In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
```

In [2]:
```
text = sns.load_dataset('titanic')
text
```

Out[2]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN	Southampton	no	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	B	Southampton	yes	True
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN	Southampton	no	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	C	Cherbourg	yes	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN	Queenstown	no	True

891 rows × 15 columns

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

Out[3]:
```
(891, 15)
```

In [4]:
```
text.isnull().sum()
```

Out[4]:
```
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
```

In [5]:
```
text.isnull()
```

Out[5]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	False	False	False	False	False	False	False	False	False	False	False	True	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	True	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	True	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False	False	True	False	False	False
887	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	True	False	False	False	False	False	False	False	True	False	False	False
889	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	False	False	True	False	False	False

891 rows × 15 columns

In [6]:
```
text.describe()
```

Out[6]:

	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
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

In [7]:
```
text
```

Out[7]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN	Southampton	no	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	B	Southampton	yes	True
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN	Southampton	no	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	C	Cherbourg	yes	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN	Queenstown	no	True

891 rows × 15 columns

In [8]:
```
text.drop('deck',axis=1,inplace=True)
```

In [9]:
```
text
```

Out[9]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	Southampton	no	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	Southampton	no	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	Southampton	yes	True
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	Southampton	no	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	Cherbourg	yes	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	Queenstown	no	True

891 rows × 14 columns

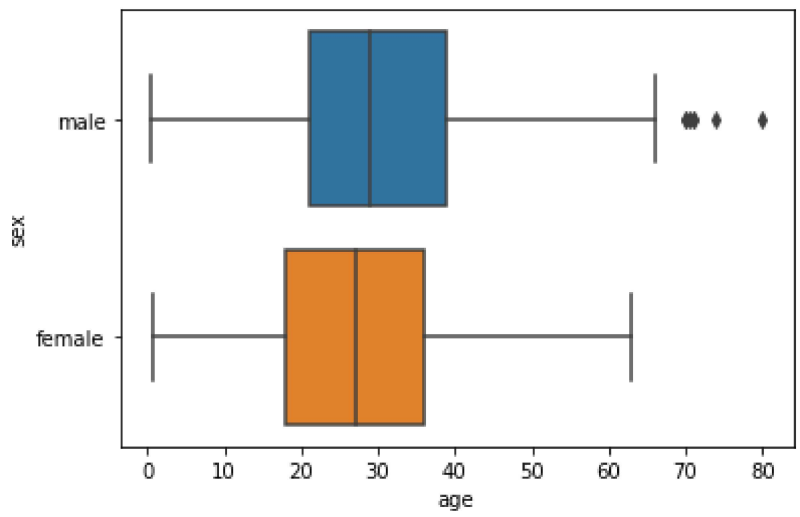
In [10]:
```
x = text
x1 = x
x1.dropna(inplace=True)
```

In [11]:
```
text.isnull().sum()
```

Out[11]:
```
survived       0
pclass         0
sex            0
age            0
sibsp          0
parch          0
fare           0
embarked       0
class          0
who            0
adult_male     0
embark_town    0
alive          0
alone          0
dtype: int64
```

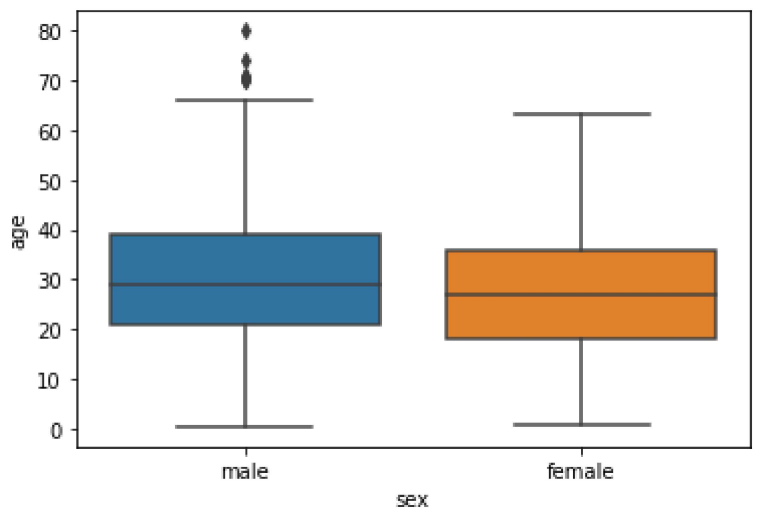
In [12]:
```
sns.boxplot(x="age",y="sex",data=x1)
```

Out[12]: <AxesSubplot:xlabel='age', ylabel='sex'>



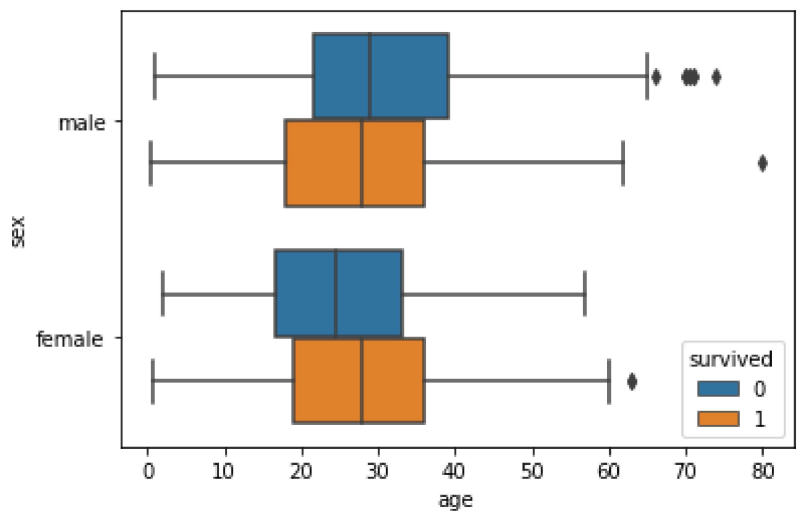
In [13]:
```
sns.boxplot(x="sex",y="age",data=x1)
```

Out[13]: <AxesSubplot:xlabel='sex', ylabel='age'>



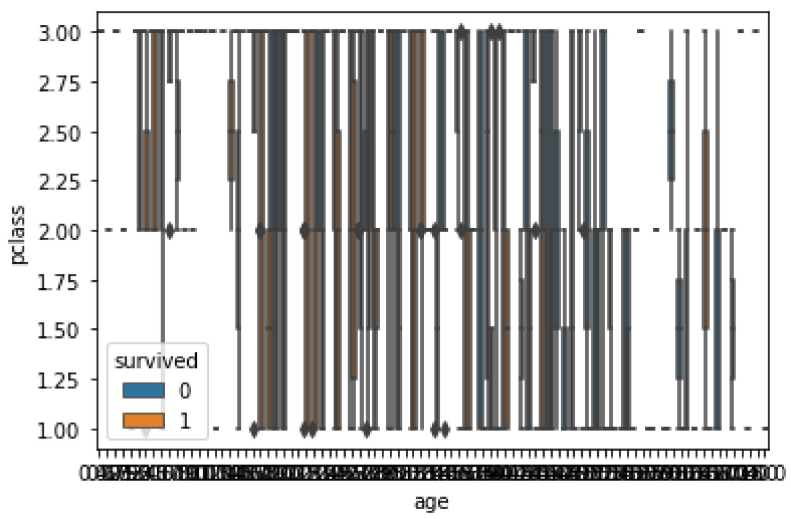
In [14]:
```
sns.boxplot(x="age",y="sex",hue='survived',data=x1)
```

Out[14]: <AxesSubplot:xlabel='age', ylabel='sex'>



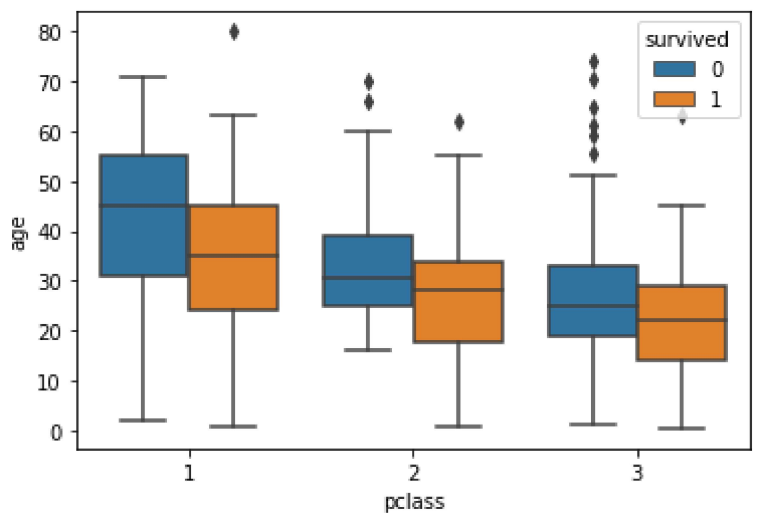
In [15]:
```
sns.boxplot(x="age",y="pclass",hue='survived',data=x1)
```

Out[15]: <AxesSubplot:xlabel='age', ylabel='pclass'>



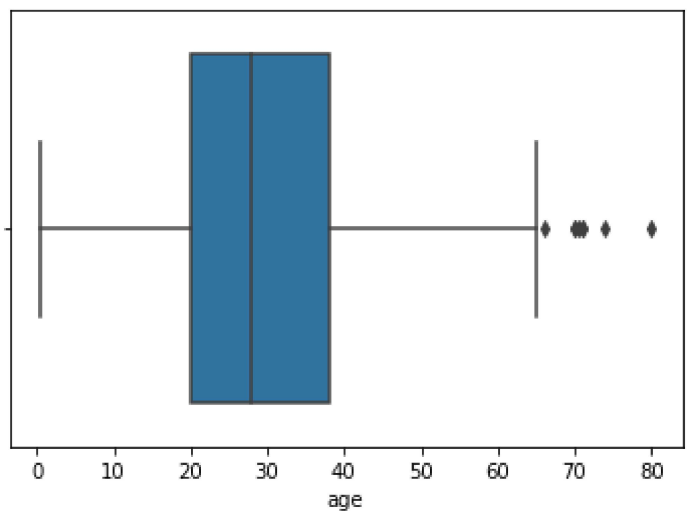
In [16]:
```
sns.boxplot(x="pclass",y="age",hue='survived',data=x1)
```

Out[16]: <AxesSubplot:xlabel='pclass', ylabel='age'>



In [17]:
```
sns.boxplot(x='age',data=x1)
```

Out[17]: <AxesSubplot:xlabel='age'>



In [18]:
```
data=sorted(x1['age'])
data
```

