

```
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
dataset = sns.load_dataset('titanic')
```

```
titanic
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked
0 Third	0	3	male	22.0	1	0	7.2500	S
1 First	1	1	female	38.0	1	0	71.2833	C
2 Third	1	3	female	26.0	0	0	7.9250	S
3 First	1	1	female	35.0	1	0	53.1000	S
4 Third	0	3	male	35.0	0	0	8.0500	S
...
886 Second	0	2	male	27.0	0	0	13.0000	S
887 First	1	1	female	19.0	0	0	30.0000	S
888 Third	0	3	female	NaN	1	2	23.4500	S
889 First	1	1	male	26.0	0	0	30.0000	C
890 Third	0	3	male	32.0	0	0	7.7500	Q

	who	adult_male	deck	embark_town	alive	alone
0	man	True	NaN	Southampton	no	False
1	woman	False	C	Cherbourg	yes	False
2	woman	False	NaN	Southampton	yes	True
3	woman	False	C	Southampton	yes	False
4	man	True	NaN	Southampton	no	True
...
886	man	True	NaN	Southampton	no	True
887	woman	False	B	Southampton	yes	True
888	woman	False	NaN	Southampton	no	False
889	man	True	C	Cherbourg	yes	True
890	man	True	NaN	Queenstown	no	True

```
[891 rows x 15 columns]
```

```
titanic.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   survived              891 non-null    int64
1   pclass                891 non-null    int64
2   sex                   891 non-null    object
3   age                   714 non-null    float64
4   sibsp                 891 non-null    int64
5   parch                 891 non-null    int64
6   fare                  891 non-null    float64
7   embarked              889 non-null    object
8   class                 891 non-null    category
9   who                   891 non-null    object
10  adult_male            891 non-null    bool
11  deck                  203 non-null    category
12  embark_town           889 non-null    object
13  alive                 891 non-null    object
14  alone                 891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB

```

```
x=titanic["fare"]
```

```

x
0      7.2500
1     71.2833
2      7.9250
3     53.1000
4      8.0500
...
886    13.0000
887    30.0000
888    23.4500
889    30.0000
890     7.7500
Name: fare, Length: 891, dtype: float64

```

```
titanic.describe()
```

	survived	pclass	age	sibsp	parch
fare					
count	891.000000	891.000000	714.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594
std	0.486592	0.836071	14.526497	1.102743	0.806057
	49.693429				

min	0.000000	1.000000	0.420000	0.000000	0.000000
0.000000					
25%	0.000000	2.000000	20.125000	0.000000	0.000000
7.910400					
50%	0.000000	3.000000	28.000000	0.000000	0.000000
14.454200					
75%	1.000000	3.000000	38.000000	1.000000	0.000000
31.000000					
max	1.000000	3.000000	80.000000	8.000000	6.000000
512.329200					

```
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

RangeIndex: 891 entries, 0 to 890

Data columns (total 15 columns):

#	Column	Non-Null Count	Dtype
0	survived	891 non-null	int64
1	pclass	891 non-null	int64
2	sex	891 non-null	object
3	age	714 non-null	float64
4	sibsp	891 non-null	int64
5	parch	891 non-null	int64
6	fare	891 non-null	float64
7	embarked	889 non-null	object
8	class	891 non-null	category
9	who	891 non-null	object
10	adult_male	891 non-null	bool
11	deck	203 non-null	category
12	embark_town	889 non-null	object
13	alive	891 non-null	object
14	alone	891 non-null	bool

```
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
```

```
memory usage: 80.7+ KB
```

```
titanic_cleaned=titanic.drop(['pclass','embarked','deck','embark_town'],axis=1)
```

```
titanic_cleaned.head(15)
```

adult_male	survived	sex	age	sibsp	parch	fare	class	who
0	0	male	22.0	1	0	7.2500	Third	man
1	1	female	38.0	1	0	71.2833	First	woman
2	1	female	26.0	0	0	7.9250	Third	woman
3	1	female	35.0	1	0	53.1000	First	woman

4	0	male	35.0	0	0	8.0500	Third	man
True								
5	0	male	NaN	0	0	8.4583	Third	man
True								
6	0	male	54.0	0	0	51.8625	First	man
True								
7	0	male	2.0	3	1	21.0750	Third	child
False								
8	1	female	27.0	0	2	11.1333	Third	woman
False								
9	1	female	14.0	1	0	30.0708	Second	child
False								
10	1	female	4.0	1	1	16.7000	Third	child
False								
11	1	female	58.0	0	0	26.5500	First	woman
False								
12	0	male	20.0	0	0	8.0500	Third	man
True								
13	0	male	39.0	1	5	31.2750	Third	man
True								
14	0	female	14.0	0	0	7.8542	Third	child
False								

	alive	alone
0	no	False
1	yes	False
2	yes	True
3	yes	False
4	no	True
5	no	True
6	no	True
7	no	False
8	yes	False
9	yes	False
10	yes	False
11	yes	True
12	no	True
13	no	False
14	no	True

```
titanic_cleaned.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
#   Column      Non-Null Count  Dtype
---  -
0   survived    891 non-null    int64
1   sex         891 non-null    object
2   age         714 non-null    float64
```

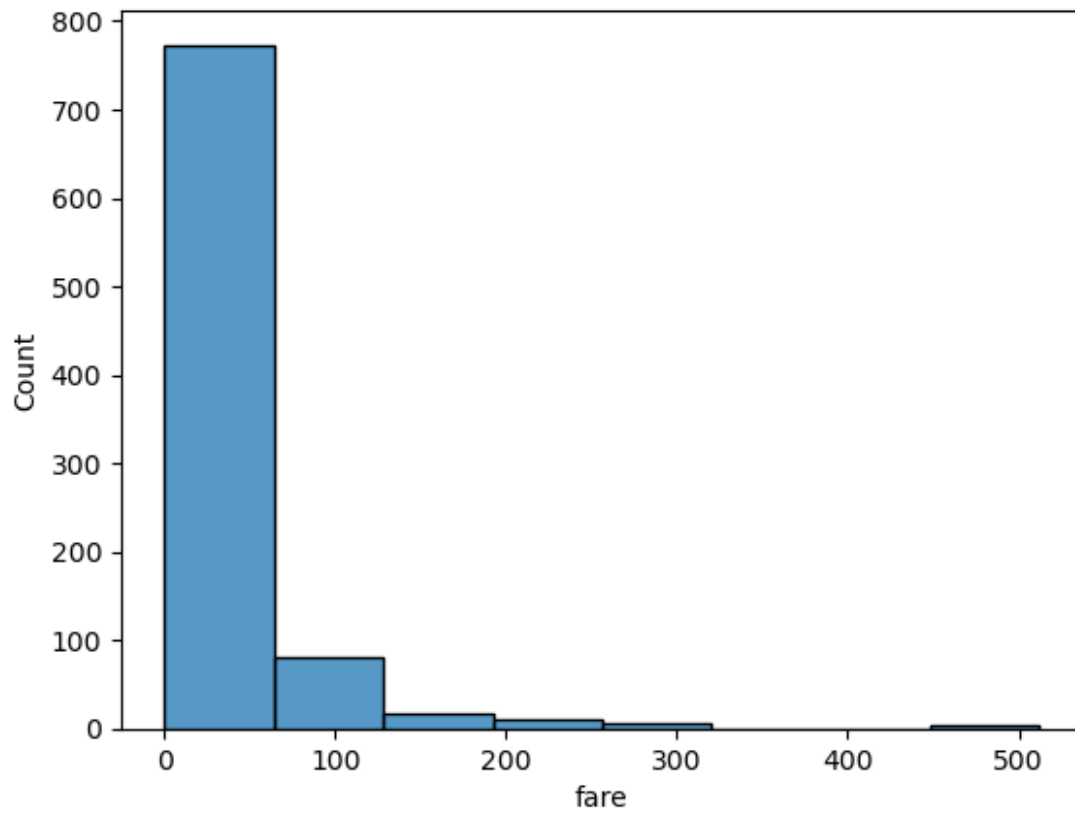
```
3  sibsp      891 non-null    int64
4  parch      891 non-null    int64
5  fare       891 non-null    float64
6  class      891 non-null    category
7  who        891 non-null    object
8  adult_male 891 non-null    bool
9  alive      891 non-null    object
10 alone      891 non-null    bool
dtypes: bool(2), category(1), float64(2), int64(3), object(3)
memory usage: 58.6+ KB
```

```
titanic_cleaned.isnull().sum()
```

