8) Data Visualization I

1. Use the inbuilt dataset 'titanic'. The dataset contains 891 rows and contains information about the passengers who boarded the unfortunate Titanic ship. Use the Seaborn library to see if we can find any patterns in the data.
2. Write a code to check how the price of the ticket (column name: 'fare') for each passenger

is distributed by plotting a histogram.import pandas as pd import matplotlib.pyplot as plt import seaborn as sbn import numpy as np

In [2]:
```
df = pd.read_csv('titanic.csv')
df.head()
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [3]:
```
df.isna().sum()
```

Out[3]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [4]:
```
df.drop(columns=['Cabin'], inplace=True)
```

In [5]:
```
df['Age'].fillna(df['Age'].median(), inplace=True)
```

```
C:\Users\swapn\AppData\Local\Temp\ipykernel_6872\1933487976.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


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

In [6]:
```
df['Fare'].fillna(df['Fare'].median(), inplace=True)
```

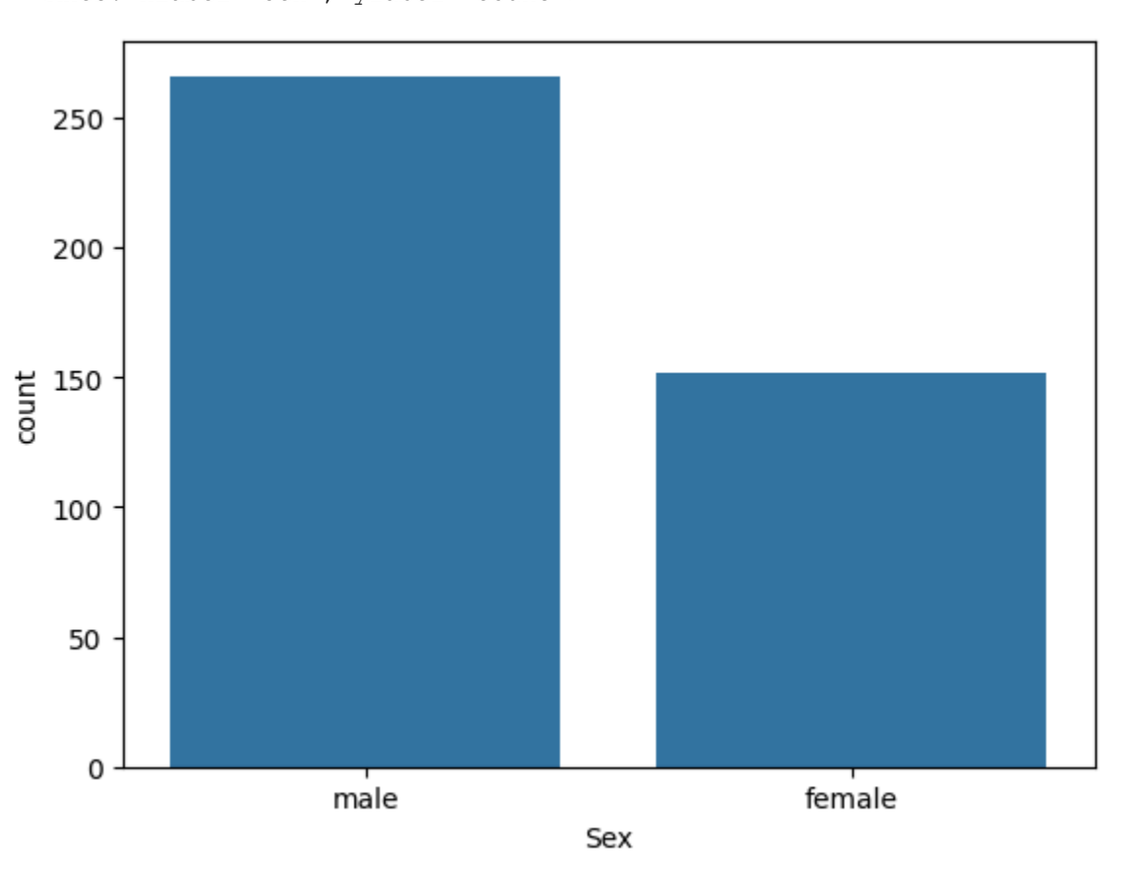
```
C:\Users\swapn\AppData\Local\Temp\ipykernel_6872\309285478.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


