1.Importing Libraries

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

2.Imporing 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	Embarke
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.shape
```

Out[4]:
```
(891, 12)
```

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.describe()
```

Out[6]:

	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 [7]:

```
#Dropping the unwanted columns from the dataset
df.drop(['Name','SibSp','Parch','Ticket'],axis=1,inplace=True)
df.head()
```

Out[7]:

	PassengerId	Survived	Pclass	Sex	Age	Fare	Cabin	Embarked
0	1	0	3	male	22.0	7.2500	NaN	S
1	2	1	1	female	38.0	71.2833	C85	C
2	3	1	3	female	26.0	7.9250	NaN	S
3	4	1	1	female	35.0	53.1000	C123	S
4	5	0	3	male	35.0	8.0500	NaN	S

In [8]:
```
corr = df.corr()
corr
```

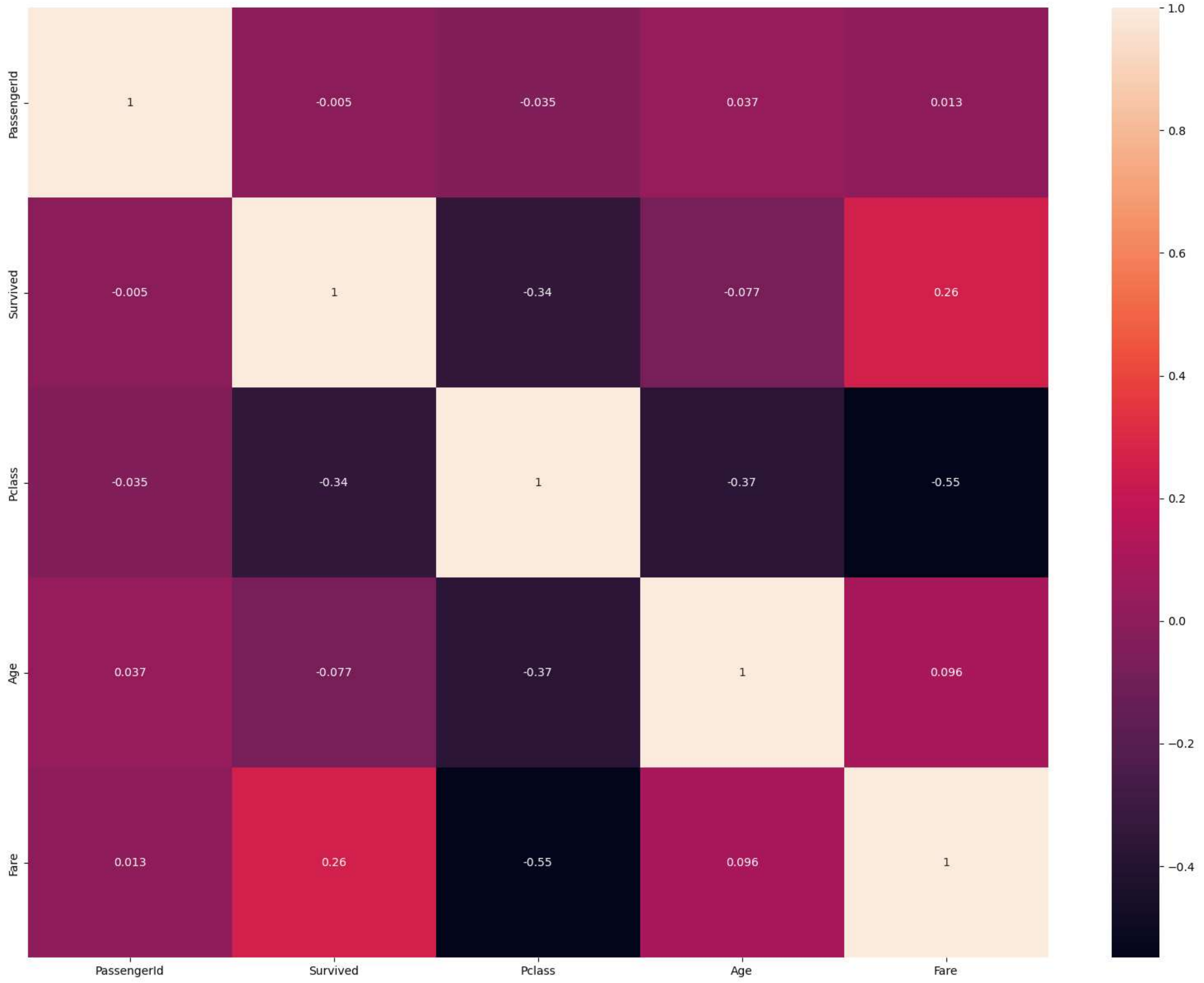
```
C:\Users\shaik\AppData\Local\Temp\ipykernel_21200\2438084875.py:1: FutureWarning: The d
efault value of numeric_only in DataFrame.corr is deprecated. In a future version, it w
ill default to False. Select only valid columns or specify the value of numeric_only to
silence this warning.
  corr = df.corr()
```

Out[8]:

	PassengerId	Survived	Pclass	Age	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	0.096067
Fare	0.012658	0.257307	-0.549500	0.096067	1.000000