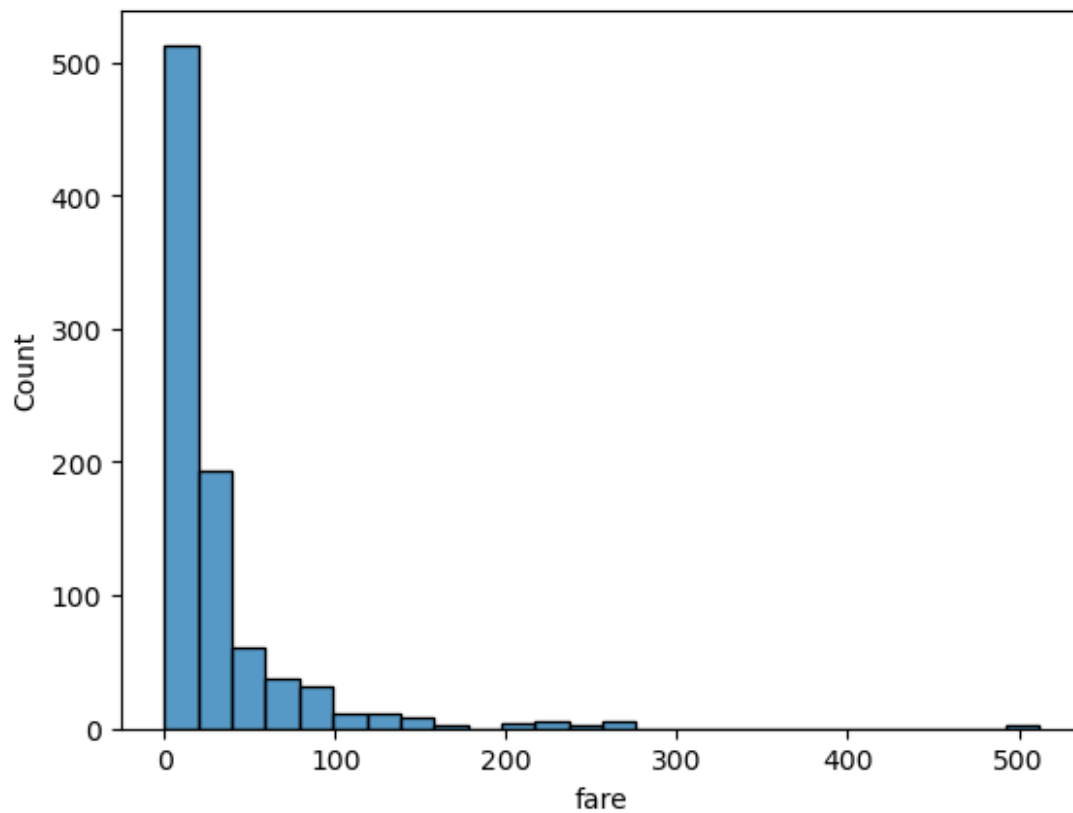
```
survived      0
sex            0
age           177
sibsp         0
parch         0
fare          0
class         0
who           0
adult_male    0
alive         0
alone         0
dtype: int64
```

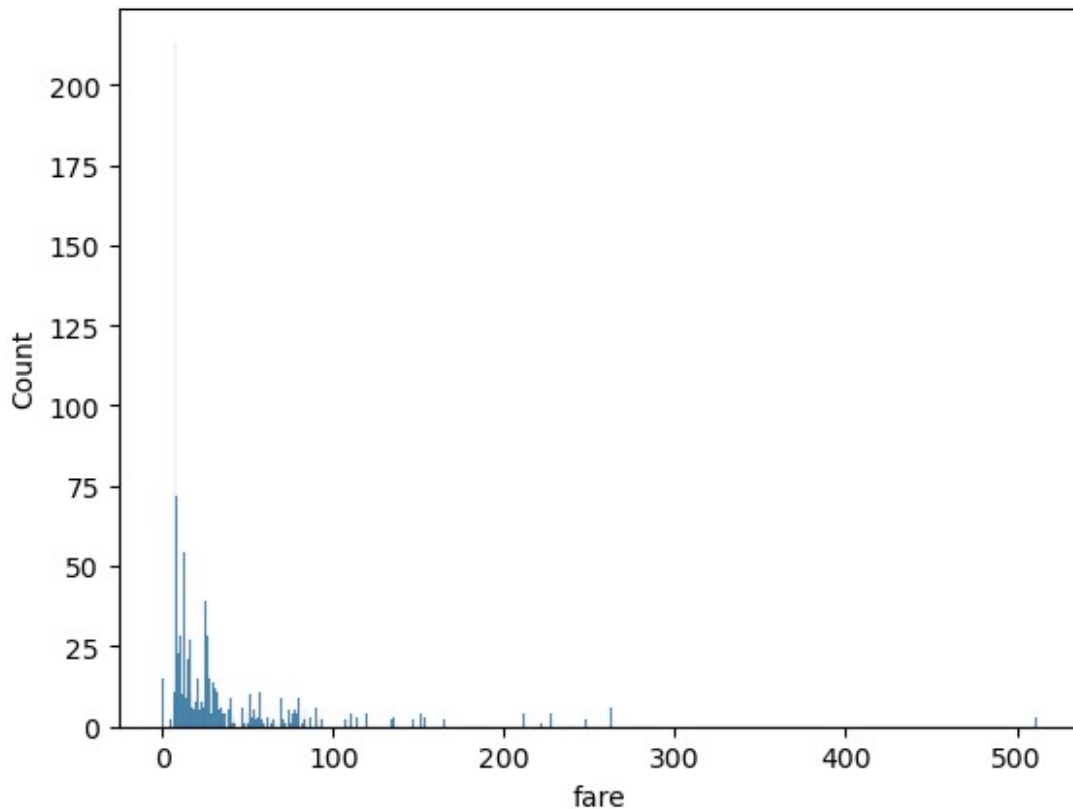
```
sns.histplot(data=titanic,x="fare",bins=8)
```

```
<Axes: xlabel='fare', ylabel='Count'>
```



```
sns.histplot(data=titanic,x="fare",binwidth=20)  
<Axes: xlabel='fare', ylabel='Count'>
```





```
sns.distplot(x = dataset['age'], bins = 10)
```

C:\Users\KJCOEMR\AppData\Local\Temp\ipykernel_4920\3209197554.py:1:
UserWarning:

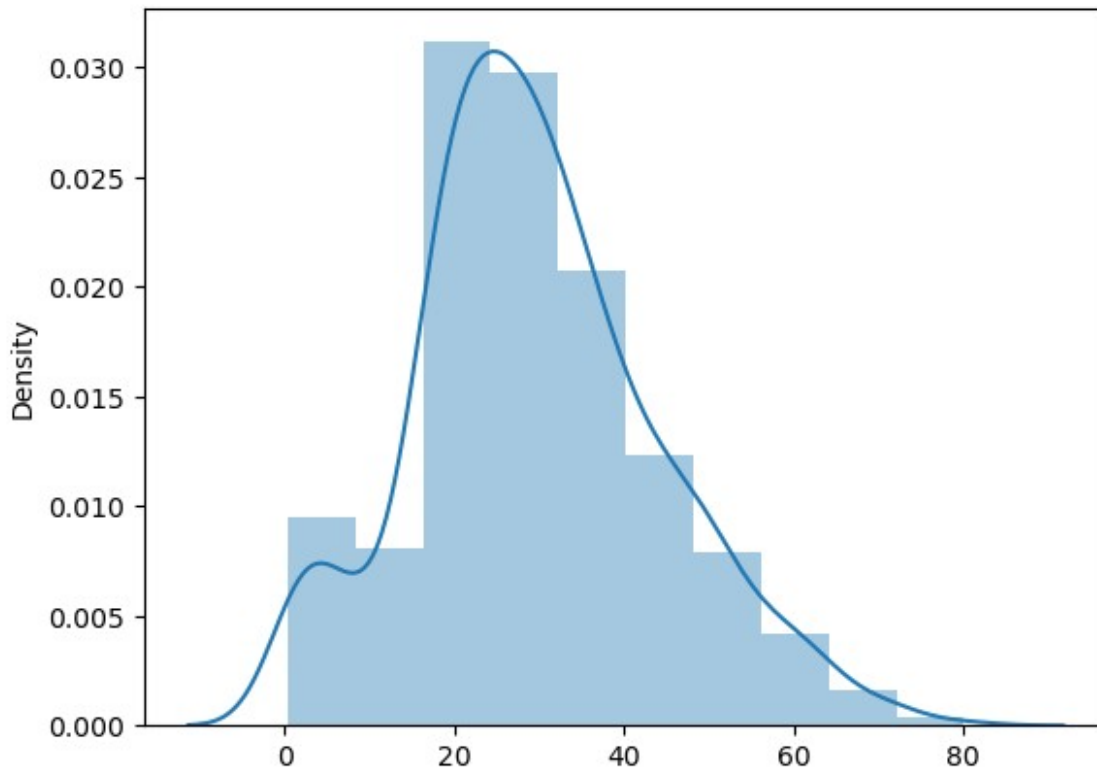
`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