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

In [7]:
```
df.isna().sum()
```

Out[7]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

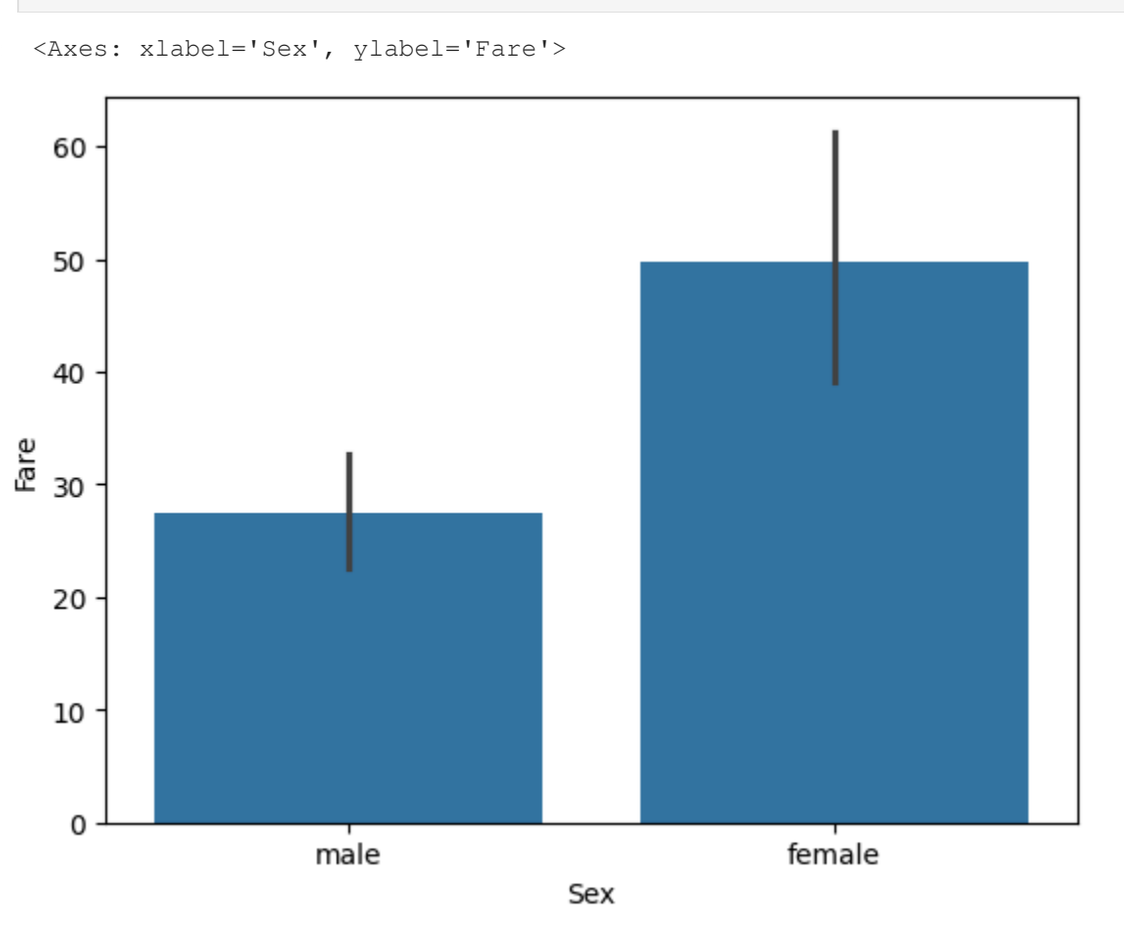
In [8]:
```
sbn.countplot(x='Sex', data=df)
```

Out[8]: `<Axes: xlabel='Sex', ylabel='count'>`



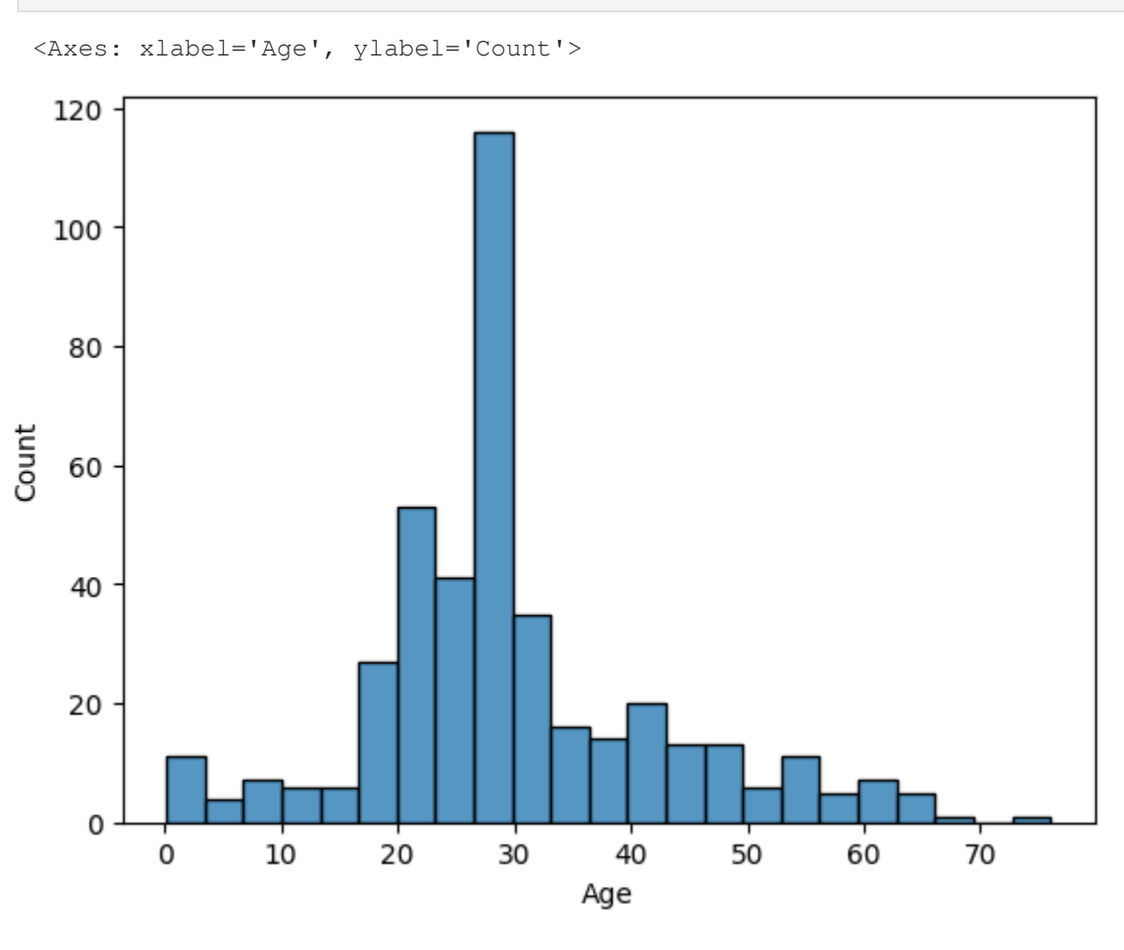
In [9]:
```
sbn.barplot(data=df, x='Sex', y='Fare')
```

Out[9]: `<Axes: xlabel='Sex', ylabel='Fare'>`



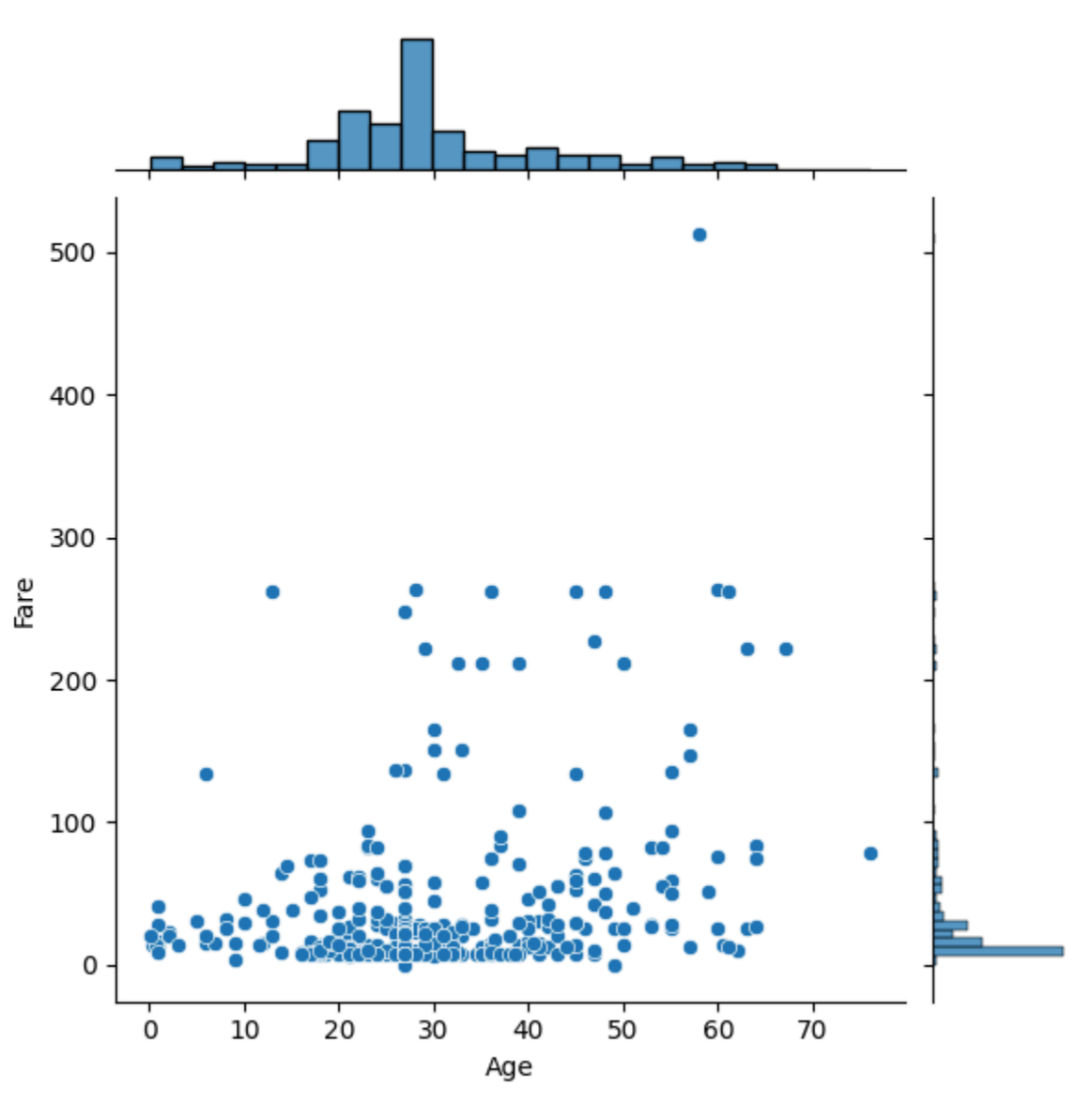
In [10]:
```
sbn.histplot(data=df, x='Age')