```
Out[18]: [0.42,
 0.67,
 0.75,
 0.75,
 0.83,
 0.83,
 0.92,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 2.0,
 2.0,
 2.0,
 2.0,
 2.0,
 2.0,
 2.0,
 2.0,
 2.0,
 2.0,
 3.0,
 3.0,
 3.0,
 3.0,
 3.0,
 3.0,
 4.0,
 4.0,
 4.0,
 4.0,
 4.0,
 4.0,
 4.0,
 4.0,
 4.0,
 4.0,
 5.0,
 5.0,
 5.0,
 5.0,
 6.0,
 6.0,
 6.0,
 7.0,
 7.0,
 7.0,
 8.0,
 8.0,
 8.0,
 8.0,
 9.0,
 9.0,
 9.0,
 9.0,
 9.0,
 9.0,
 9.0,
 9.0,
 10.0,
 10.0,
 11.0,
 11.0,
 11.0,
 11.0,
 12.0,
 13.0,
 13.0,
 14.0,
 14.0,
 14.0,
 14.0,
 14.0,
 14.0,
 14.5,
 15.0,
 15.0,
 15.0,
 15.0,
 15.0,
 16.0,
 16.0,
 16.0,
 16.0,
 16.0,
 16.0,
 16.0,
 16.0,
 16.0,
 16.0,
```