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>

```
sns.distplot(x = dataset['age'], bins = 10)
```

<Axes: ylabel='Density'>



```
sns.distplot(dataset['age'], bins = 10, kde=False)
```

C:\Users\KJC0EMR\AppData\Local\Temp\ipykernel_4920\3517108427.py:1:
UserWarning:

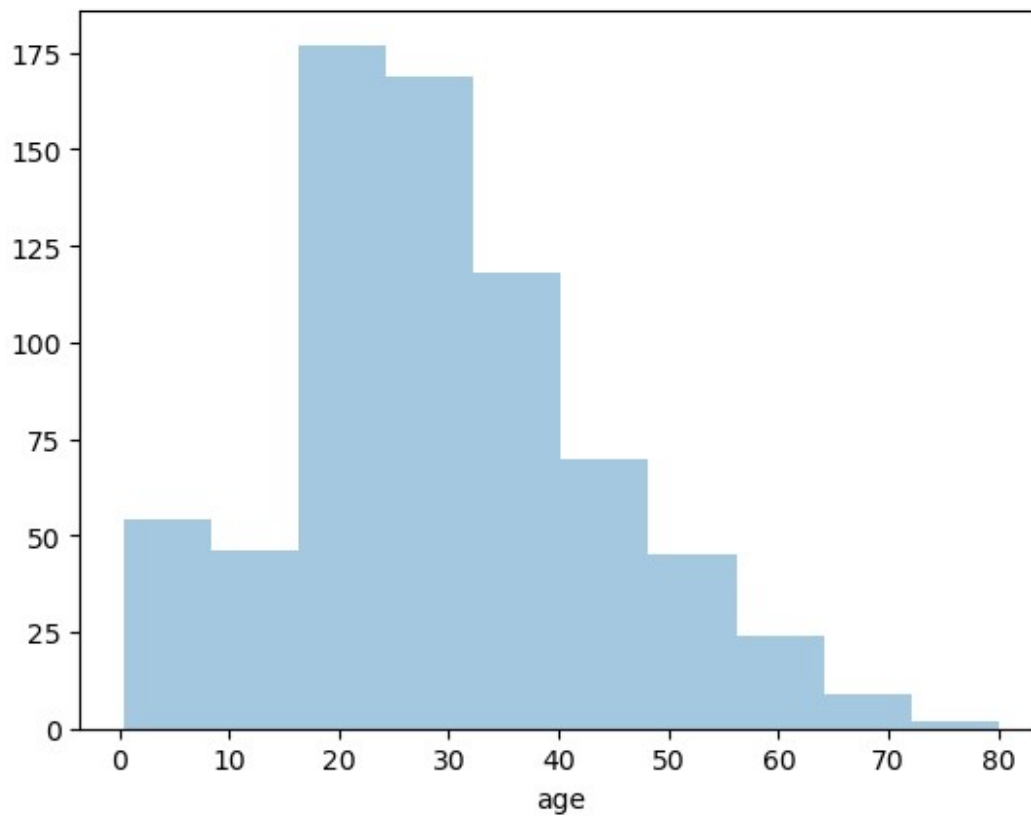
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

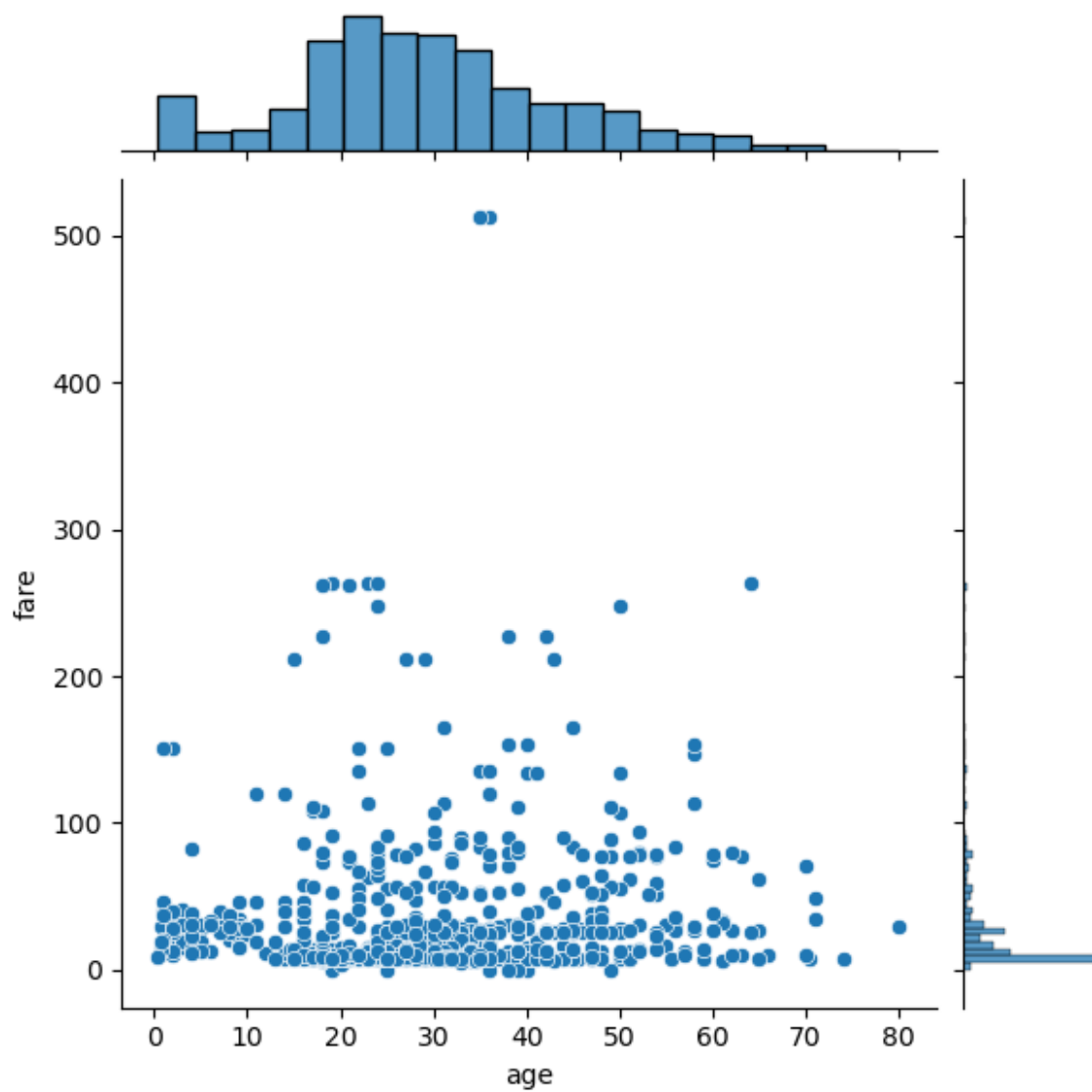
For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

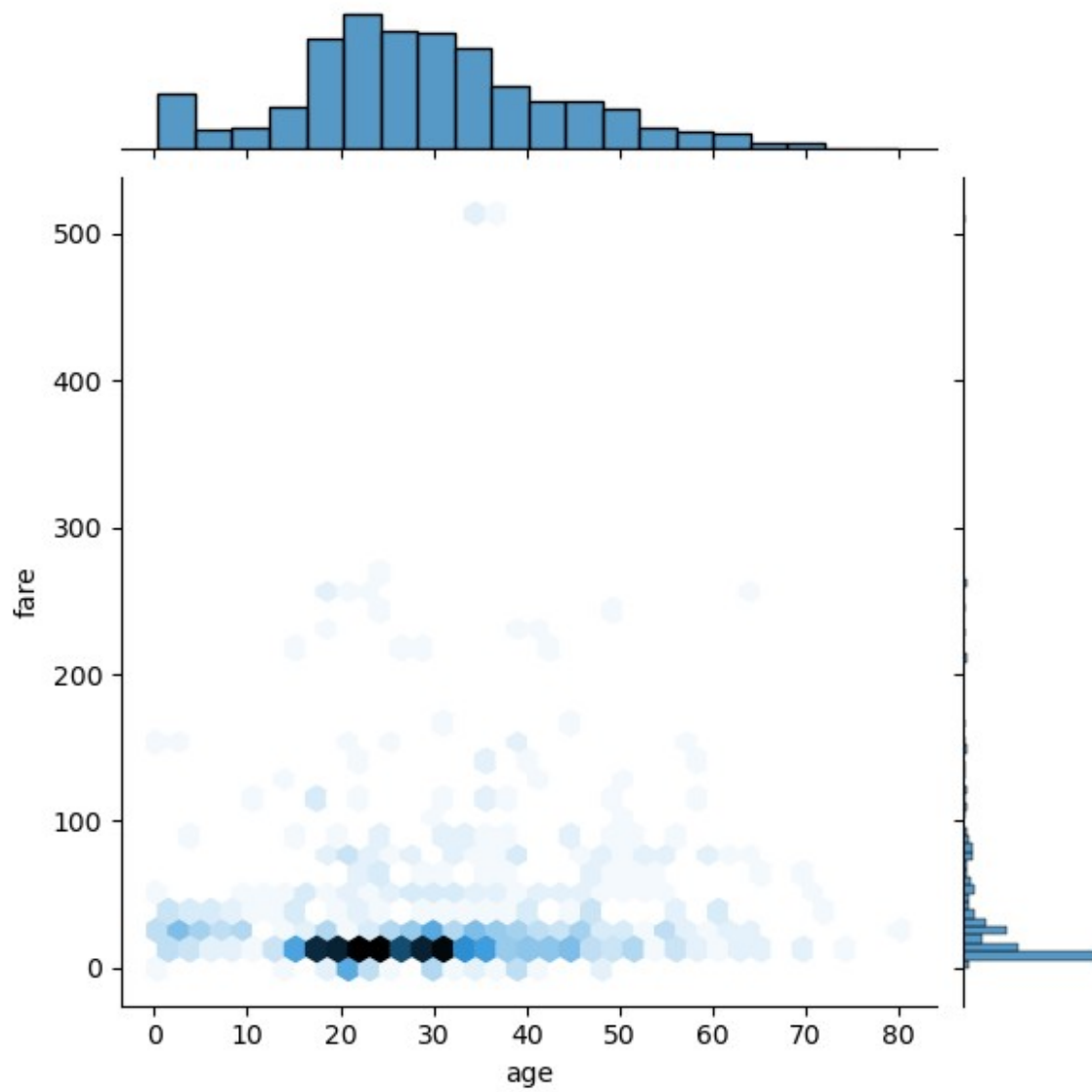
```
sns.distplot(dataset['age'], bins = 10, kde=False)
```

<Axes: xlabel='age'>



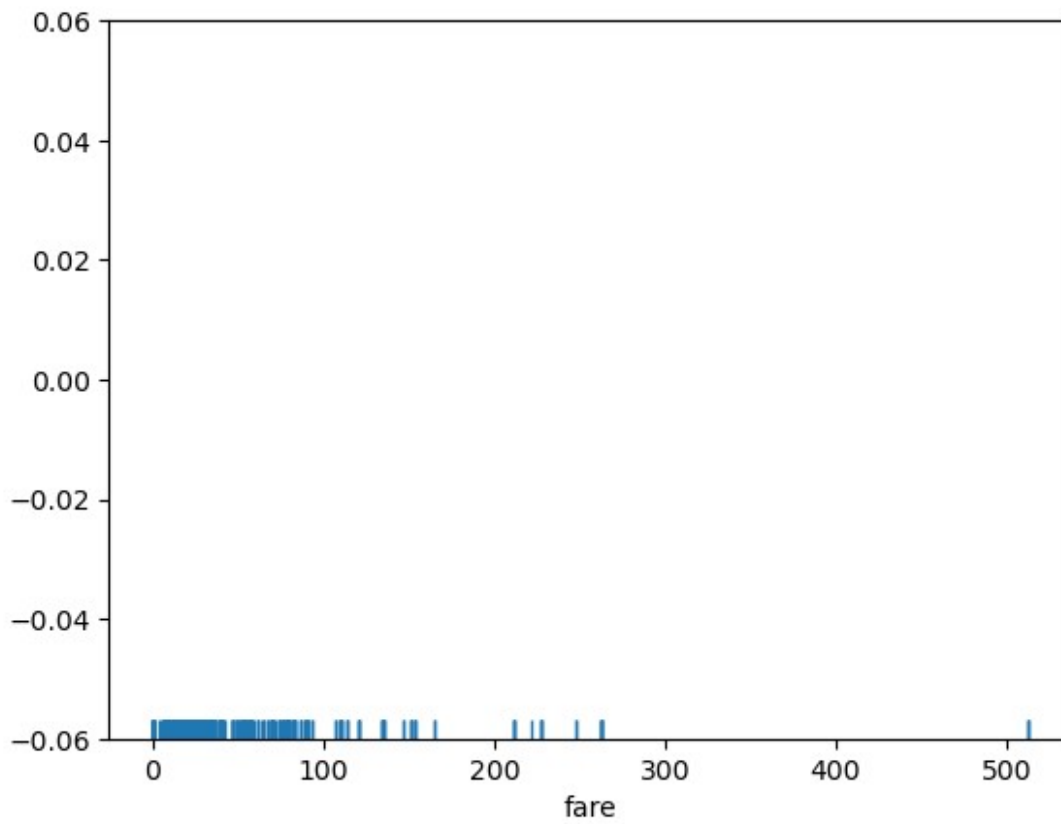
```
# For Plot 1
sns.jointplot(x = dataset['age'], y = dataset['fare'], kind =
'scatter')
# For Plot 2
sns.jointplot(x = dataset['age'], y = dataset['fare'], kind = 'hex')
<seaborn.axisgrid.JointGrid at 0x1f0fe78e960>
```



