

Name: Pranav Mehendale

Roll No.: TCOD34

Batch: T11

Assignment 8 Group A

Data Visualizaiton 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 sns
import numpy as np
```

```
%matplotlib inline
```

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

```
dataset.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S

	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

```
dataset.info()
```

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

```
RangeIndex: 891 entries, 0 to 890
```

```
Data columns (total 15 columns):
```

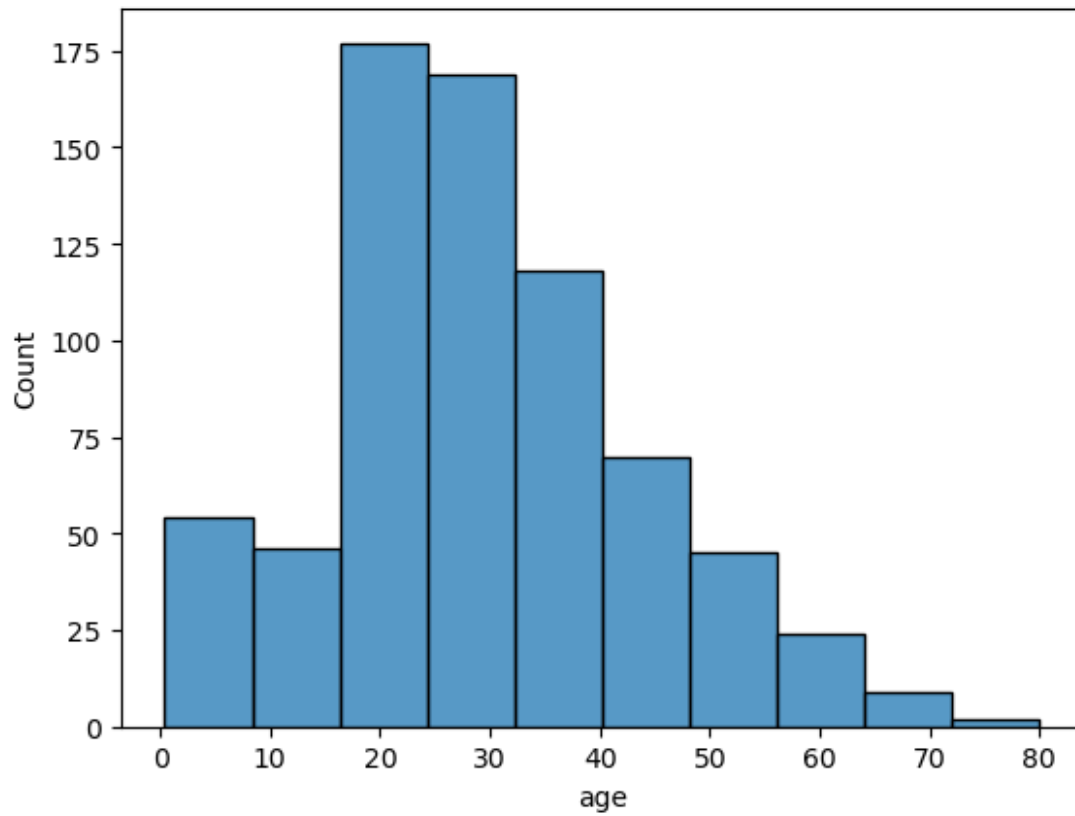