```
16.0,
16.0,
16.0,
16.0,
16.0,
16.0,
16.0,
17.0,
17.0,
17.0,
17.0,
17.0,
17.0,
17.0,
17.0,
17.0,
17.0,
17.0,
17.0,
17.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
18.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
19.0,
20.0,
20.0,
20.0,
20.0,
20.0,
20.0,
20.0,
20.0,
20.0,
20.0,
20.0,
20.0,
20.0,
20.0,
20.0,
20.5,
21.0,
21.0,
21.0,
21.0,
21.0,
21.0,
```

```
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            21.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            22.0,
            23.0,
            23.0,
            23.0,
            23.0,
            23.0,
            23.0,
            23.0,
            23.0,
            23.0,
            23.0,
            23.0,
            23.0,
            23.0,
            23.0,
            23.0,
            23.5,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.0,
            24.5,
            25.0,
```

25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
25.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
26.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
27.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.0,
28.5,
28.5,
29.0,
29.0,
29.0,
29.0,
29.0,
29.0,
29.0,
29.0,

29.0,
29.0,
29.0,
29.0,
29.0,
29.0,
29.0,
29.0,
29.0,
29.0,
29.0,
29.0,
29.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.0,
30.5,
30.5,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
31.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.0,
32.5,
32.5,
33.0,
33.0,
33.0,
33.0,
33.0,
33.0,
33.0,
33.0,
33.0,
33.0,
33.0,
33.0,
33.0,
33.0,
33.0,
33.0,
33.0,
34.0,
34.0,

```
            34.0,
            34.0,
            34.0,
            34.0,
            34.0,
            34.0,
            34.0,
            34.0,
            34.0,
            34.0,
            34.0,
            34.0,
            34.0,
            34.5,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            35.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.0,
            36.5,
            37.0,
            37.0,
            37.0,
            37.0,
            37.0,
            37.0,
            38.0,
            38.0,
            38.0,
            38.0,
            38.0,
            38.0,
            38.0,
            38.0,
            38.0,
            38.0,
            39.0,
            39.0,
            39.0,
            39.0,
            39.0,
            39.0,
            39.0,
            39.0,
            39.0,
            39.0,
            39.0,
            39.0,
            39.0,
            39.0,
            40.0,
            40.0,
            40.0,
            40.0,
            40.0,
            40.0,
            40.0,
            40.0,
```

40.0,
40.0,
40.0,
40.0,
40.0,
40.5,
40.5,
41.0,
41.0,
41.0,
41.0,
41.0,
41.0,
41.0,
42.0,
42.0,
42.0,
42.0,
42.0,
42.0,
42.0,
42.0,
42.0,
42.0,
42.0,
42.0,
42.0,
42.0,
42.0,
43.0,
43.0,
43.0,
43.0,
43.0,
44.0,
44.0,
44.0,
44.0,
44.0,
44.0,
44.0,
44.0,
44.0,
45.0,
45.0,
45.0,
45.0,
45.0,
45.0,
45.0,
45.0,
45.0,
45.0,
45.0,
45.0,
45.0,
45.5,
45.5,
46.0,
46.0,
46.0,
47.0,
47.0,
47.0,
47.0,
47.0,
47.0,
47.0,
47.0,
47.0,
48.0,
48.0,
48.0,
48.0,
48.0,
48.0,
48.0,
48.0,
48.0,
49.0,
49.0,
49.0,
49.0,
49.0,
49.0,
50.0,
50.0,
50.0,
50.0,
50.0,
50.0,
50.0,
50.0,
50.0,
50.0,
50.0,
51.0,
51.0,

