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

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

Out[2]:

	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 Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [3]:
```
df.shape
```

Out[3]: (891, 12)

In [4]:
```
df.head()
```

Out[4]:

	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

In [5]:
```
df.tail()
```

Out[5]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [6]:
```
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 [7]:
```
df.describe()
```

Out[7]:

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

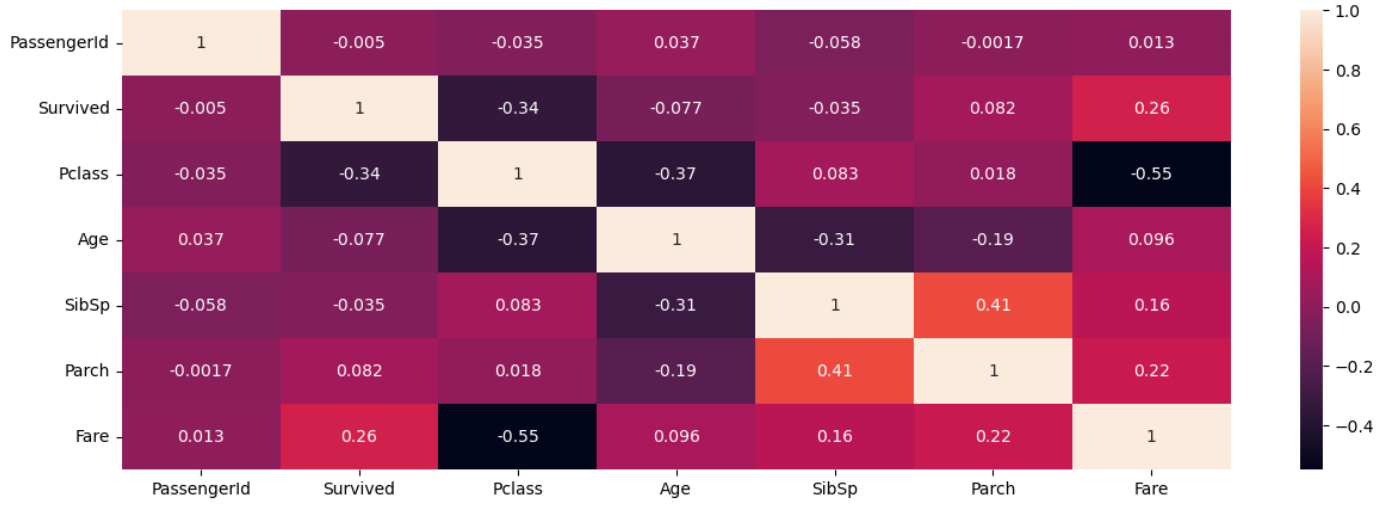
```
C:\Users\tejbh\AppData\Local\Temp\ipykernel_25916\3182140910.py:1: FutureWarning: The de
fault value of numeric_only in DataFrame.corr is deprecated. In a future version, it wil
l default to False. Select only valid columns or specify the value of numeric_only to si
lence this warning.
  corr=df.corr()
```

Out[8]:

	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 [9]:
```
plt.subplots(figsize=(15,5))
sns.heatmap(corr,annot=True)
```

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



In [10]:
```
df.Sex.value_counts()
```

Out[10]:
```
male      577
female    314
Name: Sex, dtype: int64
```

In [11]:
```
df.Survived.value_counts()
```

Out[11]:
```
0    549
1    342
Name: Survived, dtype: int64
```

In [12]:
```
df.Embarked.value_counts()
```

Out[12]:
```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

In [13]:
```
df.isnull().sum()
```

Out[13]:
```
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 [14]:
```
df = df.drop(columns = ['Name' , 'Ticket' , 'Cabin'])
```

In [15]:
```
mean_age = df['Age'].mean()
df['Age'].fillna(mean_age, inplace=True)
```

In [16]:
```
df["Embarked"].fillna(df["Embarked"].mode()[0],inplace=True)
```

In [17]:
```
print(df.isnull().sum())
```

```
PassengerId    0
Survived       0
Pclass         0
Sex            0
```

```
Age            0
SibSp          0
Parch          0
Fare           0
Embarked       0
dtype: int64
```

In [18]:
```
df
```

Out[18]:

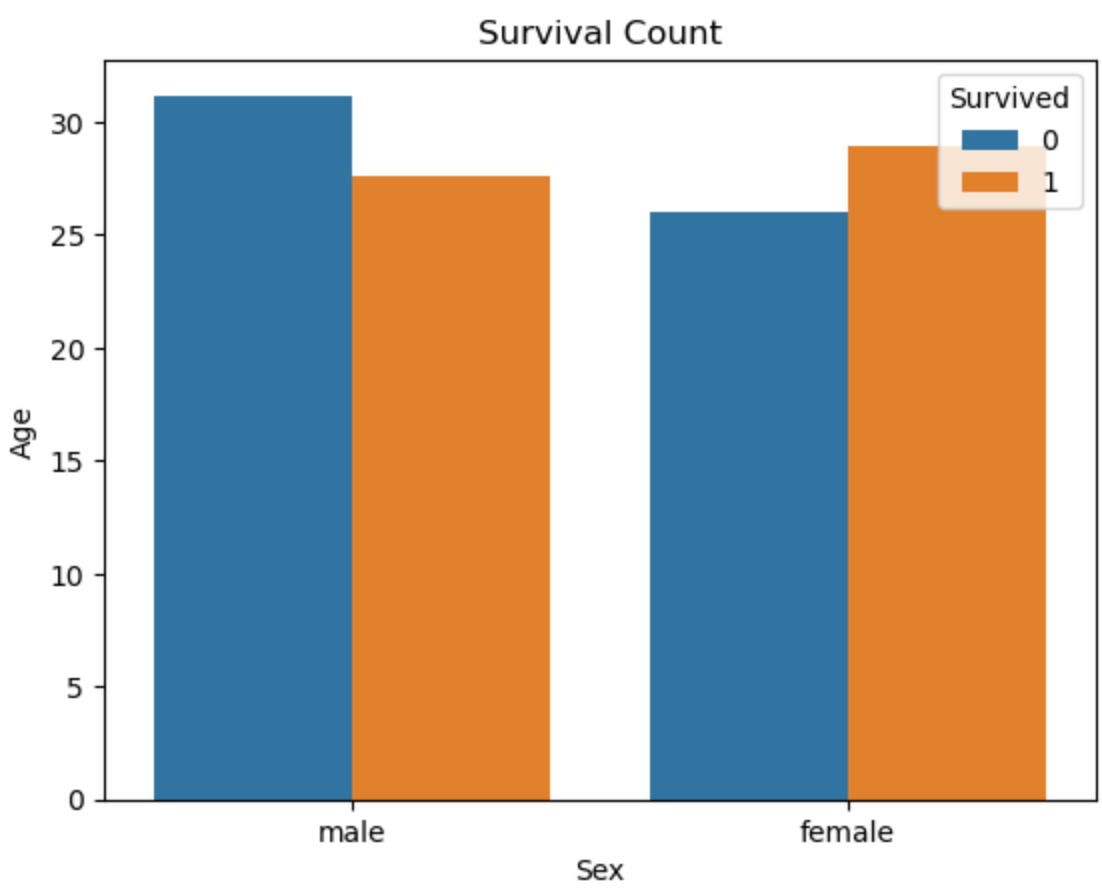
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	male	22.000000	1	0	7.2500	S
1	2	1	1	female	38.000000	1	0	71.2833	C
2	3	1	3	female	26.000000	0	0	7.9250	S
3	4	1	1	female	35.000000	1	0	53.1000	S
4	5	0	3	male	35.000000	0	0	8.0500	S
...	...	...	...	...	...	...	...	...	...
886	887	0	2	male	27.000000	0	0	13.0000	S
887	888	1	1	female	19.000000	0	0	30.0000	S
888	889	0	3	female	29.699118	1	2	23.4500	S
889	890	1	1	male	26.000000	0	0	30.0000	C
890	891	0	3	male	32.000000	0	0	7.7500	Q

891 rows × 9 columns

In [19]:
```
plt.title('Survival Count')
sns.barplot(data=df,x=df.Sex,y=df.Age,hue=df.Survived,ci=None)
```

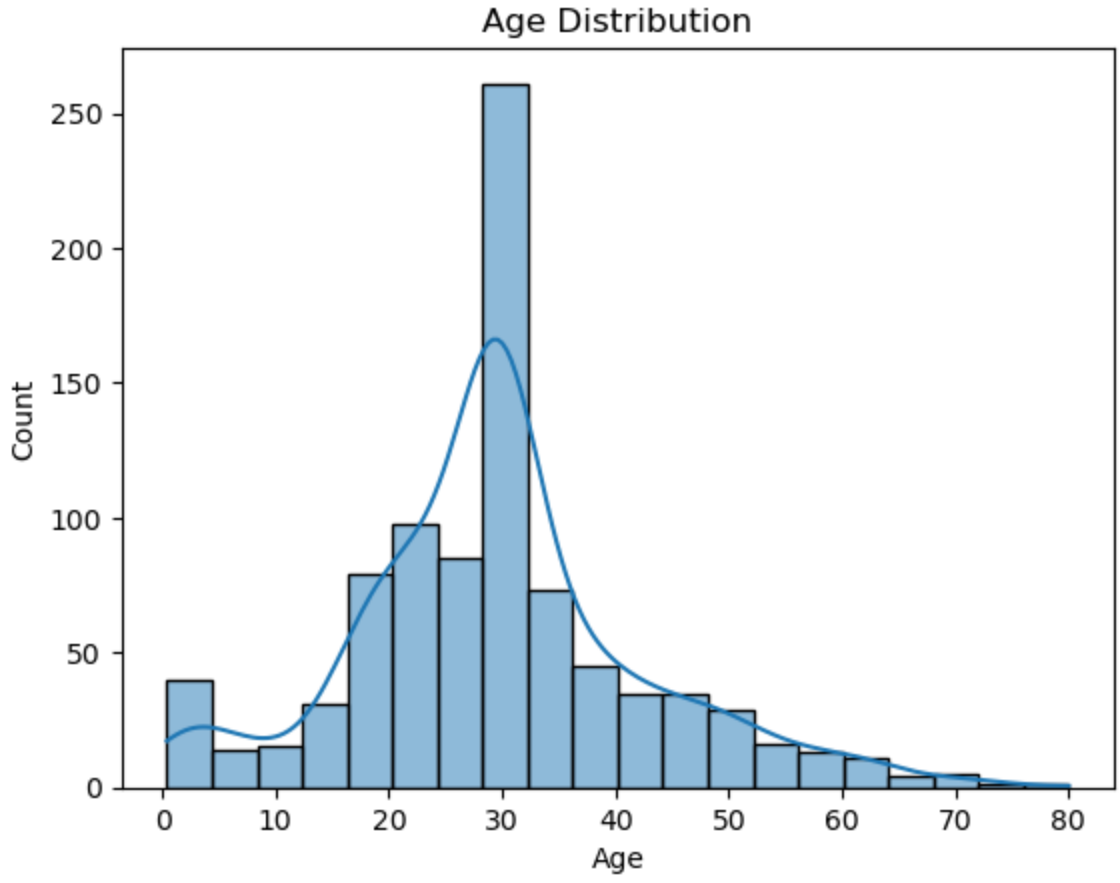
```
C:\Users\tejbh\AppData\Local\Temp\ipykernel_25916\3205349858.py:2: FutureWarning:

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

  sns.barplot(data=df,x=df.Sex,y=df.Age,hue=df.Survived,ci=None)
```

Out[19]:
```
<Axes: title={'center': 'Survival Count'}, xlabel='Sex', ylabel='Age'>
```



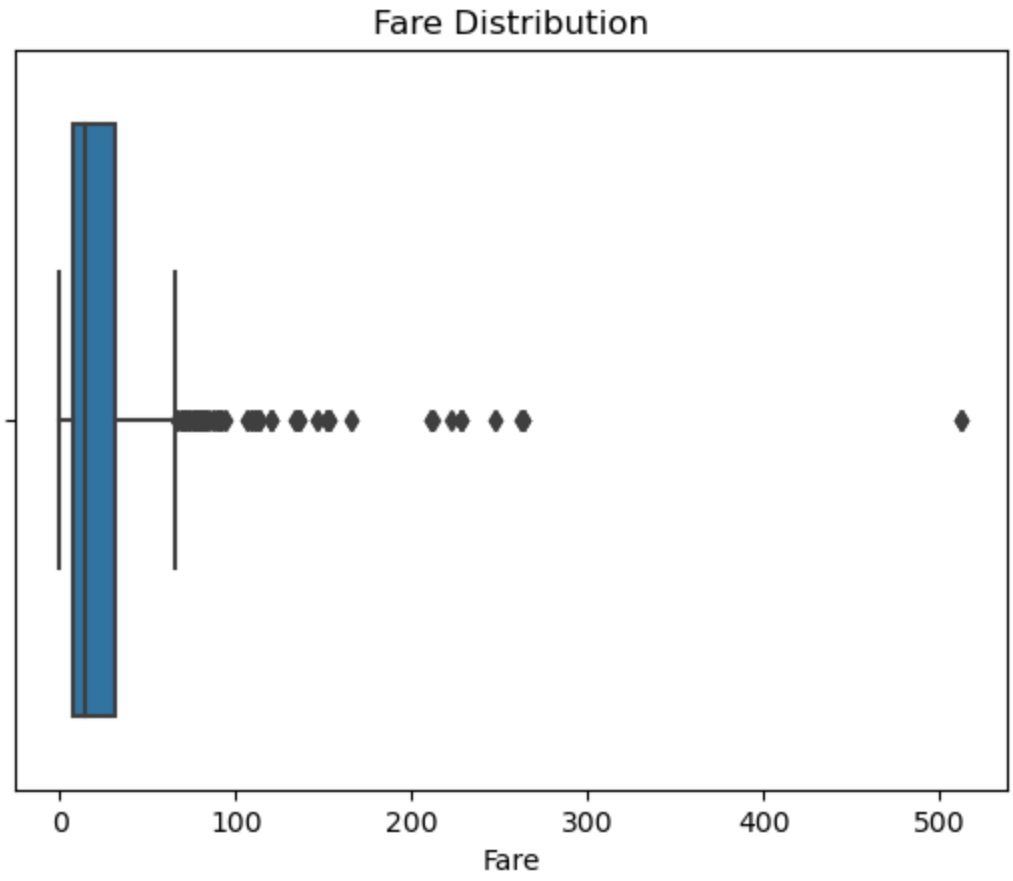
In [20]:
```
sns.histplot(data=df, x='Age', bins=20, kde=True)
plt.title('Age Distribution')
```

Out[20]:
```
Text(0.5, 1.0, 'Age Distribution')
```



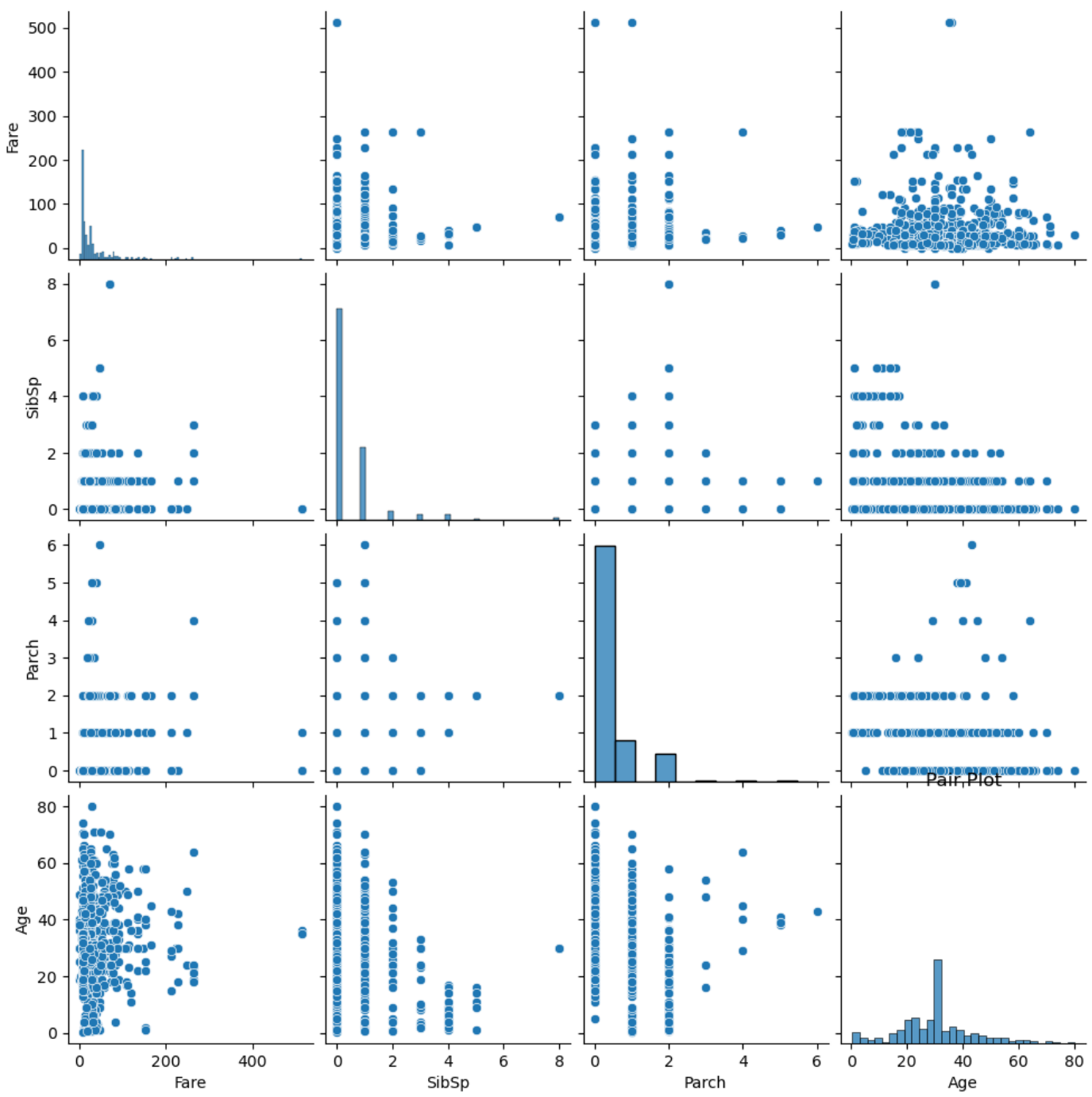
In [21]:
```
sns.boxplot(data=df, x='Fare')
plt.title('Fare Distribution')
```

Out[21]: Text(0.5, 1.0, 'Fare Distribution')



In [22]:
```
sns.pairplot(data=df[['Fare', 'SibSp', 'Parch','Age']])
plt.title('Pair Plot')
```

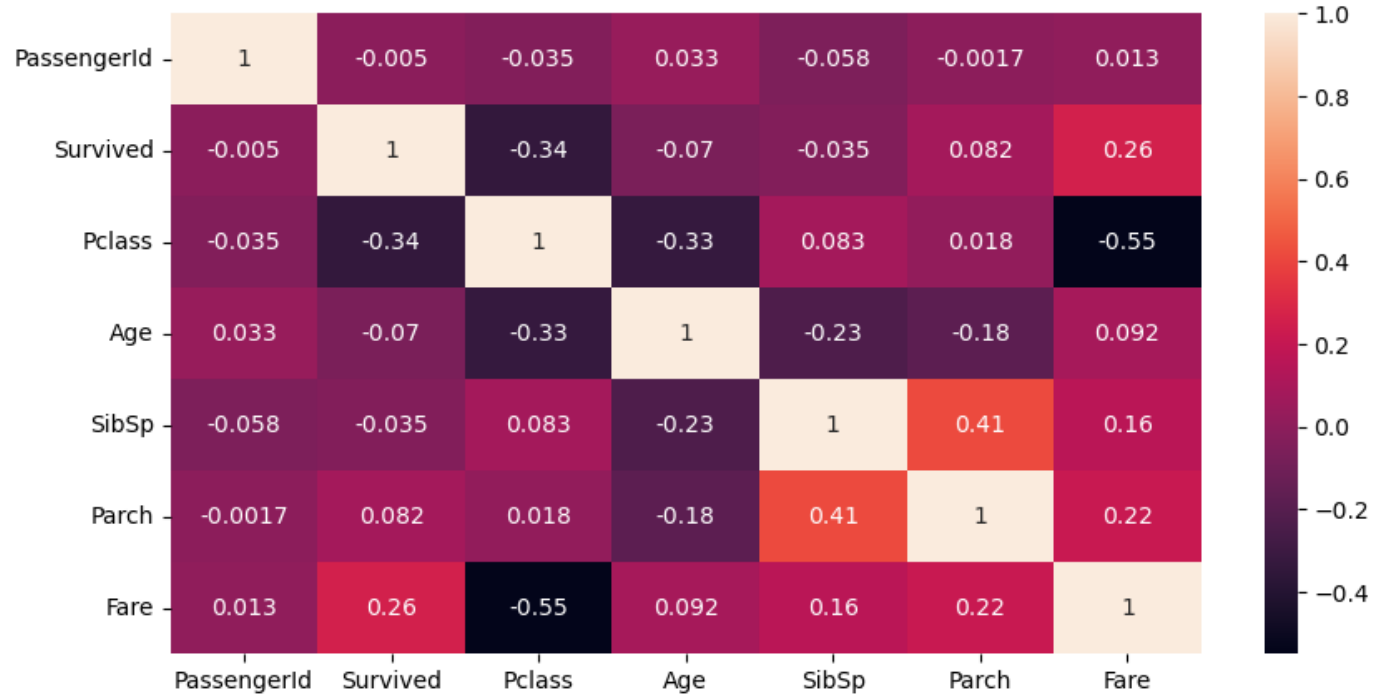
Out[22]: Text(0.5, 1.0, 'Pair Plot')



In [23]:
```
corr=df.corr()
plt.subplots(figsize=(10,5))
sns.heatmap(corr,annot=True)
```

```
C:\Users\tejbh\AppData\Local\Temp\ipykernel_25916\1909905835.py:1: FutureWarning: The de
fault value of numeric_only in DataFrame.corr is deprecated. In a future version, it wil
l default to False. Select only valid columns or specify the value of numeric_only to si
lence this warning.
  corr=df.corr()
```

Out[23]:
```
<Axes: >
```



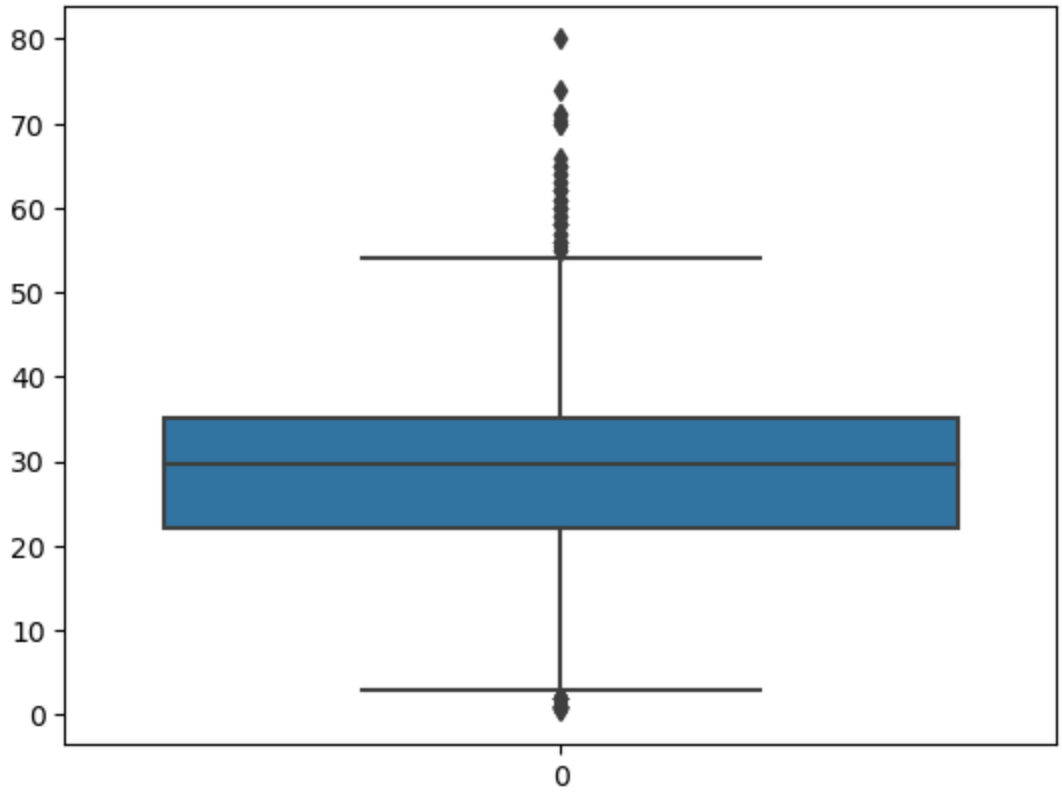
In [24]:
```
df.median()
```

```
C:\Users\tejbh\AppData\Local\Temp\ipykernel_25916\530051474.py:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
  df.median()
```

Out[24]:
```
PassengerId    446.000000
Survived         0.000000
Pclass           3.000000
Age             29.699118
SibSp            0.000000
Parch            0.000000
Fare            14.454200
dtype: float64
```

In [25]:
```
sns.boxplot(df.Age)
```

Out[25]:
```
<Axes: >
```



In [26]:
```
from scipy import stats
```

In [27]:
```
z_scores = np.abs(stats.zscore(df['Age']))
max_threshold=3
outliers = df['Age'][z_scores > max_threshold]
```

In [28]:
```
print(outliers)
```

```
96     71.0
116    70.5
493    71.0
630    80.0
672    70.0
745    70.0
851    74.0
Name: Age, dtype: float64
```

In [29]:
```
q1=df.Age.quantile(0.25)
q3=df.Age.quantile(0.75)
IQR=q3-q1

upperlim=q3+1.5*IQR
lowerlim=q1-1.5*IQR

df["Age"]=np.where(df.Age>upperlim,29.699118,df.Age)
df["Age"]=np.where(df.Age<lowerlim,29.699118,df.Age)
df
```

Out[29]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	male	22.000000	1	0	7.2500	S
1	2	1	1	female	38.000000	1	0	71.2833	C
2	3	1	3	female	26.000000	0	0	7.9250	S
3	4	1	1	female	35.000000	1	0	53.1000	S
4	5	0	3	male	35.000000	0	0	8.0500	S

...	...	...	...	...	...	...	...	...	...
886	887	0	2	male	27.000000	0	0	13.0000	S
887	888	1	1	female	19.000000	0	0	30.0000	S
888	889	0	3	female	29.699118	1	2	23.4500	S
889	890	1	1	male	26.000000	0	0	30.0000	C
890	891	0	3	male	32.000000	0	0	7.7500	Q

891 rows × 9 columns

In [30]:
```
q1=df.Fare.quantile(0.25)
q3=df.Fare.quantile(0.75)
IQR=q3-q1

upperlim=q3+1.5*IQR
lowerlim=q1-1.5*IQR

df["Fare"]=np.where(df.Fare>upperlim,14.454200,df.Fare)
```

In [31]:
```
df
```

Out[31]:

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	1	0	3	male	22.000000	1	0	7.2500	S
1	2	1	1	female	38.000000	1	0	14.4542	C
2	3	1	3	female	26.000000	0	0	7.9250	S
3	4	1	1	female	35.000000	1	0	53.1000	S
4	5	0	3	male	35.000000	0	0	8.0500	S
...	...	...	...	...	...	...	...	...	...
886	887	0	2	male	27.000000	0	0	13.0000	S
887	888	1	1	female	19.000000	0	0	30.0000	S
888	889	0	3	female	29.699118	1	2	23.4500	S
889	890	1	1	male	26.000000	0	0	30.0000	C
890	891	0	3	male	32.000000	0	0	7.7500	Q

891 rows × 9 columns

In [32]:
```
x=df.iloc[:,2:]
x
```

Out[32]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	male	22.000000	1	0	7.2500	S
1	1	female	38.000000	1	0	14.4542	C
2	3	female	26.000000	0	0	7.9250	S
3	1	female	35.000000	1	0	53.1000	S
4	3	male	35.000000	0	0	8.0500	S
...	...	...	...	...	...	...	...

886	2	male	27.000000	0	0	13.0000	S
887	1	female	19.000000	0	0	30.0000	S
888	3	female	29.699118	1	2	23.4500	S
889	1	male	26.000000	0	0	30.0000	C
890	3	male	32.000000	0	0	7.7500	Q

891 rows × 7 columns

In [33]:
```
y=df["Survived"]
y
```

Out[33]:
```
0       0
1       1
2       1
3       1
4       0
       ..
886     0
887     1
888     0
889     1
890     0
Name: Survived, Length: 891, dtype: int64
```

In [34]:
```
from sklearn.preprocessing import LabelEncoder
```

In [35]:
```
le=LabelEncoder()
x["Sex"]=le.fit_transform(x["Sex"])
x.head()
```

Out[35]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1	22.0	1	0	7.2500	S
1	1	0	38.0	1	0	14.4542	C
2	3	0	26.0	0	0	7.9250	S
3	1	0	35.0	1	0	53.1000	S
4	3	1	35.0	0	0	8.0500	S

In [36]:
```
embarked=pd.get_dummies(x["Embarked"],drop_first=True)
x=pd.concat([x,embarked],axis=1)
x.drop(["Embarked"],axis=1,inplace=True)
x.head()
```

Out[36]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Q	S
0	3	1	22.0	1	0	7.2500	0	1
1	1	0	38.0	1	0	14.4542	0	0
2	3	0	26.0	0	0	7.9250	0	1
3	1	0	35.0	1	0	53.1000	0	1
4	3	1	35.0	0	0	8.0500	0	1

In [37]:
```
from sklearn.preprocessing import StandardScaler
```

In [38]:
```
sc=StandardScaler()
x=sc.fit_transform(x)
x
```

Out[38]:
```
array([[ 0.82737724,  0.73769513, -0.70858401, ..., -0.79755374,
        -0.30756234,  0.61583843],
       [-1.56610693, -1.35557354,  0.92494776, ..., -0.23055642,
        -0.30756234, -1.62380254],
       [ 0.82737724, -1.35557354, -0.30020106, ..., -0.74442873,
        -0.30756234,  0.61583843],
       ...,
       [ 0.82737724, -1.35557354,  0.07746307, ...,  0.47744647,
        -0.30756234,  0.61583843],
       [-1.56610693,  0.73769513, -0.30020106, ...,  0.99295581,
        -0.30756234, -1.62380254],
       [ 0.82737724,  0.73769513,  0.31237335, ..., -0.75820188,
         3.25137334, -1.62380254]])
```

In [39]:
```
from sklearn.model_selection import train_test_split
```

In [40]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

In [41]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[41]:
```
((623, 8), (268, 8), (623,), (268,))
```

In [42]:
```
x_train
```

Out[42]:
```
array([[-1.56610693e+00,  7.37695132e-01,  2.25219232e+00, ...,
         7.21427989e-01, -3.07562343e-01,  6.15838425e-01],
       [-1.56610693e+00, -1.35557354e+00,  2.04800085e+00, ...,
        -2.30556425e-01, -3.07562343e-01, -1.62380254e+00],
       [ 8.27377244e-01,  7.37695132e-01,  7.74631084e-02, ...,
         2.32304862e+00, -3.07562343e-01,  6.15838425e-01],
       ...,
       [ 8.27377244e-01,  7.37695132e-01,  7.74630724e-02, ...,
        -7.59516233e-01,  3.25137334e+00, -1.62380254e+00],
       [ 8.27377244e-01, -1.35557354e+00,  7.20756290e-01, ...,
         1.28898315e-03, -3.07562343e-01,  6.15838425e-01],
       [-3.69364841e-01,  7.37695132e-01,  7.74631084e-02, ...,
         1.70128926e+00, -3.07562343e-01,  6.15838425e-01]])
```