#	Column	Non-Null Count	Dtype
0	survived	891 non-null	int64
1	pclass	891 non-null	int64
2	sex	891 non-null	object
3	age	714 non-null	float64
4	sibsp	891 non-null	int64
5	parch	891 non-null	int64
6	fare	891 non-null	float64
7	embarked	889 non-null	object
8	class	891 non-null	category
9	who	891 non-null	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)
```

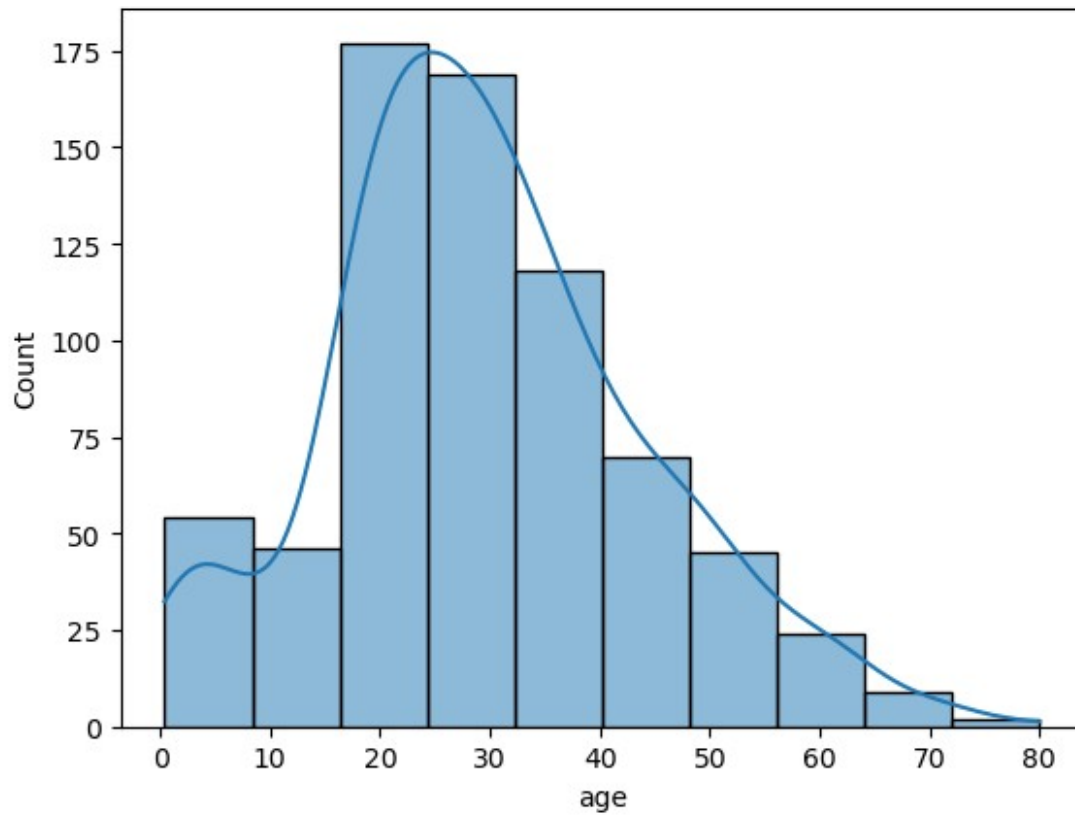
```
memory usage: 80.7+ KB
```

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

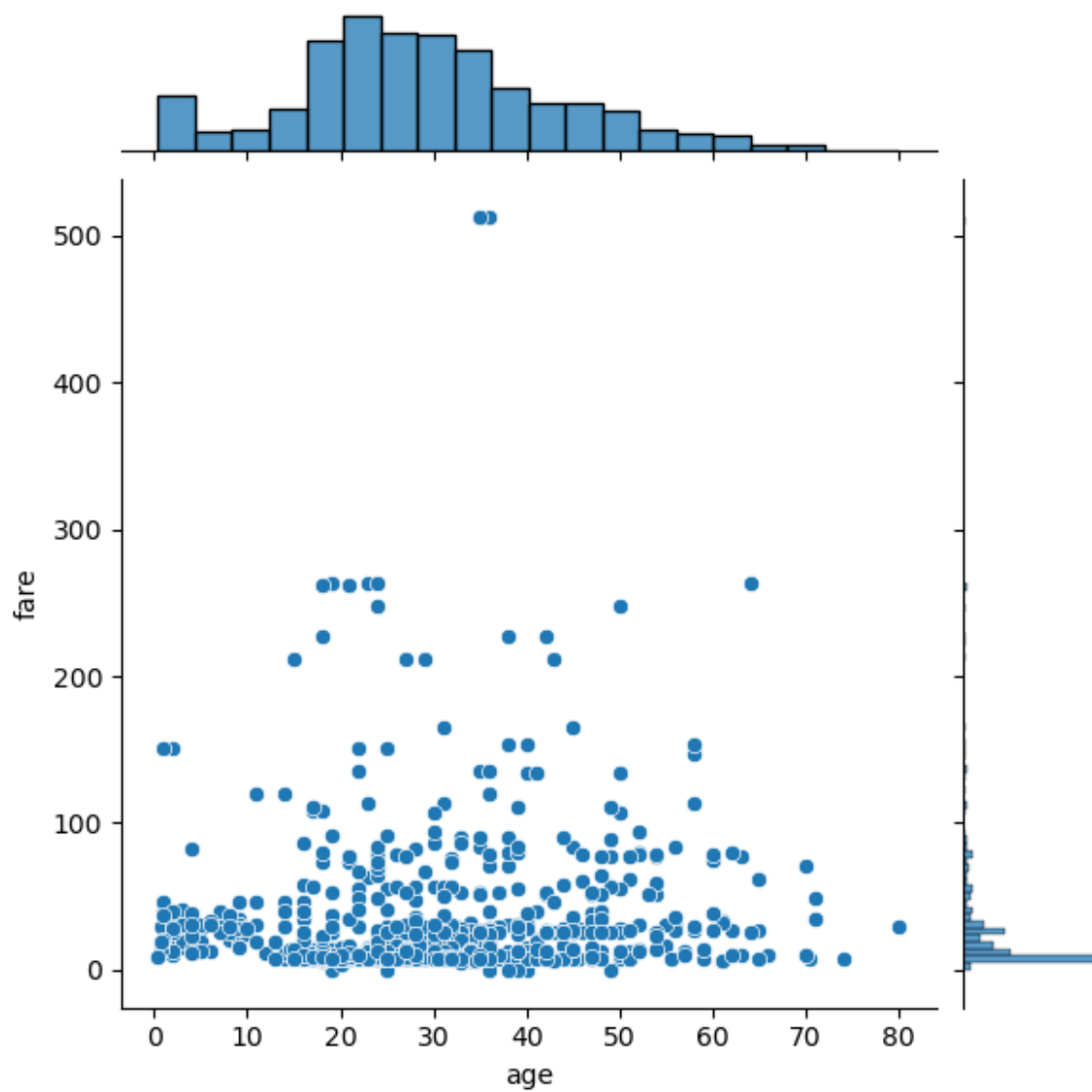
```
<Axes: xlabel='age', ylabel='Count'>
```

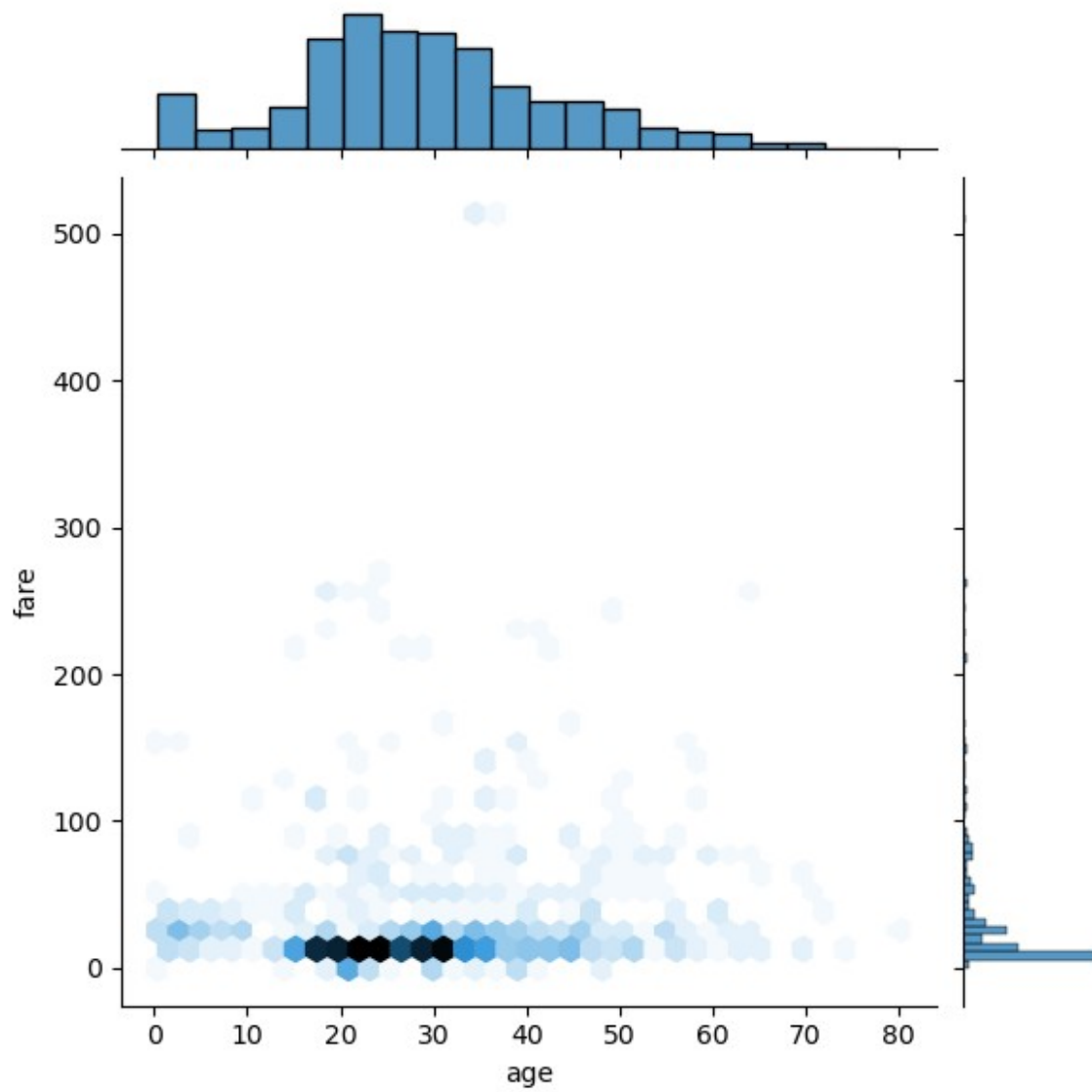


```
sns.histplot(dataset['age'], bins = 10, kde=True)  
<Axes: xlabel='age', ylabel='Count'>
```



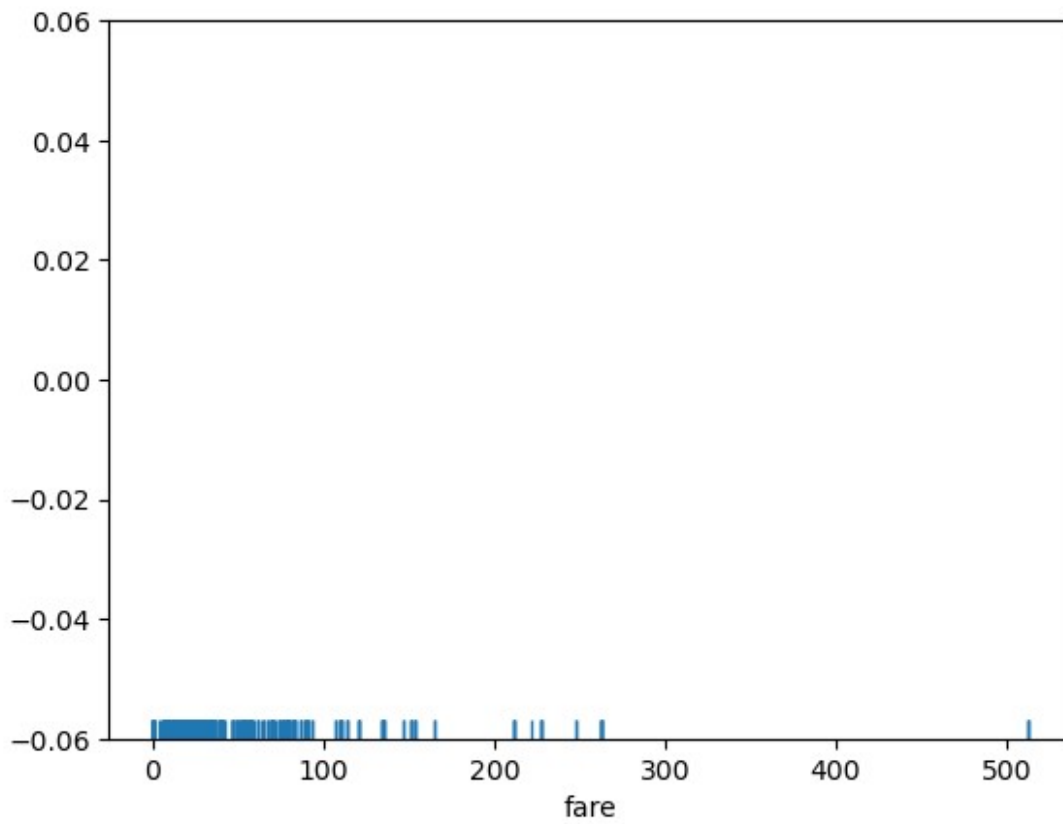
```
sns.jointplot(x=dataset['age'], y=dataset['fare'], kind='scatter')
sns.jointplot(x=dataset['age'], y=dataset['fare'], kind='hex')
<seaborn.axisgrid.JointGrid at 0x1d862a6ca50>
```



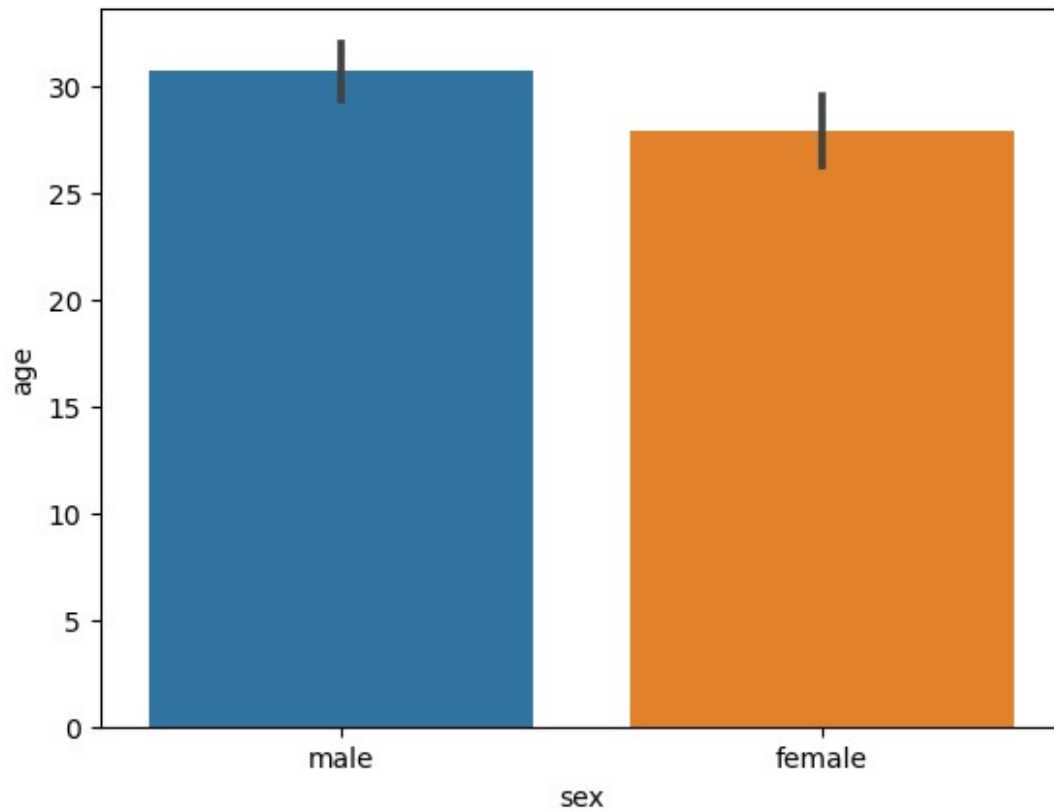