```

Out[10]: `<Axes: xlabel='Age', ylabel='Count'>`



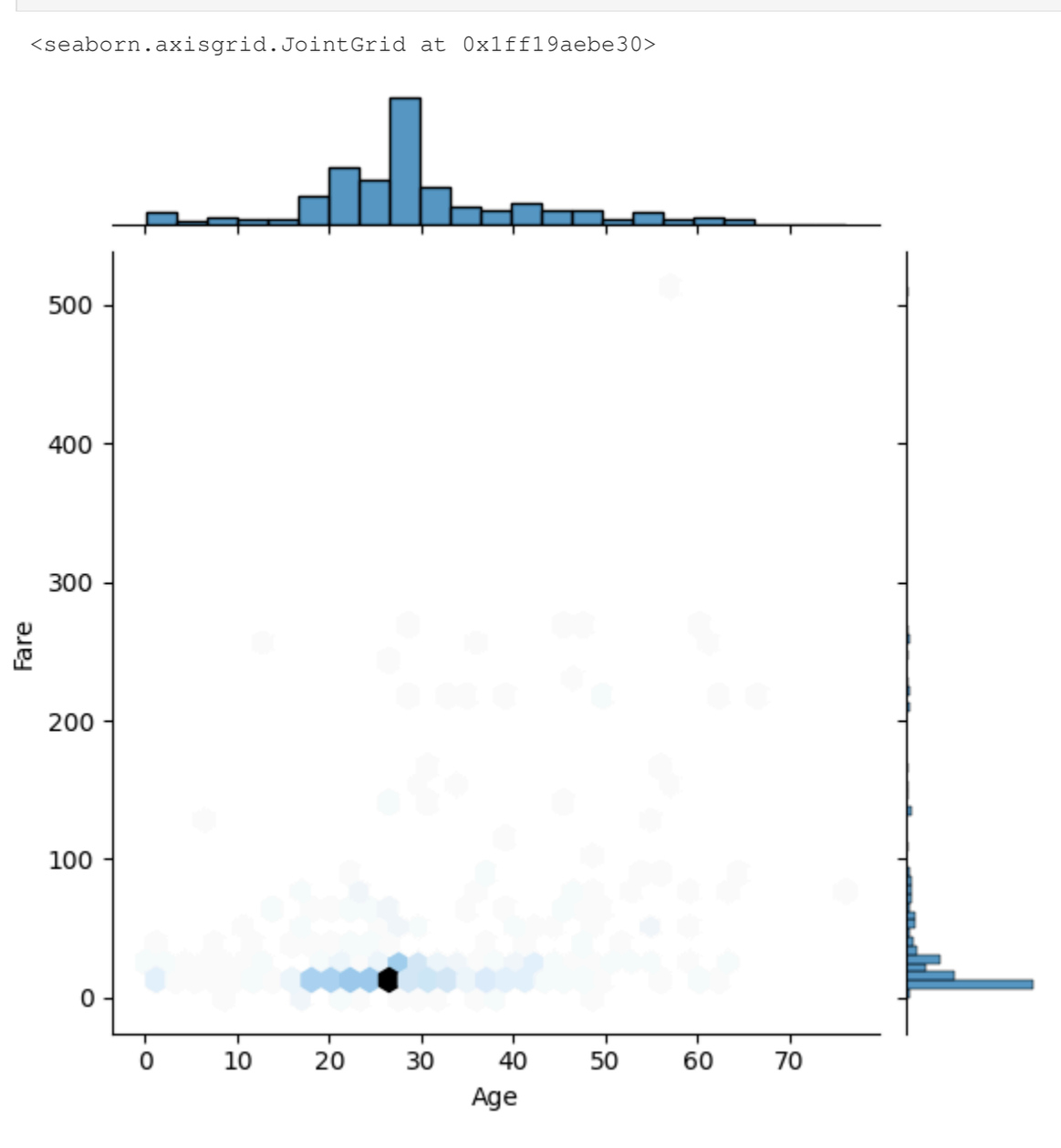
In [11]:
```
sbn.jointplot(data=df, x='Age', y='Fare')
```

Out[11]: `<seaborn.axisgrid.JointGrid at 0x1ff1bf23a40>`



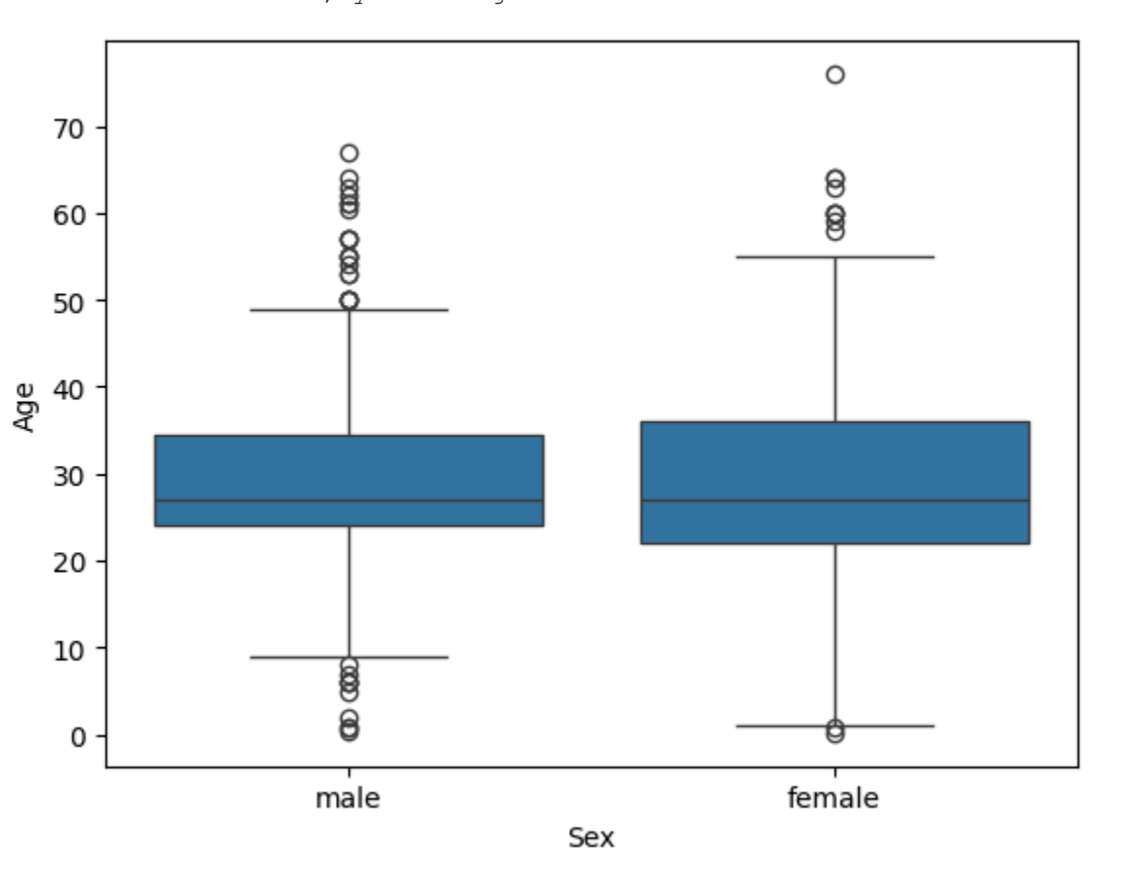
In [12]:
```
sbn.jointplot(data=df, x='Age', y='Fare', kind='hex')
```

Out[12]: `<seaborn.axisgrid.JointGrid at 0x1ff19aebe30>`



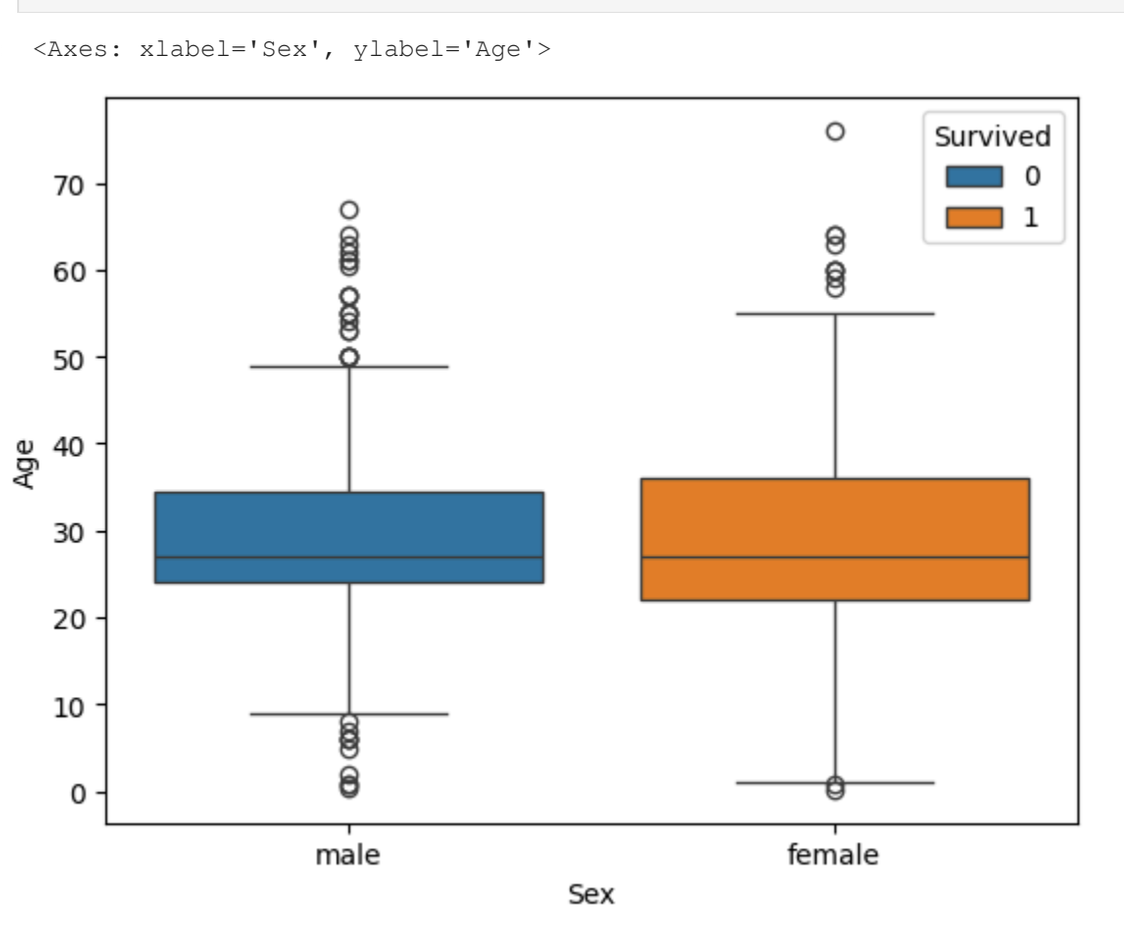
In [13]:
```
sbn.boxplot(data=df, x='Sex', y='Age')
```

Out[13]: `<Axes: xlabel='Sex', ylabel='Age'>`



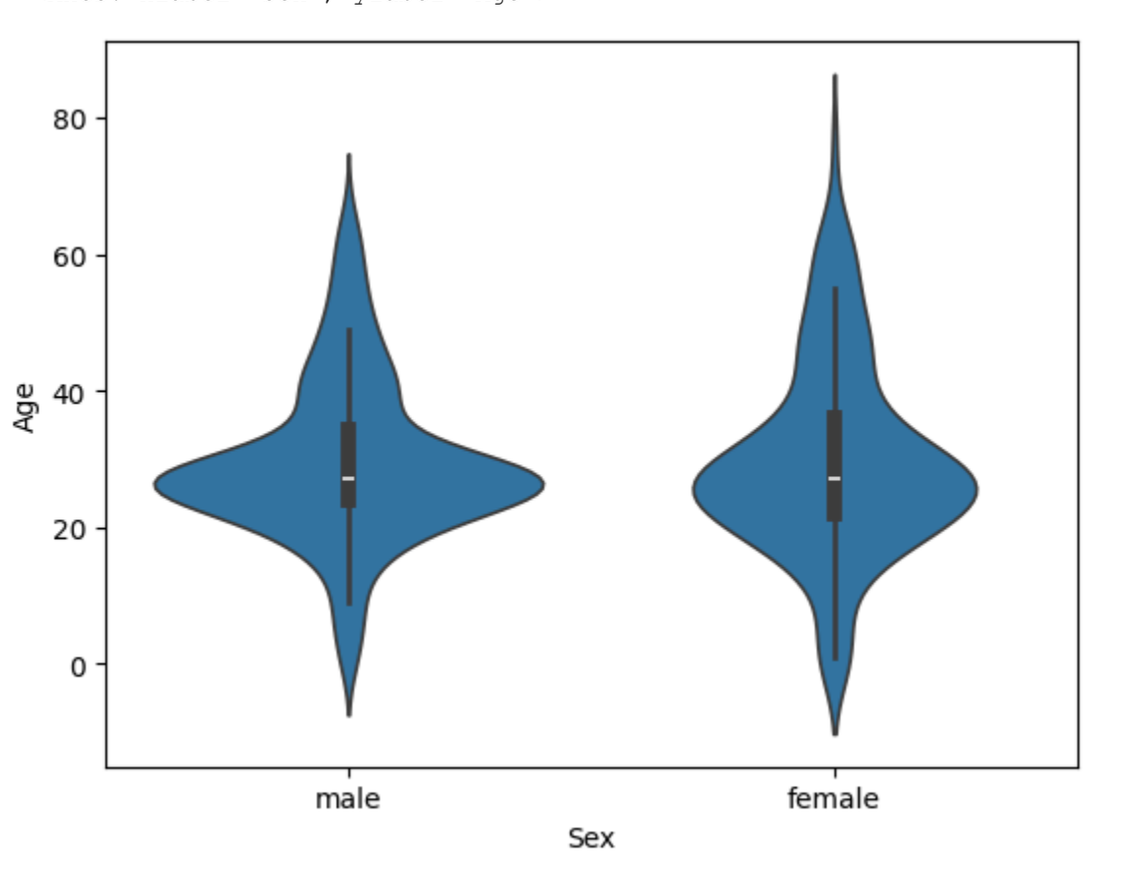
In [23]:
```
sbn.boxplot(data=df, x='Sex', y='Age', hue='Survived')
```

Out[23]: `<Axes: xlabel='Sex', ylabel='Age'>`



In [15]:
```
sbn.violinplot(data=df, x='Sex', y='Age')
```

Out[15]: `<Axes: xlabel='Sex', ylabel='Age'>`

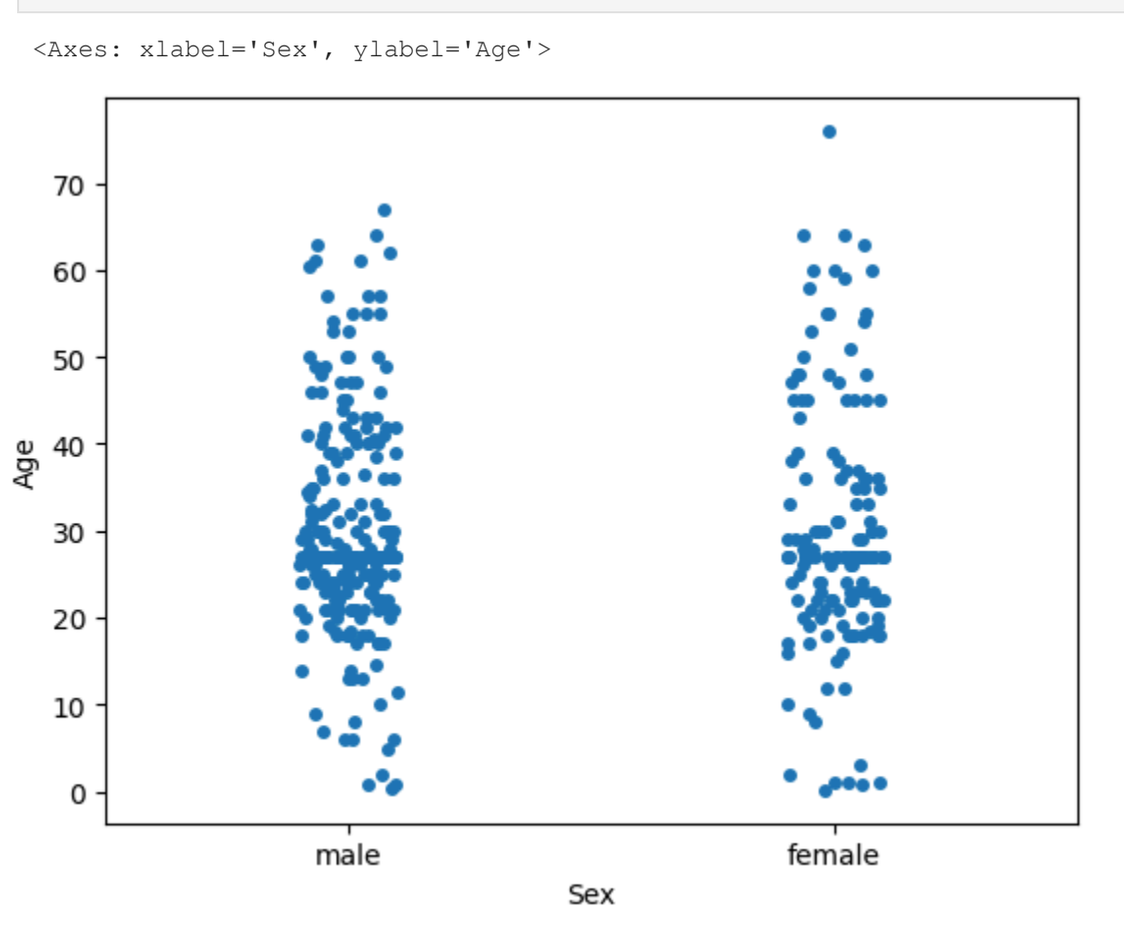


In [16]:
```
sbn.violinplot(data=df, x='Sex', y='Age', hue='Survived')
```

Out[16]: `<Axes: xlabel='Sex', ylabel='Age'>`

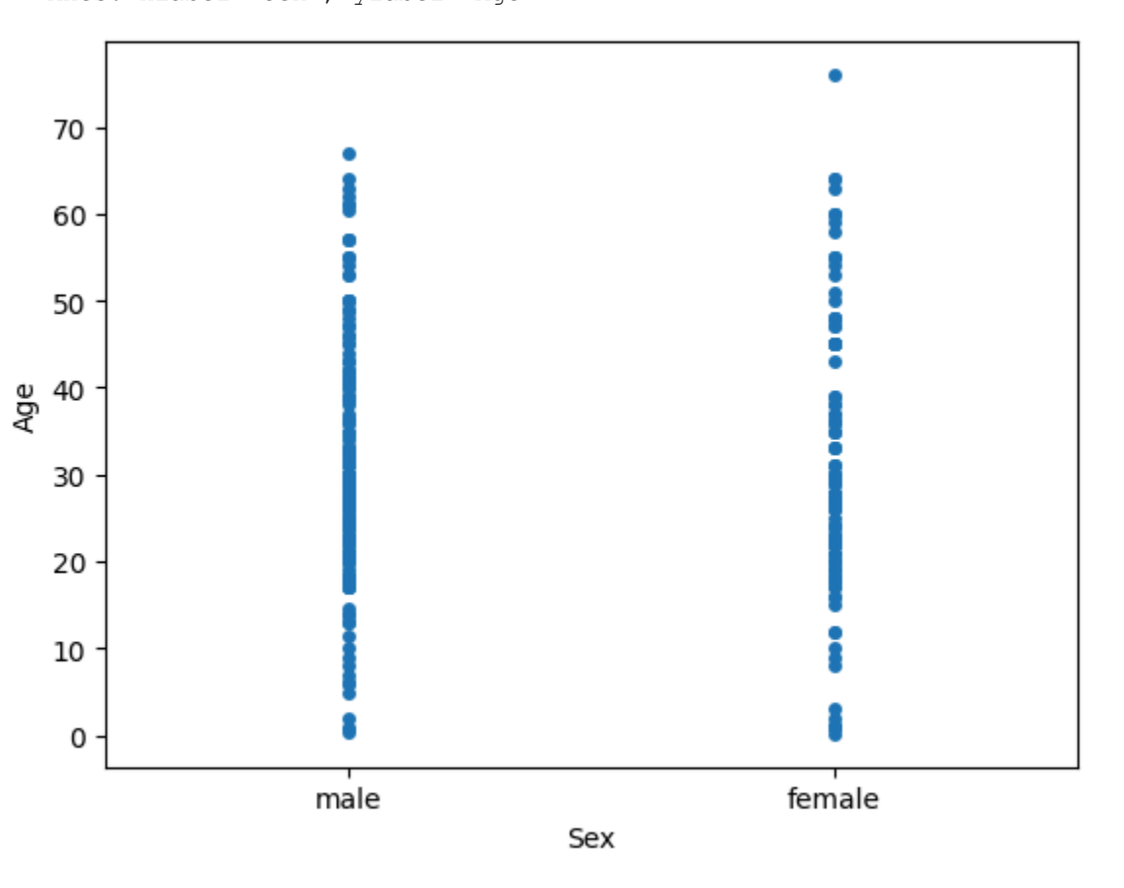
In [17]:
```
sbn.stripplot(data=df, x='Sex', y='Age')
```

Out[17]: `<Axes: xlabel='Sex', ylabel='Age'>`



In [18]:
```
sbn.stripplot(data=df, x='Sex', y='Age', jitter=False)
```

Out[18]: `<Axes: xlabel='Sex', ylabel='Age'>`

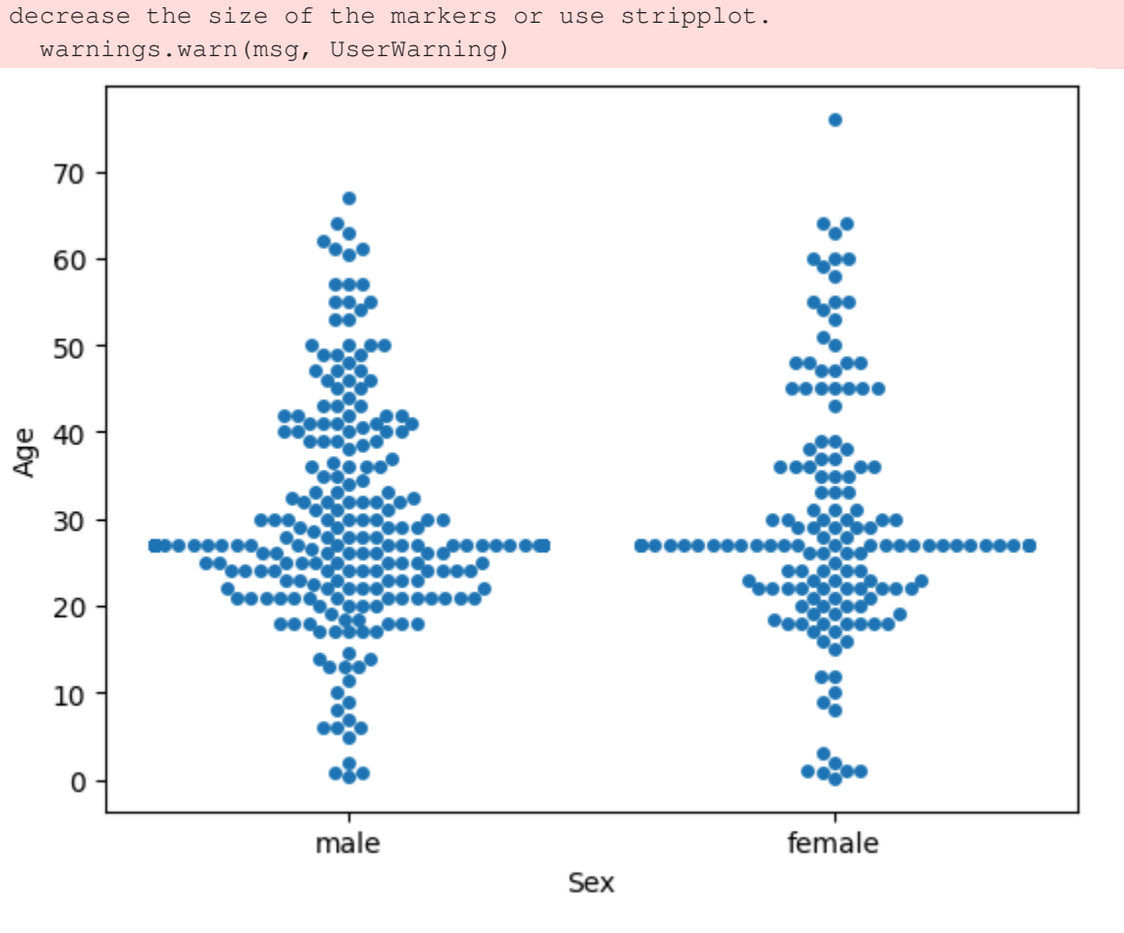


In [19]:
```
sbn.swarmplot(data=df, x='Sex', y='Age')
```

```
C:\Users\swapn\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCache\local-packages\Python312\site-packages\seaborn\categorical.py:3399: UserWarning: 10.5% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

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

```
C:\Users\swapn\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCache\local-packages\Python312\site-packages\seaborn\categorical.py:3399: UserWarning: 18.8% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

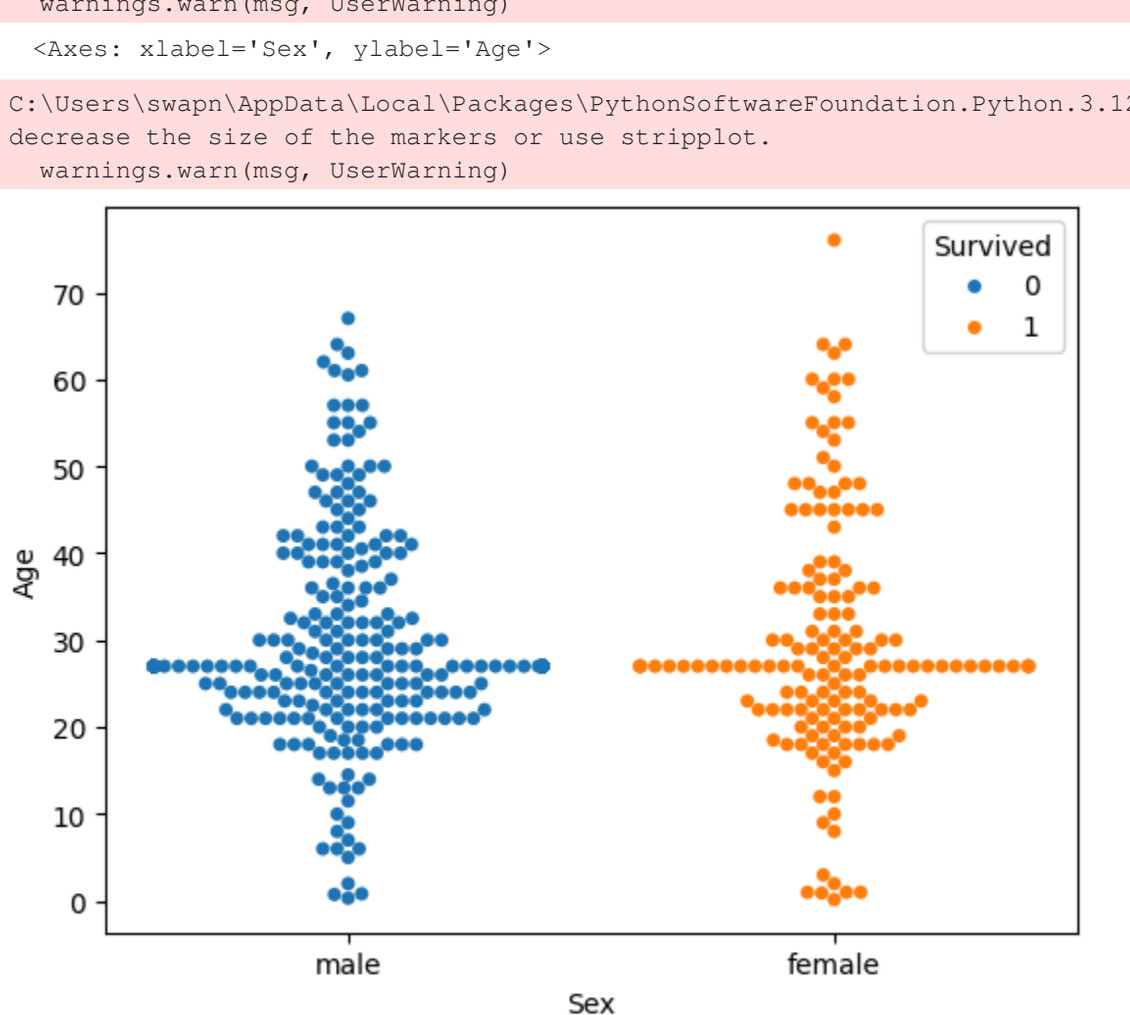


In [24]:
```
sbn.swarmplot(data=df, x='Sex', y='Age',hue='Survived')
```

```
C:\Users\swapn\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCache\local-packages\Python312\site-packages\seaborn\categorical.py:3399: UserWarning: 10.5% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

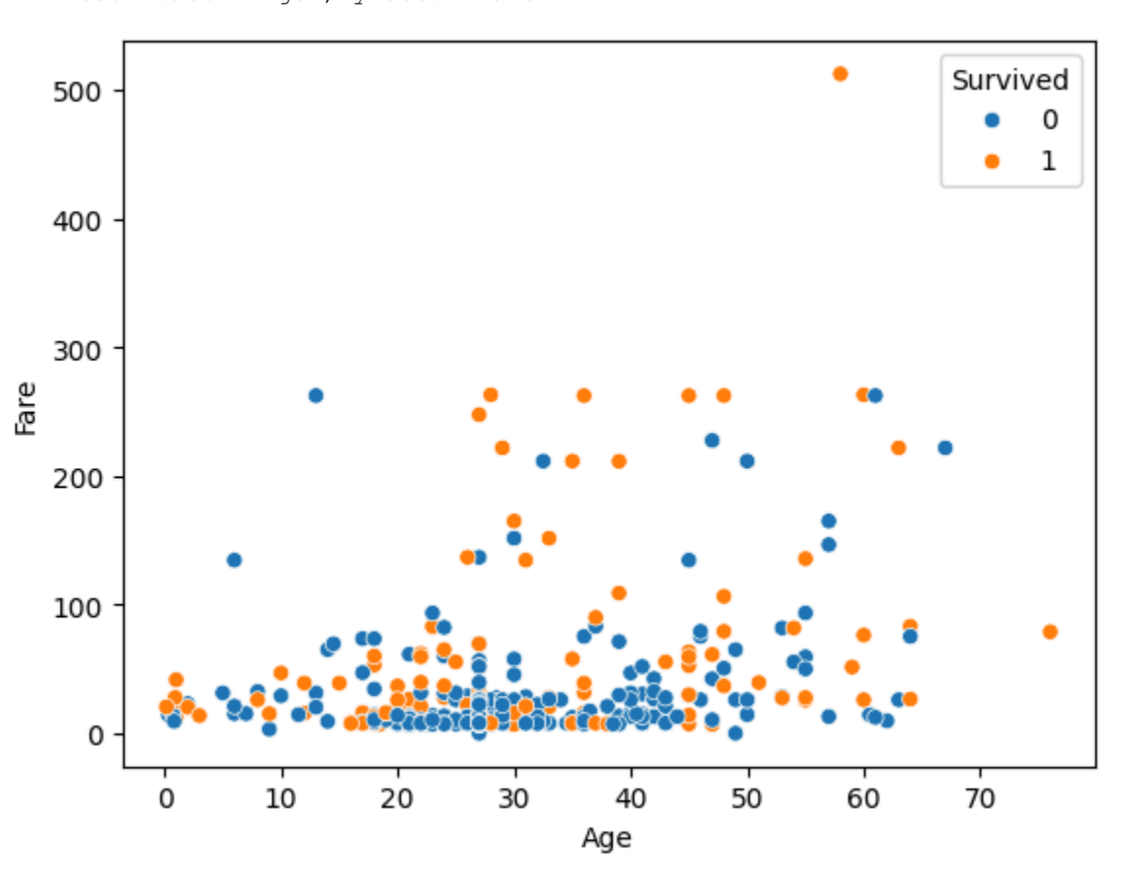
Out[24]: `<Axes: xlabel='Sex', ylabel='Age'>`

```
C:\Users\swapn\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCache\local-packages\Python312\site-packages\seaborn\categorical.py:3399: UserWarning: 18.8% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



In [25]:
```
sbn.scatterplot(data=df, x='Age', y='Fare', hue='Survived')
```

Out[25]: `<Axes: xlabel='Age', ylabel='Fare'>`



In [42]:
```
fig, axes = plt.subplots(2, 2, sharex=True, figsize=(10,10))
fig.suptitle('Bigger 1 row x 2 columns axes with no data')
sbn.scatterplot(ax=axes[0][0],data=df, x='Age', y='Fare', hue='Survived')
axes[0,0].set_ylabel("sfe")
```

Out[42]: `Text(0, 0.5, 'sfe')`