In [9]:
```
plt.subplots(figsize=(20,15))
sns.heatmap(corr,annot=True)
```

Out[9]:
```
<Axes: >
```



3.Checking for Null Values

In [10]:

```
df.isnull().any()
```

Out[10]:

```
PassengerId    False
Survived       False
Pclass         False
Sex            False
Age             True
Fare           False
Cabin           True
Embarked        True
dtype: bool
```

In [11]:

```
df.isnull().sum()
```

Out[11]:

```
PassengerId      0
Survived         0
Pclass           0
Sex              0
Age            177
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [12]:

```
df.dropna()
```

Out[12]:

	PassengerId	Survived	Pclass	Sex	Age	Fare	Cabin	Embarked
1	2	1	1	female	38.0	71.2833	C85	C
3	4	1	1	female	35.0	53.1000	C123	S
6	7	0	1	male	54.0	51.8625	E46	S
10	11	1	3	female	4.0	16.7000	G6	S
11	12	1	1	female	58.0	26.5500	C103	S
...	...	...	...	...	...	...	...	...
871	872	1	1	female	47.0	52.5542	D35	S
872	873	0	1	male	33.0	5.0000	B51 B53 B55	S
879	880	1	1	female	56.0	83.1583	C50	C
887	888	1	1	female	19.0	30.0000	B42	S
889	890	1	1	male	26.0	30.0000	C148	C

183 rows × 8 columns

In [13]:

```
df.shape
```

Out[13]:

```
(891, 8)
```

In [14]:
```
df['Age'].fillna(0)
```

Out[14]:
```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888     0.0
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

In [15]:
```
df['Age'].ffill()
```

Out[15]:
```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888    19.0
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

In [16]:
```
df['Age'].bfill()
```

Out[16]:
```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888    26.0
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

In [17]:

```
df['Age'].fillna(df['Age'].median(),inplace = True)
df['Age']
```

Out[17]:

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ... 
886    27.0
887    19.0
888    28.0
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

In [18]:

```
df.isnull().sum()
```

Out[18]:

```
PassengerId      0
Survived         0
Pclass           0
Sex              0
Age              0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [19]:

```
df[['Cabin','Embarked']].head()
```

Out[19]:

	Cabin	Embarked
0	NaN	S
1	C85	C
2	NaN	S
3	C123	S
4	NaN	S

In [20]:

```
df[['Cabin','Embarked']].isnull().sum()
```

Out[20]:

```
Cabin       687
Embarked      2
dtype: int64
```

In [21]:

```
df['Embarked'].value_counts()
```

Out[21]:

```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [22]:

```
df['Embarked'] = df['Embarked'].fillna(df['Embarked'].value_counts().index[0])
```

In [23]:

```
df['Embarked'].isnull().sum()
df['Embarked'].value_counts()
```

Out[23]:

```
S    646
C    168
Q     77
Name: Embarked, dtype: int64
```

In [24]:

```
df['Cabin'] = df['Cabin'].fillna('Unknown')
df['Cabin'].value_counts()
df['Cabin'].isnull().sum()
```

Out[24]:

```
0
```

In [25]:

```
df.Cabin.nunique()
```

Out[25]:

```
148
```

In [26]:

```
df.Cabin.value_counts()
```

Out[26]:

```
Unknown        687
C23 C25 C27      4
G6               4
B96 B98          4
C22 C26          3
              ... 
E34              1
C7               1
C54              1
E36              1
C148             1
Name: Cabin, Length: 148, dtype: int64
```