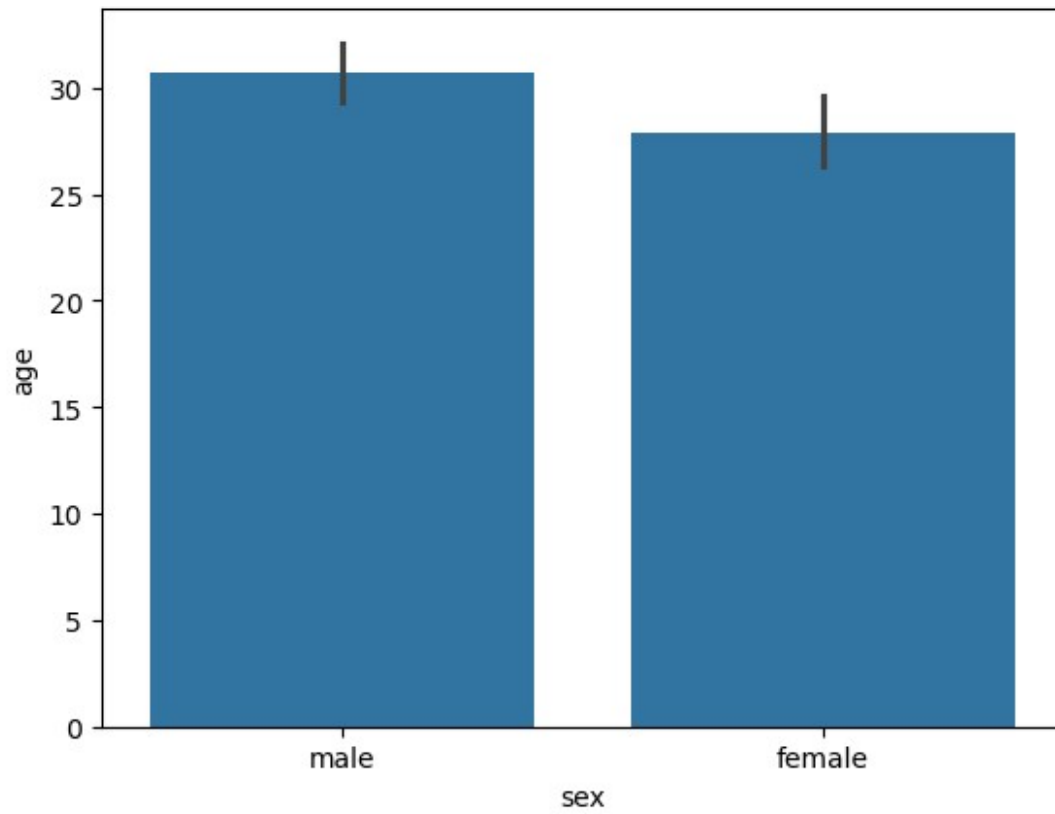


```
sns.rugplot(dataset['fare'])
```

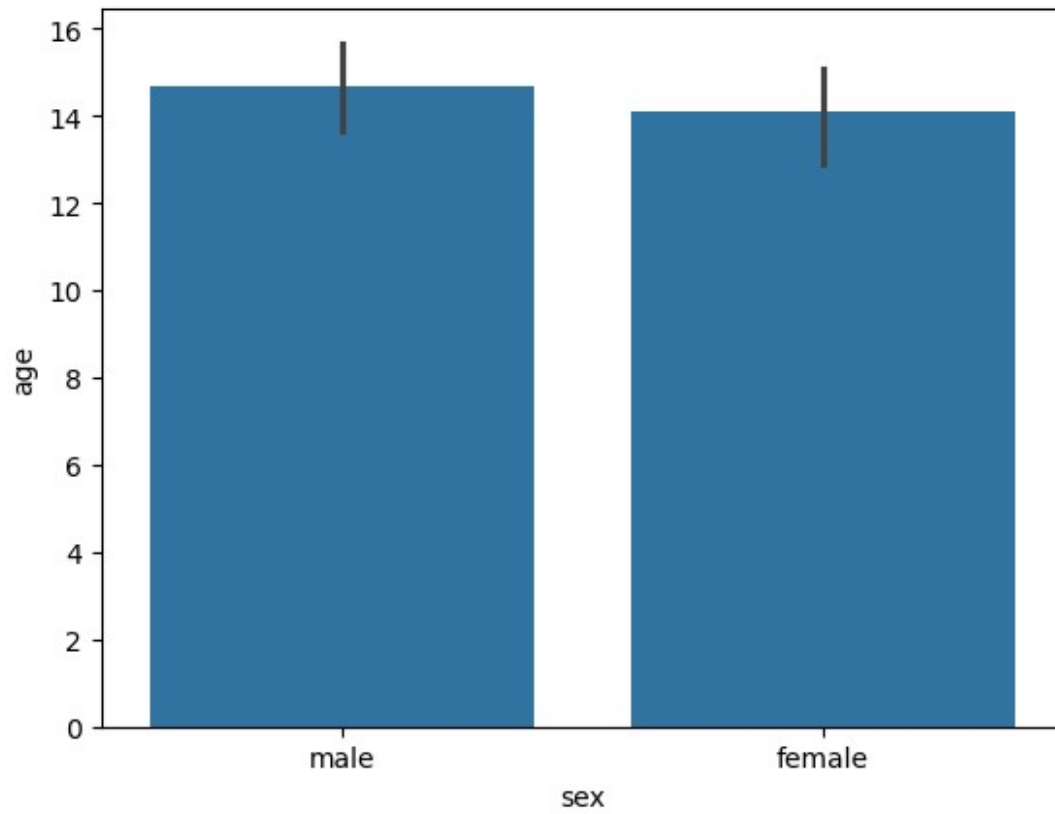
```
<Axes: xlabel='fare'>
```



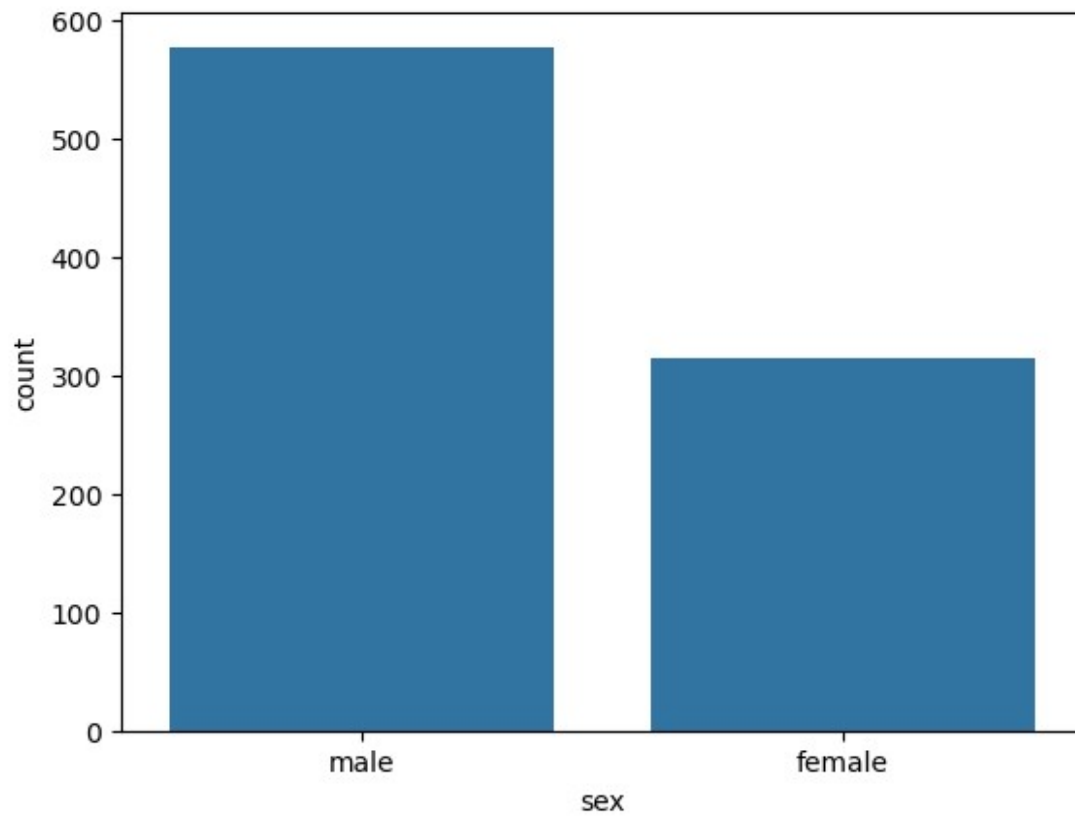
```
sns.barplot(x='sex', y='age', data=dataset)  
<Axes: xlabel='sex', ylabel='age'>
```



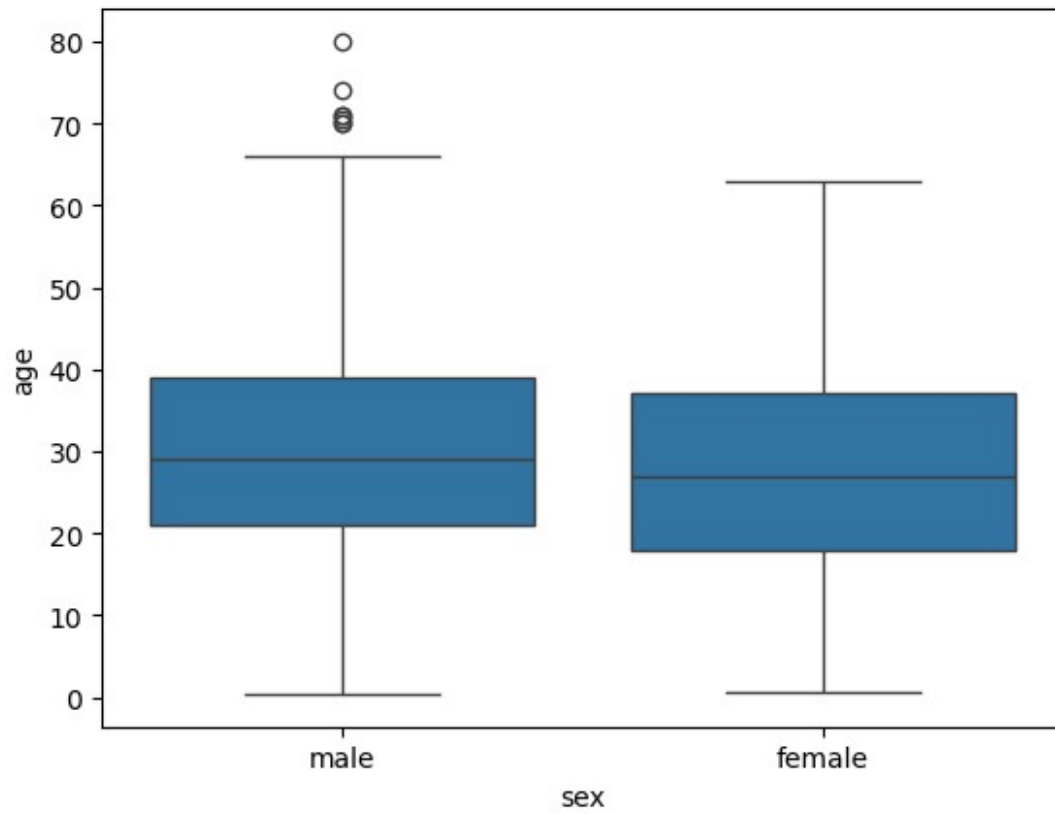
```
sns.barplot(x='sex', y='age', data=dataset, estimator=np.std)  
<Axes: xlabel='sex', ylabel='age'>
```



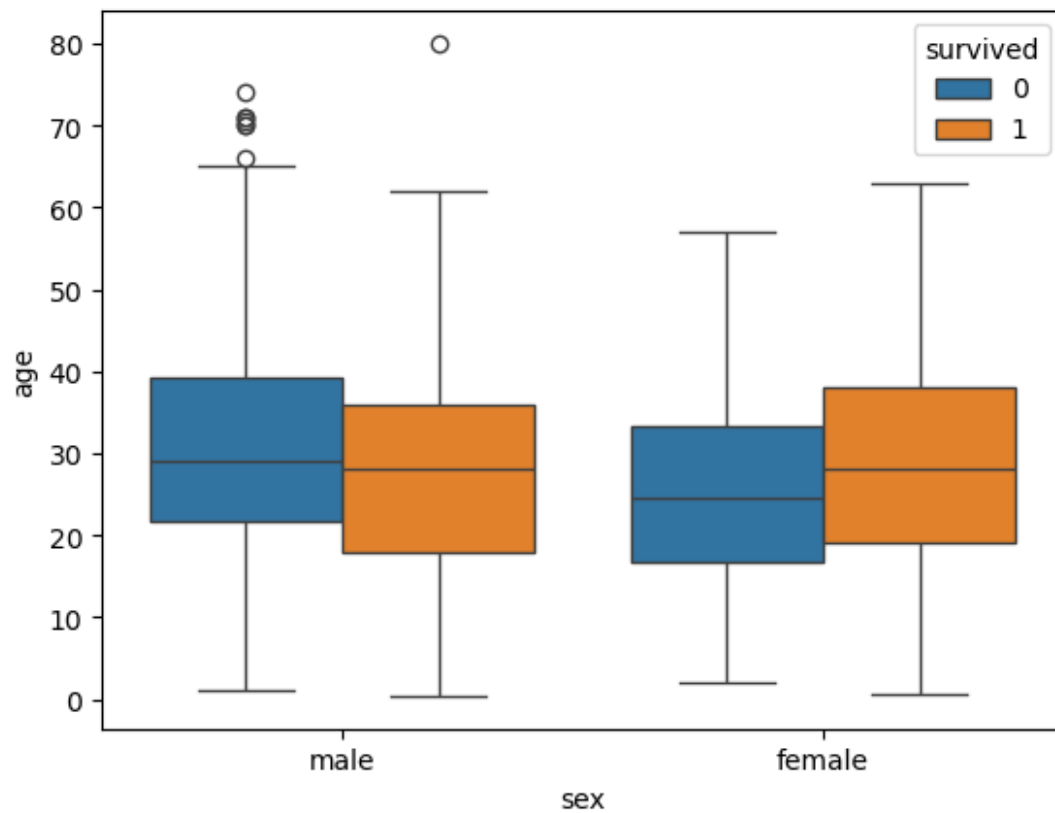
```
sns.countplot(x='sex', data=dataset)  
<Axes: xlabel='sex', ylabel='count'>
```



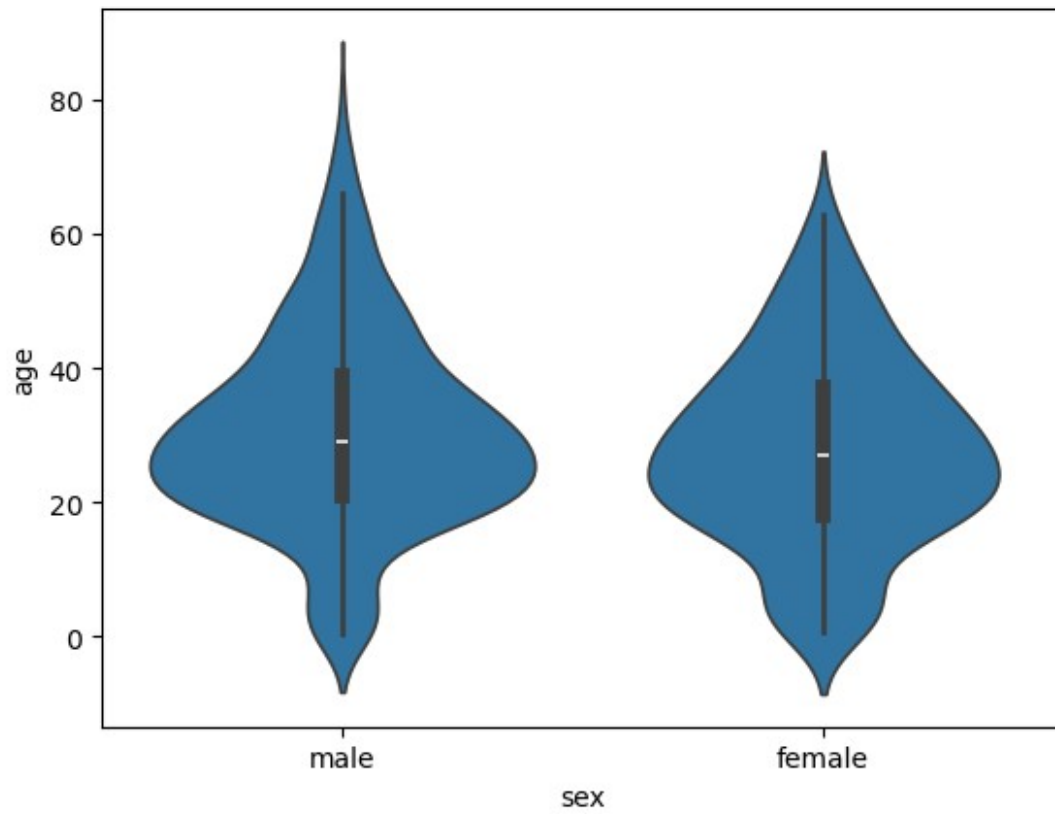
```
sns.boxplot(x='sex', y='age', data=dataset)  
<Axes: xlabel='sex', ylabel='age'>
```

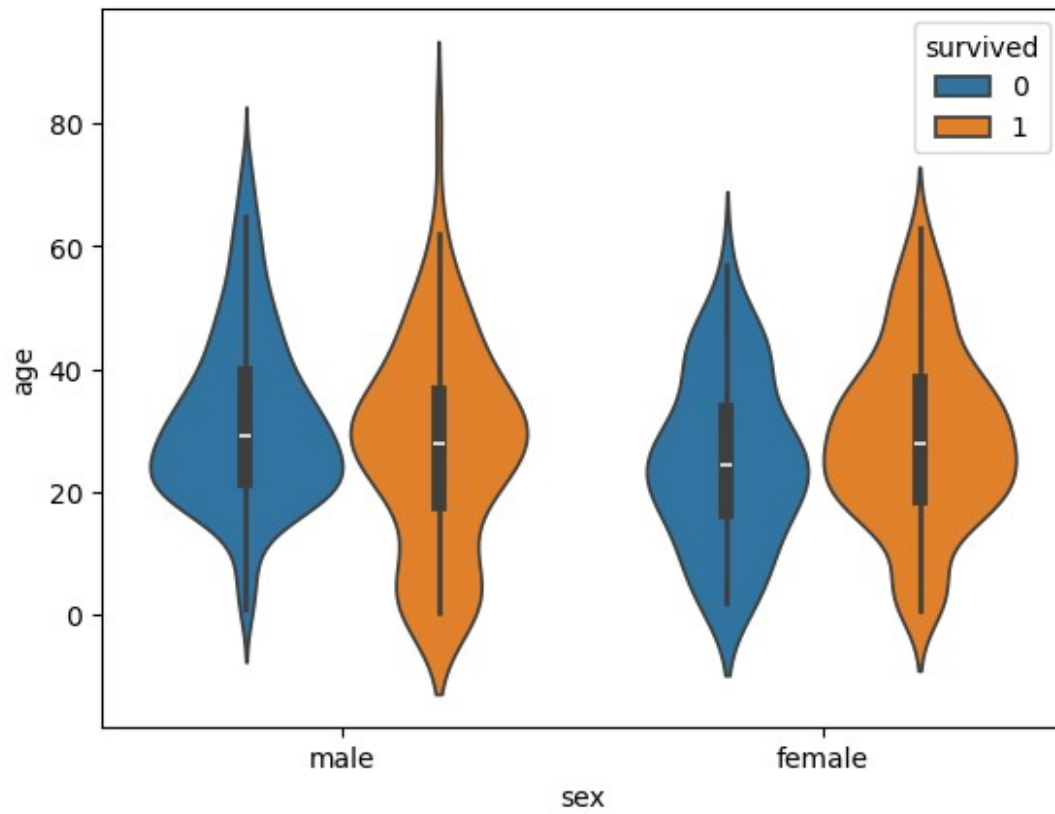
```
sns.boxplot(x='sex', y='age', data=dataset, hue="survived")  
<Axes: xlabel='sex', ylabel='age'>
```



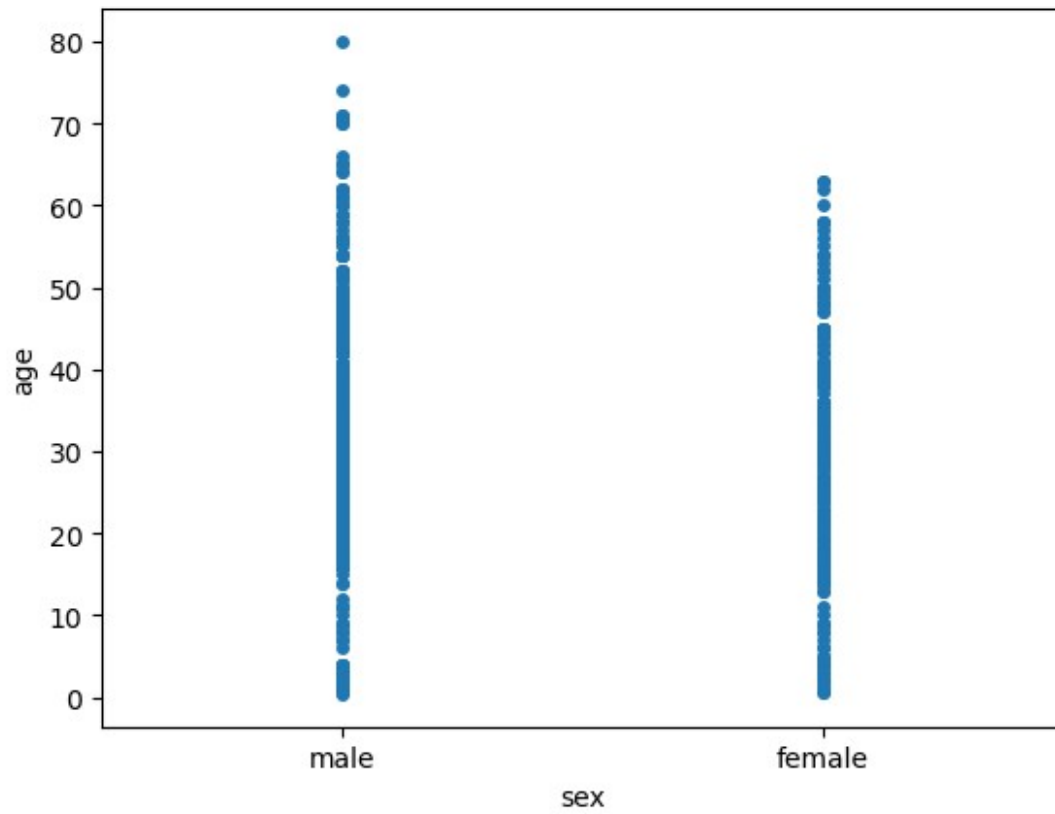
```
sns.violinplot(x='sex', y='age', data=dataset)  
<Axes: xlabel='sex', ylabel='age'>
```



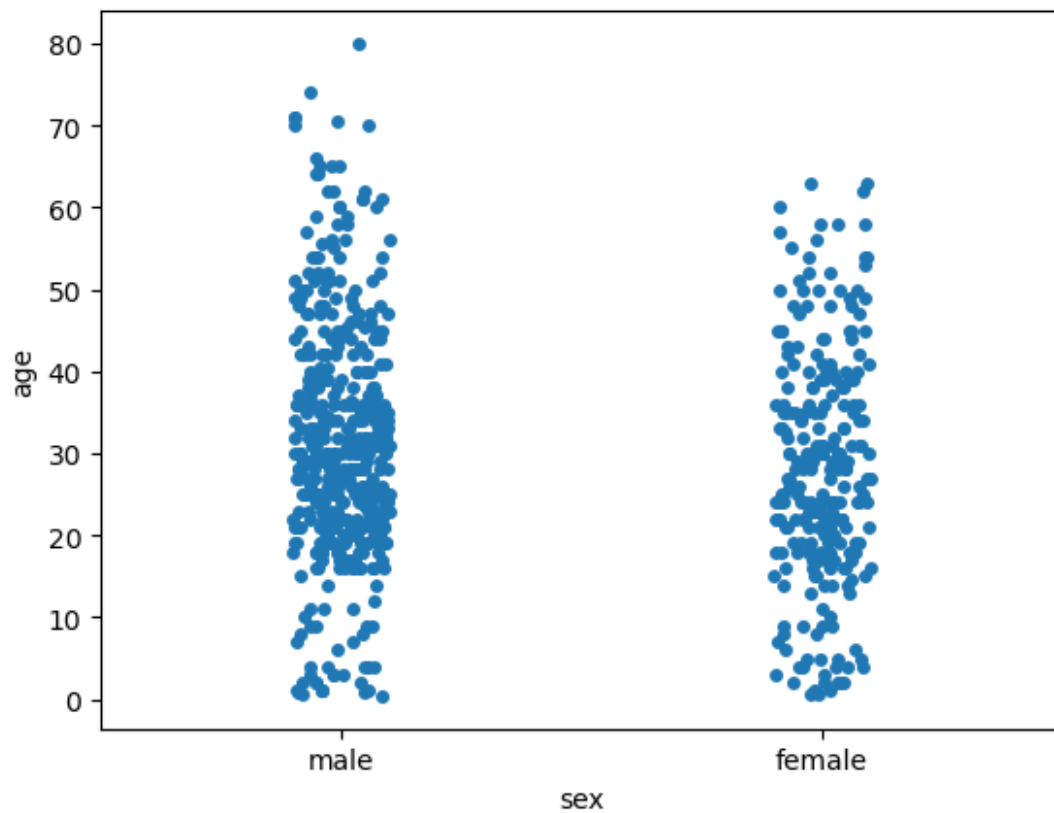
```
sns.violinplot(x='sex', y='age', data=dataset, hue='survived')  
<Axes: xlabel='sex', ylabel='age'>
```



```
sns.stripplot(x='sex', y='age', data=dataset, jitter=False)  
<Axes: xlabel='sex', ylabel='age'>
```

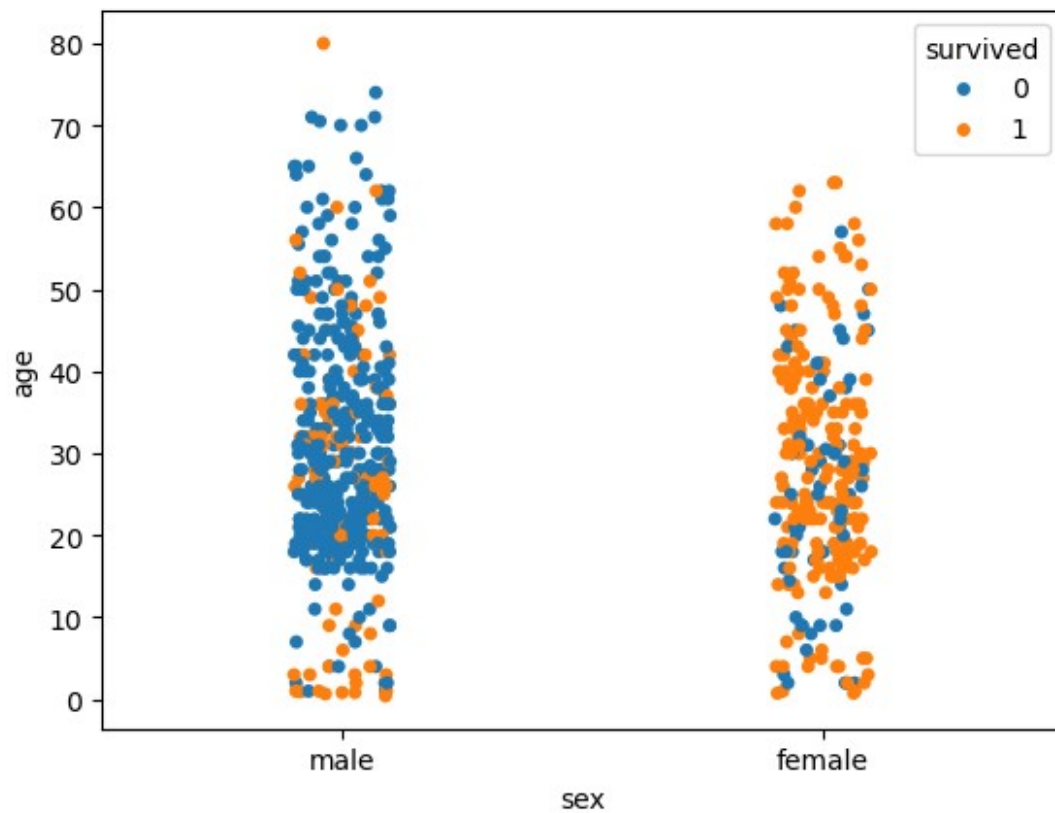


```
sns.stripplot(x='sex', y='age', data=dataset, jitter=True)  
<Axes: xlabel='sex', ylabel='age'>
```

