```
#importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

```
df = pd.read_csv("Titanic-Dataset.csv")
print(df)
```

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         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
..                                                 ...     ...   ...    ...
886                              Montvila, Rev. Juozas    male  27.0      0
887                       Graham, Miss. Margaret Edith  female  19.0      0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN      1
889                              Behr, Mr. Karl Howell    male  26.0      0
890                                Dooley, Mr. Patrick    male  32.0      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
..     ...               ...      ...   ...      ...
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]
```

```
df.shape
```

```
(891, 12)
```

```
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
```

```
corr=df.corr()
corr
```

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

	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

```
plt.subplots(figsize=(20,15))
sns.heatmap(corr,annot=True)
```

```
# Print the number of null values in each column
df.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
```

```
# Print the number of null values in each column
print(df.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
```

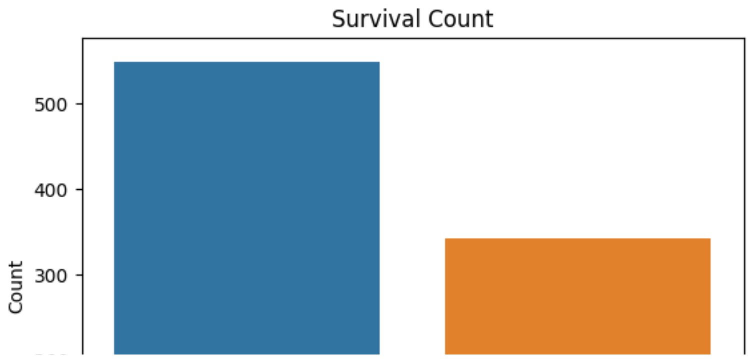
```
df["Age"].fillna(df["Age"].mean(),inplace=True)
df["Cabin"].fillna(df["Cabin"].mode()[0],inplace=True)
df["Embarked"].fillna(df["Embarked"].mode()[0],inplace=True)
df.isnull().sum()


    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
```

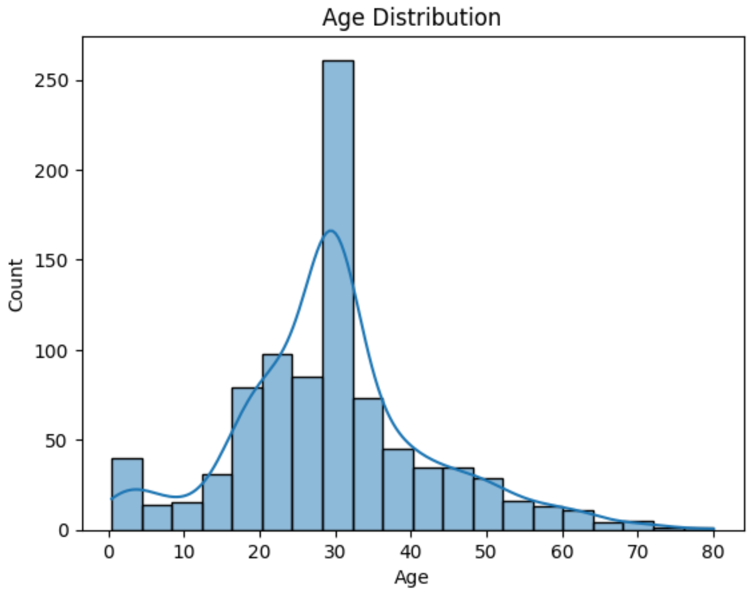
```
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence	female	38.0	1	0	PC 17599	71.2

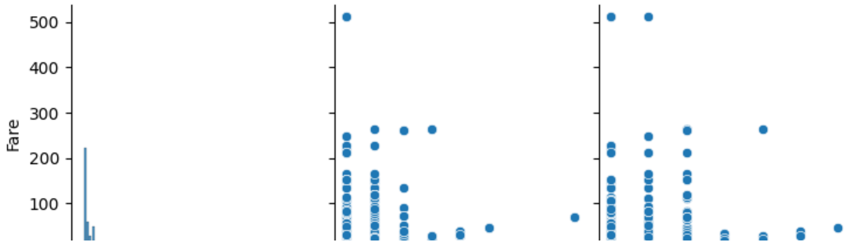
```
#data visualization
sns.countplot(data=df, x='Survived')
plt.title('Survival Count')
plt.xlabel('Survived')
plt.ylabel('Count')
plt.show()
```