```
51.0,
51.0,
51.0,
51.0,
51.0,
52.0,
52.0,
52.0,
52.0,
52.0,
52.0,
52.0,
53.0,
54.0,
54.0,
54.0,
54.0,
54.0,
54.0,
54.0,
54.0,
54.0,
55.0,
55.0,
55.5,
56.0,
56.0,
56.0,
56.0,
57.0,
57.0,
58.0,
58.0,
58.0,
58.0,
58.0,
59.0,
59.0,
60.0,
60.0,
60.0,
60.0,
60.0,
61.0,
61.0,
61.0,
62.0,
62.0,
62.0,
63.0,
63.0,
64.0,
64.0,
65.0,
65.0,
65.0,
66.0,
70.0,
70.0,
70.5,
71.0,
71.0,
74.0,
80.0]
```

In [19]:
```
q1 = np.percentile(data,25)
q3 = np.percentile(data,75)

print(q1)
print(q3)
```

```
20.0
38.0
```

In [20]:
```
IQR = q3-q1
minimum = q1-(1.5*IQR)
maximum = q3+(1.5*IQR)
print(IQR,minimum,maximum)
```

```
18.0 -7.0 65.0
```

In [21]:
```
outliers = []
data = sorted(x1['age'])
def detect_outliers_iqr(data):
    q1 = np.percentile(data, 25)
    q3 = np.percentile(data, 75)

    IQR = q3-q1
    lwr_bound = q1-(1.5*IQR)
    upr_bound = q3+(1.5*IQR)

    for i in data:
        if (i<lwr_bound or i>upr_bound):
            outliers.append(i)
    return outliers
marks_outliers = detect_outliers_iqr(data)
print("Outliers from IQR method: ", marks_outliers)
```

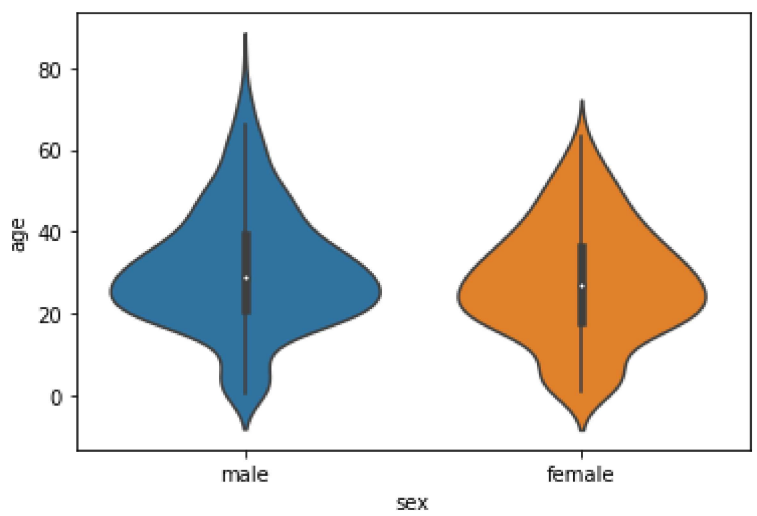
```
Outliers from IQR method:  [66.0, 70.0, 70.0, 70.5, 71.0, 71.0, 74.0, 80.0]
```

In [ ]:

In [22]:
```
#extra (types of boxplot)
```

In [23]:
```
sns.violinplot(x='sex',y='age',data=x1)
```

Out[23]: <AxesSubplot:xlabel='sex', ylabel='age'>

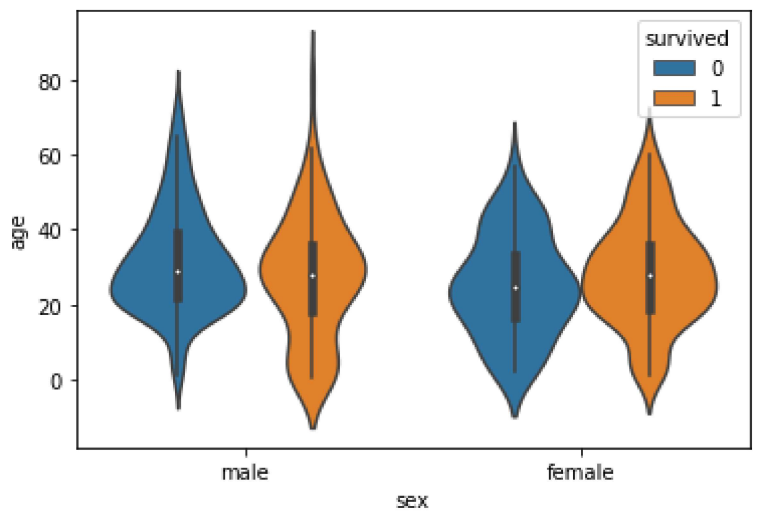


In [24]:
```
sns.violinplot(x='sex',y='age',hue='survived',split=True,data=x1)
```

Out[24]: <AxesSubplot:xlabel='sex', ylabel='age'>

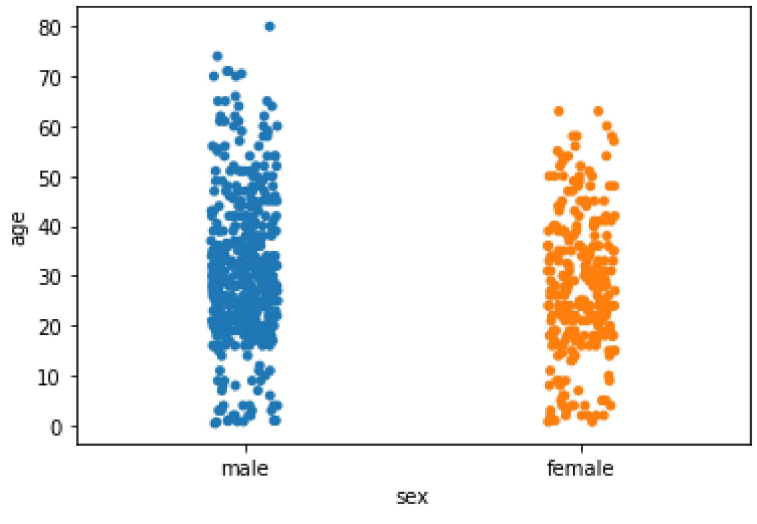
In [25]:
```
sns.violinplot(x='sex',y='age',hue='survived',data=x1)
```

Out[25]: <AxesSubplot:xlabel='sex', ylabel='age'>



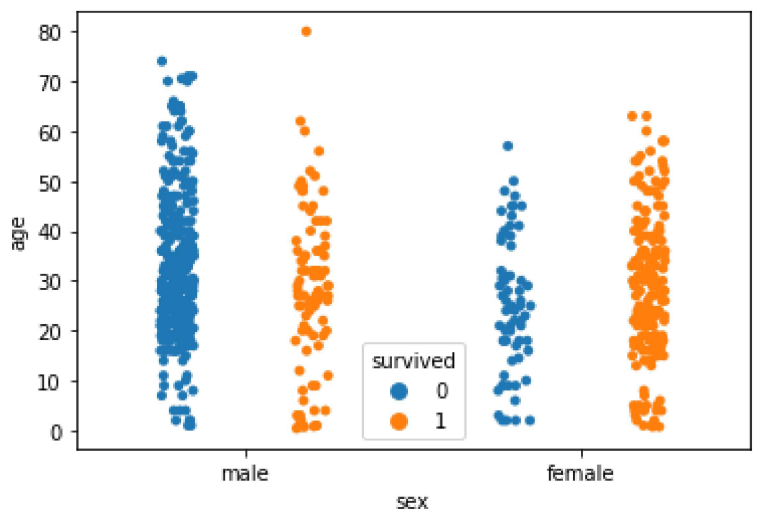
In [26]:
```
sns.stripplot(x='sex',y='age',data=x1)
```

Out[26]: <AxesSubplot:xlabel='sex', ylabel='age'>



In [27]:
```
sns.stripplot(x='sex',y='age',data=x1,dodge=True,hue='survived')
```

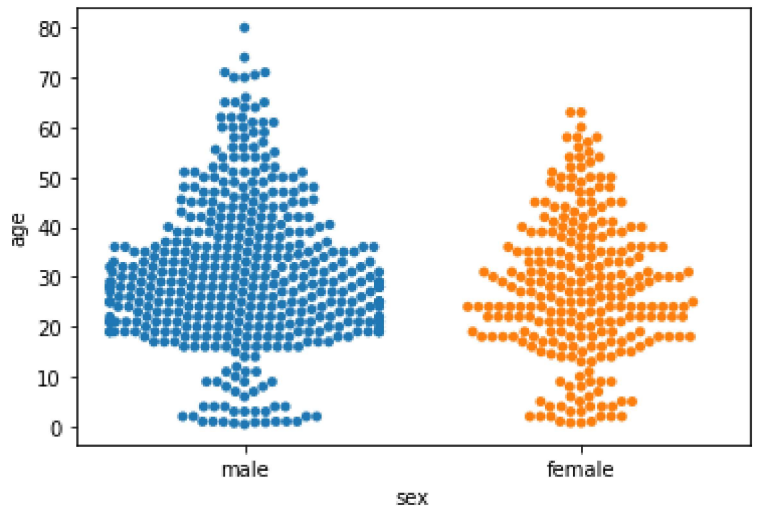
Out[27]: <AxesSubplot:xlabel='sex', ylabel='age'>



In [28]:
```
sns.swarmplot(x='sex',y='age',data=x1)
```

```
C:\Users\user\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 7.5% of the points cannot be placed; you may want to de
crease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[28]: <AxesSubplot:xlabel='sex', ylabel='age'>



In [29]:
```
df = x1[['sex','age']]
df
```

Out[29]:

	sex	age
0	male	22.0
1	female	38.0
2	female	26.0
3	female	35.0
4	male	35.0
...	...	...
885	female	39.0
886	male	27.0
887	female	19.0
889	male	26.0
890	male	32.0

712 rows × 2 columns

In [30]:
```
d_male = df[df['sex']=='male']
d_male
```

Out[30]:

	sex	age
0	male	22.0
4	male	35.0
6	male	54.0
7	male	2.0
12	male	20.0
...	...	...
883	male	28.0
884	male	25.0
886	male	27.0
889	male	26.0
890	male	32.0

453 rows × 2 columns

In [31]:
```
d_male_age = d_male['age']
```

In [32]:
```
d_male_age
```

Out[32]:
```
0      22.0
4      35.0
6      54.0
7       2.0
12     20.0
       ...
883    28.0
884    25.0
886    27.0
889    26.0
890    32.0
Name: age, Length: 453, dtype: float64
```

In [33]:
```
outliers = []
data = sorted(d_male_age)
def detect_outliers_iqr(data):
    q1 = np.percentile(data, 25)
    q3 = np.percentile(data, 75)
    print(q1,q3)

    IQR = q3-q1
    lwr_bound = q1-(1.5*IQR)
    upr_bound = q3+(1.5*IQR)
    print(lwr_bound,upr_bound)

    for i in data:
        if (i<lwr_bound or i>upr_bound):
            outliers.append(i)
    return outliers
marks_outliers = detect_outliers_iqr(data)
print("Outliers from IQR method: ", marks_outliers)
```

```
21.0 39.0
-6.0 66.0
Outliers from IQR method:  [70.0, 70.0, 70.5, 71.0, 71.0, 74.0, 80.0]
```

In [ ]:

In [ ]: