ASSIGNMENT-3

September 21, 2023

KADE NAVANEESWAR GOWD VIT-AP

1 Import the Libraries

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

2 Importing the data set

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

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

[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	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

[4]:
```
df.describe()
```

[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

[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
```

[6]:
```
df.corr()
```

```
C:\Users\mb419\AppData\Local\Temp\ipykernel_21236\1134722465.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()
```

[6]:
```
             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
```

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

```
C:\Users\mb419\AppData\Local\Temp\ipykernel_21236\60082530.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().Fare.sort_values(ascending=False)
```

[7]:
```
Fare           1.000000
Survived       0.257307
Parch          0.216225
SibSp          0.159651
Age            0.096067
PassengerId    0.012658
Pclass        -0.549500
Name: Fare, dtype: float64
```

3 Checking for Null values

[8]:
```
df.isnull().any()
```

[8]:
```
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
```

[9]:
```
df.isnull().sum()
```

[9]:
```
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
```

4 Filling the Null values

[10]:
```
# Fill missing values in numerical columns with the mean
numerical_cols = ['Age', 'Fare']
for col in numerical_cols:
    df[col].fillna(df[col].mean(), inplace=True)

# Fill missing values in categorical columns with the mode
categorical_cols = ['Embarked', 'Cabin']
for col in categorical_cols:
    df[col].fillna(df[col].mode()[0], inplace=True)
```

[11]:
```
df.isnull().any()
```

[11]:
```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age            False
```

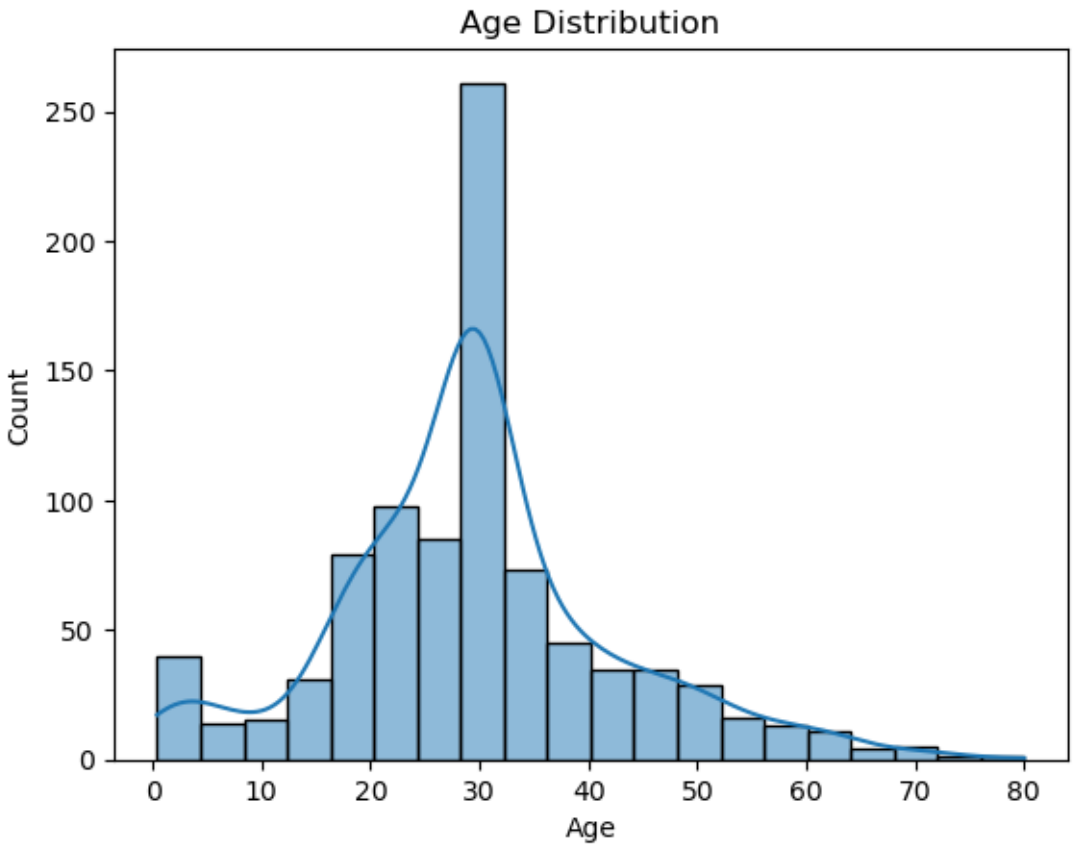
```
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin          False
Embarked       False
dtype: bool
```

[12]:
```
df.isnull().sum()
```

[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
```

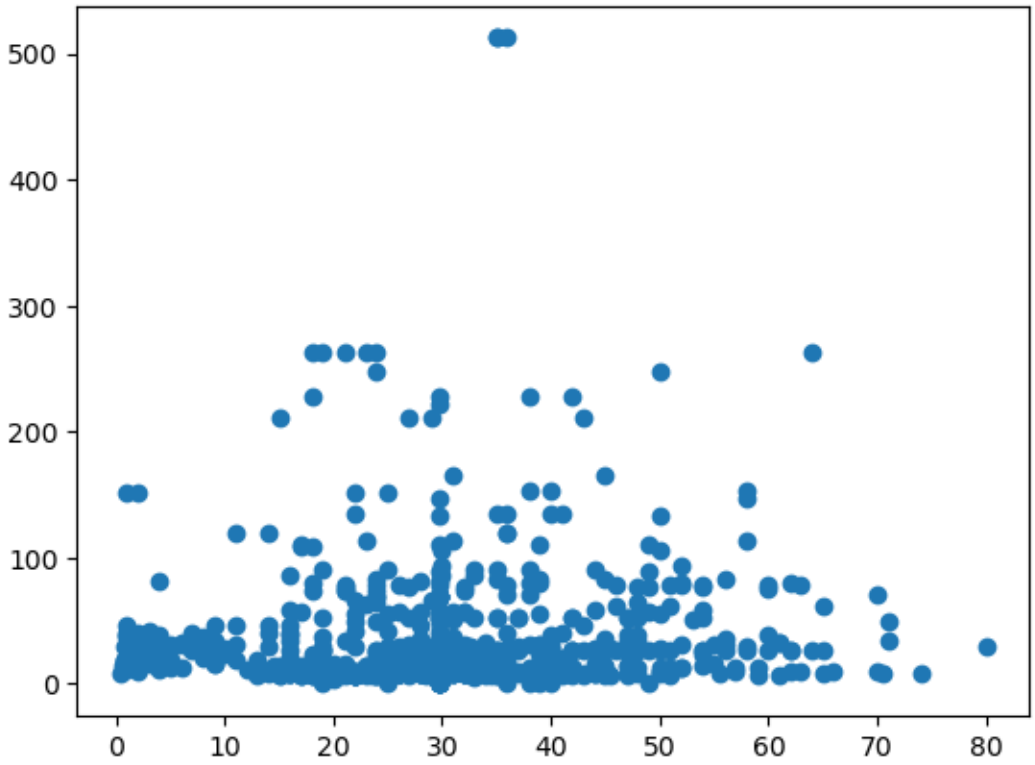
5 Data Visualization

[13]:
```
sns.histplot(df['Age'], bins=20, kde=True)
plt.title('Age Distribution')
plt.show()
```



[14]:
```
plt.scatter(df["Age"],df["Fare"])
```

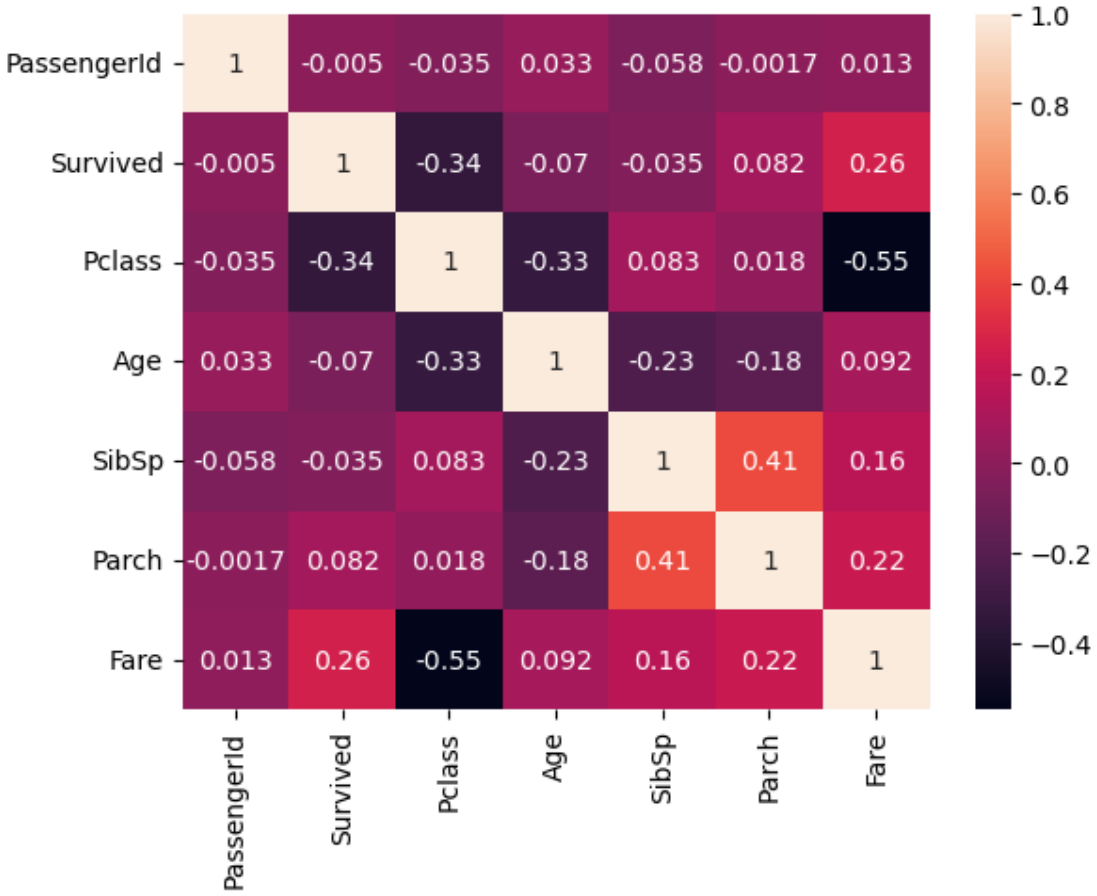
[14]:
```
<matplotlib.collections.PathCollection at 0x1591345cb90>
```



[15]:
```
sns.heatmap(df.corr(),annot=True)
```

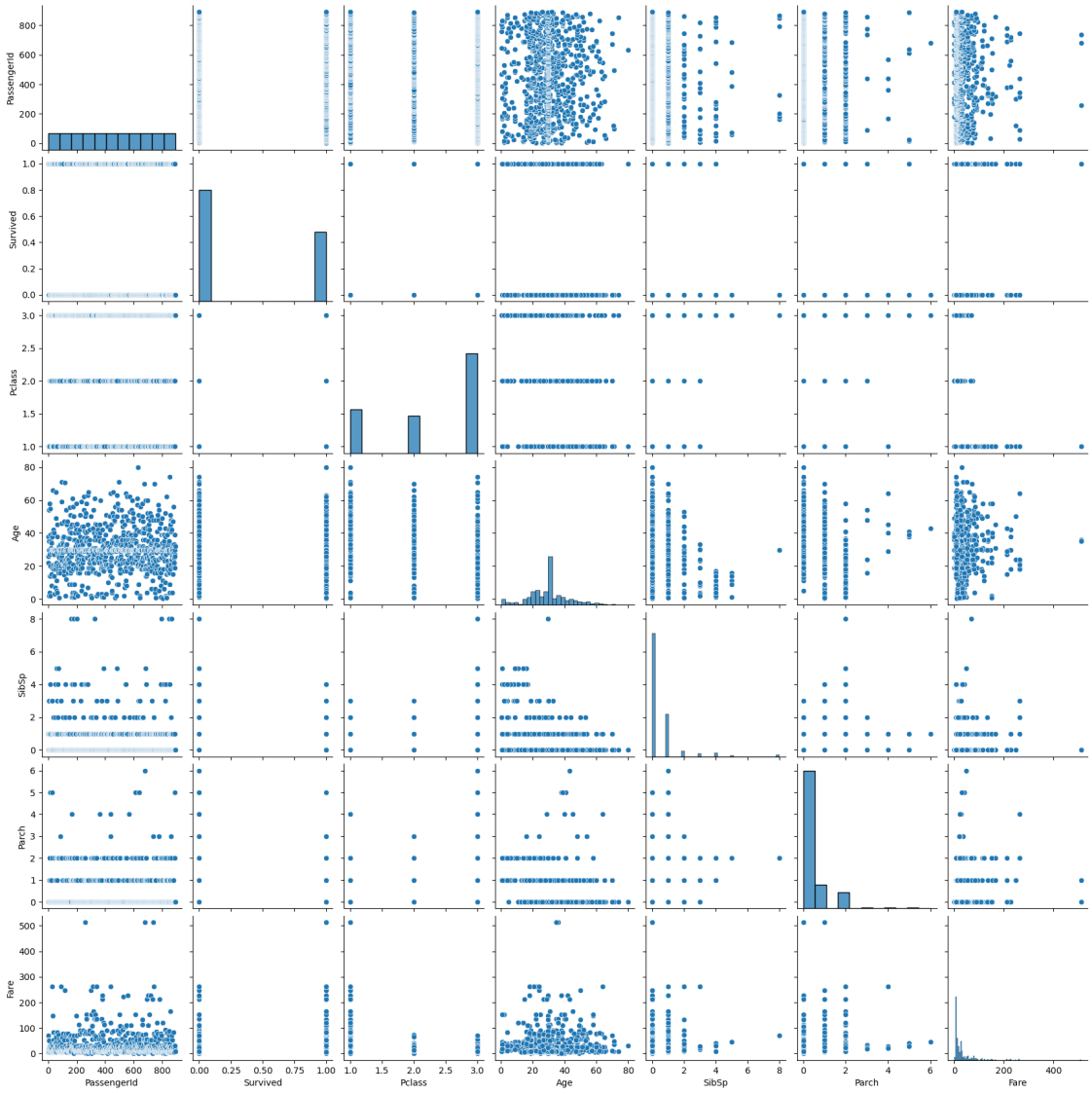
```
C:\Users\mb419\AppData\Local\Temp\ipykernel_21236\4277794465.py:1:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(df.corr(),annot=True)
```

[15]: <Axes: >

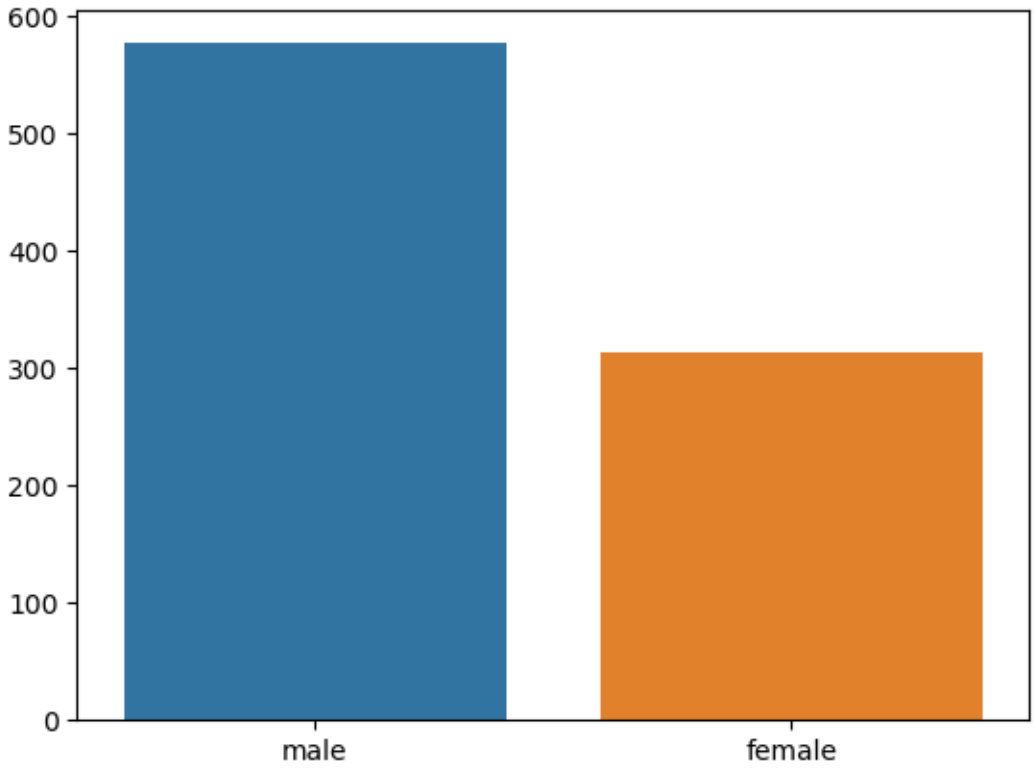


[16]:
```
sns.pairplot(df)
```

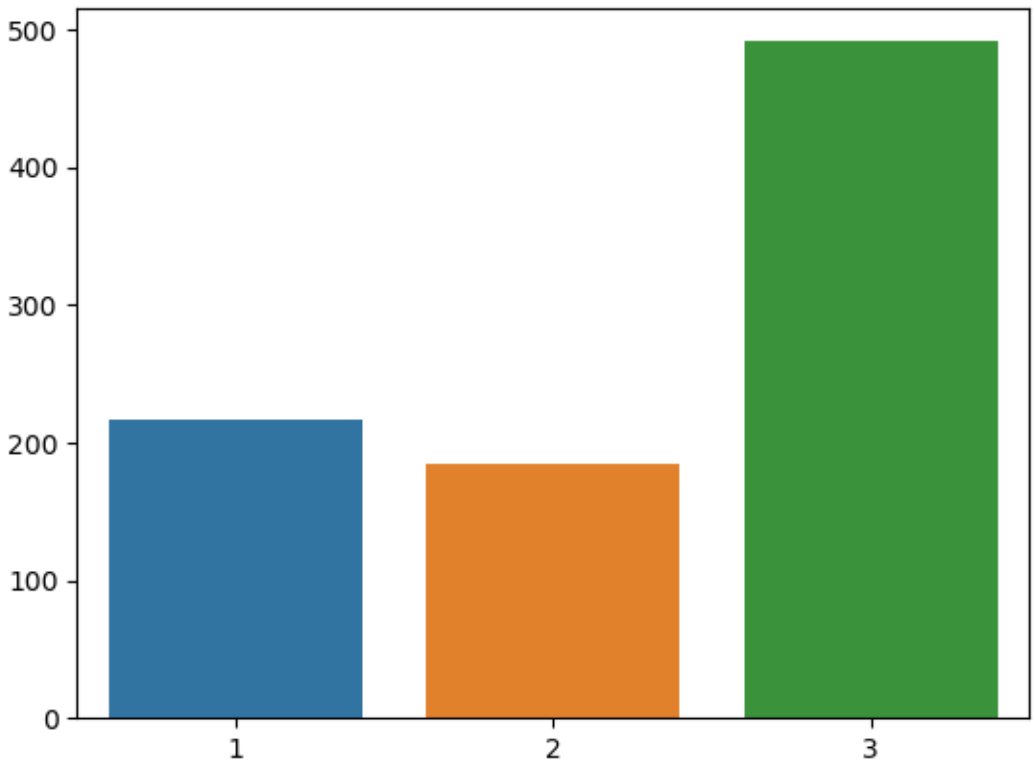
[16]: <seaborn.axisgrid.PairGrid at 0x15913501510>



[17]:
```
gender_count = df['Sex'].value_counts()
sns.barplot(x=gender_count.index, y=gender_count.values)
plt.show()
```



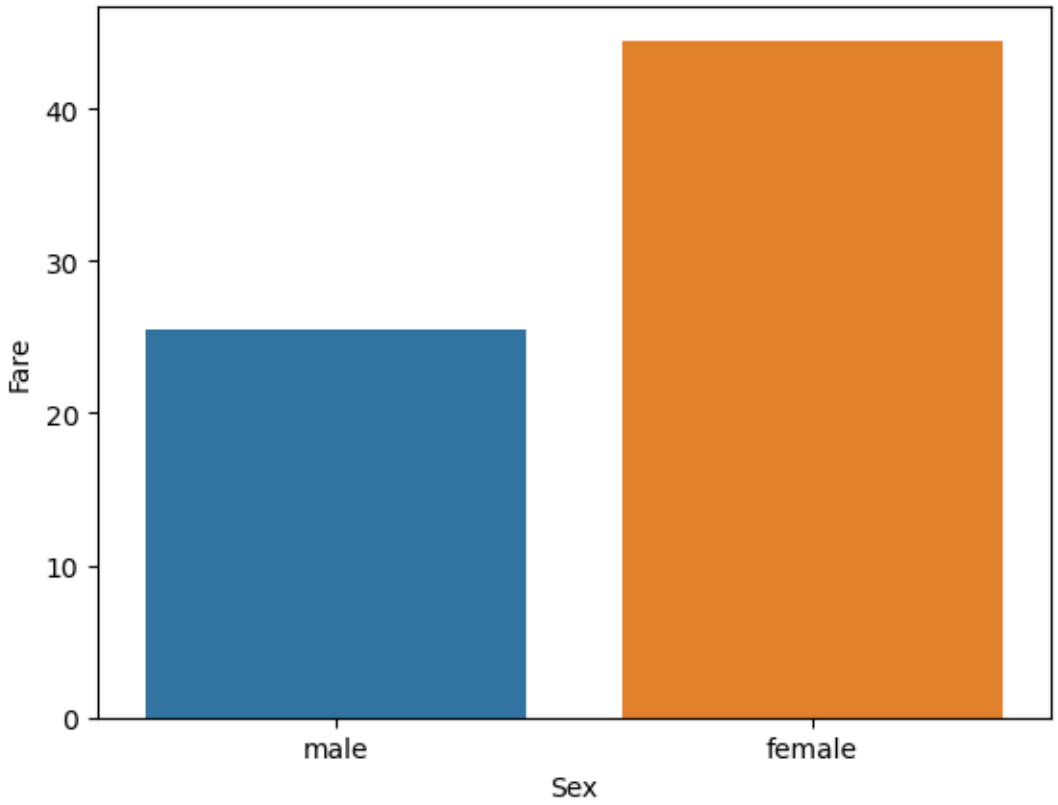
[18]:
```
passenger_class = df['Pclass'].value_counts()
sns.barplot(x=passenger_class.index, y=passenger_class.values)
plt.show()
```



[19]:
```
sns.barplot(x=df["Sex"],y=df["Fare"],ci=0)
```

```
C:\Users\mb419\AppData\Local\Temp\ipykernel_21236\1722039900.py:1:
FutureWarning:

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

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

[19]:
```
<Axes: xlabel='Sex', ylabel='Fare'>
```



6 Outlier Detection

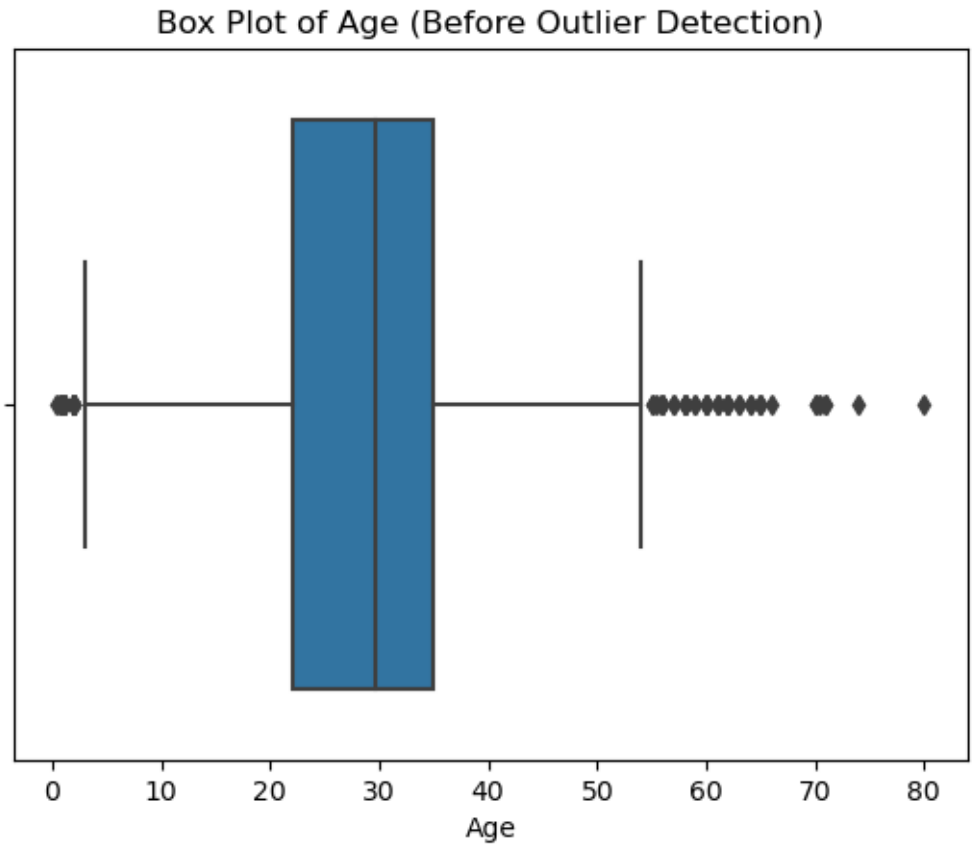
[20]:
```
df.head()
```

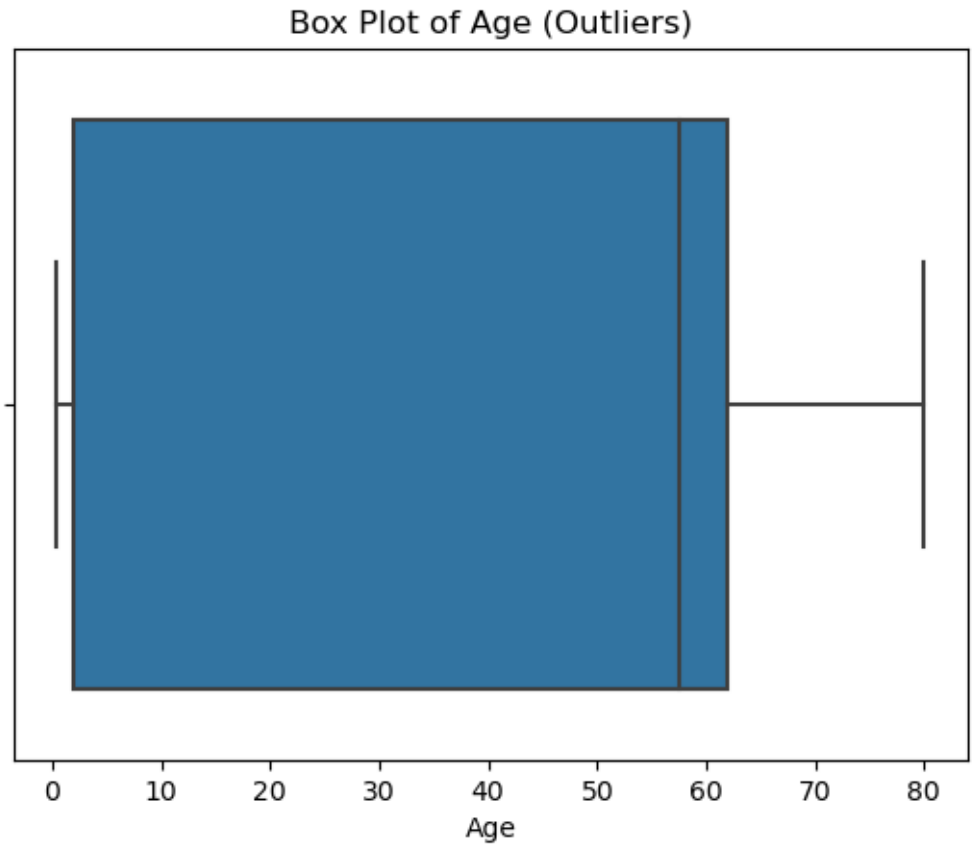
[20]:

	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	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th…	female	38.0	1					
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0					
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1					
4	5	0	3	Allen, Mr. William Henry	male	35.0	0					

```
1      0          PC 17599  71.2833      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
```

[21]:
```
sns.boxplot(x=df['Age'])
plt.title('Box Plot of Age (Before Outlier Detection)')
plt.show()

# Calculate the IQR (Interquartile Range)
Q1 = df['Age'].quantile(0.25)
Q3 = df['Age'].quantile(0.75)
IQR = Q3 - Q1

# Define the lower and upper bounds for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Detect outliers
outliers = df[(df['Age'] < lower_bound) | (df['Age'] > upper_bound)]

# Visualize the outliers
sns.boxplot(x=outliers['Age'])
plt.title('Box Plot of Age (Outliers)')
plt.show()

# Remove outliers (if desired)
#df_cleaned = df[(df['Age'] >= lower_bound) & (df['Age'] <= upper_bound)]
```


Box Plot of Age (Before Outlier Detection)
0
10
20
30
40
50
60
70
80
Age



7 Splitting Dependent and Independent variables

[22]:
```
df.head()
```

[22]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	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					

```
0      0         A/5 21171   7.2500  B96 B98        S
1      0          PC 17599  71.2833      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
```

[23]:
```
x = df.drop('Survived', axis=1)
x.head()
```

[23]:
```
   PassengerId  Pclass                                                Name  \
0            1       3                             Braund, Mr. Owen Harris
1            2       1   Cumings, Mrs. John Bradley (Florence Briggs Th…
2            3       3                              Heikkinen, Miss. Laina
3            4       1        Futrelle, Mrs. Jacques Heath (Lily May Peel)
4            5       3                            Allen, Mr. William Henry

      Sex   Age  SibSp  Parch            Ticket     Fare    Cabin Embarked
0    male  22.0      1      0         A/5 21171   7.2500  B96 B98        S
1  female  38.0      1      0          PC 17599  71.2833      C85        C
2  female  26.0      0      0  STON/O2. 3101282   7.9250  B96 B98        S
3  female  35.0      1      0            113803  53.1000     C123        S
4    male  35.0      0      0            373450   8.0500  B96 B98        S
```

[24]:
```
x.shape
```

[24]:
```
(891, 11)
```

[25]:
```
type(x)
```

[25]:
```
pandas.core.frame.DataFrame
```

[26]:
```
y = df['Survived']
y.head()
```

[26]:
```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

[27]:
```
x.head()
```

[27]:
```
   PassengerId  Pclass                                                Name  \
0            1       3                             Braund, Mr. Owen Harris
1            2       1   Cumings, Mrs. John Bradley (Florence Briggs Th…
2            3       3                              Heikkinen, Miss. Laina
3            4       1        Futrelle, Mrs. Jacques Heath (Lily May Peel)
```

```
4            5       3                              Allen, Mr. William Henry

      Sex   Age  SibSp  Parch            Ticket     Fare    Cabin Embarked
0    male  22.0      1      0         A/5 21171   7.2500  B96 B98        S
1  female  38.0      1      0          PC 17599  71.2833      C85        C
2  female  26.0      0      0  STON/O2. 3101282   7.9250  B96 B98        S
3  female  35.0      1      0            113803  53.1000     C123        S
4    male  35.0      0      0            373450   8.0500  B96 B98        S
```

8 Performing Encoding

[28]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

[29]:
```
x["Embarked"]=le.fit_transform(x["Embarked"])
x["Sex"]=le.fit_transform(x["Sex"])
x["Ticket"]=le.fit_transform(x["Ticket"])
x["Cabin"]=le.fit_transform(x["Cabin"])
x["Name"]=le.fit_transform(x["Name"])
```

[30]:
```
x.head()
```

[30]:
```
   PassengerId  Pclass  Name  Sex   Age  SibSp  Parch  Ticket     Fare  Cabin  \
0            1       3   108    1  22.0      1      0     523   7.2500     47
1            2       1   190    0  38.0      1      0     596  71.2833     81
2            3       3   353    0  26.0      0      0     669   7.9250     47
3            4       1   272    0  35.0      1      0      49  53.1000     55
4            5       3    15    1  35.0      0      0     472   8.0500     47

   Embarked
0         2
1         0
2         2
3         2
4         2
```

9 Feature Scalling

[33]:
```
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
```

[34]:
```
X_Scaled=ms.fit_transform(x)
```

[35]:
```
X_Scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
```

[36]:
```
X_Scaled.head()
```

[36]:
```
   PassengerId  Pclass      Name  Sex       Age  SibSp  Parch    Ticket  \
0     0.000000     1.0  0.121348  1.0  0.271174  0.125    0.0  0.769118
1     0.001124     0.0  0.213483  0.0  0.472229  0.125    0.0  0.876471
2     0.002247     1.0  0.396629  0.0  0.321438  0.000    0.0  0.983824
3     0.003371     0.0  0.305618  0.0  0.434531  0.125    0.0  0.072059
4     0.004494     1.0  0.016854  1.0  0.434531  0.000    0.0  0.694118

       Fare     Cabin  Embarked
0  0.014151  0.321918       1.0
1  0.139136  0.554795       0.0
2  0.015469  0.321918       1.0
3  0.103644  0.376712       1.0
4  0.015713  0.321918       1.0
```

10 Splitting Data into Train and Test

[37]:
```
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)
```

[38]:
```
print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

```
(712, 11) (179, 11) (712,) (179,)
```

[39]:
```
y_train
```

[39]:
```
140    0
439    0
817    0
378    0
491    0
      ..
835    1
192    1
629    0
559    1
684    0
Name: Survived, Length: 712, dtype: int64
```

[40]:
```
y_test
```

[40]:
```
495    0
648    0
278    0
31     1
255    1
      ..
```

```
780    1
837    0
215    1
833    0
372    0
Name: Survived, Length: 179, dtype: int64
```

[41]:
```
print(x_train.head(),x_test.head(),y_train.head(),y_test.head())
```

```
     PassengerId  Pclass      Name  Sex       Age  SibSp     Parch    Ticket  \
140     0.157303     1.0  0.111236  0.0  0.367921  0.000  0.333333  0.298529
439     0.493258     0.5  0.502247  1.0  0.384267  0.000  0.000000  0.804412
817     0.917978     0.5  0.566292  1.0  0.384267  0.125  0.166667  0.908824
378     0.424719     1.0  0.095506  1.0  0.246042  0.000  0.000000  0.269118
491     0.551685     1.0  0.978652  1.0  0.258608  0.000  0.000000  0.954412

         Fare     Cabin  Embarked
140  0.029758  0.321918       0.0
439  0.020495  0.321918       1.0
817  0.072227  0.321918       0.0
378  0.007832  0.321918       0.0
491  0.014151  0.321918       1.0        PassengerId  Pclass      Name  Sex
Age  SibSp     Parch    Ticket  \
495     0.556180     1.0  0.988764  1.0  0.367921  0.000  0.000000  0.258824
648     0.728090     1.0  0.971910  1.0  0.367921  0.000  0.000000  0.911765
278     0.312360     1.0  0.765169  1.0  0.082684  0.500  0.166667  0.705882
31      0.034831     0.0  0.871910  0.0  0.367921  0.125  0.000000  0.861765
255     0.286517     1.0  0.920225  0.0  0.359135  0.000  0.333333  0.272059

         Fare     Cabin  Embarked
495  0.028221  0.321918       0.0
648  0.014737  0.321918       1.0
278  0.056848  0.321918       0.5
31   0.285990  0.280822       0.0
255  0.029758  0.321918       0.0   140    0
439    0
817    0
378    0
491    0
Name: Survived, dtype: int64 495    0
648    0
278    0
31     1
255    1
Name: Survived, dtype: int64
```