```
sns.histplot(data=df, x='Age', bins=20, kde=True)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```

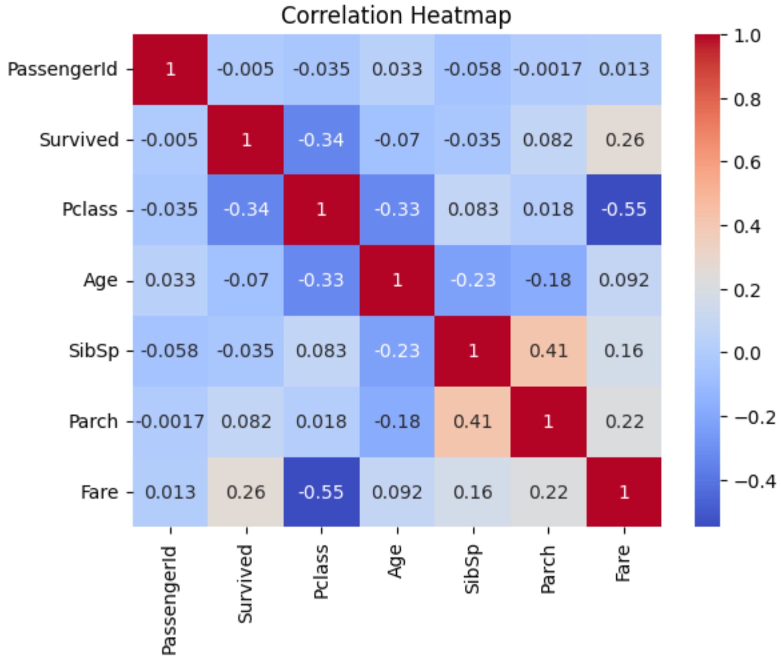


```
sns.pairplot(data=df[['Fare', 'SibSp', 'Parch']])
plt.title('Pair Plot')
plt.show()
```

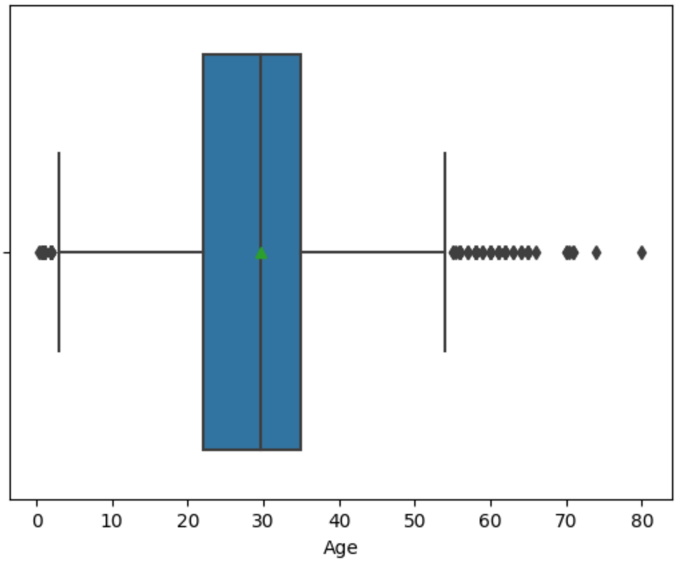


```
corr_matrix = df.corr()
sns.heatmap(corr_matrix, annot=True,cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

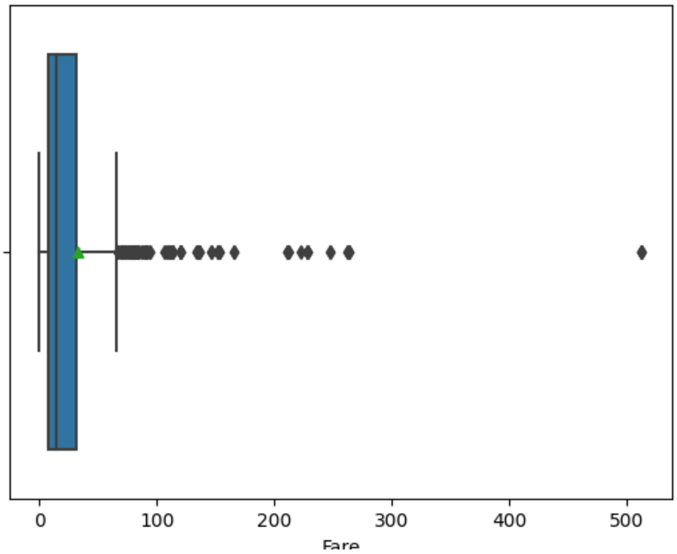
```
<ipython-input-16-8dcbd071fff3>:1: FutureWarning: The default value of numeric_only i
  corr_matrix = df.corr()
```