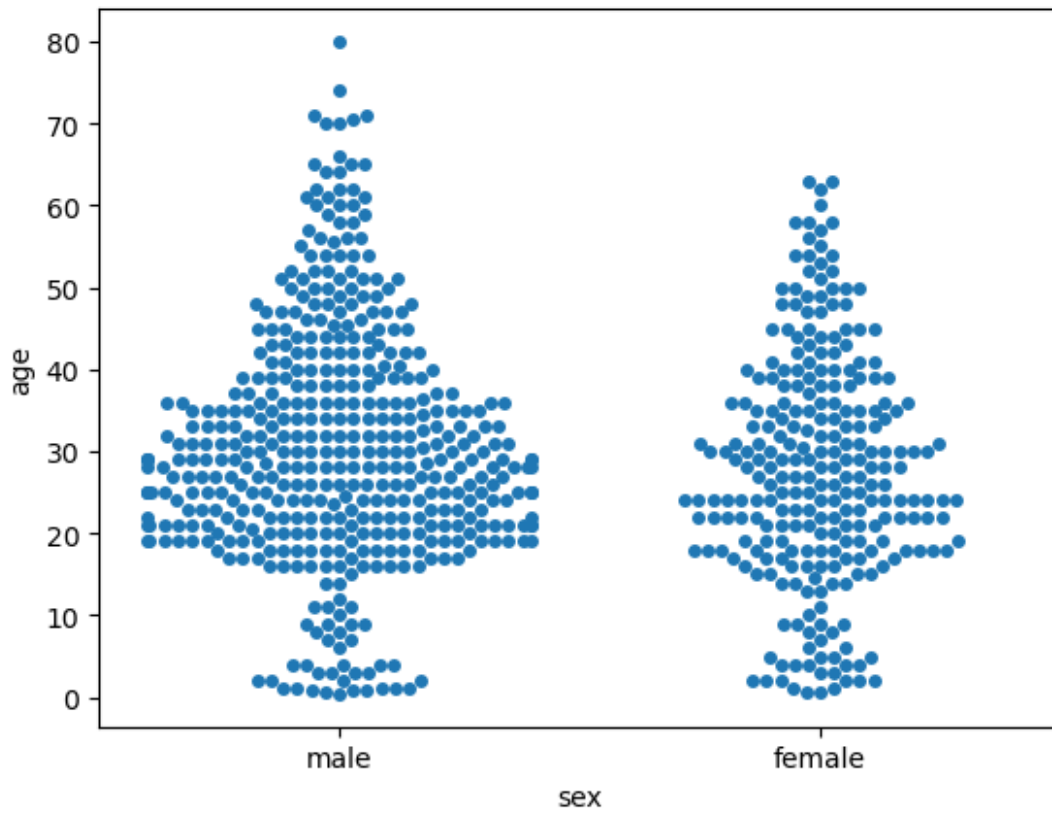


```
sns.stripplot(x='sex', y='age', data=dataset, jitter=True,  
hue='survived')
```

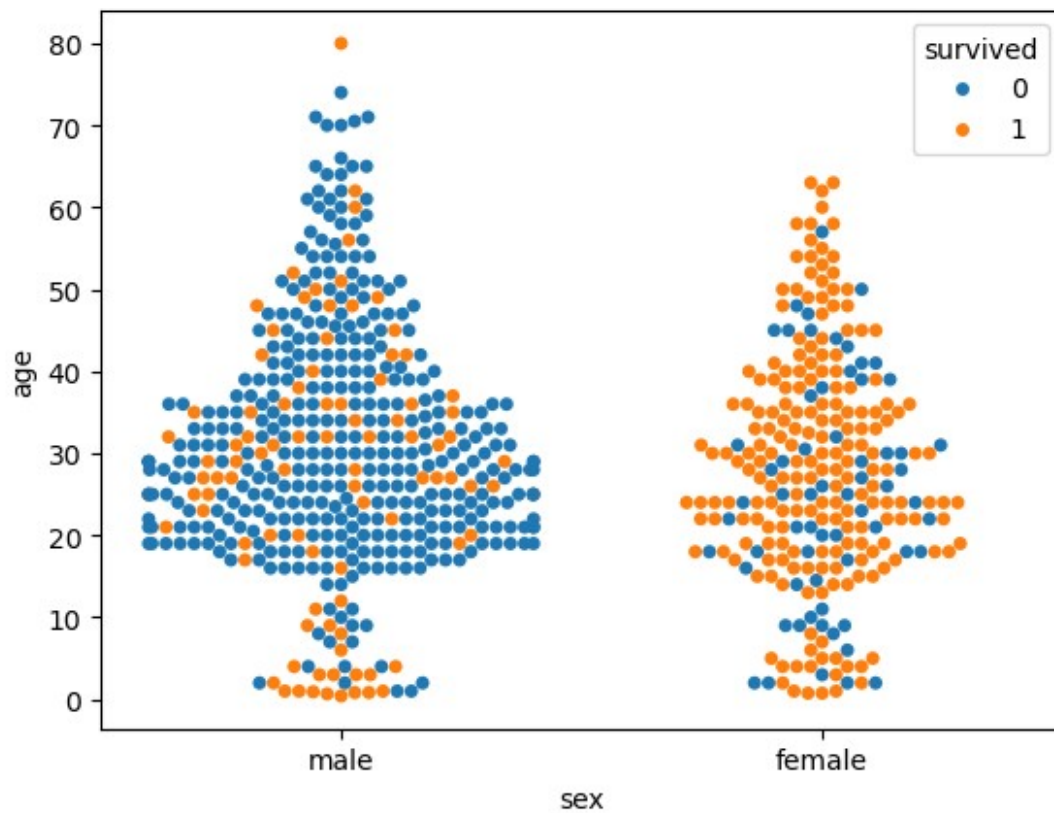
```
<Axes: xlabel='sex', ylabel='age'>
```



```
sns.swarmplot(x='sex', y='age', data=dataset)  
<Axes: xlabel='sex', ylabel='age'>
```



```
sns.swarmplot(x='sex', y='age', data=dataset, hue='survived')  
<Axes: xlabel='sex', ylabel='age'>
```

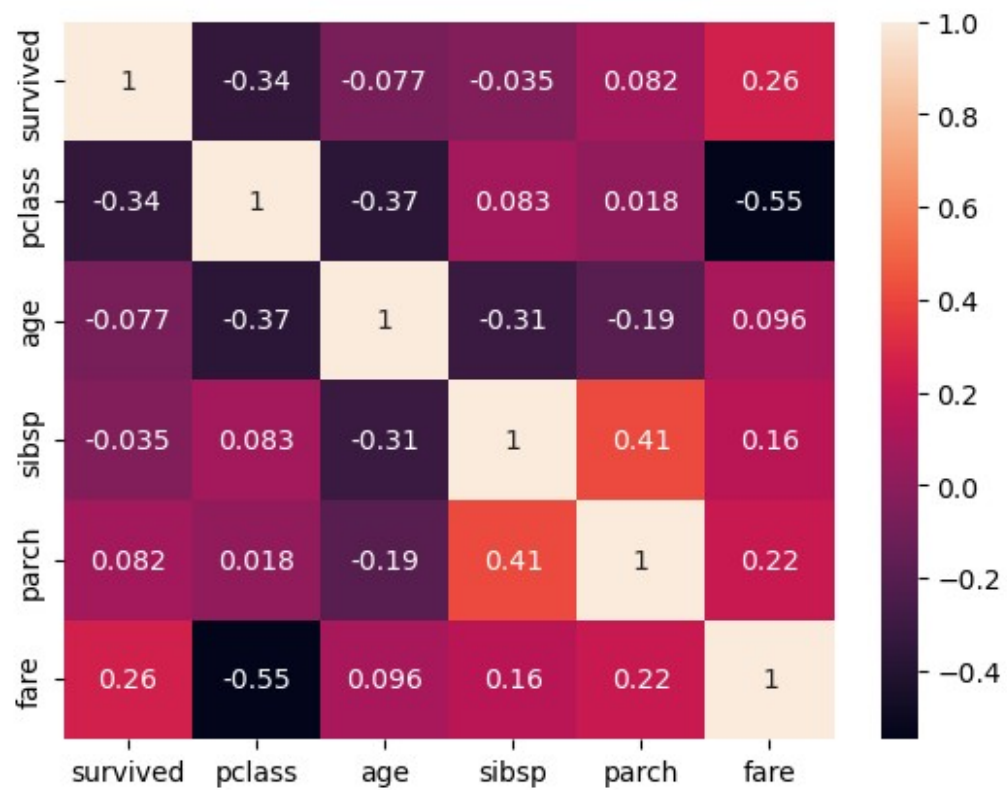



```
dataset = dataset.select_dtypes(include=['number'])
```

```
corr = dataset.corr()
```

```
sns.heatmap(corr, annot=True)
```

```
<Axes: >
```



```
sns.heatmap(corr)
```

```
<Axes: >
```