In [27]:

```
df.isnull().sum()
```

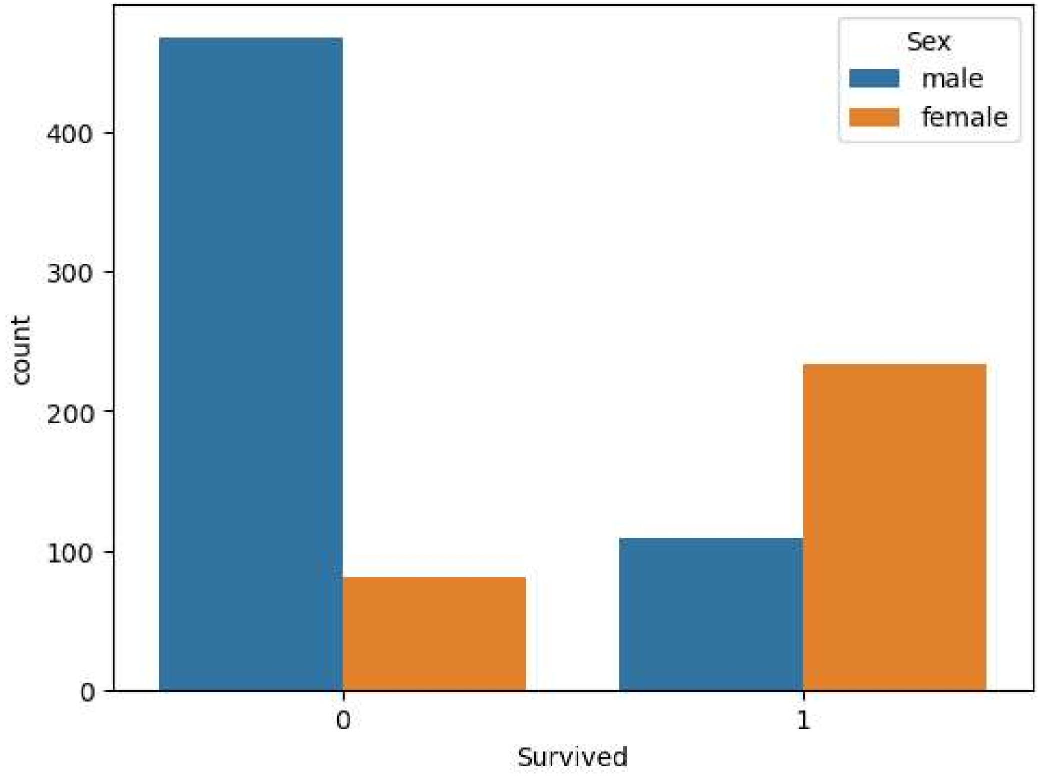
Out[27]:

```
PassengerId    0
Survived       0
Pclass         0
Sex            0
Age            0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

4.Data Visualization

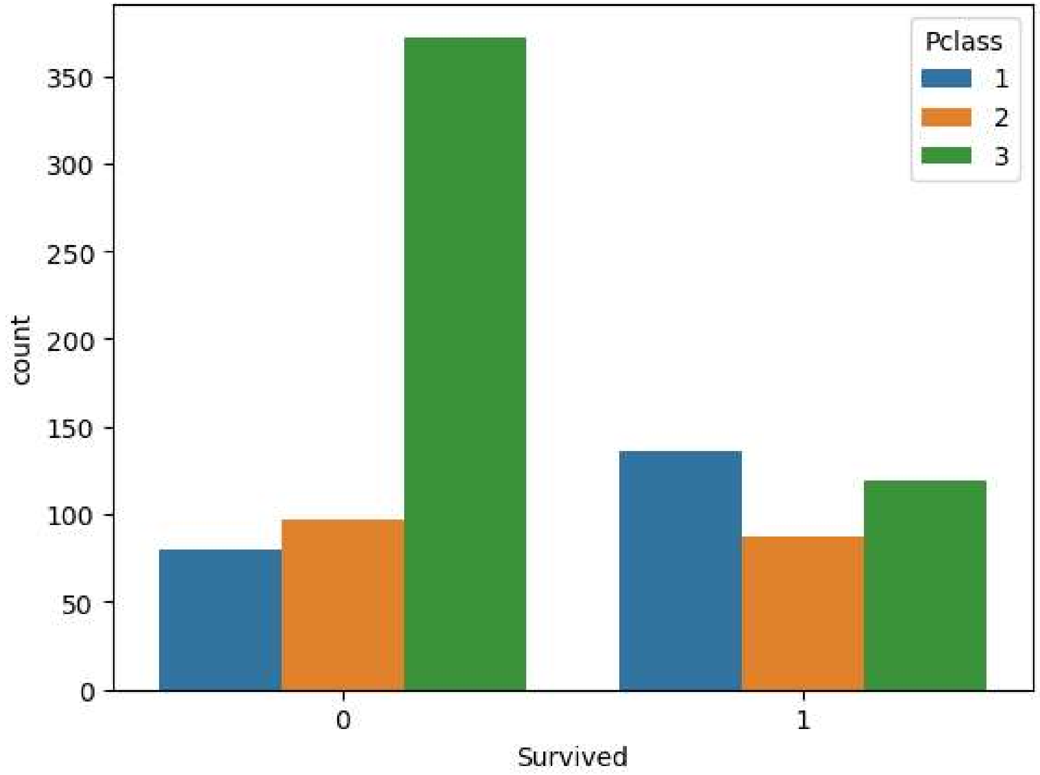
In [28]:
```
sns.countplot(x='Survived',data=df,hue = 'Sex')
```

Out[28]:
```
<Axes: xlabel='Survived', ylabel='count'>
```



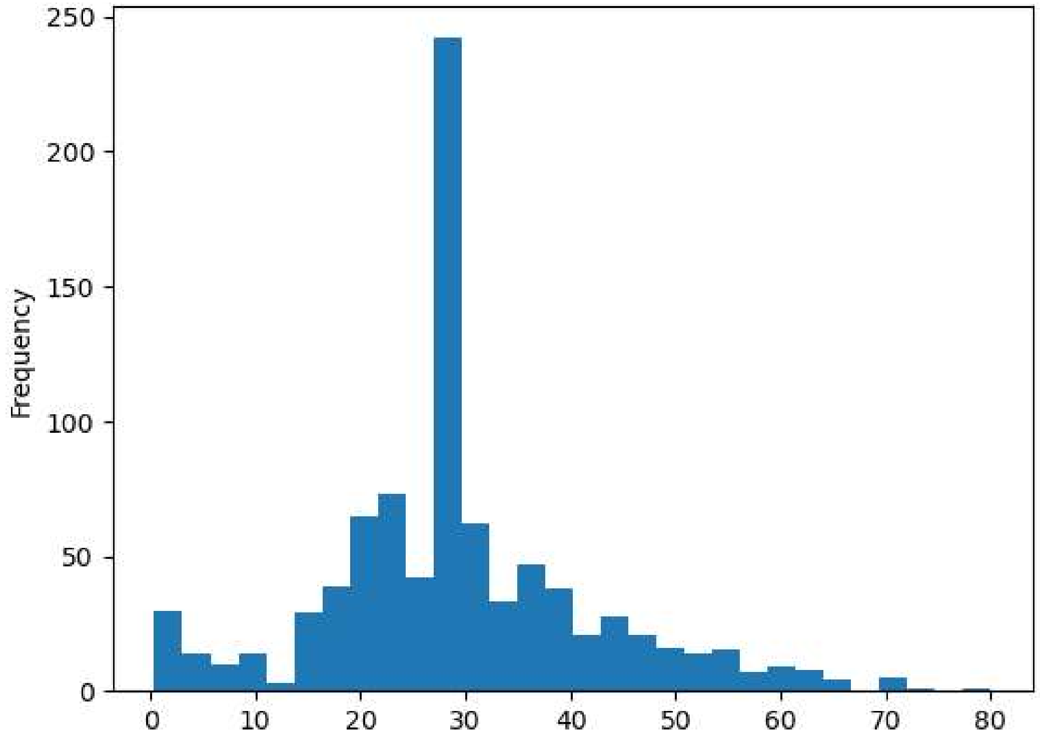
```
In [29]: sns.countplot(x='Survived',data=df,hue = 'Pclass')
```

