practical-8

March 10, 2024

[ ]:
```
import pandas as pd
import numpy as np
import seaborn as sn
import matplotlib.pyplot as plt
```

[ ]:
```
titanic = sn.load_dataset('titanic')
sn.load_dataset('titanic')
```

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

       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.isnull().sum()
```

[ ]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

[ ]:
```
dummy_column=pd.get_dummies(titanic.deck)
```

[ ]:
```
new_df=titanic.join(dummy_column).drop(['deck'],axis=1)
```

[ ]:
```
new_df
```

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

       who  …  embark_town alive  alone  A  B  C  D  E  F  G
0      man  …  Southampton    no  False  0  0  0  0  0  0  0
1    woman  …    Cherbourg   yes  False  0  0  1  0  0  0  0
2    woman  …  Southampton   yes   True  0  0  0  0  0  0  0
3    woman  …  Southampton   yes  False  0  0  1  0  0  0  0
4      man  …  Southampton    no   True  0  0  0  0  0  0  0
..     …  …          …   …    … .. .. .. .. .. .. ..
886    man  …  Southampton    no   True  0  0  0  0  0  0  0
887  woman  …  Southampton   yes   True  0  1  0  0  0  0  0
888  woman  …  Southampton    no  False  0  0  0  0  0  0  0
889    man  …    Cherbourg   yes   True  0  0  1  0  0  0  0
```

```
890    man  …   Queenstown    no   True  0  0  0  0  0  0  0

[891 rows x 21 columns]
```

[ ]:
```
titanic['age'].dropna()
```

[ ]:
```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       …
885    39.0
886    27.0
887    19.0
889    26.0
890    32.0
Name: age, Length: 714, dtype: float64
```

[ ]:
```
titanic['embarked'].dropna()
```

[ ]:
```
0      S
1      C
2      S
3      S
4      S
      ..
886    S
887    S
888    S
889    C
890    Q
Name: embarked, Length: 889, dtype: object
```

[ ]:
```
titanic['embark_town'].dropna()
```

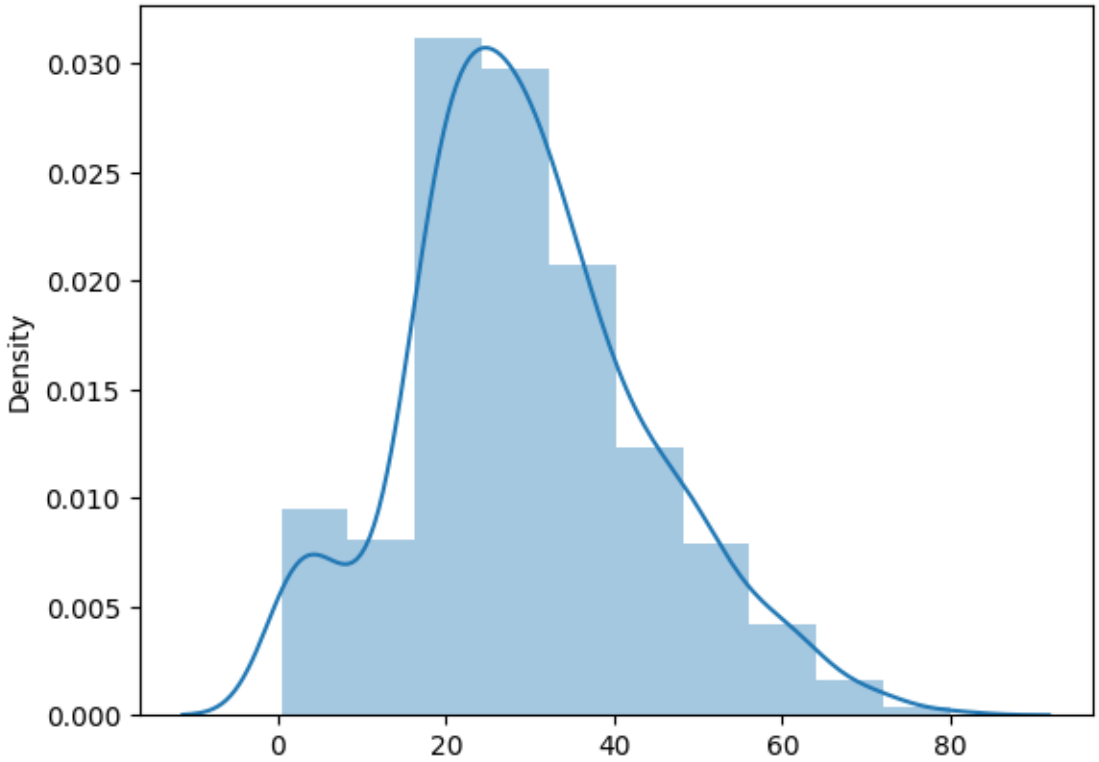
[ ]:
```
0      Southampton
1        Cherbourg
2      Southampton
3      Southampton
4      Southampton
          …
886    Southampton
887    Southampton
888    Southampton
889      Cherbourg
890     Queenstown
```

```
Name: embark_town, Length: 889, dtype: object
```

[ ]:
```
sn.distplot(x = titanic['age'], bins = 10)
```

```
<ipython-input-22-7e8a7e9f4975>: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

  sn.distplot(x = titanic['age'], bins = 10)
```