```
#outliner detection
# Create a box plot of the Age column
sns.boxplot(x='Age', showmeans=True, data=df)
plt.show()
```



```
sns.boxplot(x='Fare', showmeans=True, data=df)
plt.show()
```



```
#Splitting Dependent and Independent variables
# Split the data into dependent and independent variables
X = df.drop(['Survived'], axis=1)
y = df['Survived']
X.head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	B96 B98
1	2	1	Cumings, Mrs. John Bradley (Florence	female	38.0	1	0	PC 17599	71.2833	C85

```
y.head()
```

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

```
#Perform Encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
X["Sex"]=le.fit_transform(X["Sex"])
X["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
```

```
X["Sex"].value_counts()
```

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

```
X["Sex"].nunique()
```

```
2
```

```
X.Sex.value_counts()
```

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

```
#One Hot encoding on geography column
X.shape
```

```
(891, 11)
```

```
Sex=pd.get_dummies(X["Sex"],drop_first=True)
Sex
```

	1
0	1
1	0
2	0
3	0
4	1
...	...
886	1
887	0
888	0
889	1
890	1

891 rows × 1 columns

```
#concat
X=pd.concat([X,Sex],axis=1)
X.head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	B96 B98
1	2	1	Cumings, Mrs. John Bradley (Florence	0	38.0	1	0	PC 17599	71.2833	C85

```
X.drop(["Sex"],axis=1,inplace=True)
X.head(10)
```

	PassengerId	Pclass	Name	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	3	Braund, Mr. Owen Harris	22.000000	1	0	A/5 21171	7.2500	B96 B98
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.000000	1	0	PC 17599	71.2833	C85
2	3	3	Heikkinen, Miss. Laina	26.000000	0	0	STON/O2. 3101282	7.9250	B96 B98
			Futrelle, Mrs.						

```
X.shape
```

```
(891, 11)
```

```
#feature scaling
scale = StandardScaler()
```

```
X[['Age', 'Fare']] = scale.fit_transform(X[['Age', 'Fare']])
X.head()
```

	PassengerId	Pclass	Name	Age	SibSp	Parch	Ticket	Fare	Cabin	Embar
0	1	3	Braund, Mr. Owen Harris	-0.592481	1	0	A/5 21171	-0.502445	B96 B98	
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0.638789	1	0	PC 17599	0.786845	C85	
2	3	3	Heikkinen, Miss. Laina	-0.284663	0	0	STON/O2. 3101282	-0.488854	B96 B98	
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0.407926	1	0	113803	0.420730	C123	
4	5	3	Allen, Mr. William Henry	0.407926	0	0	373450	-0.486337	B96 B98	

```
#splitting data into train and test
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=0)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(623, 11)
(268, 11)
(623,)
(268,)
```

```
a=[1,2,3,4,5,6]
b=[1,0,1,5,6,3]
for i in range(5):
                  a_train,a_test,b_train,b_test=train_test_split(a,b,test_size=0.3,random_state=100)
print("with random state",a_train)
```

```
with random state [5, 4, 6, 1]
```