```
Out[29]: <Axes: xlabel='Survived', ylabel='count'>
```



```
In [30]: df['Age'].dropna().plot.hist(bins=30)
```

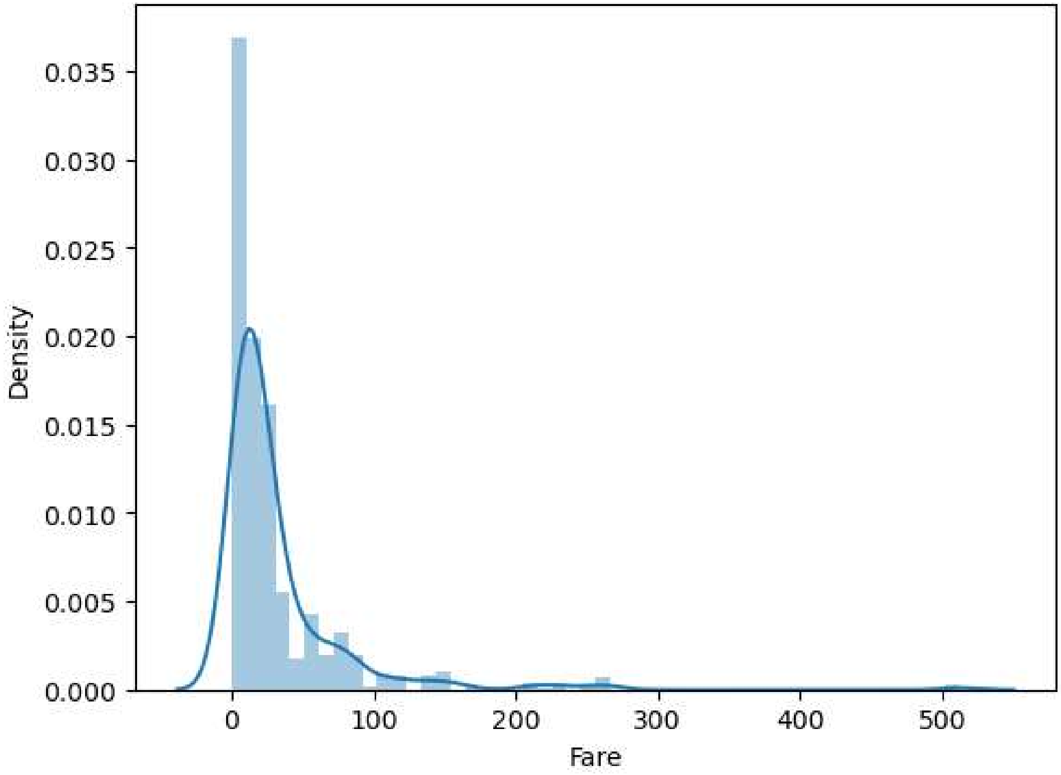
```
Out[30]: <Axes: ylabel='Frequency'>
```



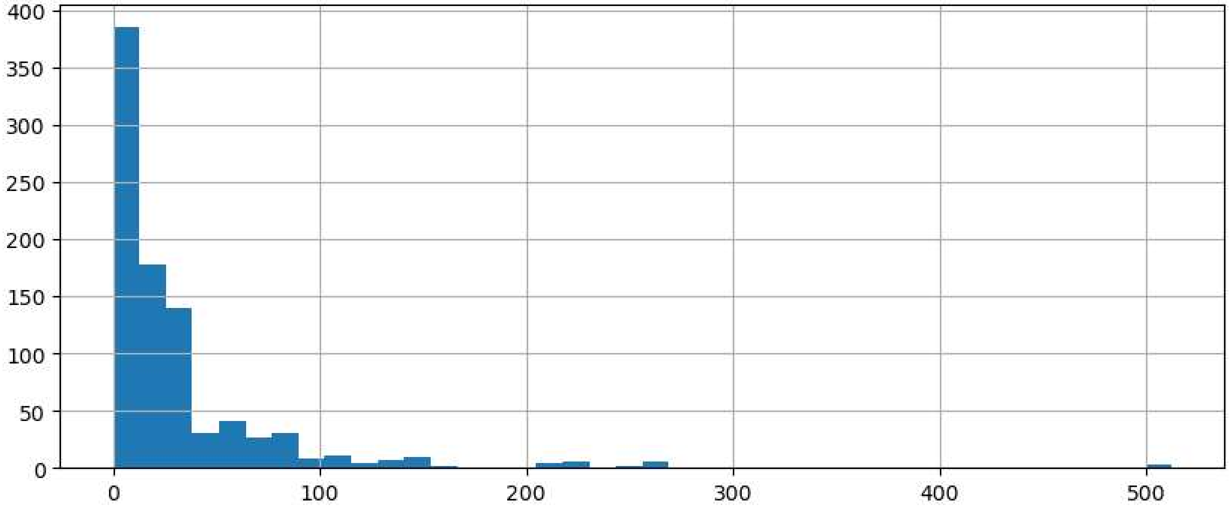
In [31]:
```
sns.distplot(df['Fare'])
```