[ ]:
```
<Axes: ylabel='Density'>
```

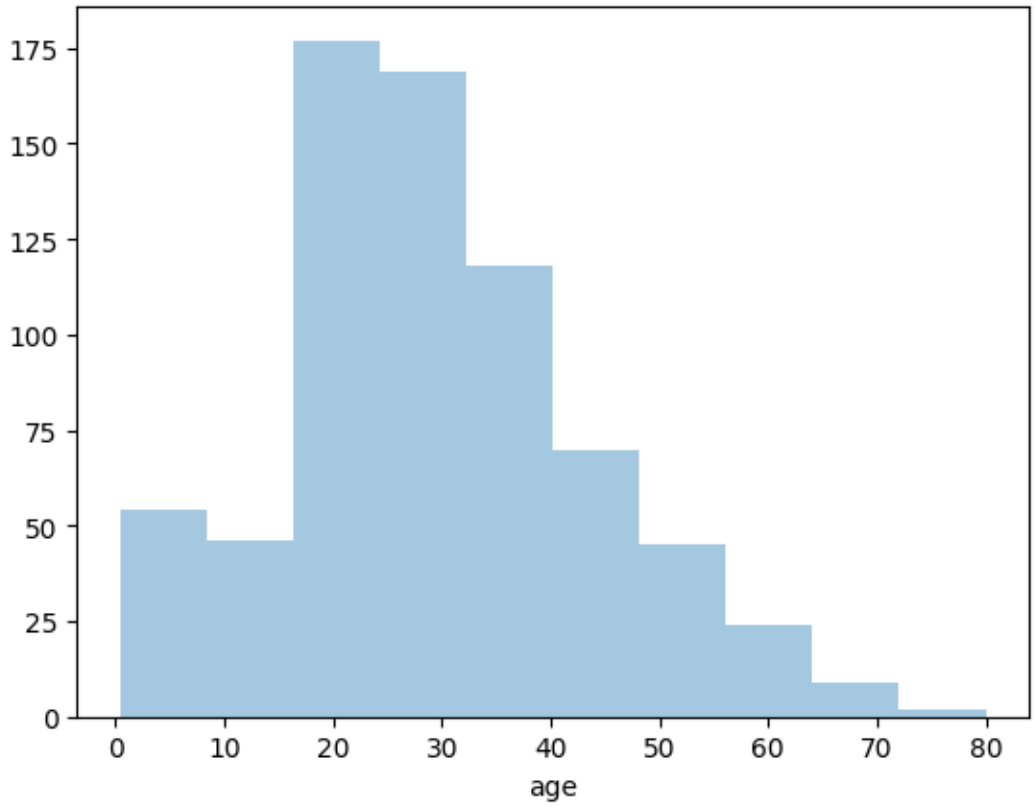


[ ]:
```
sn.distplot(titanic['age'], bins = 10,kde=False)
```

```
<ipython-input-23-8a2d2ece3bbf>:1: UserWarning:

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

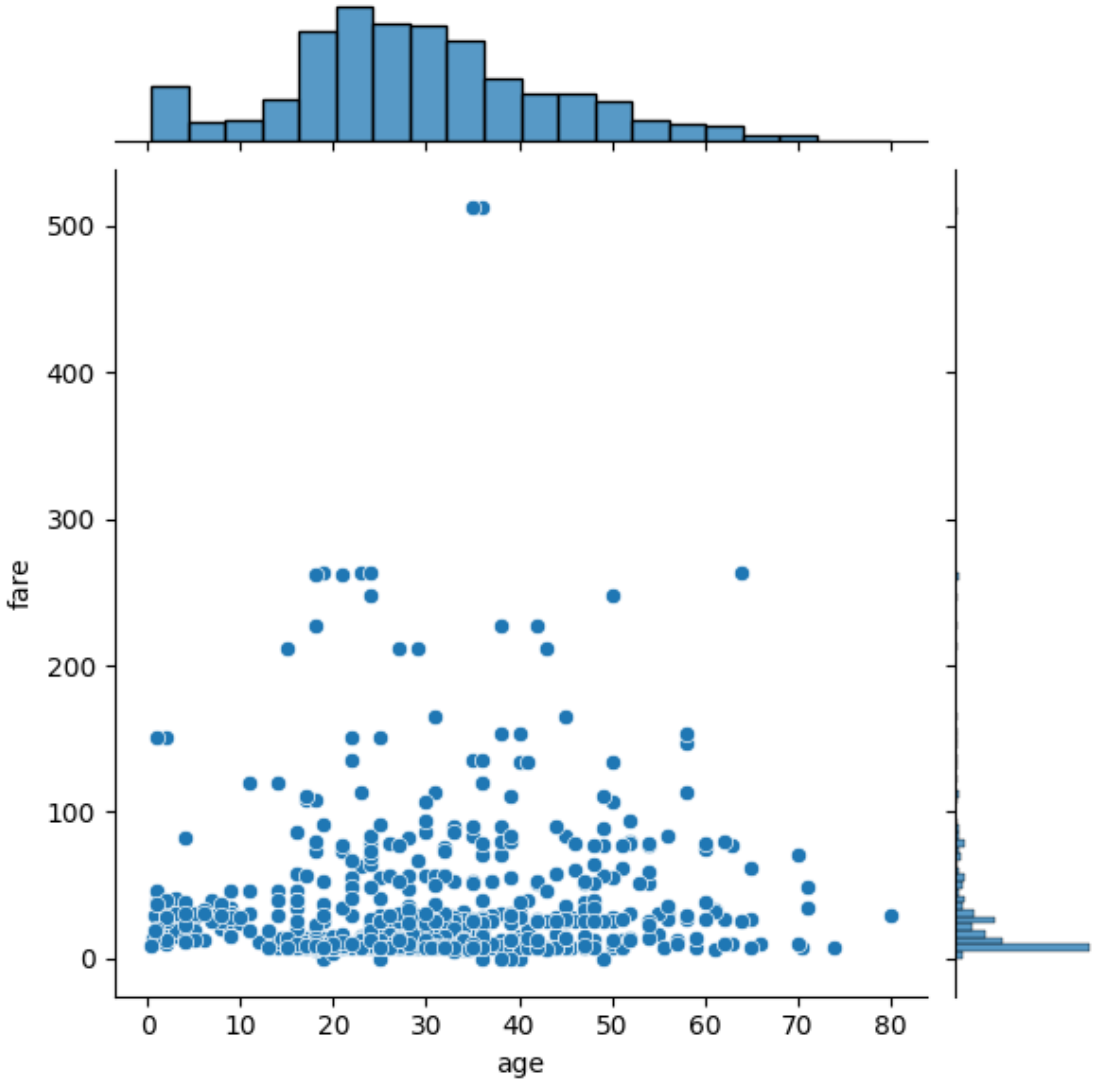
```
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