```
sns.rugplot(dataset['fare'])
```

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



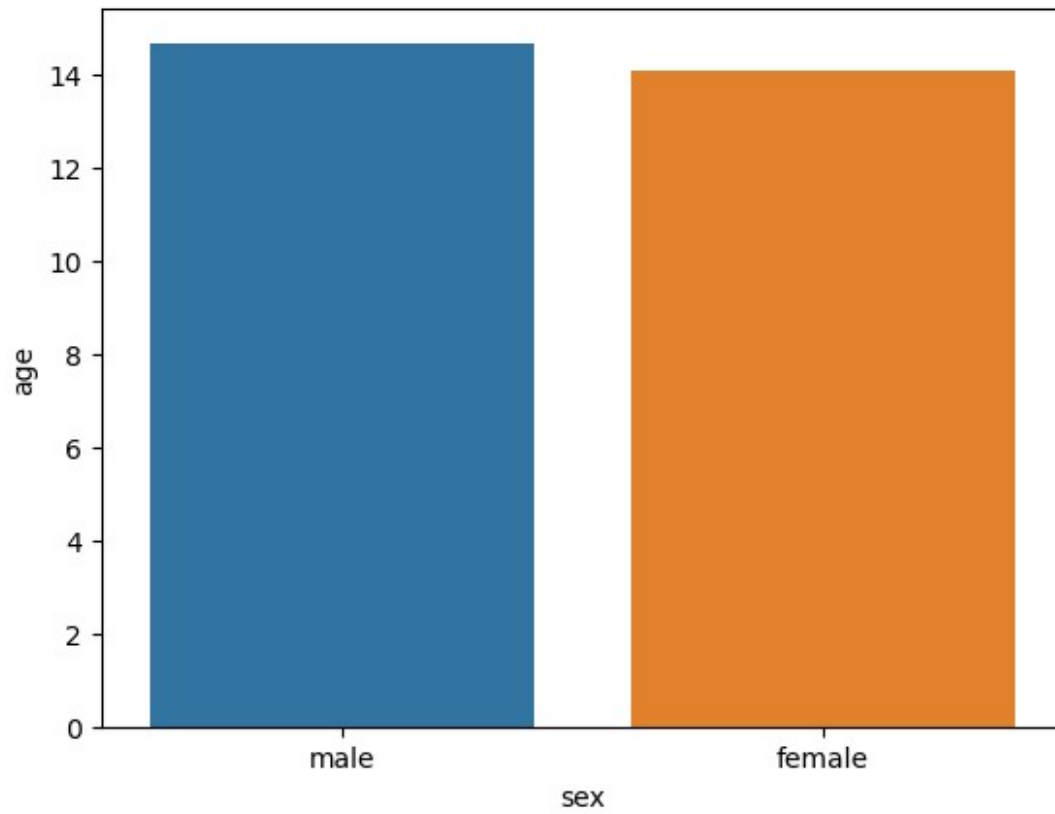
```
sns.barplot(x='sex', y='age', data=dataset)  
<Axes: xlabel='sex', ylabel='age'>
```



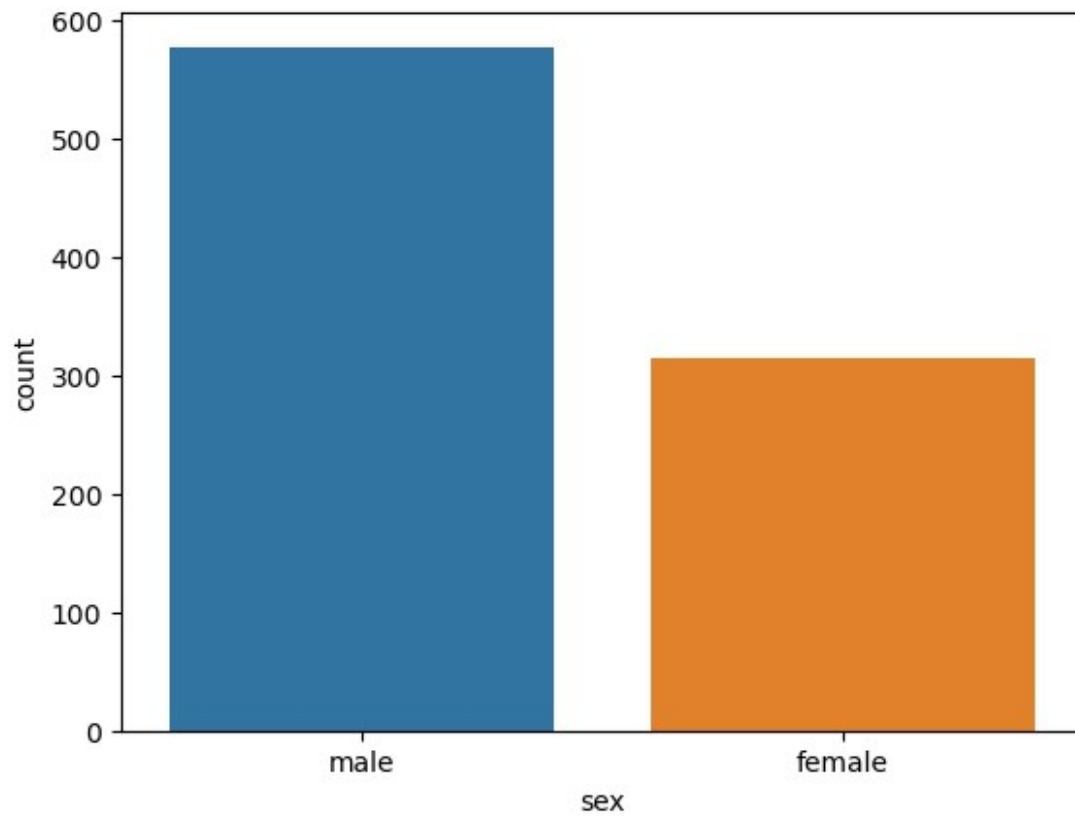
```
sns.barplot(x='sex', y='age', data=dataset, estimator=np.std)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\numpy\lib\nanfunctions.py:1556: RuntimeWarning: All-NaN slice encountered  
    return function_base._ureduce(a,
```

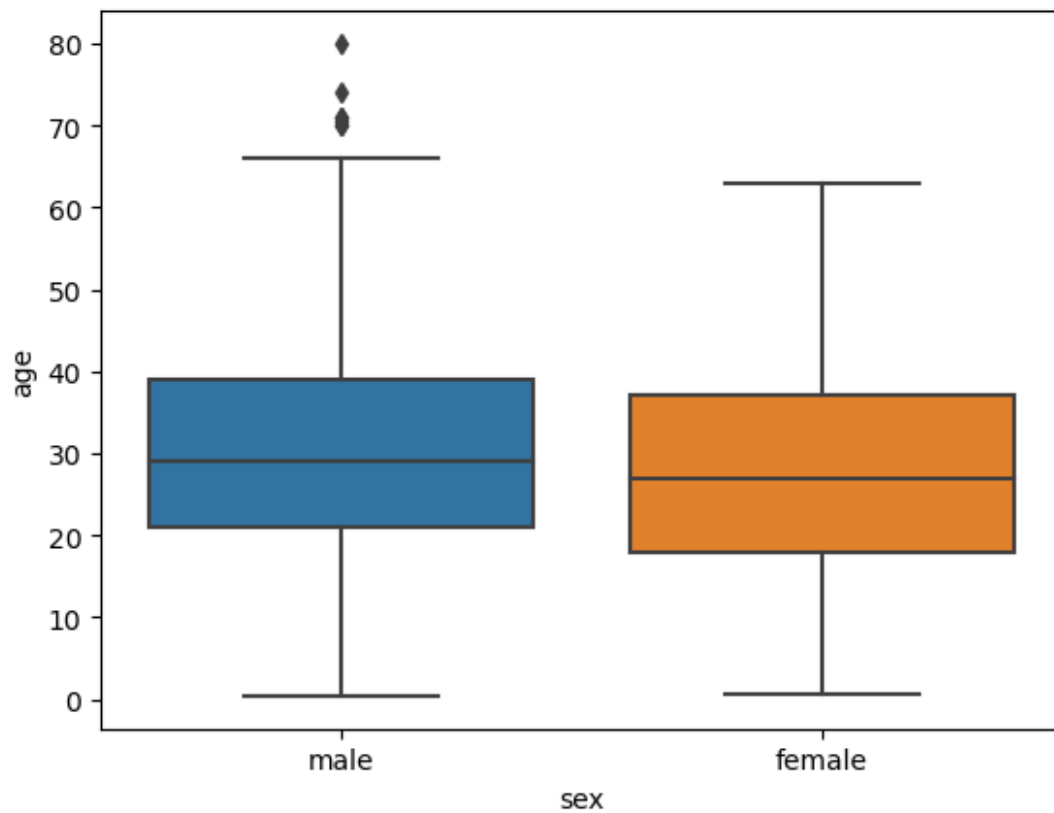
```
<Axes: xlabel='sex', ylabel='age'>
```

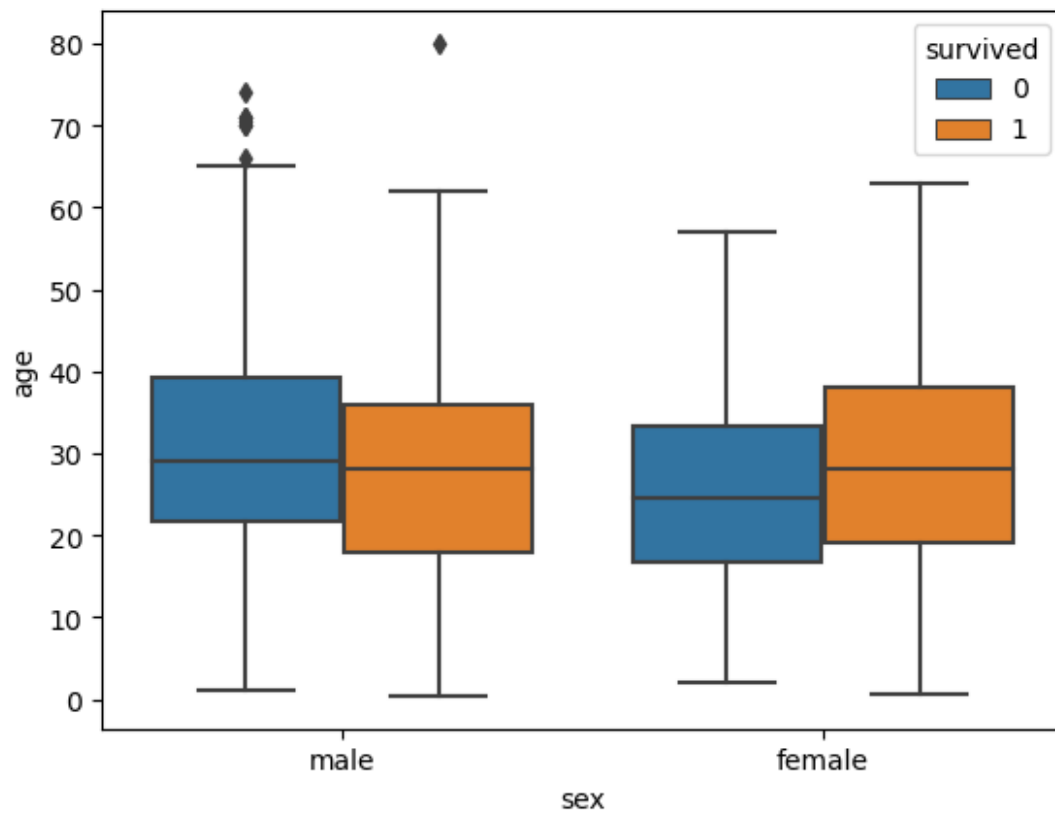
```
sns.countplot(x='sex', data=dataset)  
<Axes: xlabel='sex', ylabel='count'>
```



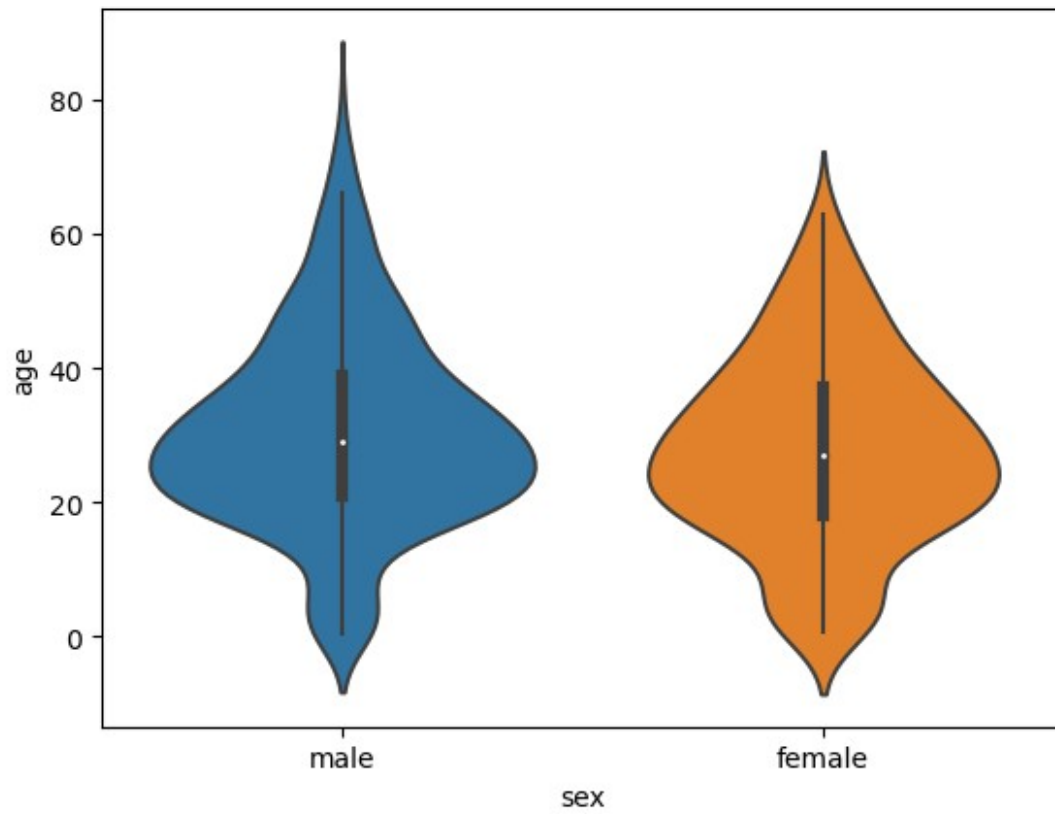
```
sns.boxplot(x='sex', y='age', data=dataset)  
<Axes: xlabel='sex', ylabel='age'>
```



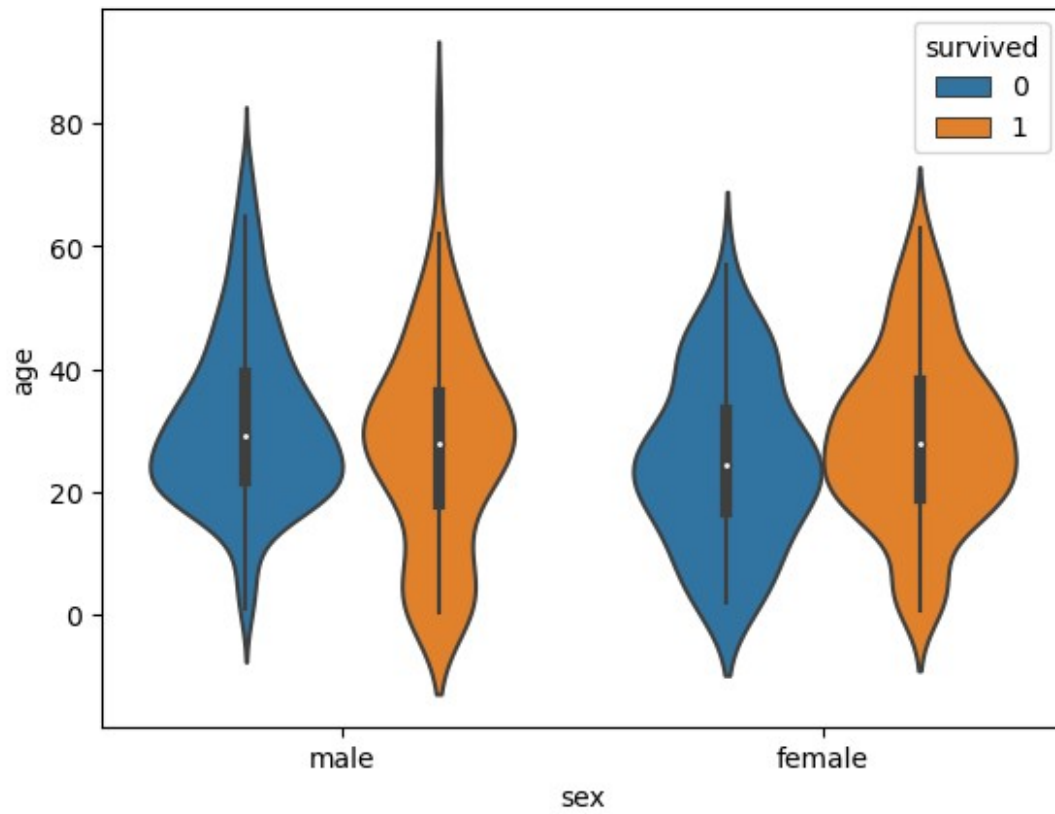
```
sns.boxplot(x='sex', y='age', data=dataset, hue="survived")  
<Axes: xlabel='sex', ylabel='age'>
```



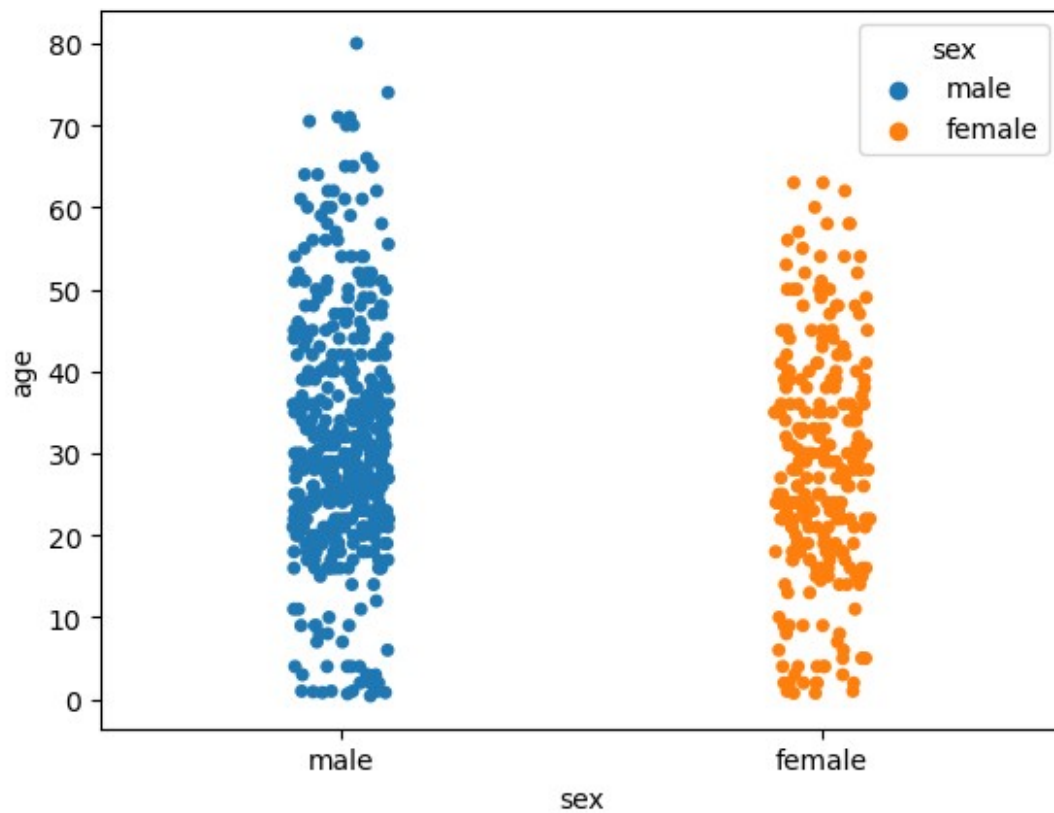
```
sns.violinplot(x='sex', y='age', data=dataset)  
<Axes: xlabel='sex', ylabel='age'>
```



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

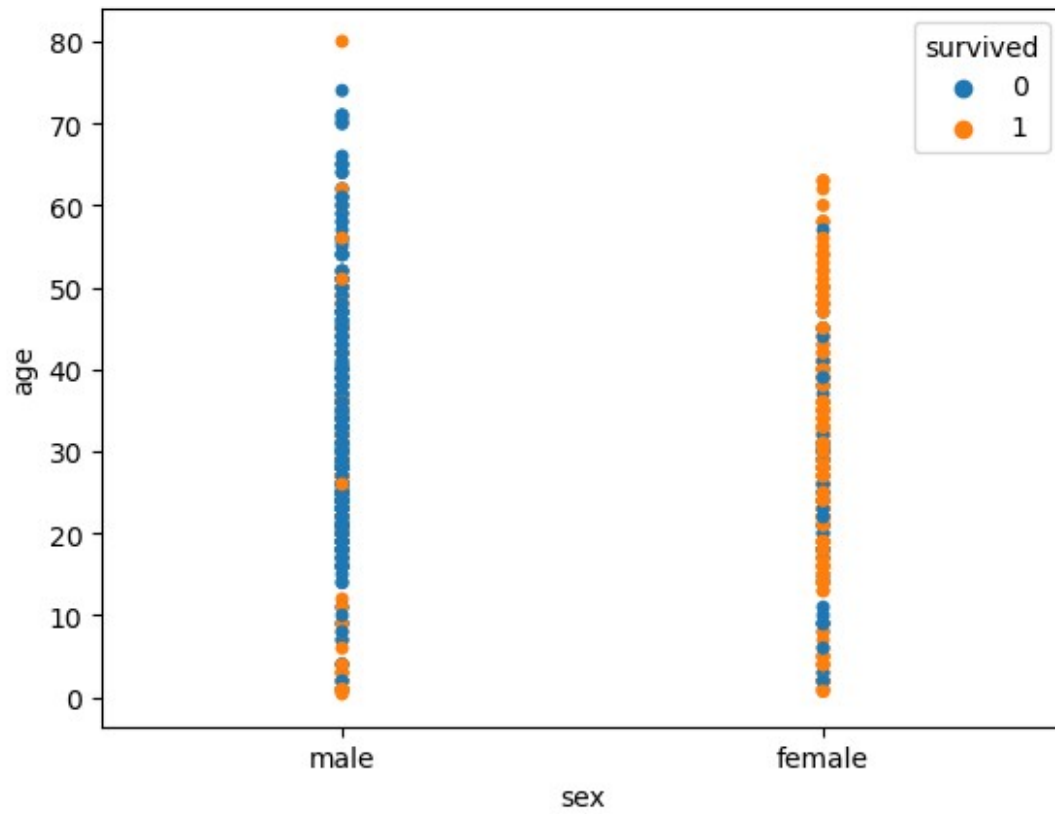


```
sns.stripplot(x='sex', y='age', data=dataset, jitter=True, hue='sex')  
<Axes: xlabel='sex', ylabel='age'>
```



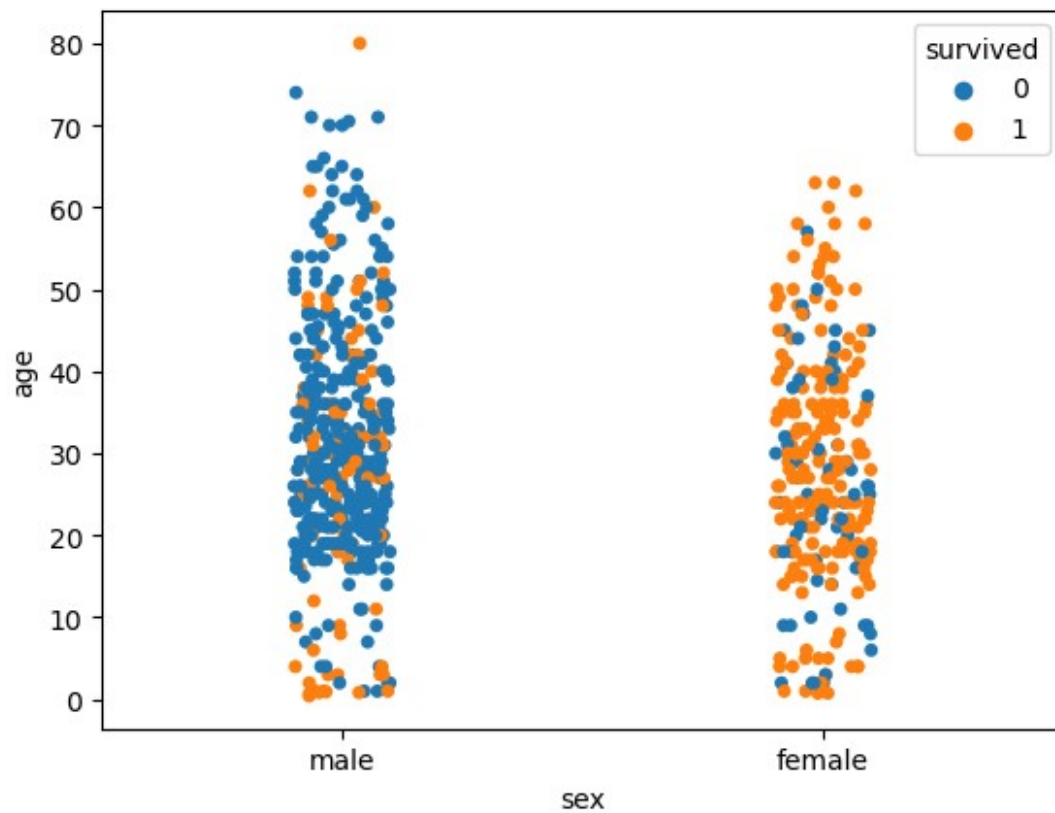
```
sns.stripplot(x='sex', y='age', data=dataset, jitter=False,  
hue='survived')
```

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

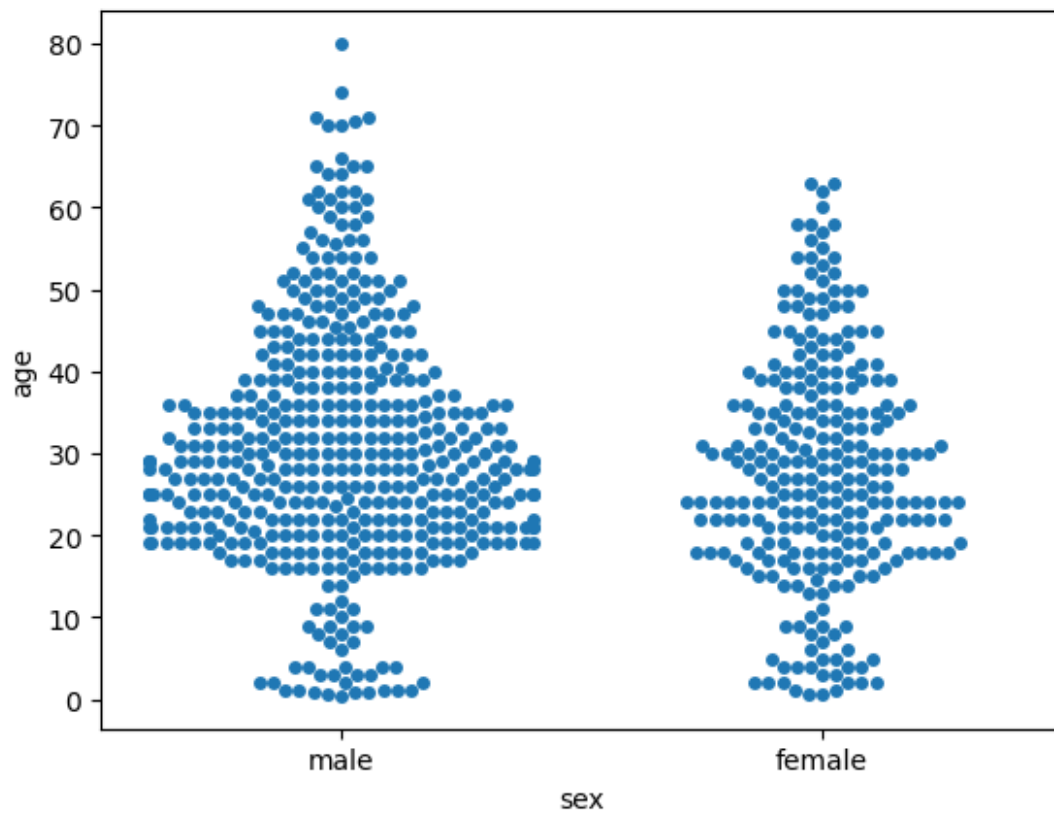


```
sns.stripplot(x='sex', y='age', data=dataset, jitter=True,  
hue='survived')
```

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

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



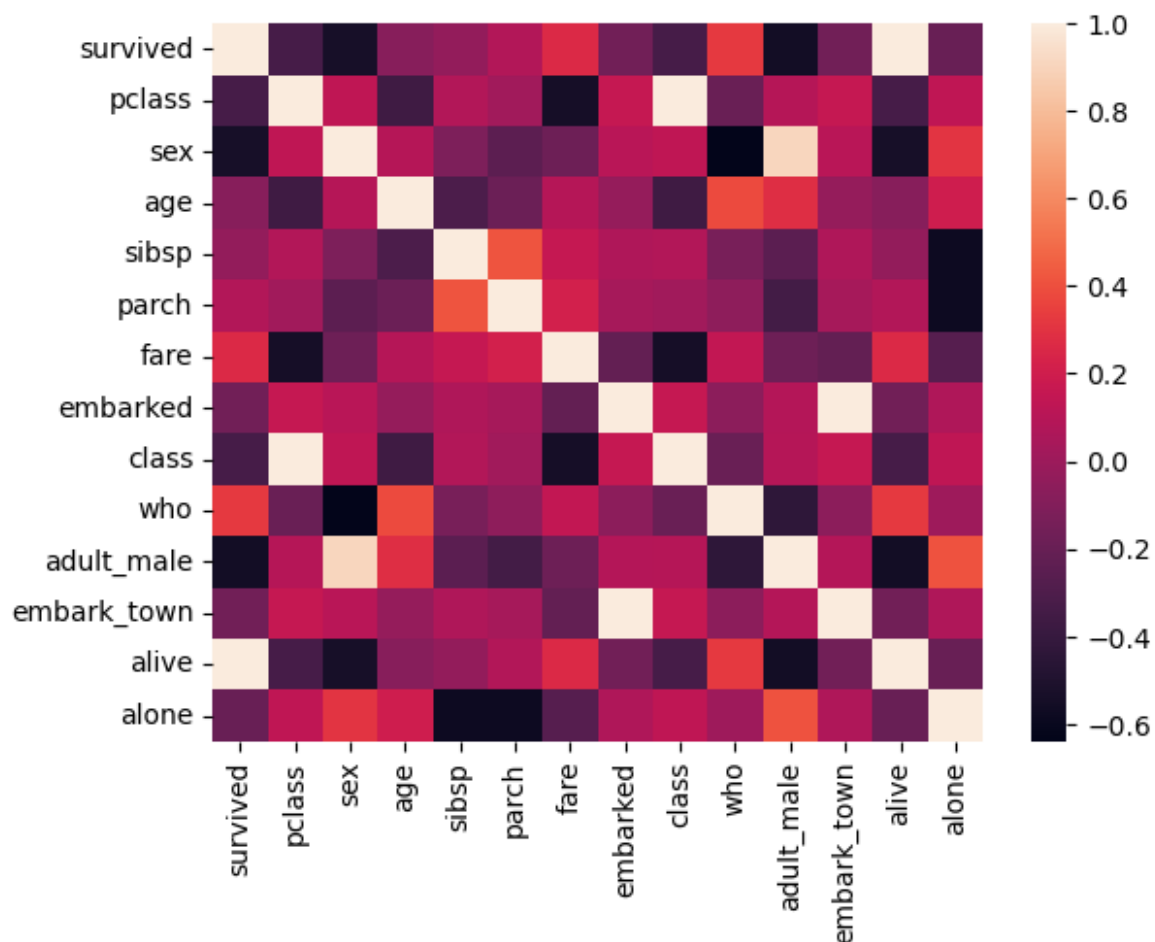
```
sns.swarmplot(x='sex', y='age', data=dataset, hue='survived')  
<Axes: xlabel='sex', ylabel='age'>
```


	adult_male	deck	embark_town	alive	alone
0	True	NaN	2	0	0
1	False	C	0	1	0
2	False	NaN	2	1	1
3	False	C	2	1	0
4	True	NaN	2	0	1

```
dataset.drop('deck', axis=1, inplace=True)
```

```
corr = dataset.corr()
sns.heatmap(corr)
```

<Axes: >



```
sns.heatmap(corr,annot=True, cmap='crest_r')
```

<Axes: >