```
C:\Users\shaik\AppData\Local\Temp\ipykernel_21200\3425841524.py: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 (https://gist.github.c
om/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df['Fare'])
```

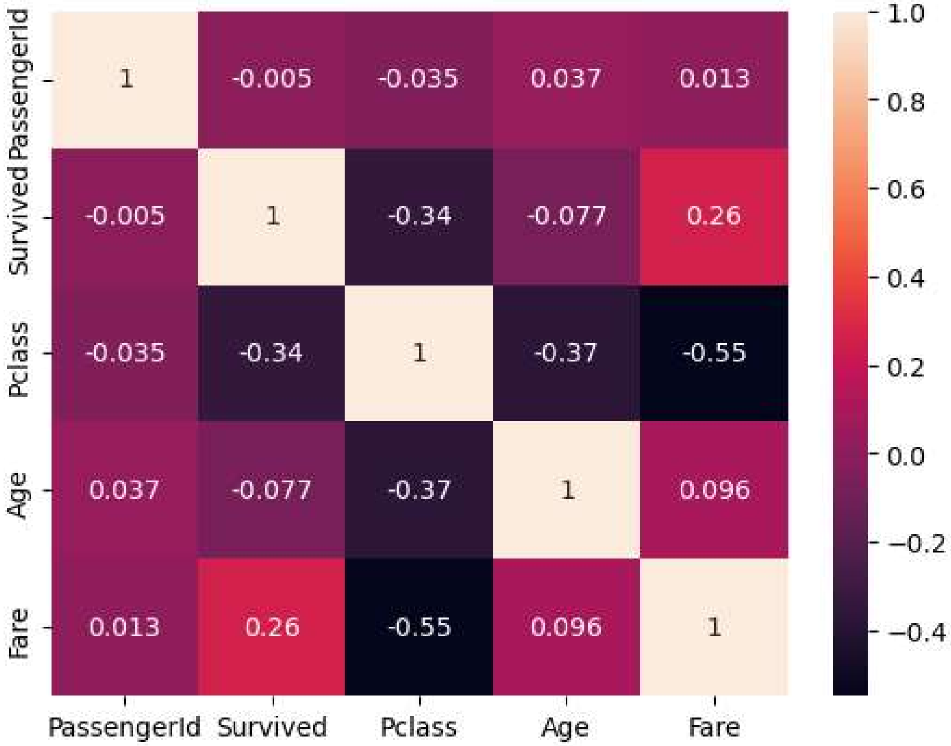
Out[31]:
```
<Axes: xlabel='Fare', ylabel='Density'>
```



```
In [32]: df['Fare'].hist(bins=40,figsize=(10,4))

Out[32]: <Axes: >
```



```
In [33]: sns.heatmap(corr,annot=True)

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



5.Outlier Detection

In [34]:

```
df.describe()
```

Out[34]:

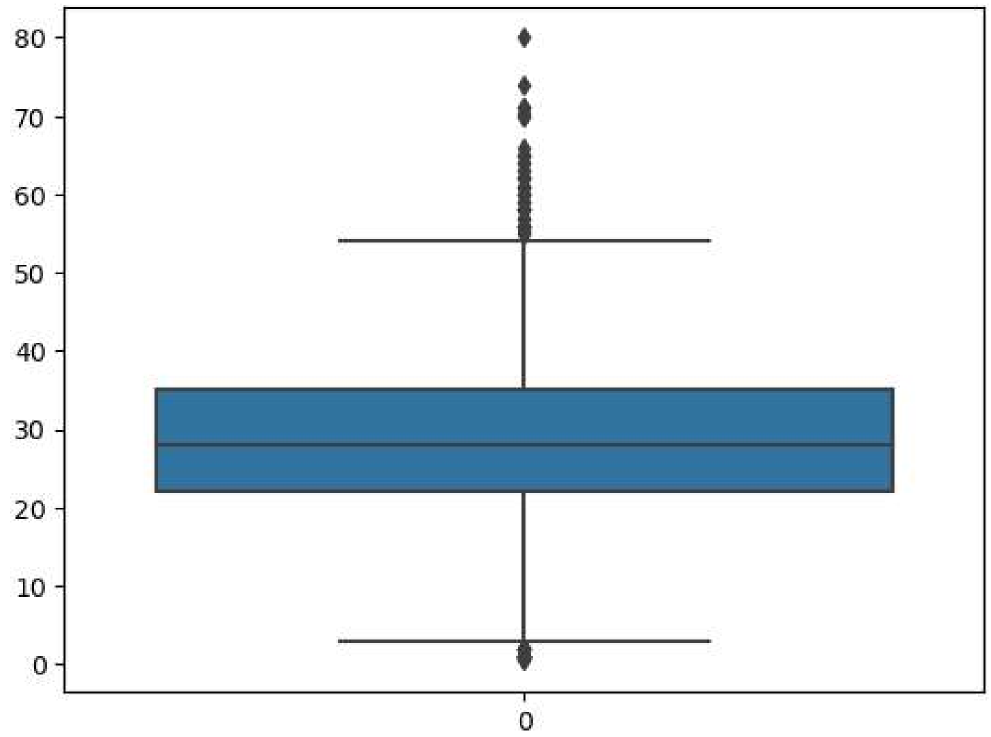
	PassengerId	Survived	Pclass	Age	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.361582	32.204208
std	257.353842	0.486592	0.836071	13.019697	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000
25%	223.500000	0.000000	2.000000	22.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	14.454200
75%	668.500000	1.000000	3.000000	35.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	512.329200

In [35]:

```
sns.boxplot(df.Age)
```

Out[35]:

```
<Axes: >
```



In [36]:

```
q1 = df.Age.quantile(0.25)
q3 = df.Age.quantile(0.75)
IQR = q3 - q1
IQR
```

Out[36]:

```
13.0
```

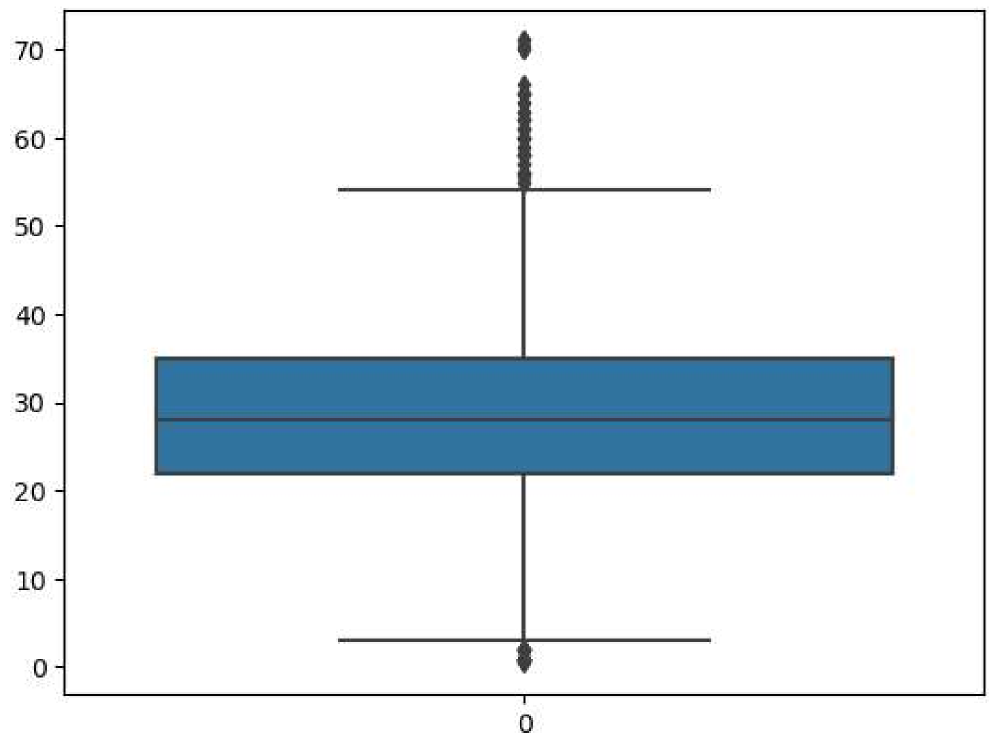
In [37]:
```
upper_limit = q3 + 3*IQR
lower_limit = q1 - 3*IQR
print(upper_limit)
print(lower_limit)
```

```
74.0
-17.0
```

In [38]:
```
df = df[df.Age<upper_limit]
```

In [39]:
```
sns.boxplot(df.Age)
```

Out[39]:
```
<Axes: >
```



In [40]:
```
#Removing outliers using z_score method
from scipy import stats
```

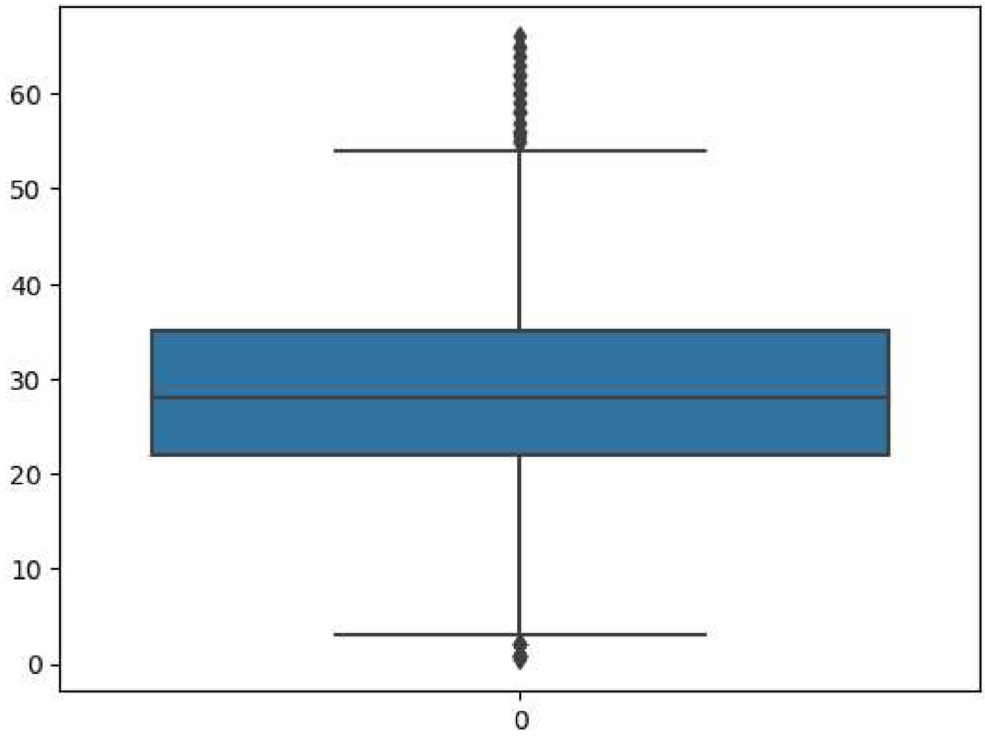
In [41]:
```
Age_zscore = stats.zscore(df.Age)
Age_zscore
```

Out[41]:
```
0     -0.565499
1      0.681741
2     -0.253689
3      0.447883
4      0.447883
         ...
886   -0.175737
887   -0.799357
888   -0.097784
889   -0.253689
890    0.214026
Name: Age, Length: 889, dtype: float64
```

In [42]:
```
df_z=df[np.abs(Age_zscore)<=3]
```

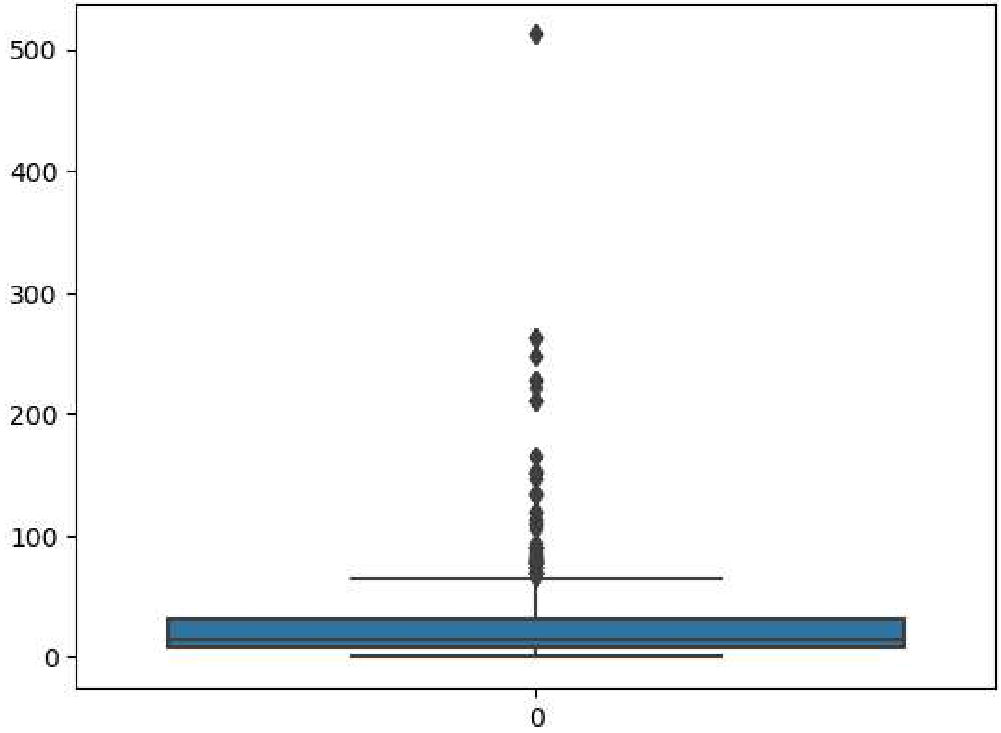
In [43]:
```
sns.boxplot(df_z['Age'])
```

Out[43]:
```
<Axes: >
```



In [44]:
```
sns.boxplot(df['Fare'])
```

Out[44]:
```
<Axes: >
```



In [45]:

```
q1=df.Fare.quantile(0.25)
q3=df.Fare.quantile(0.75)
```

In [46]:

```
q1,q3
```

Out[46]:

```
(7.925, 31.0)
```

In [47]:

```
IQR = q3-q1
```

In [48]:

```
upper_limit=q3+1.5*IQR
```

In [49]:

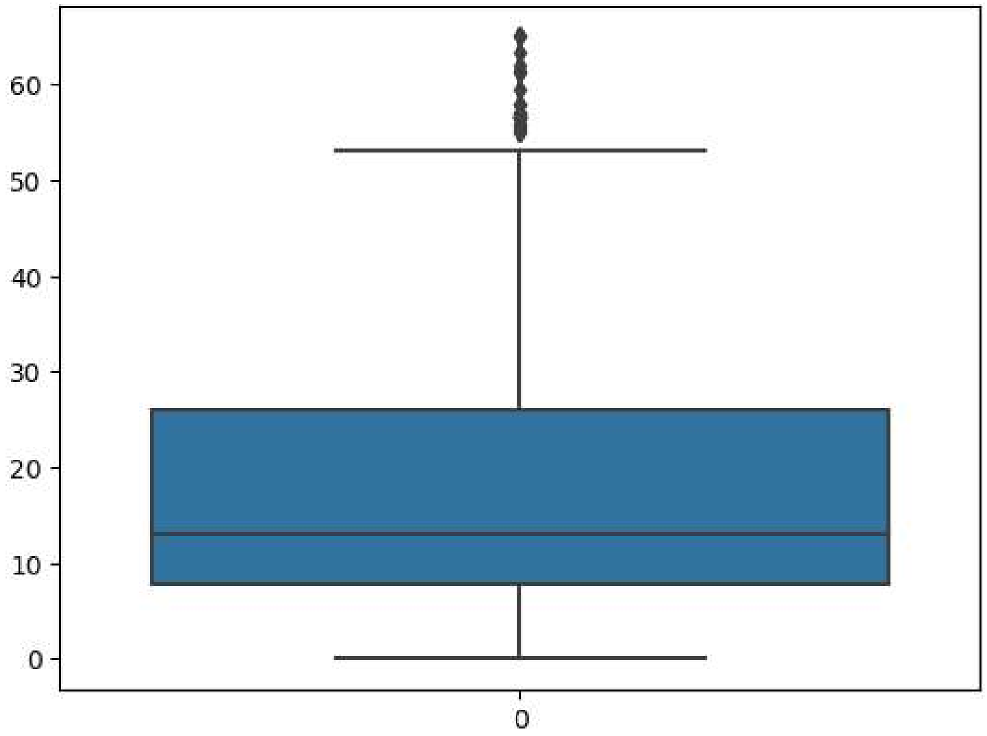
```
df = df[df['Fare']<upper_limit]
```

In [50]:

```
sns.boxplot(df['Fare'])
```

Out[50]:

```
<Axes: >
```



6.Splitting dependant and independant variables

In [51]:

```
df.drop('Cabin',axis=1,inplace=True)
```

In [52]:
```
df.head()
```

Out[52]:

	PassengerId	Survived	Pclass	Sex	Age	Fare	Embarked
0	1	0	3	male	22.0	7.2500	S
2	3	1	3	female	26.0	7.9250	S
3	4	1	1	female	35.0	53.1000	S
4	5	0	3	male	35.0	8.0500	S
5	6	0	3	male	28.0	8.4583	Q

In [53]:
```
df.shape
```

Out[53]:
```
(773, 7)
```

In [54]:
```
x=df.iloc[:,2:]
y=df.iloc[:,1:2]
```

In [55]:
```
x.shape
```

Out[55]:
```
(773, 5)
```

In [56]:
```
y.shape
```

Out[56]:
```
(773, 1)
```

7.Encoding

In [ ]:
```

```

In [57]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

In [58]:
```
x['Sex']=le.fit_transform(x['Sex'])
```

In [59]:
```
x.head()
```

Out[59]:

	Pclass	Sex	Age	Fare	Embarked
0	3	1	22.0	7.2500	S
2	3	0	26.0	7.9250	S
3	1	0	35.0	53.1000	S
4	3	1	35.0	8.0500	S
5	3	1	28.0	8.4583	Q

In [60]:
```
x['Embarked']=le.fit_transform(x['Embarked'])
```

In [61]:
```
x.head()
```

Out[61]:

	Pclass	Sex	Age	Fare	Embarked
0	3	1	22.0	7.2500	2
2	3	0	26.0	7.9250	2
3	1	0	35.0	53.1000	2
4	3	1	35.0	8.0500	2
5	3	1	28.0	8.4583	1

In [62]:
```
print(le.classes_)
```

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

In [63]:
```
print(dict(zip(le.classes_,range(len(le.classes_)))))
```

```
{'C': 0, 'Q': 1, 'S': 2}
```

8.Feature Scaling

In [64]:
```
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
```

In [65]:
```
x_scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
x_scaled.head()
```

Out[65]:

	Pclass	Sex	Age	Fare	Embarked
0	1.0	1.0	0.305752	0.111538	1.0
1	1.0	0.0	0.362426	0.121923	1.0
2	0.0	0.0	0.489940	0.816923	1.0
3	1.0	1.0	0.489940	0.123846	1.0
4	1.0	1.0	0.390762	0.130128	0.5

9.Splitting the data into train and test

In [66]:
```
from sklearn.model_selection import train_test_split
x_train,y_train,x_test,y_test = train_test_split(x_scaled,y,test_size=0.2,random_state=0
```

In [67]:
```
x_train.shape,y_train.shape,x_test.shape,y_test.shape
```

Out[67]:
```
((618, 5), (155, 5), (618, 1), (155, 1))
```

In [68]:

```
x_train.head()
```

Out[68]:

	Pclass	Sex	Age	Fare	Embarked
768	0.5	1.0	0.376594	0.200000	1.0
419	1.0	1.0	0.702465	0.123846	1.0
118	1.0	1.0	0.277416	0.108462	1.0
252	1.0	1.0	0.263247	0.123846	1.0
157	1.0	1.0	0.291584	0.121923	1.0

In [69]:

```
y_train.head()
```

Out[69]:

	Pclass	Sex	Age	Fare	Embarked
369	1.0	1.0	0.390762	0.119231	0.5
628	1.0	1.0	0.277416	0.133269	1.0
401	1.0	1.0	0.390762	0.123846	1.0
14	0.5	0.0	0.773307	0.246154	1.0
549	0.0	1.0	0.390762	0.000000	1.0

In [ ]: