c["Pclass"],drop_first=True)
```

```
Pclass
```

	2	3
0	0	1
1	0	0
2	0	1
3	0	0
4	0	1
...	...	...

```
[132] c=pd.concat([c,Pclass],axis=1)
```

```
[133] c.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	2	3
0	0	3	1	22.0	1	0	0	1
1	1	1	0	38.0	1	0	0	0
2	1	3	0	26.0	0	0	0	1
3	1	1	0	35.0	1	0	0	0
4	0	3	1	35.0	0	0	0	1

```
[134] c.drop(["Pclass"],axis=1,inplace=True)
      c.head()
```

	Survived	Sex	Age	SibSp	Parch	2	3
0	0	1	22.0	1	0	0	1
1	1	0	38.0	1	0	0	0
2	1	0	26.0	0	0	0	1
3	1	0	35.0	1	0	0	0
4	0	1	35.0	0	0	0	1

9.Splitting Data into Train and Test

```
[136] from sklearn.model_selection import train_test_split
      c_train,c_test,y_train,y_test=train_test_split(c,y,test_size=0.3,random_state=0)
```

```
[137] c_train.shape,c_test.shape,y_train.shape,y_test.shape
```

```
((623, 7), (268, 7), (623, 2), (268, 2))
```

8.Feature Scaling

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

```
      c_train=sc.fit_transform(c_train)
```

```
[147] x_test=sc.fit_transform(x_test)
```

```
      c_train
```

```
array([[ 1.25474307,  0.72592065,  1.63377655, ..., -0.47299765,
        -0.51849697, -1.07851493],
       [ 1.25474307, -1.37756104,  1.48009931, ..., -0.47299765,
        -0.51849697, -1.07851493],
       [-0.79697591,  0.72592065, -2.20815449, ...,  1.93253327,
        -0.51849697,  0.92720089],
       ...,
       [-0.79697591,  0.72592065, -0.05667311, ..., -0.47299765,
        -0.51849697,  0.92720089],
       [ 1.25474307, -1.37756104,  0.48119724, ..., -0.47299765,
        -0.51849697,  0.92720089],
       [-0.79697591,  0.72592065,  2.32532414, ...,  0.72976781,
         1.92865159, -1.07851493]])
```