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

  sn.distplot(titanic['age'], bins = 10,kde=False)
```

[ ]:
```
<Axes: xlabel='age'>
```



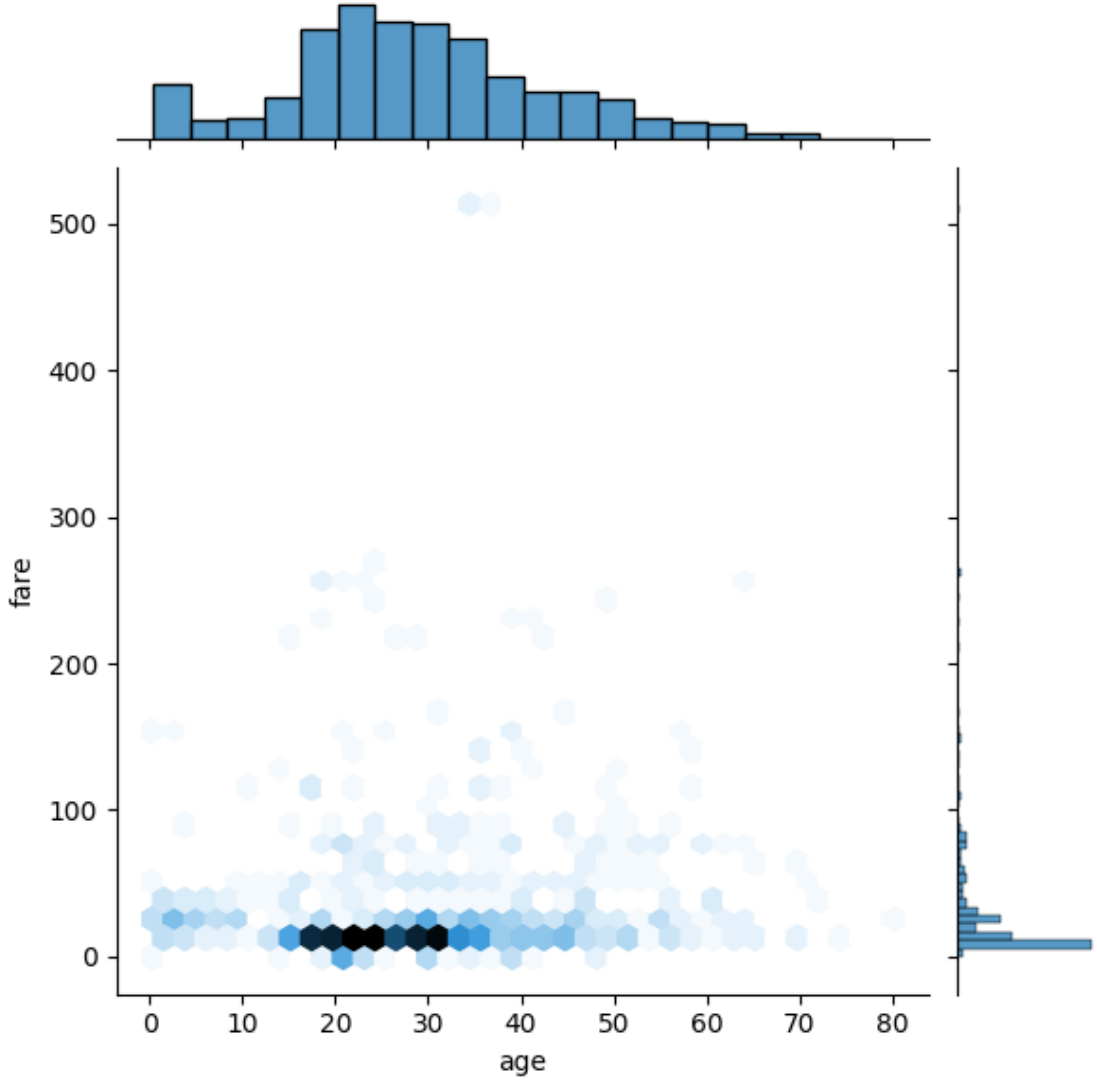
[ ]:
```
sn.jointplot(x = titanic['age'], y = titanic['fare'], kind = 'scatter')
```

[ ]:
```
<seaborn.axisgrid.JointGrid at 0x7dc5826bfbe0>
```



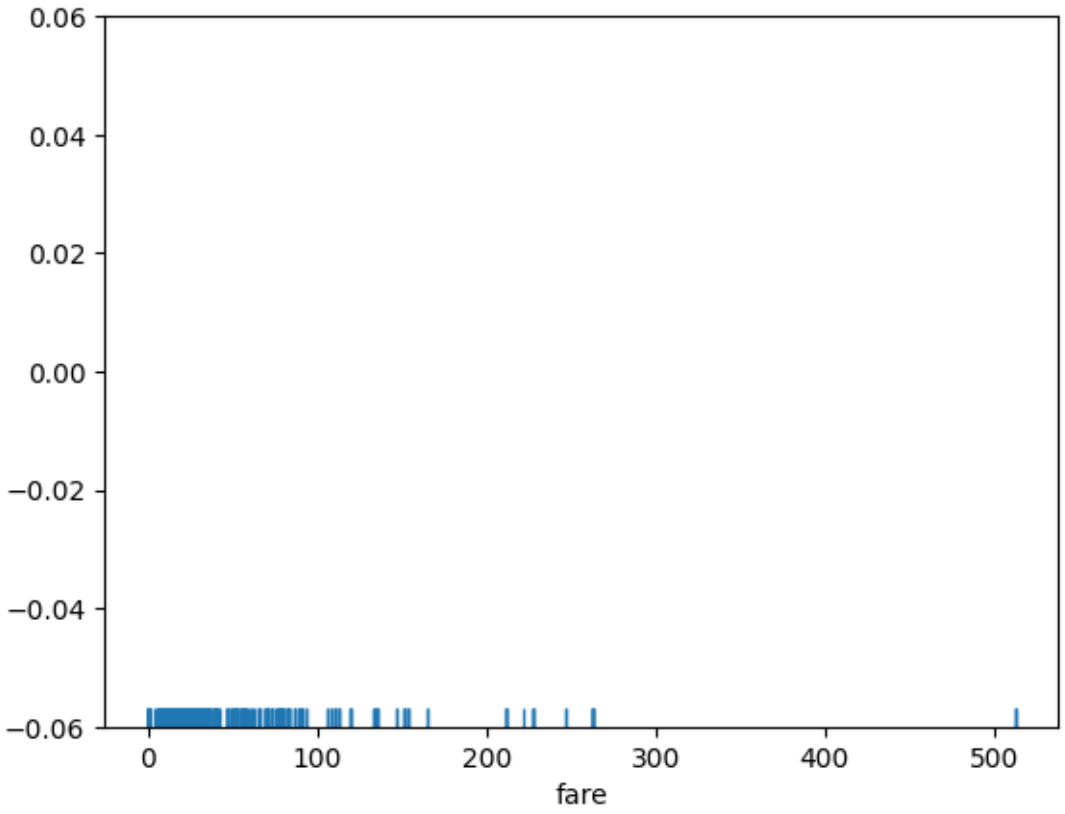
[ ]:
```
sn.jointplot(x = titanic['age'], y = titanic['fare'], kind = 'hex')
```

[ ]: <seaborn.axisgrid.JointGrid at 0x7dc5802fd660>



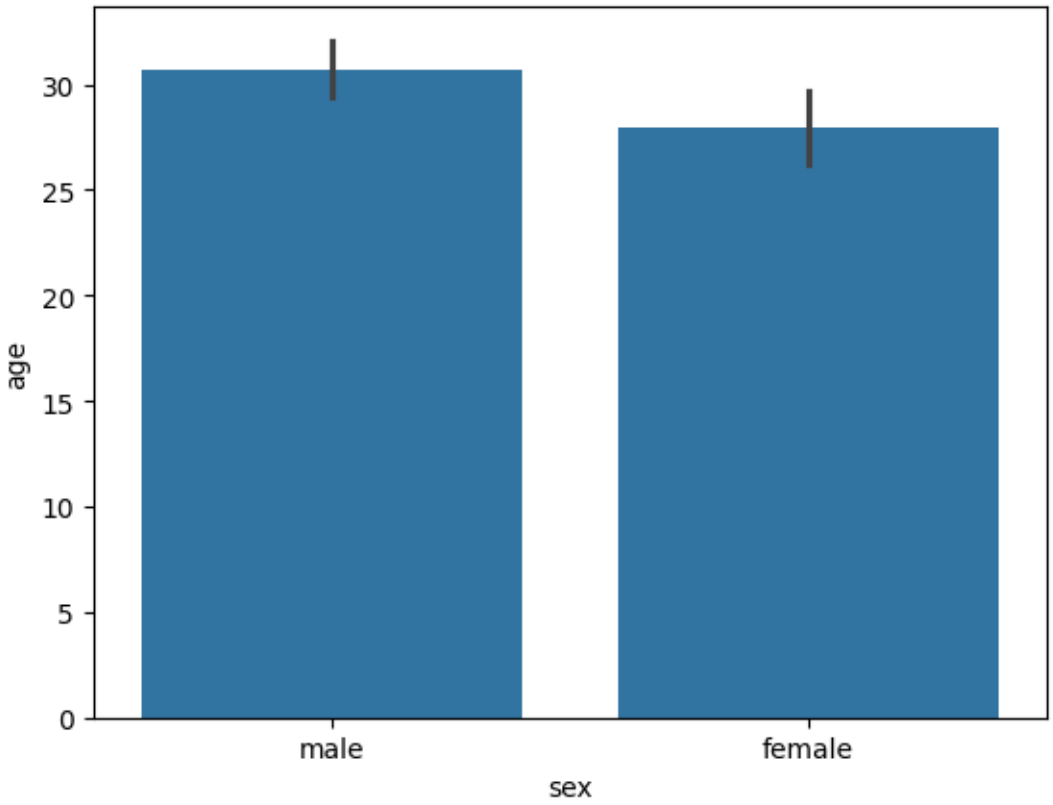
[ ]:
```
sn.rugplot(titanic['fare'])
```

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



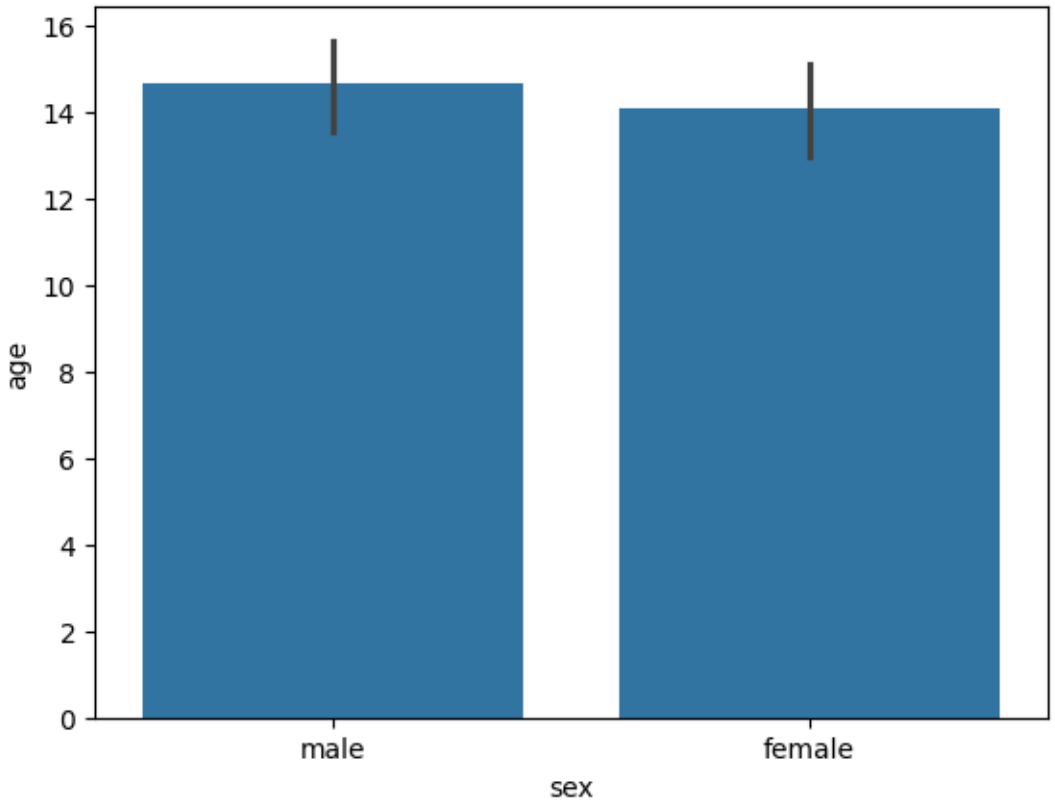
[ ]:
```
sn.barplot(x="sex", y="age", data=titanic)
```

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



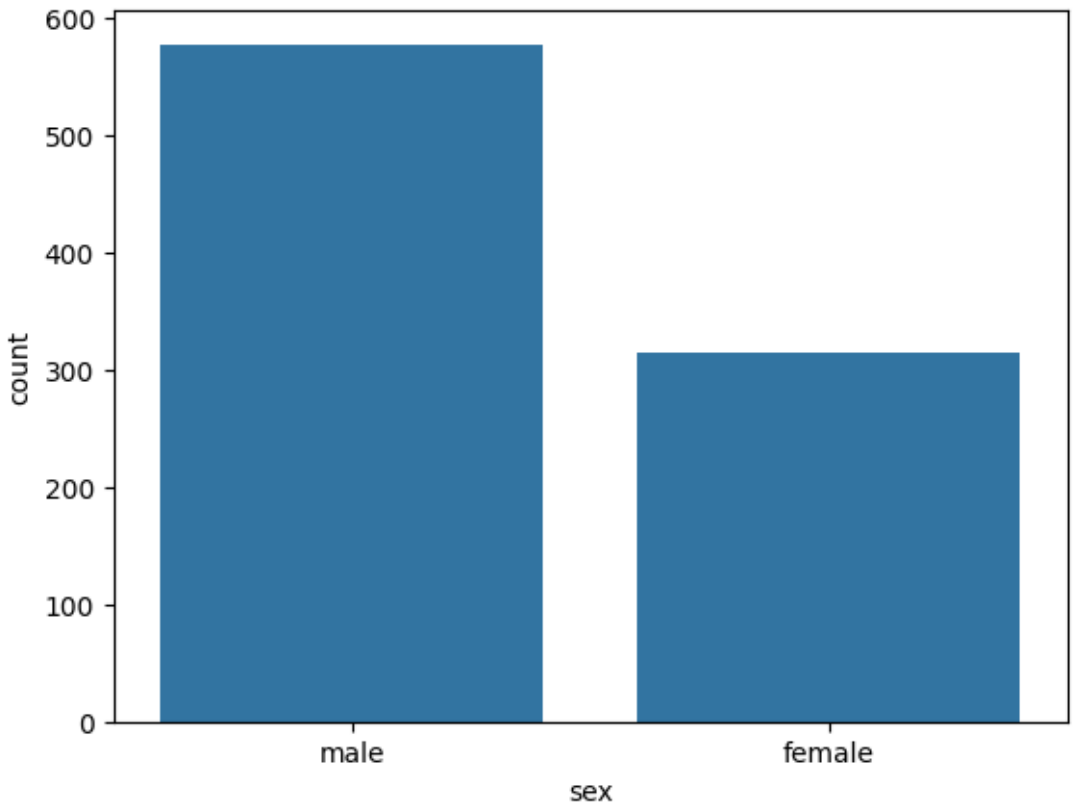
[ ]:
```
sn.barplot(x='sex', y='age', data=titanic, estimator=np.std)
```

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



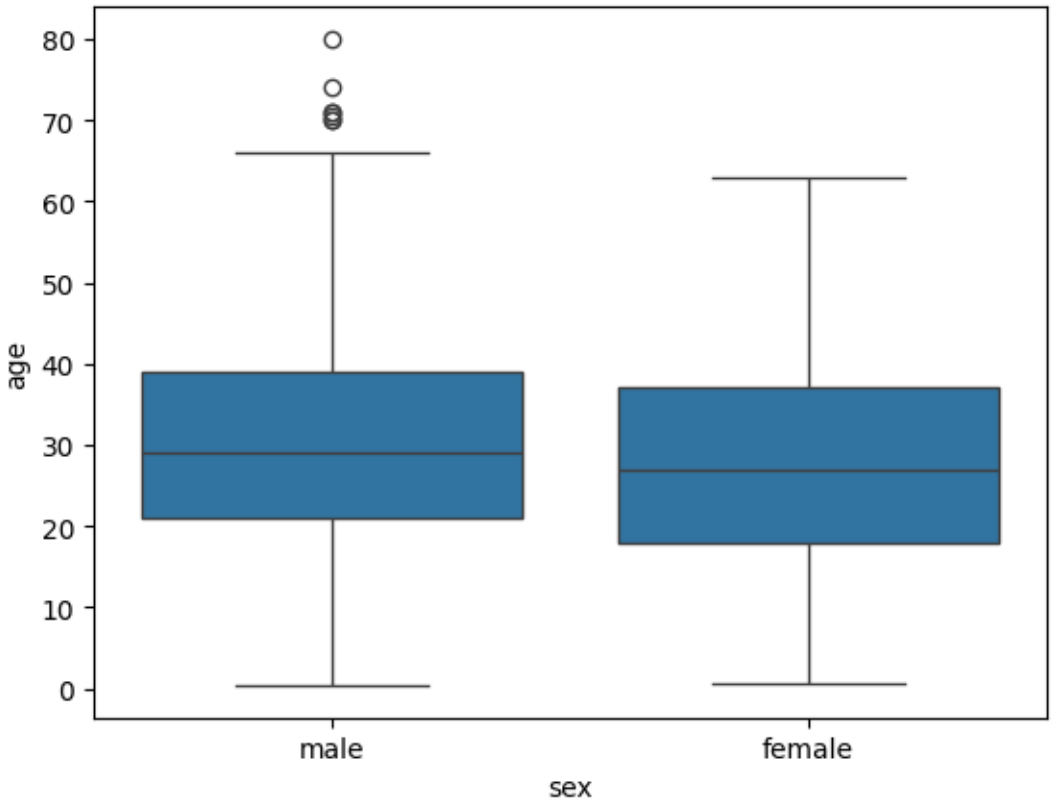
[ ]:
```
sn.countplot(x='sex', data=titanic)
```

[ ]:
```
<Axes: xlabel='sex', ylabel='count'>
```



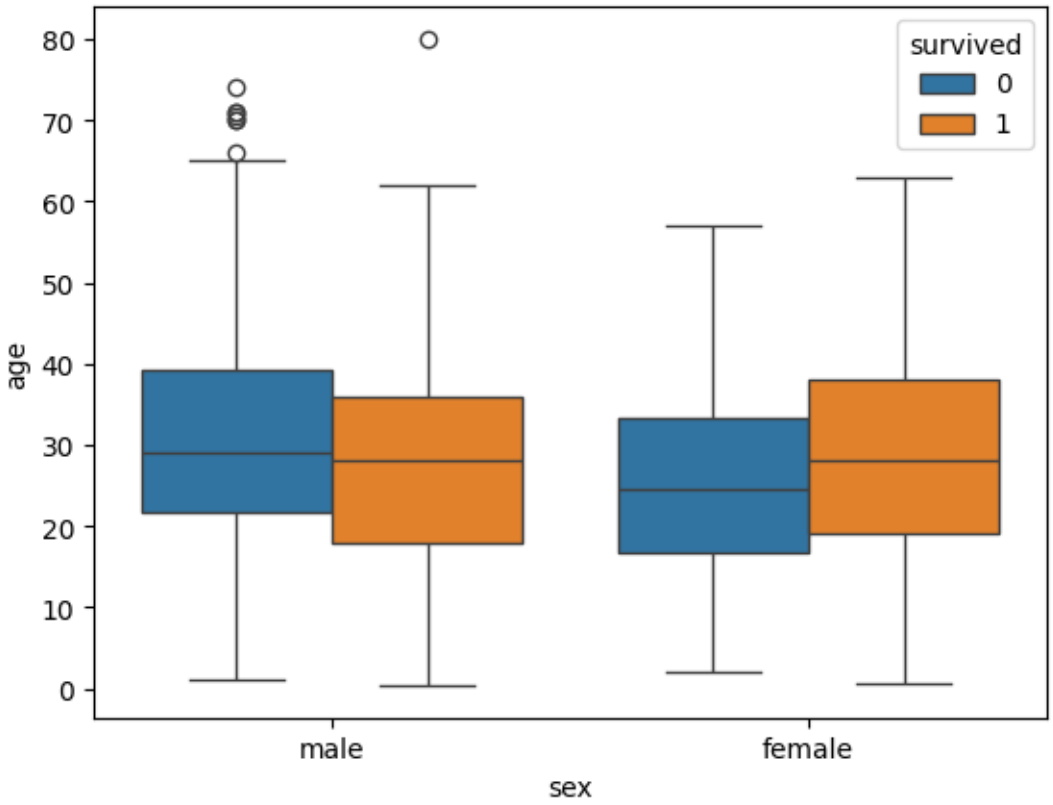
[ ]:
```
sn.boxplot(x='sex', y='age', data=titanic)
```

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



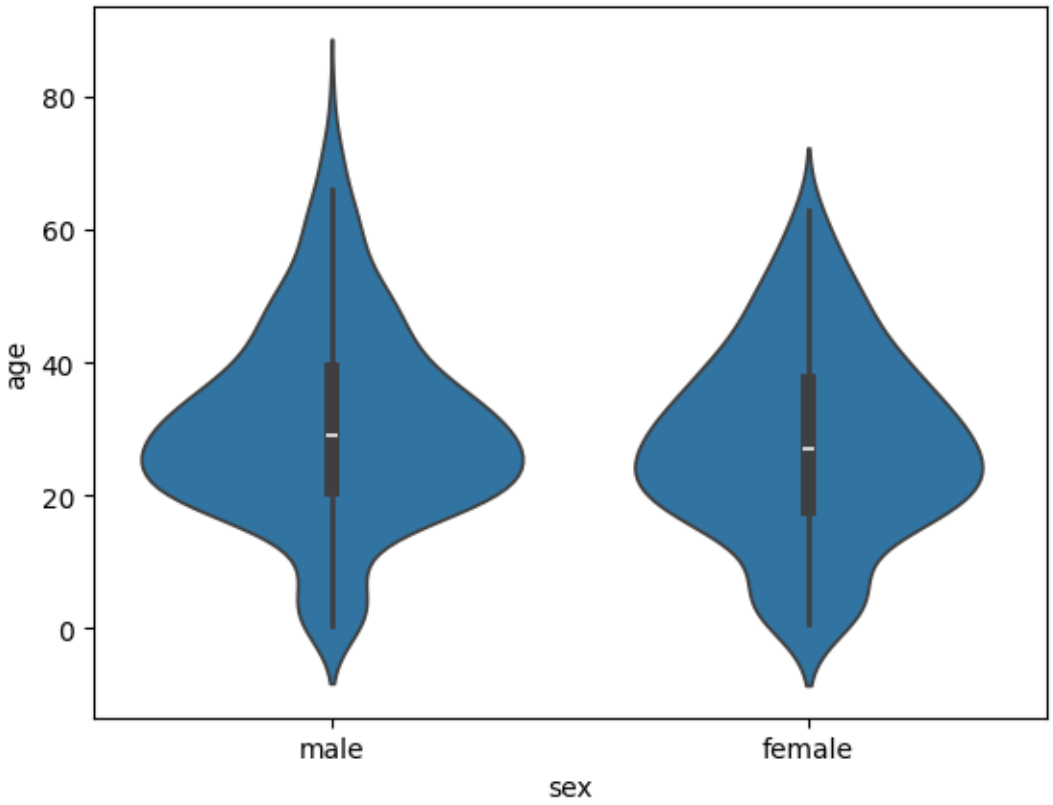
[ ]:
```
sn.boxplot(x='sex', y='age', data=titanic, hue='survived')
```

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



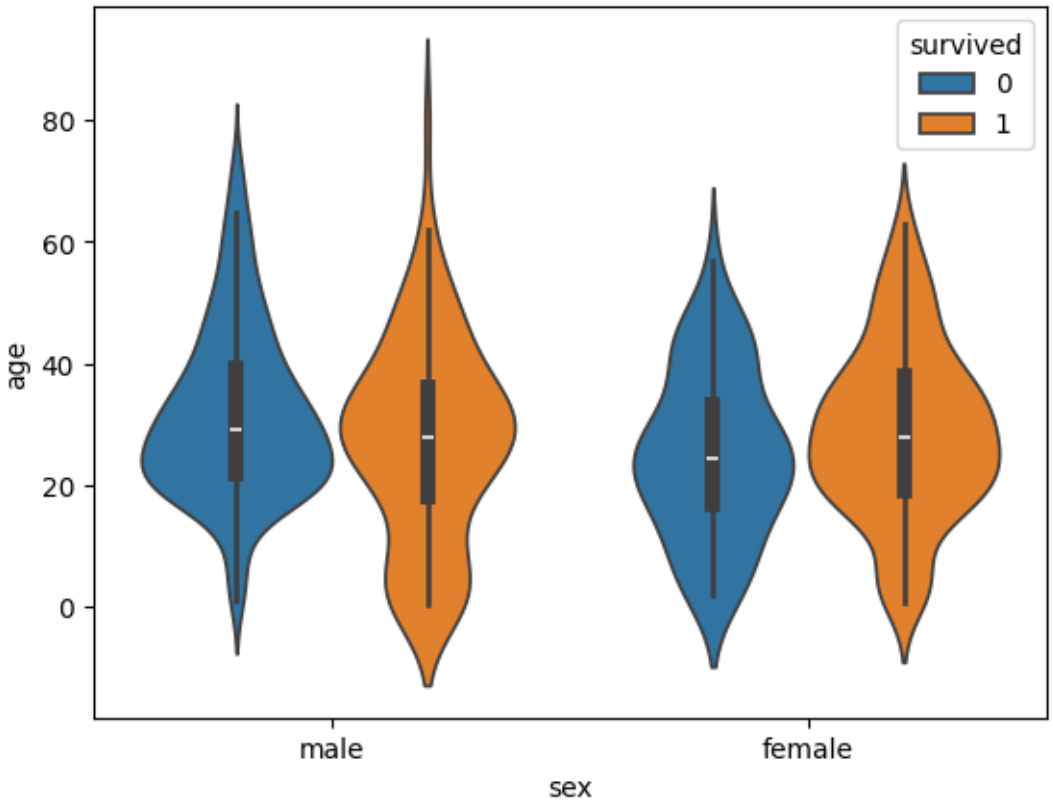
[ ]:
```
sn.violinplot(x='sex', y='age', data=titanic)
```

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



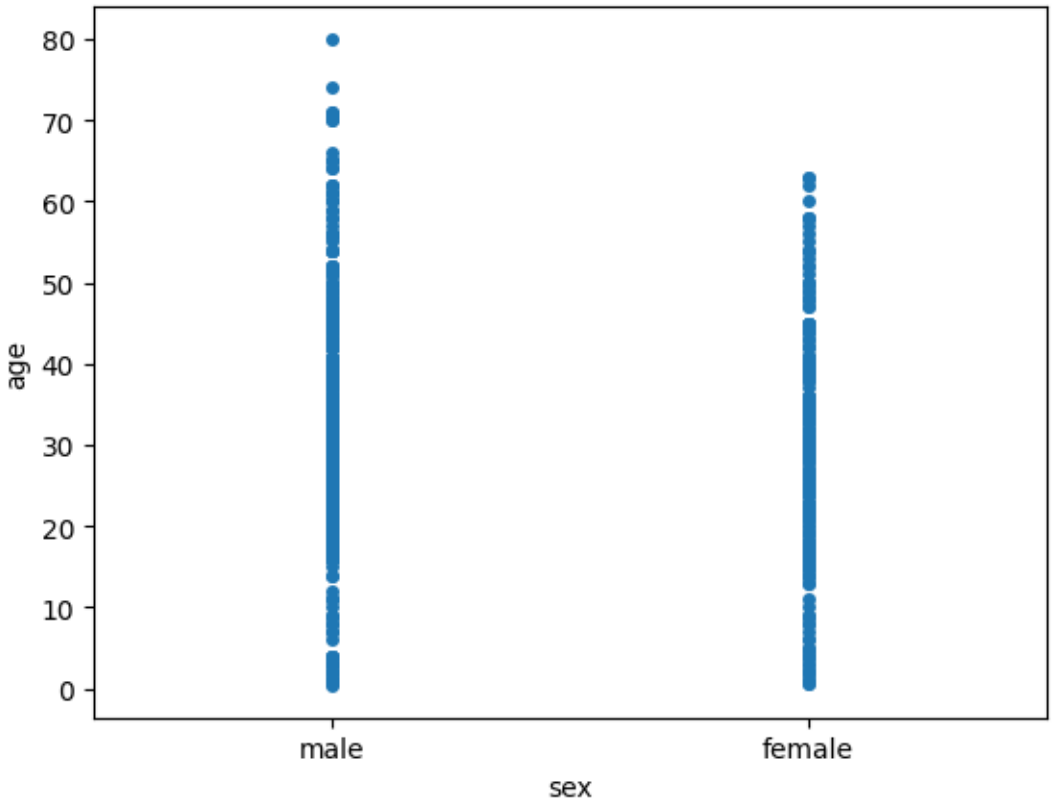
[ ]:
```
sn.violinplot(x='sex', y='age', data=titanic, hue='survived')
```

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



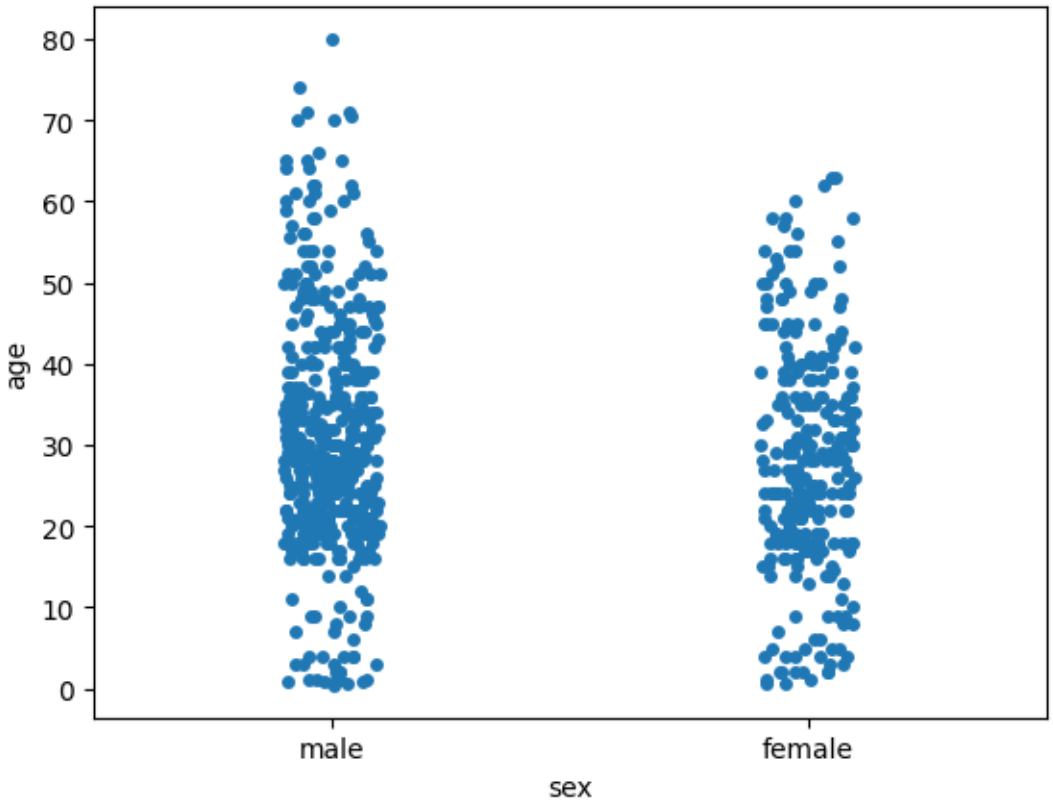
[ ]:
```
sn.stripplot(x='sex', y='age', data=titanic, jitter=False)
```

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



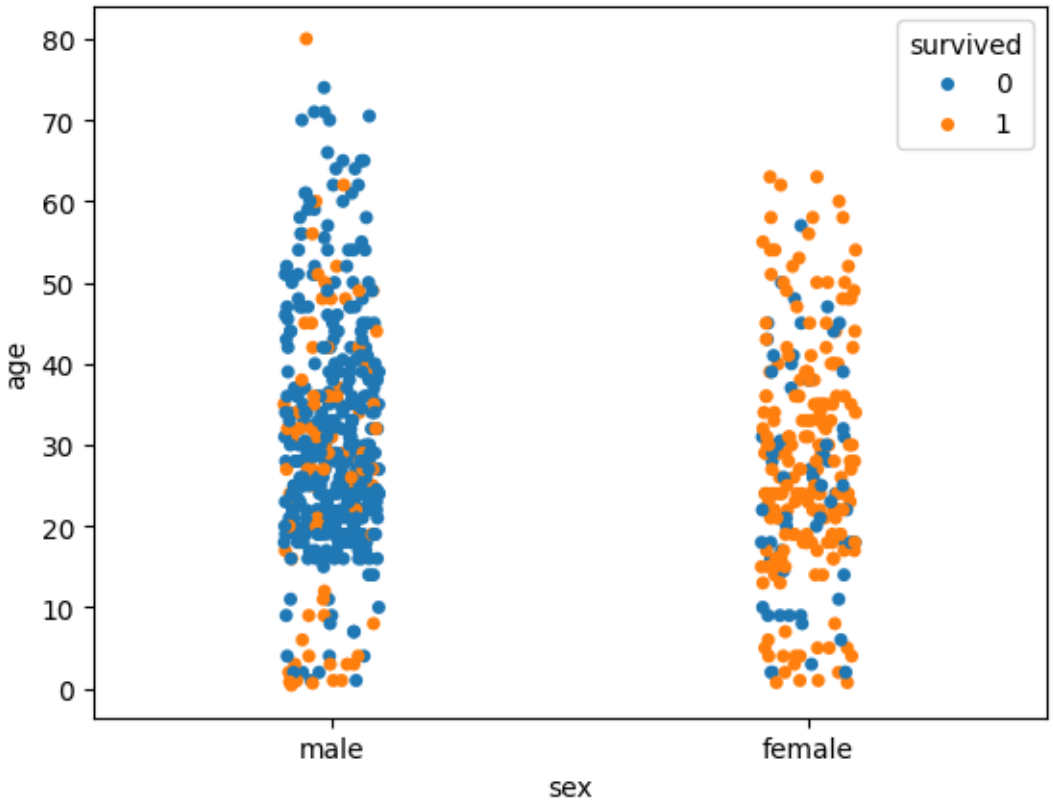
[ ]:
```
sn.stripplot(x='sex', y='age', data=titanic, jitter=True)
```

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



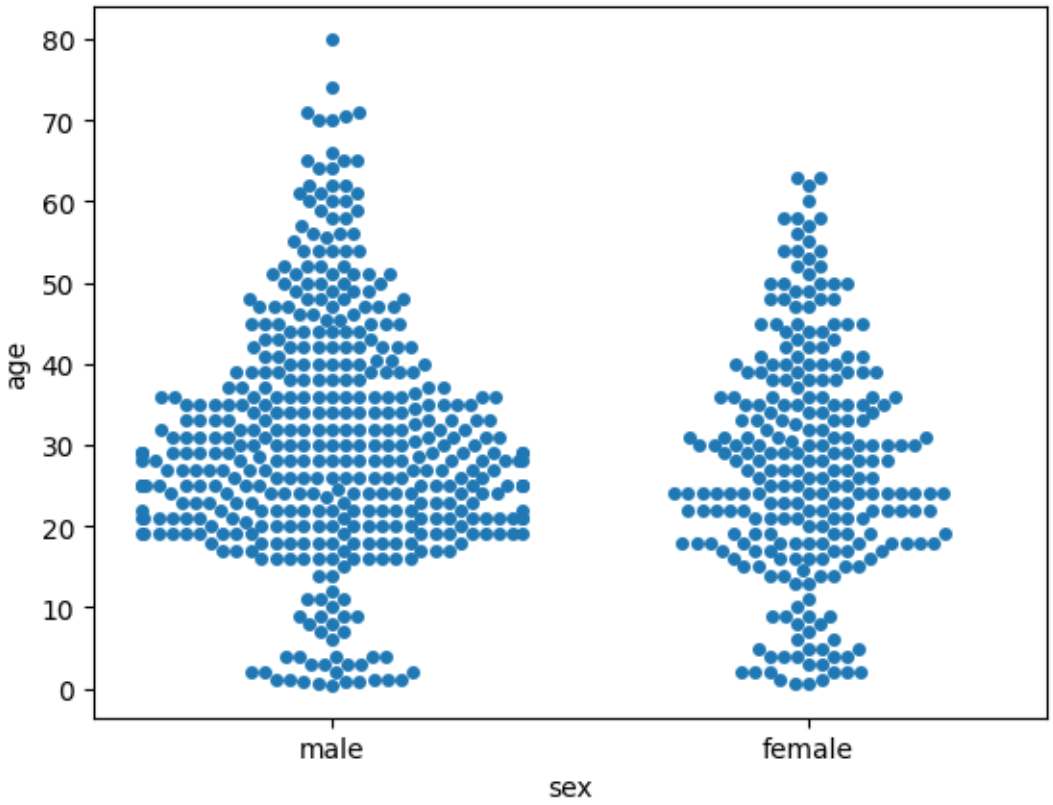
[ ]:
```
sn.stripplot(x='sex', y='age', data=titanic, jitter=True, hue='survived')
```

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



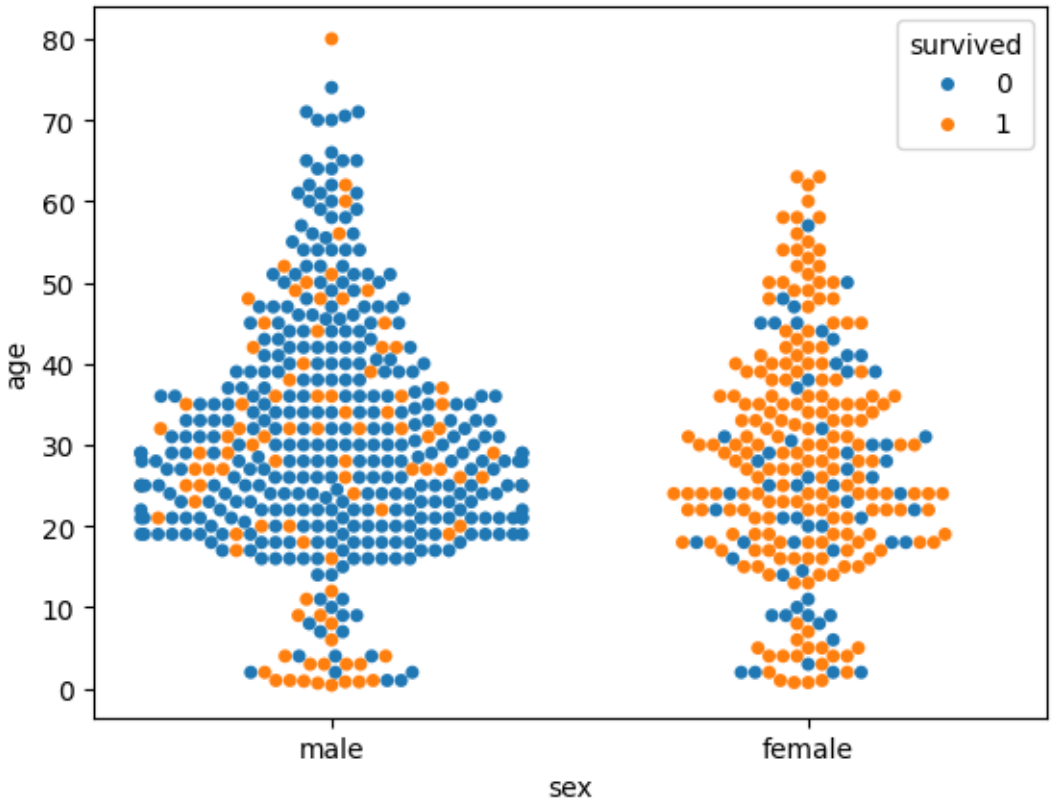
[ ]:
```
sn.swarmplot(x='sex', y='age', data=titanic)
```

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



[ ]:
```
sn.swarmplot(x='sex', y='age', data=titanic,hue='survived')
```

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



[ ]:
```
corr = titanic.corr()
```

```
<ipython-input-43-e16dc4d98a33>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  corr = titanic.corr()
```

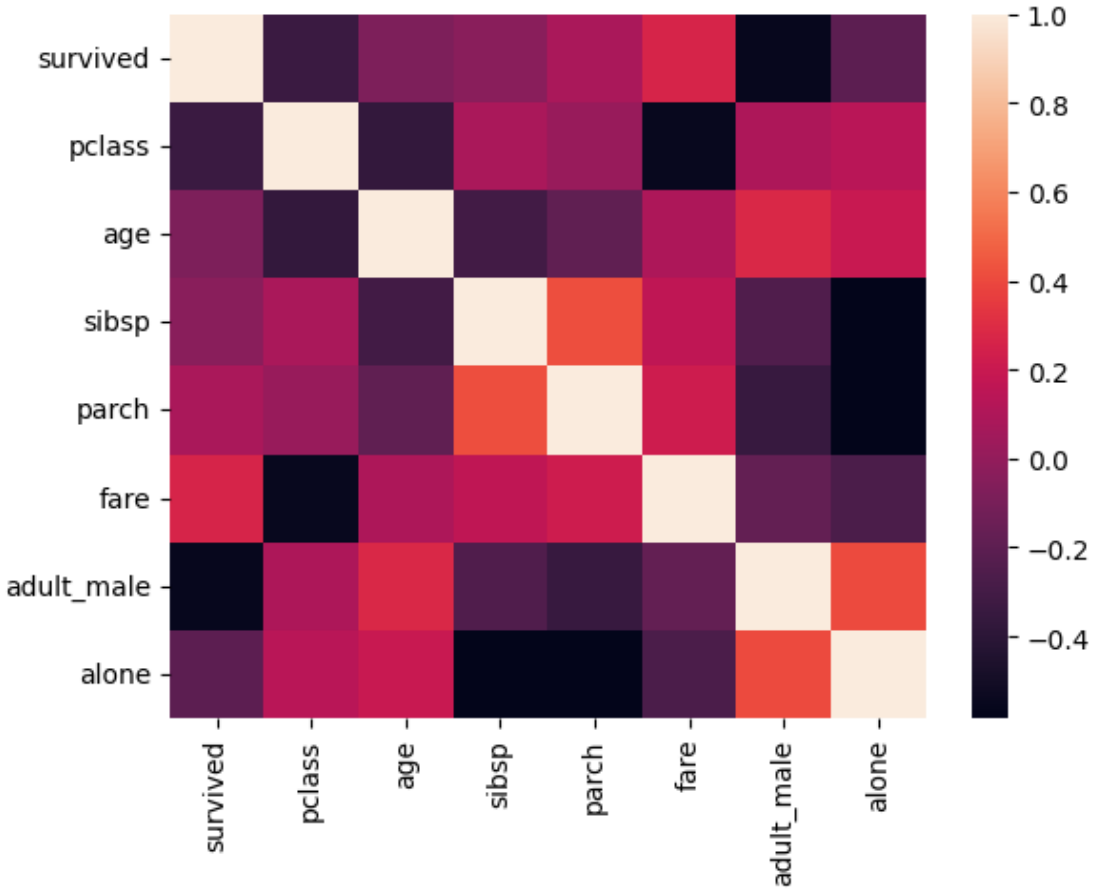
[ ]:
```
corr
```

[ ]:
```
             survived    pclass       age     sibsp     parch      fare  \
survived     1.000000 -0.338481 -0.077221 -0.035322  0.081629  0.257307
pclass      -0.338481  1.000000 -0.369226  0.083081  0.018443 -0.549500
age         -0.077221 -0.369226  1.000000 -0.308247 -0.189119  0.096067
sibsp       -0.035322  0.083081 -0.308247  1.000000  0.414838  0.159651
parch        0.081629  0.018443 -0.189119  0.414838  1.000000  0.216225
fare         0.257307 -0.549500  0.096067  0.159651  0.216225  1.000000
adult_male  -0.557080  0.094035  0.280328 -0.253586 -0.349943 -0.182024
alone       -0.203367  0.135207  0.198270 -0.584471 -0.583398 -0.271832

             adult_male     alone
```

```
survived     -0.557080 -0.203367
pclass        0.094035  0.135207
age           0.280328  0.198270
sibsp        -0.253586 -0.584471
parch        -0.349943 -0.583398
fare         -0.182024 -0.271832
adult_male    1.000000  0.404744
alone         0.404744  1.000000
```

[ ]:
```
sn.heatmap(corr)
```

[ ]:
```
<Axes: >
```

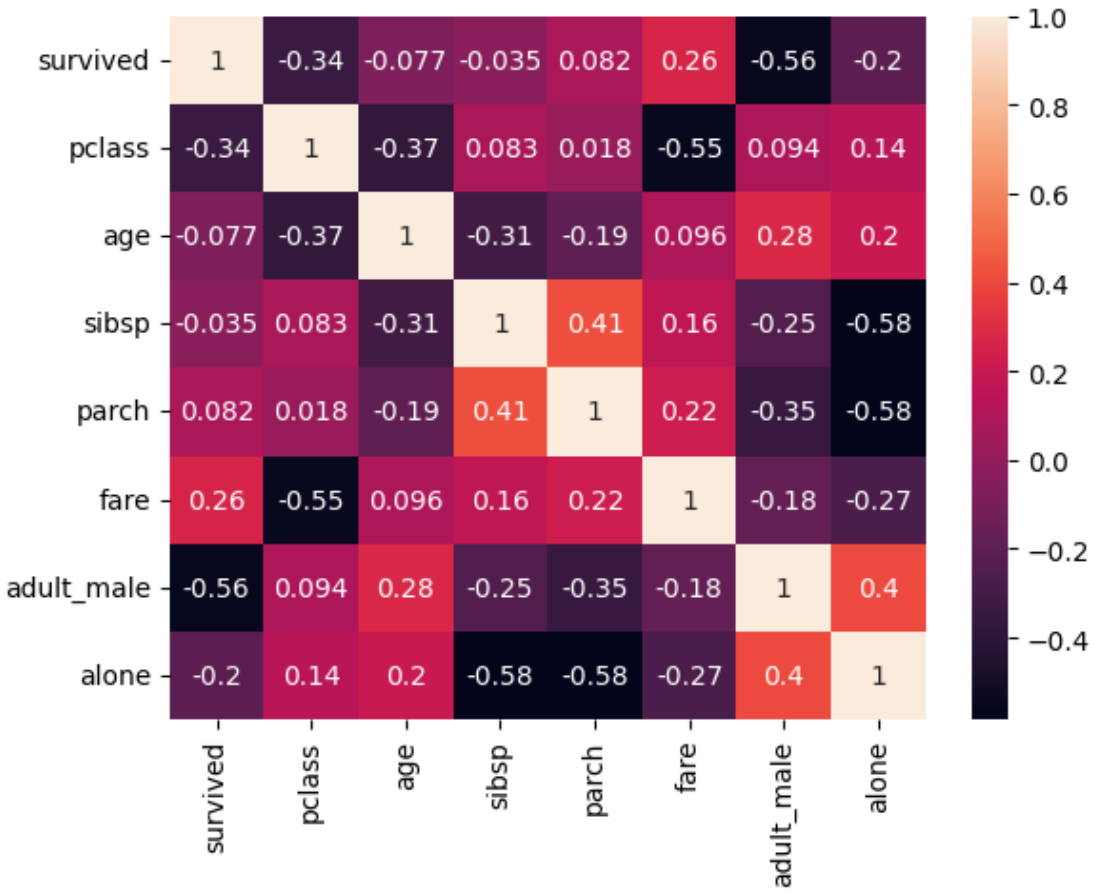


[ ]:
```
corr = titanic.corr()
sn.heatmap(corr,annot=True)
```

```
<ipython-input-49-af7ebf149f61>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
```

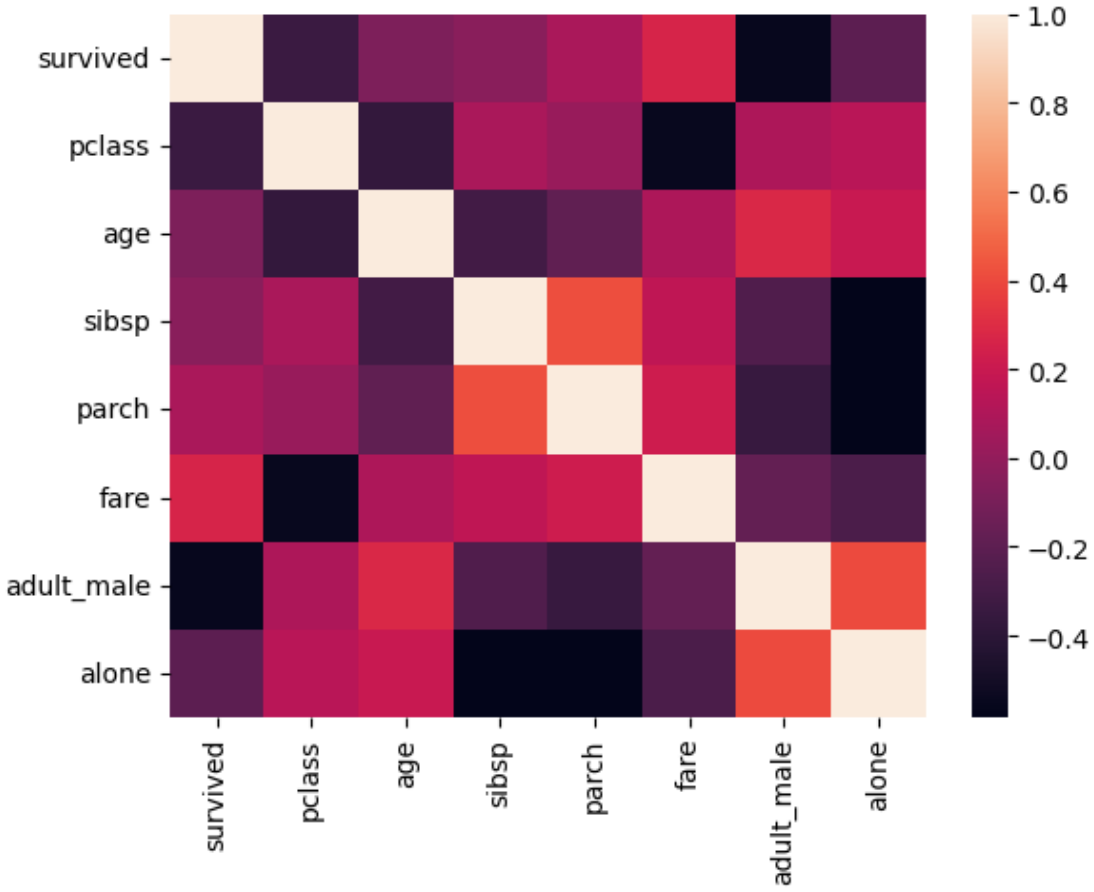
```
to silence this warning.
  corr = titanic.corr()
```

[ ]: <Axes: >



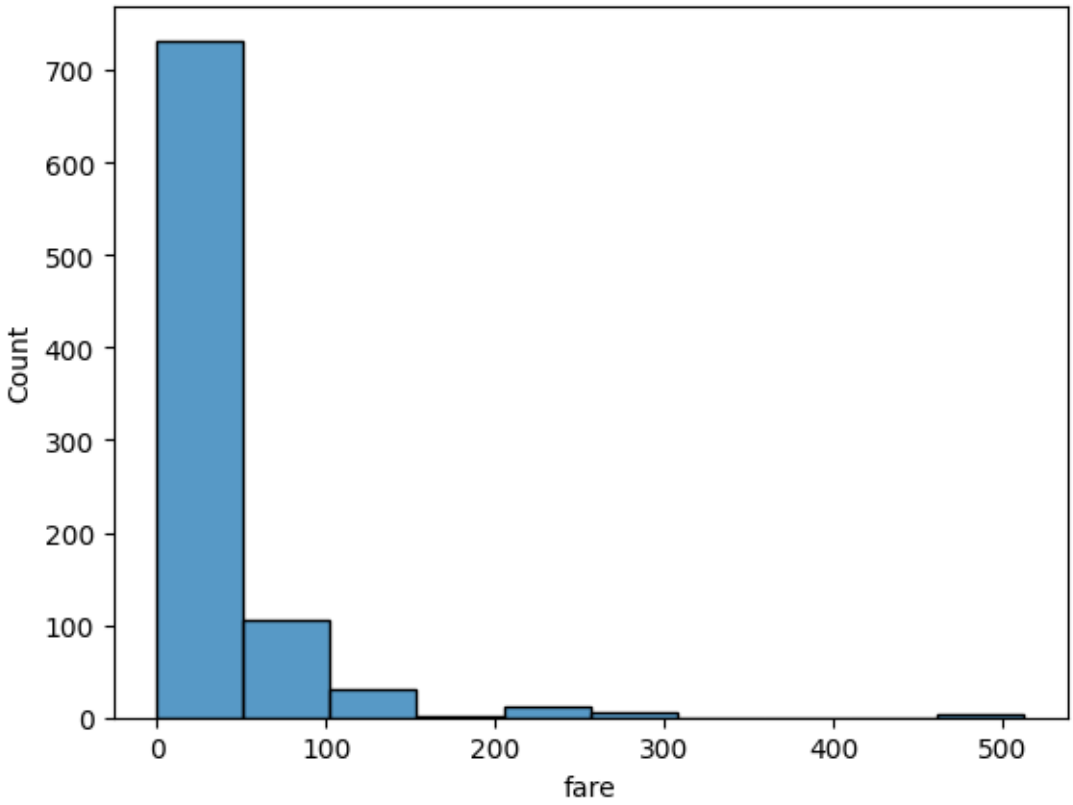
[ ]:
```
sn.heatmap(corr)
```

[ ]: <Axes: >



[ ]:
```
sn.histplot(titanic['fare'],kde=False, bins=10)
```

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


700
600
500
400
300
200
100
0
Count
0
100
200
300
400
500
fare