Madhur Jaripatke

Roll No. 48

TE A Computer

RMDSSOE, Warje, Pune

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

Importing Libraries

In [ ]:
```
pip install seaborn
```

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading the Dataset

In [3]:
```
titanic = sns.load_dataset('titanic')
```

In [4]:
```
titanic.head()
```

Out[4]:

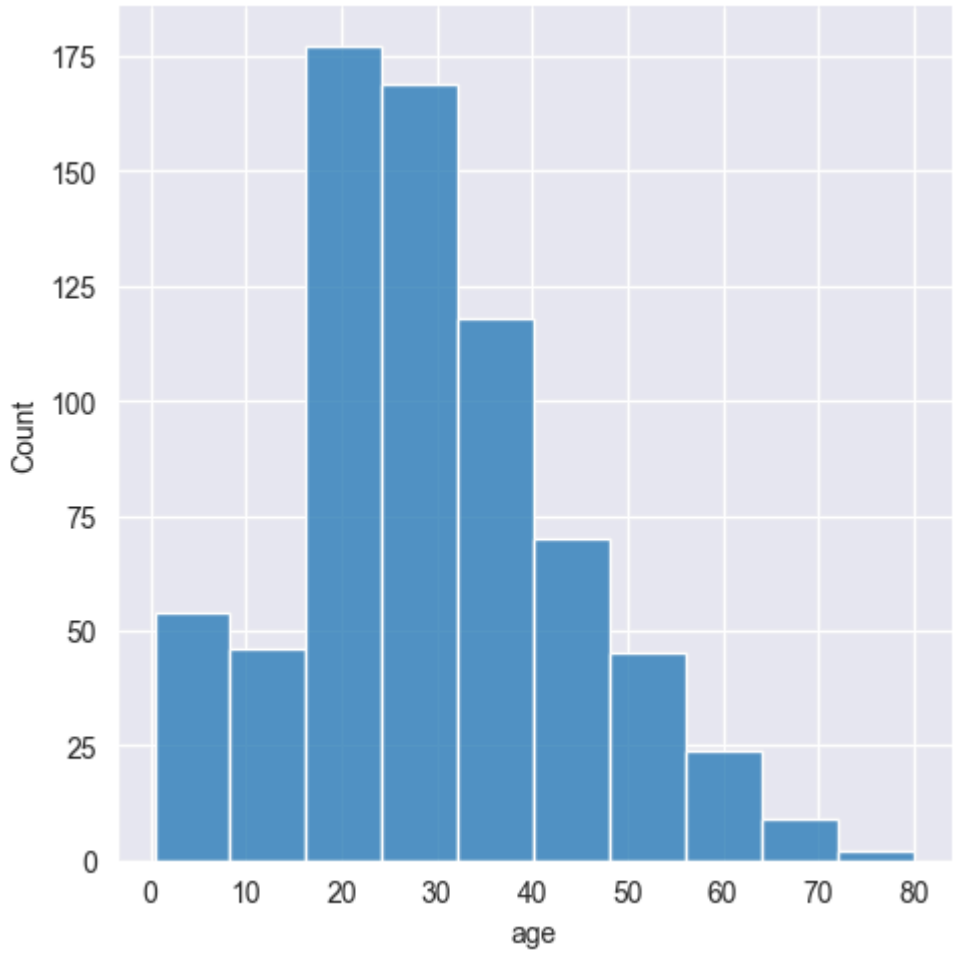
	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adul
0	0	3	male	22.0	1	0	7.2500	S	Third	man	
1	1	1	female	38.0	1	0	71.2833	C	First	woman	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	
3	1	1	female	35.0	1	0	53.1000	S	First	woman	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	

Data Visualization

Distribution of Age using distplot

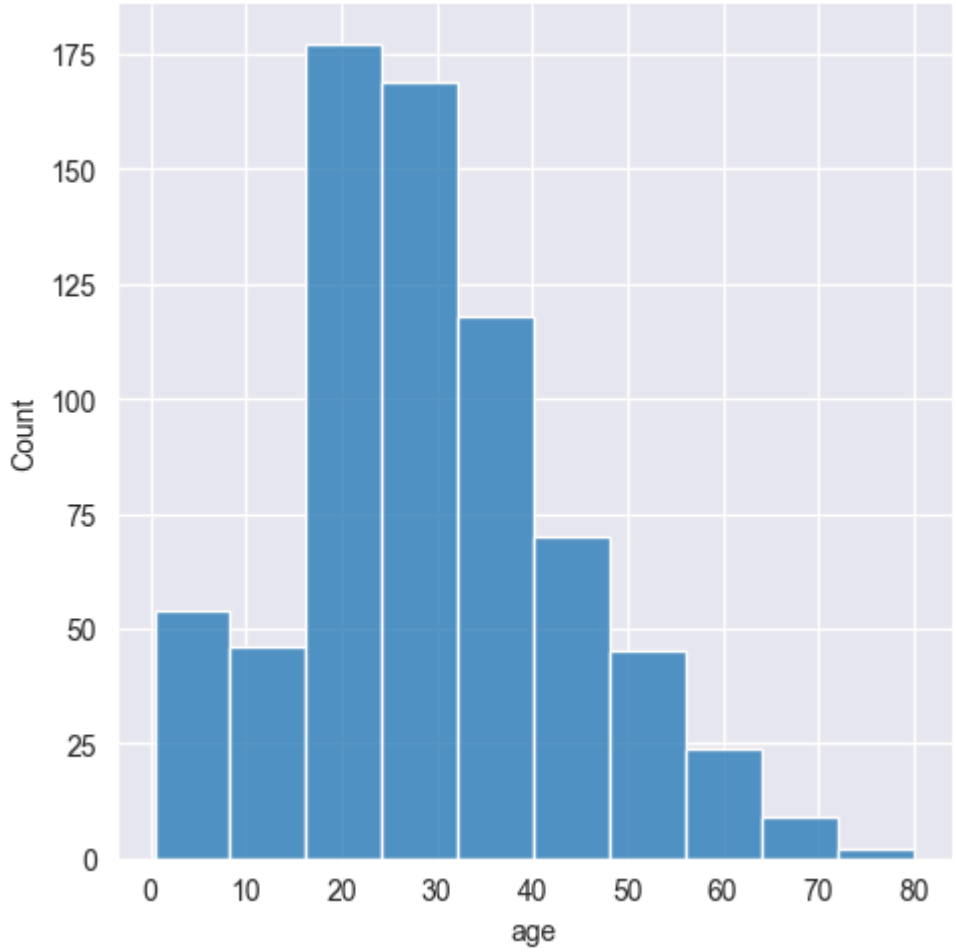
In [5]:
```
sns.displot(titanic['age'], bins = 10)
```

Out[5]: <seaborn.axisgrid.FacetGrid at 0x1bd75ec7380>



In [6]:
```
sns.displot(titanic['age'], bins = 10, kde = False)
```

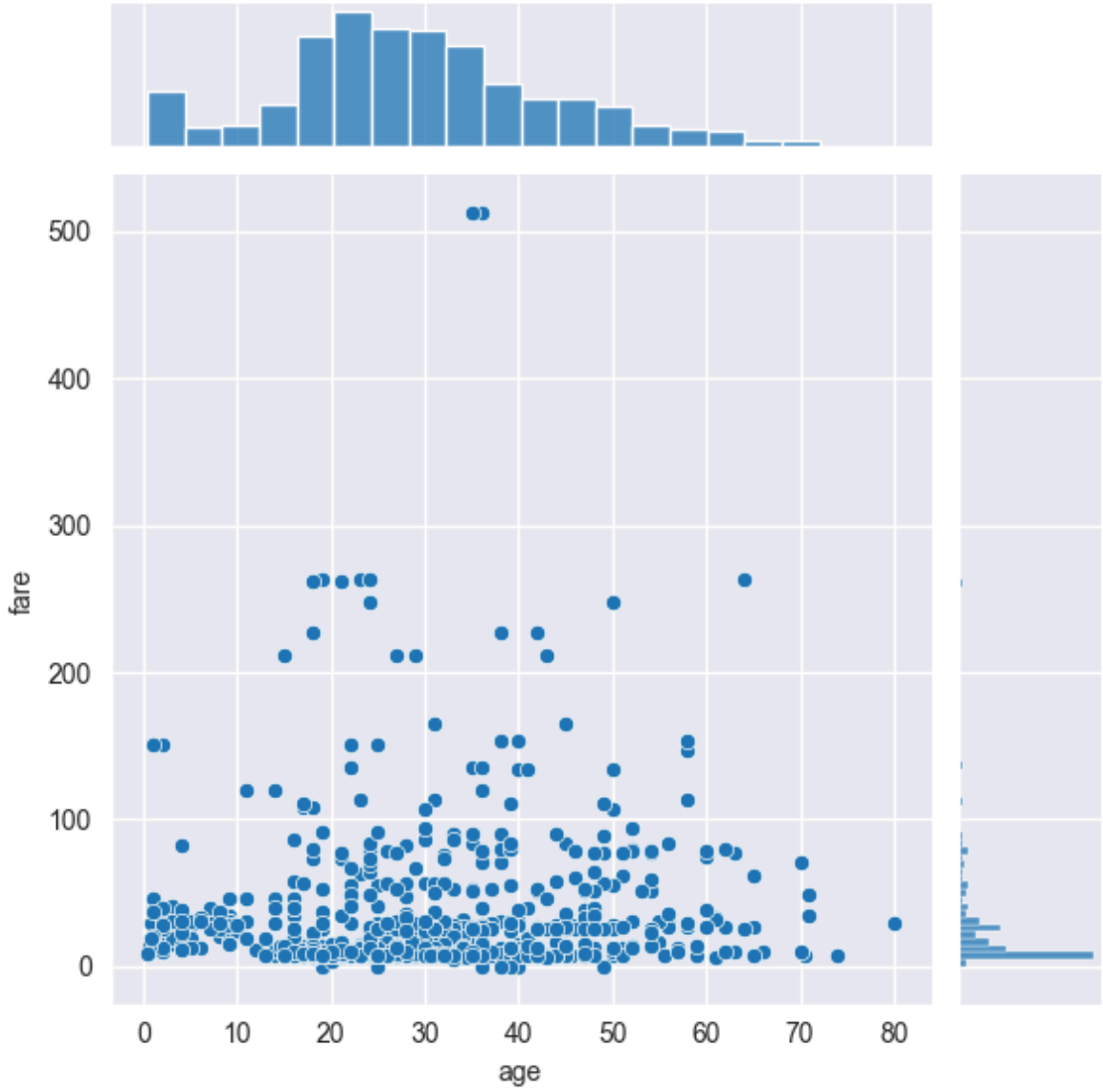
Out[6]: <seaborn.axisgrid.FacetGrid at 0x1bd171a2ea0>

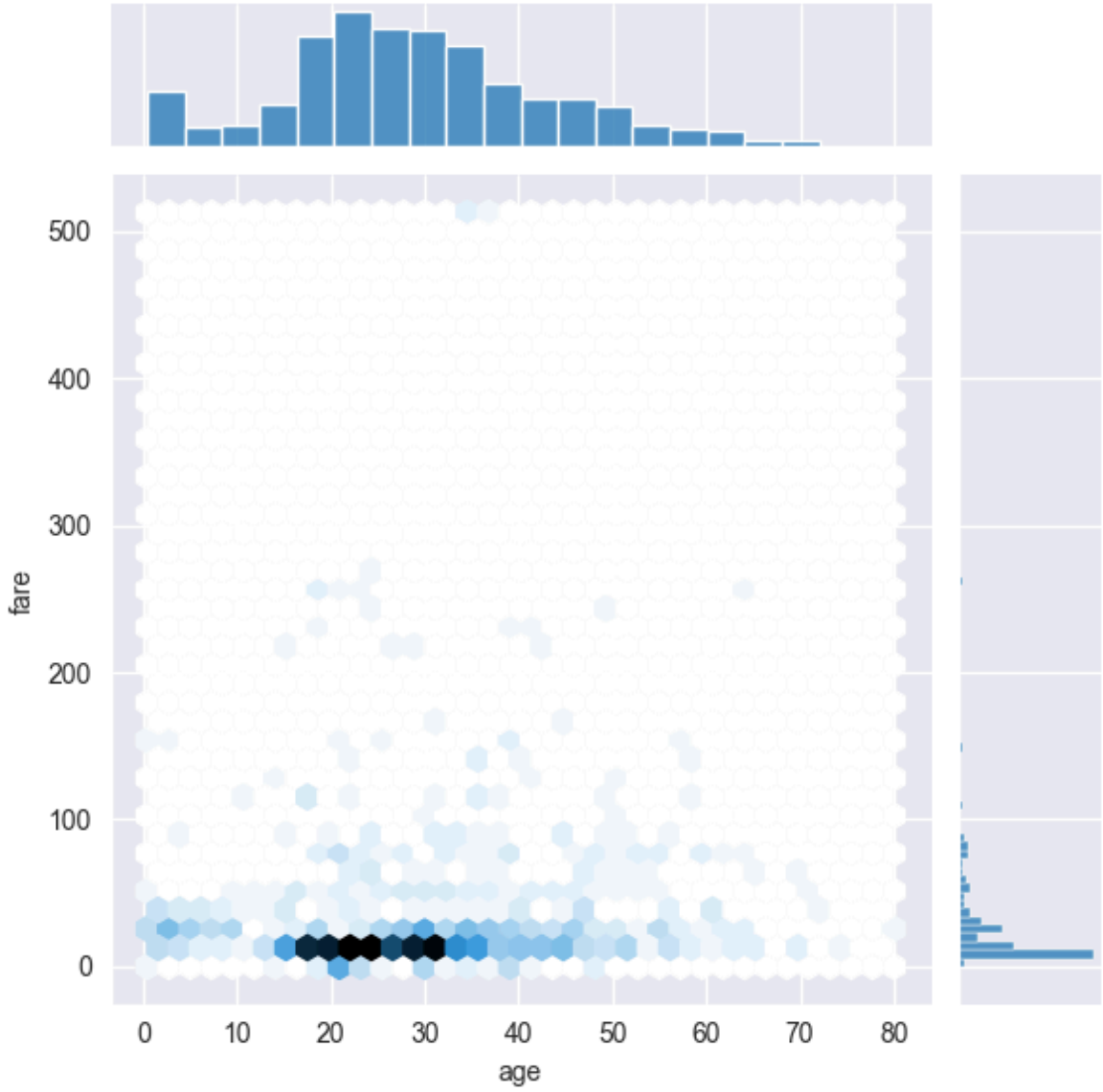


Distribution of Fare using jointplot

In [7]:
```
sns.jointplot(x = titanic['age'], y = titanic['fare'], kind = 'scatter')
sns.jointplot(x = titanic['age'], y = titanic['fare'], kind = 'hex')
```

Out[7]: <seaborn.axisgrid.JointGrid at 0x1bd1717f9e0>

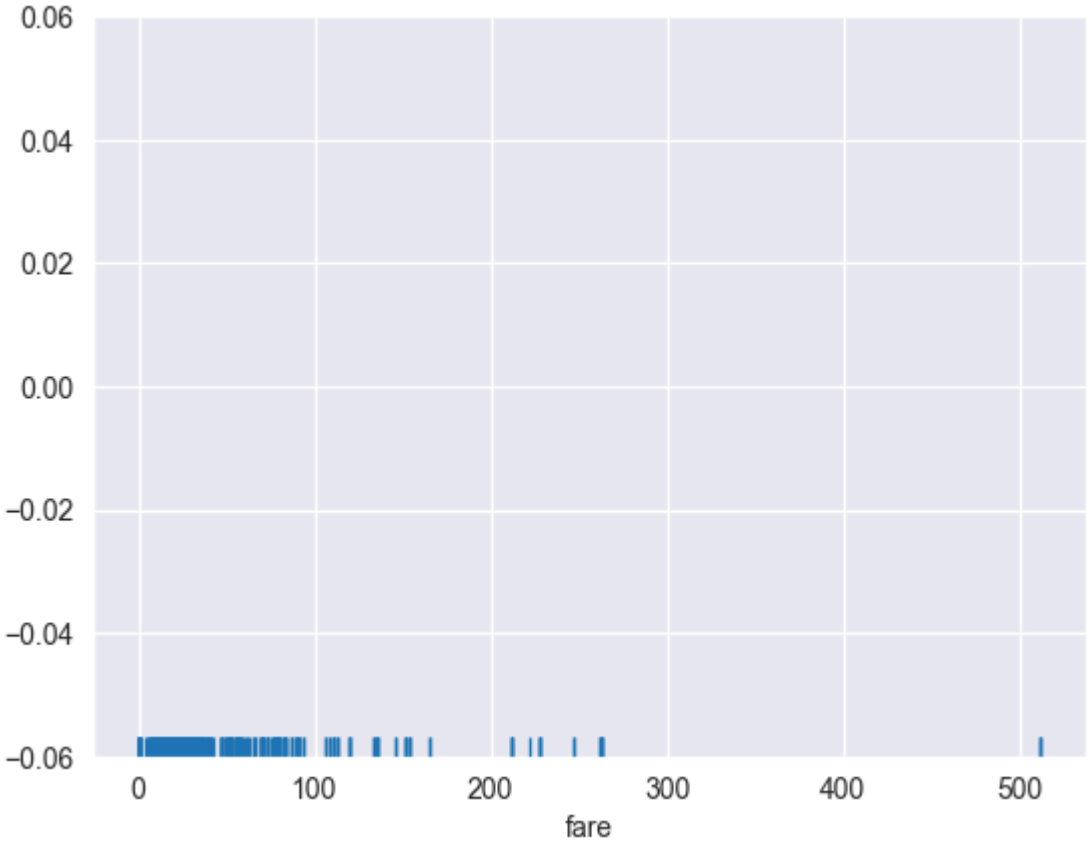

500
400
300
200
100
0
fare
0
10
20
30
40
50
60
70
80
age



Distribution of Fare using rugplot

In [8]:
```
sns.rugplot(titanic['fare'])
```

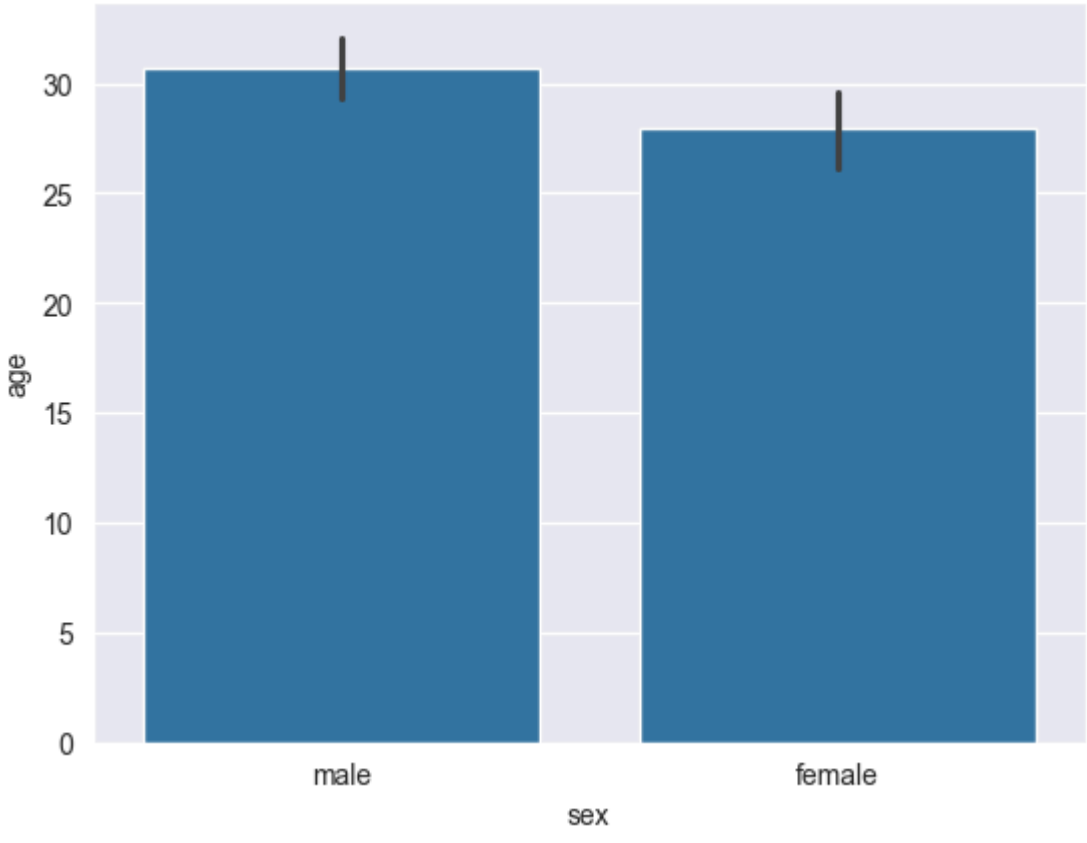
Out[8]:
```
<Axes: xlabel='fare'>
```



Distribution of Age & Gender using barplot

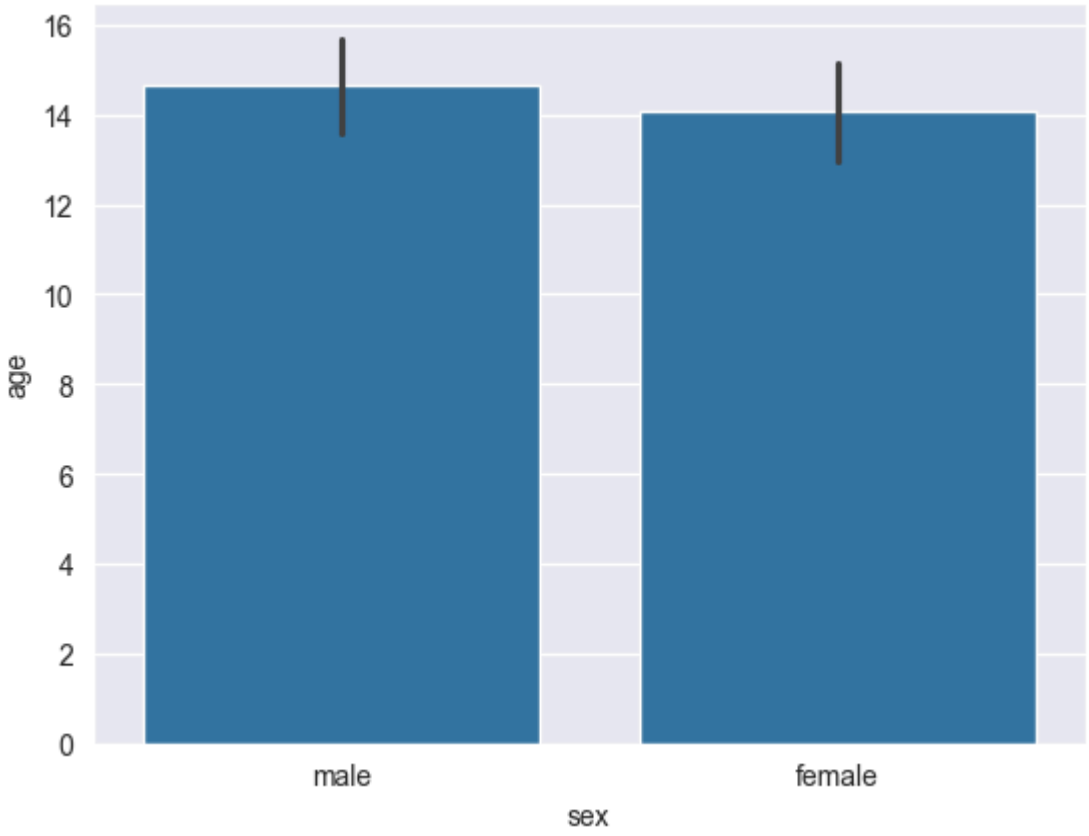
In [9]:
```
sns.barplot(x = 'sex', y = 'age', data = titanic)
```

Out[9]:
```
<Axes: xlabel='sex', ylabel='age'>
```



In [10]:
```
sns.barplot(x = 'sex', y = 'age', data = titanic, estimator = np.std)
```

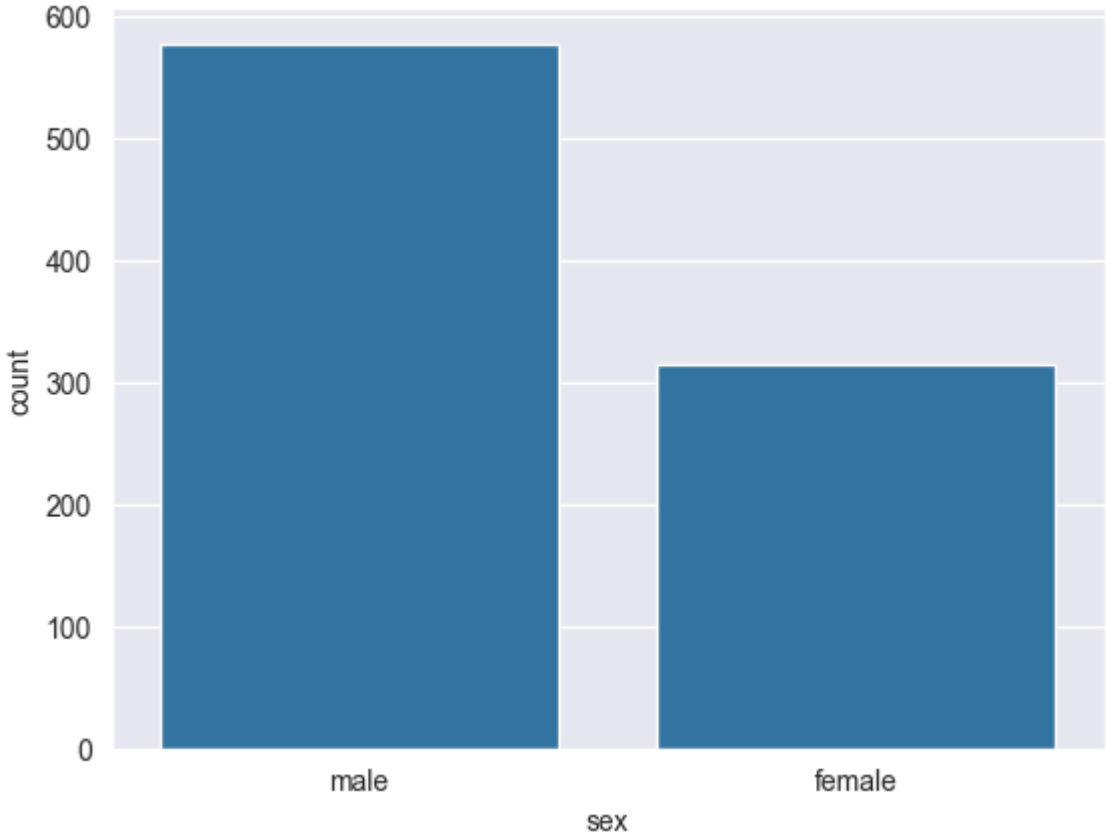
Out[10]:
```
<Axes: xlabel='sex', ylabel='age'>
```



Distribution Plots

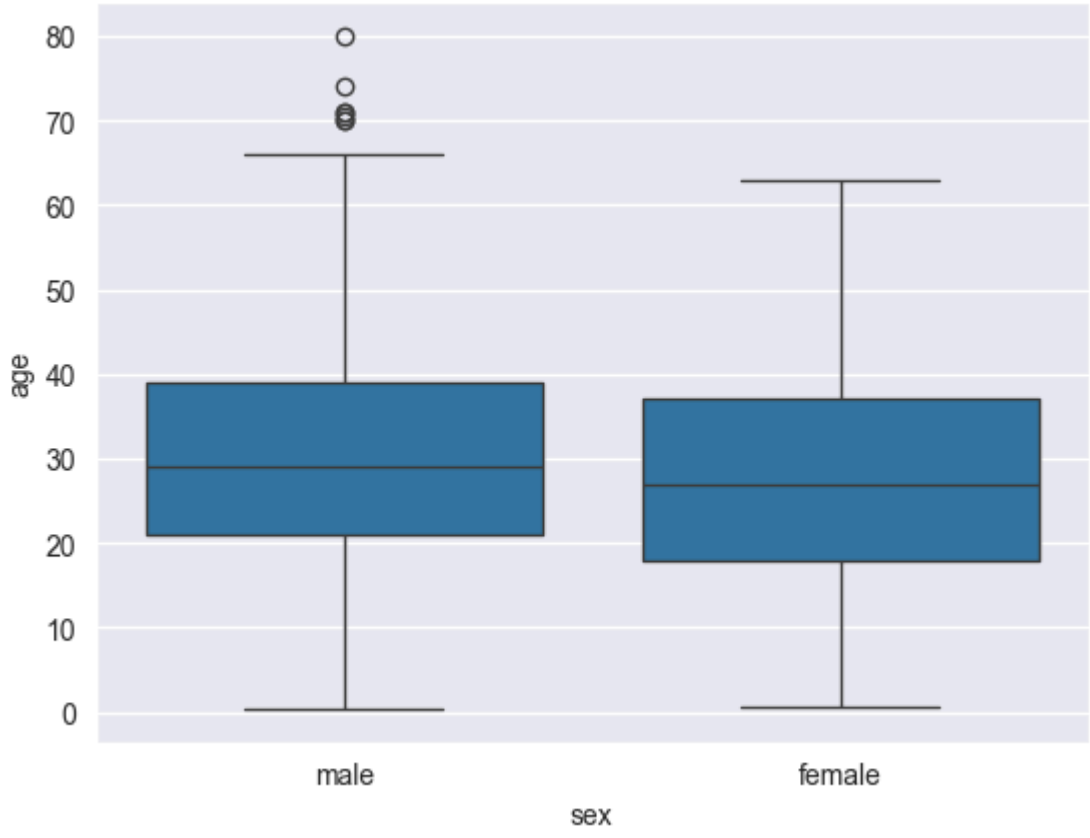
In [11]:
```
sns.countplot(x = 'sex', data = titanic)
```

Out[11]:
```
<Axes: xlabel='sex', ylabel='count'>
```



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

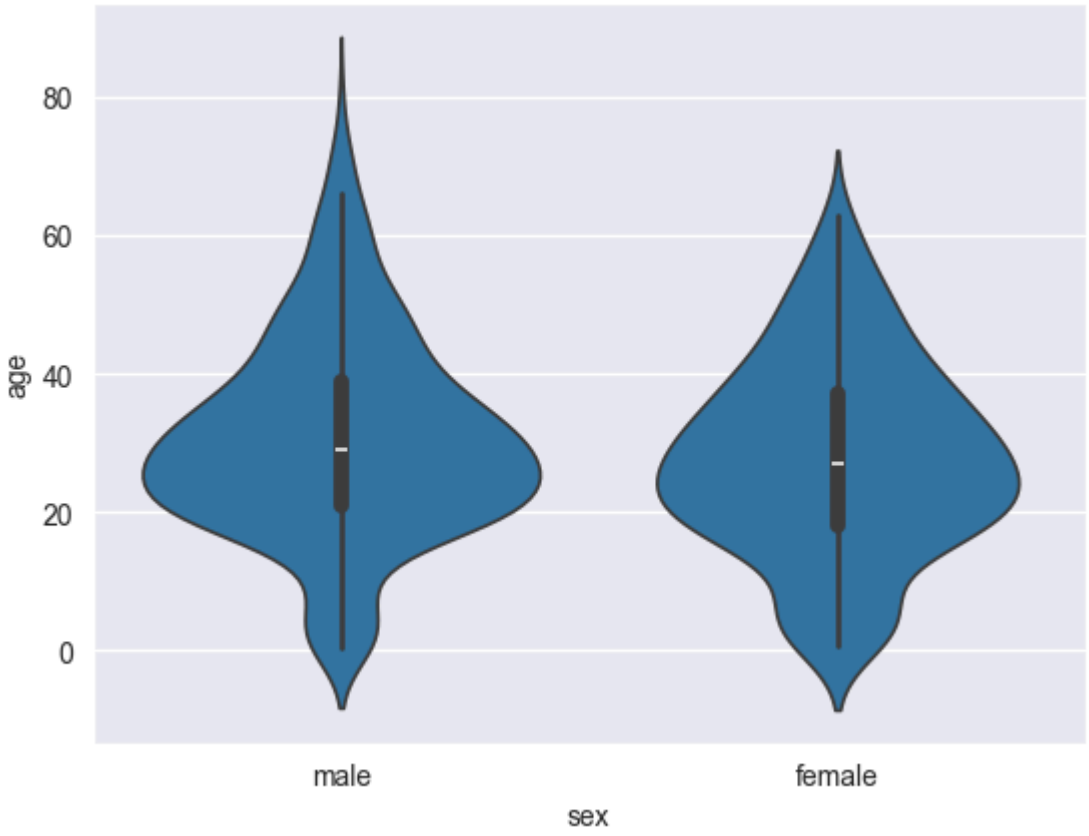
Out[12]: <Axes: xlabel='sex', ylabel='age'>



Violin Plot

In [13]:
```
sns.violinplot(x = 'sex', y = 'age', data = titanic)
```

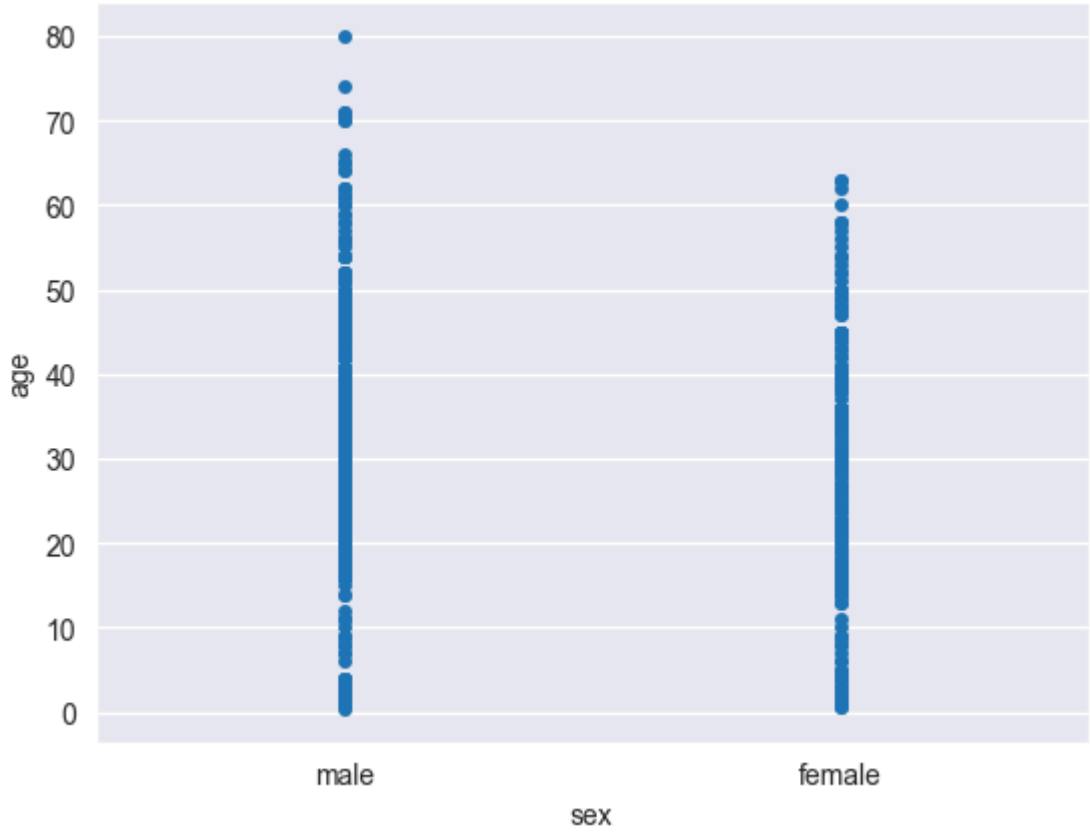
Out[13]:
```
<Axes: xlabel='sex', ylabel='age'>
```



Strip Plot

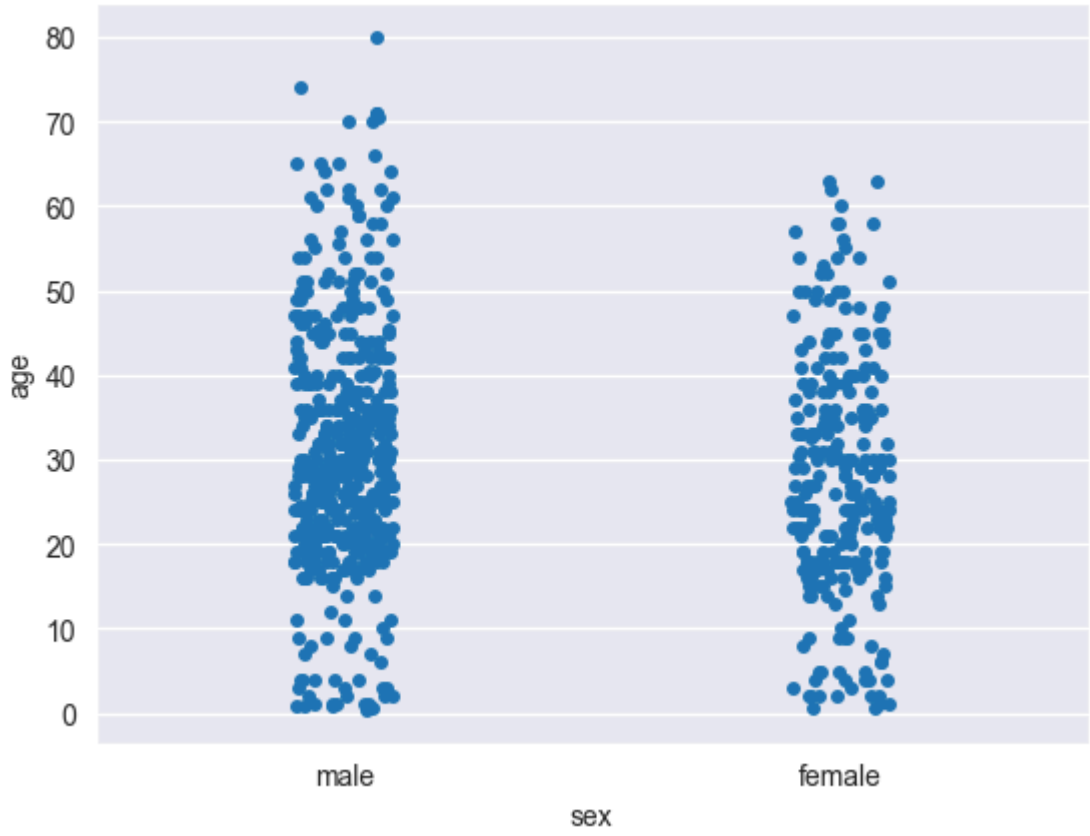
In [14]:
```
sns.stripplot(x = 'sex', y = 'age', data = titanic, jitter=False)
```

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



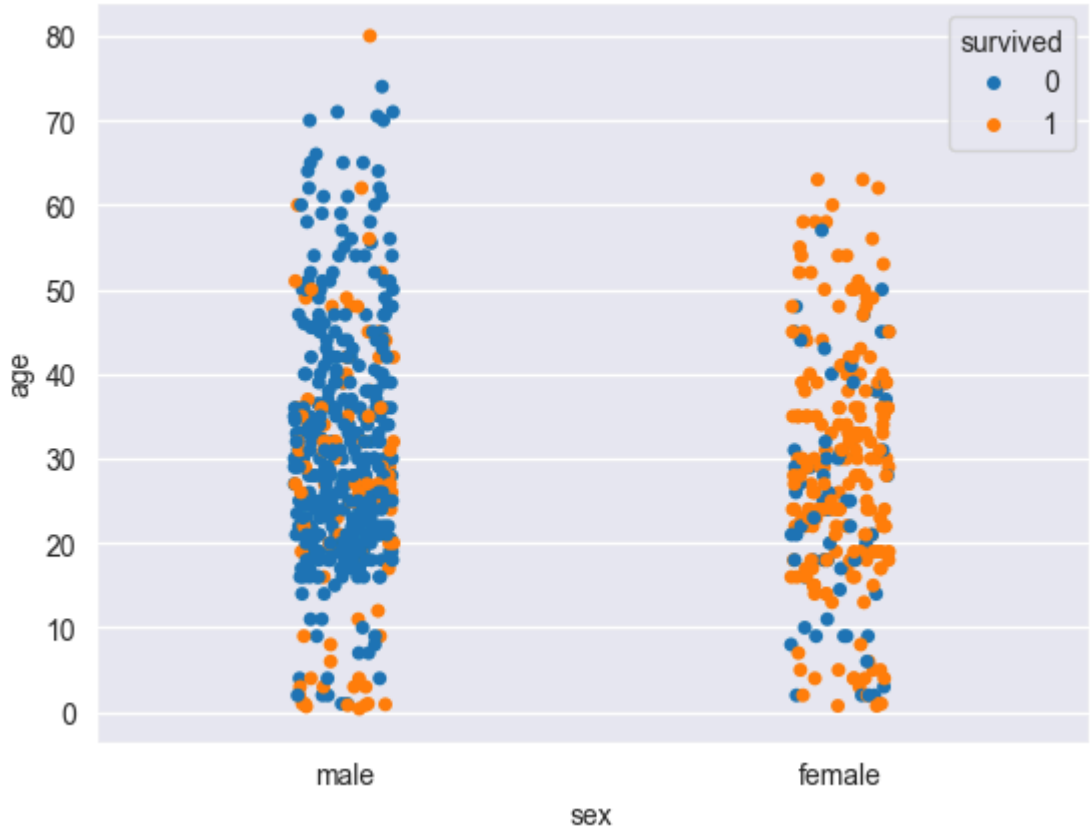
In [15]:
```
sns.stripplot(x = 'sex', y = 'age', data = titanic, jitter=True)
```

Out[15]: <Axes: xlabel='sex', ylabel='age'>



In [16]:
```
sns.stripplot(x = 'sex', y = 'age', data = titanic, jitter=True, hue = 'survived
```

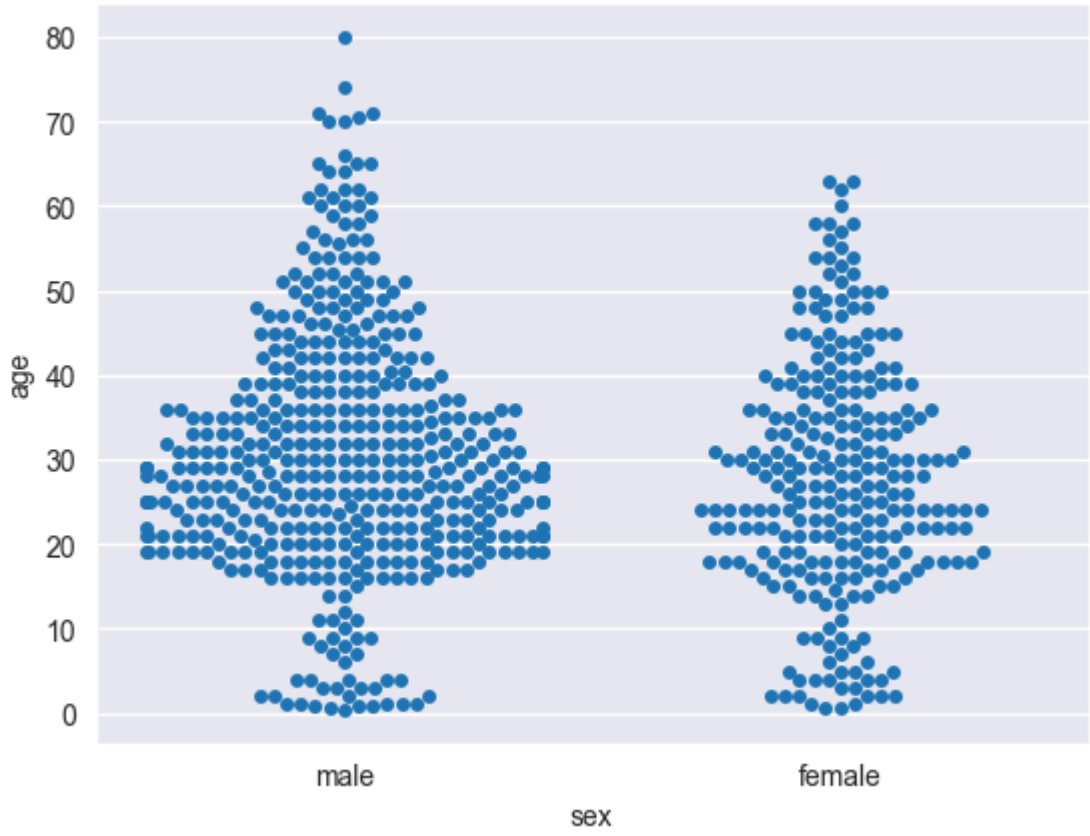
Out[16]: <Axes: xlabel='sex', ylabel='age'>



Swarm Plot

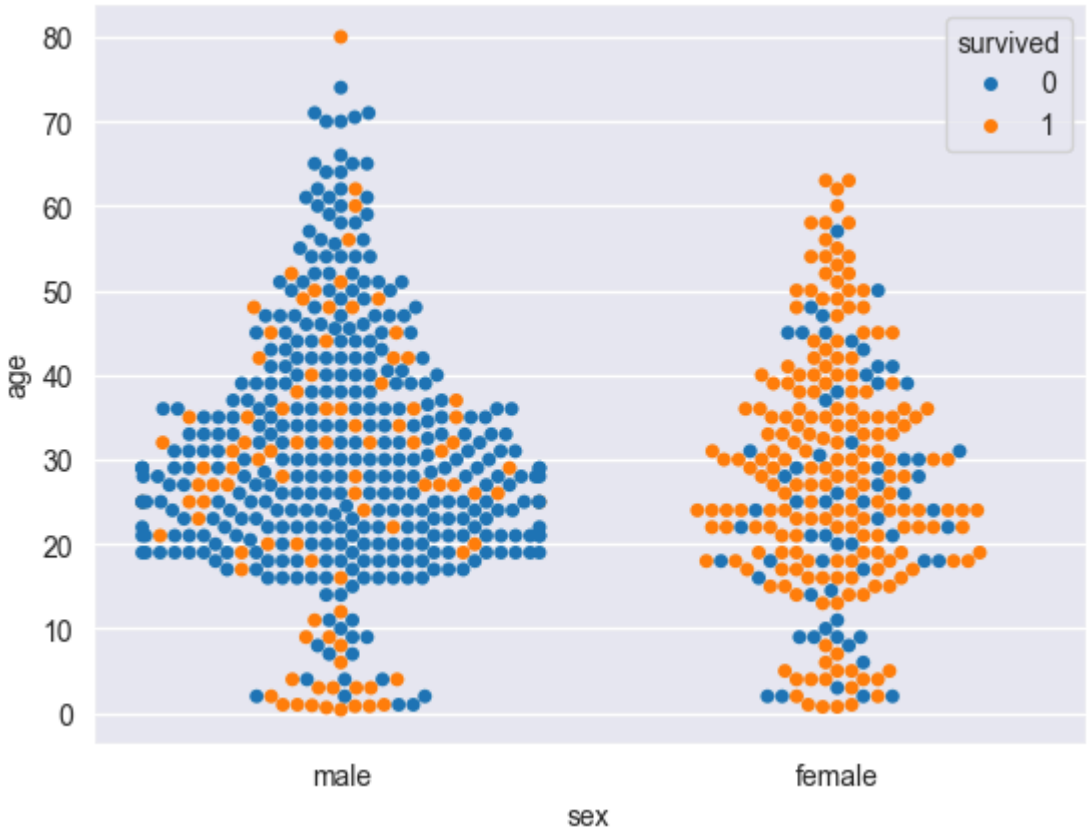
In [17]:
```
sns.swarmplot(x = 'sex', y = 'age', data = titanic)
```

Out[17]: <Axes: xlabel='sex', ylabel='age'>



In [18]:
```
sns.swarmplot(x = 'sex', y = 'age', data = titanic, hue = 'survived')
```

Out[18]:
```
<Axes: xlabel='sex', ylabel='age'>
```



Correlation

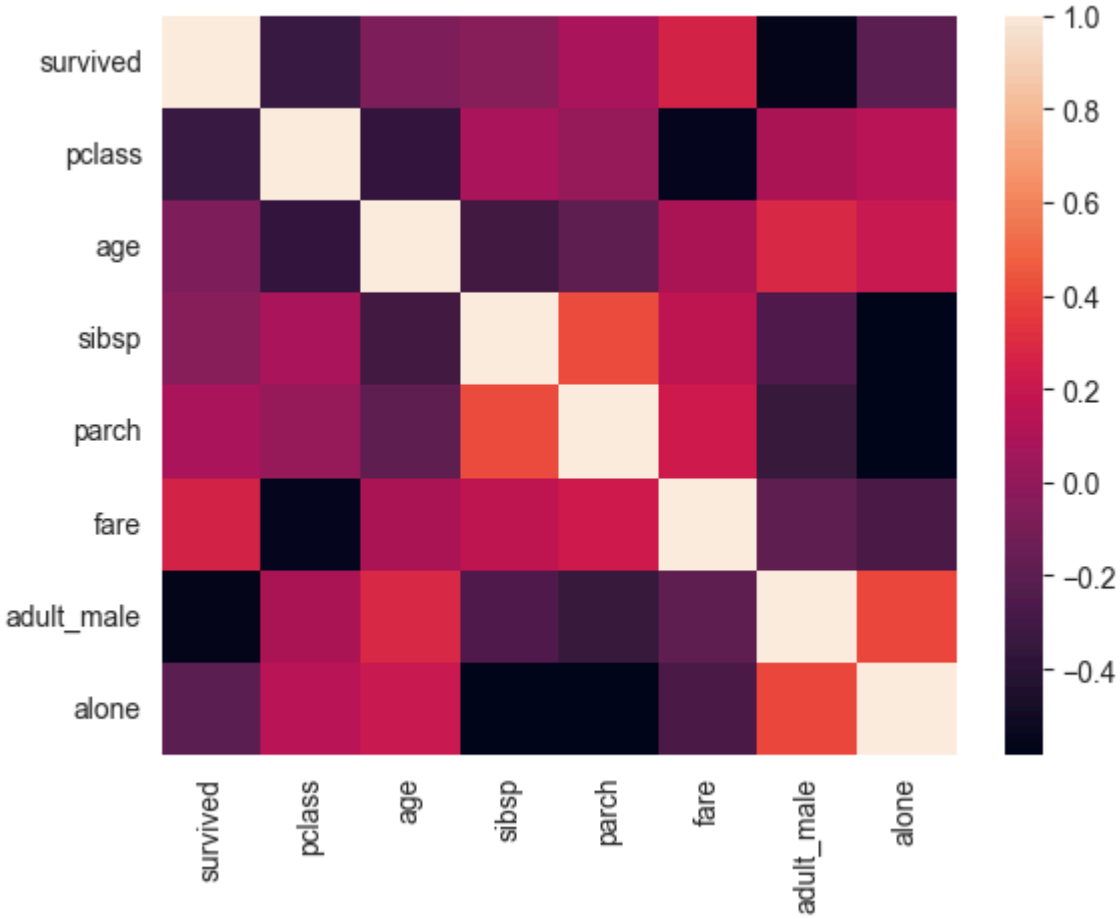
In [19]:
```
titanic.corr(numeric_only= True)
```

Out[19]:

	survived	pclass	age	sibsp	parch	fare	adult_male
survived	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307	-0.557080
pclass	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500	0.094035
age	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067	0.280328
sibsp	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651	-0.253586
parch	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225	-0.349943
fare	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000	-0.182024
adult_male	-0.557080	0.094035	0.280328	-0.253586	-0.349943	-0.182024	1.000000
alone	-0.203367	0.135207	0.198270	-0.584471	-0.583398	-0.271832	0.404744

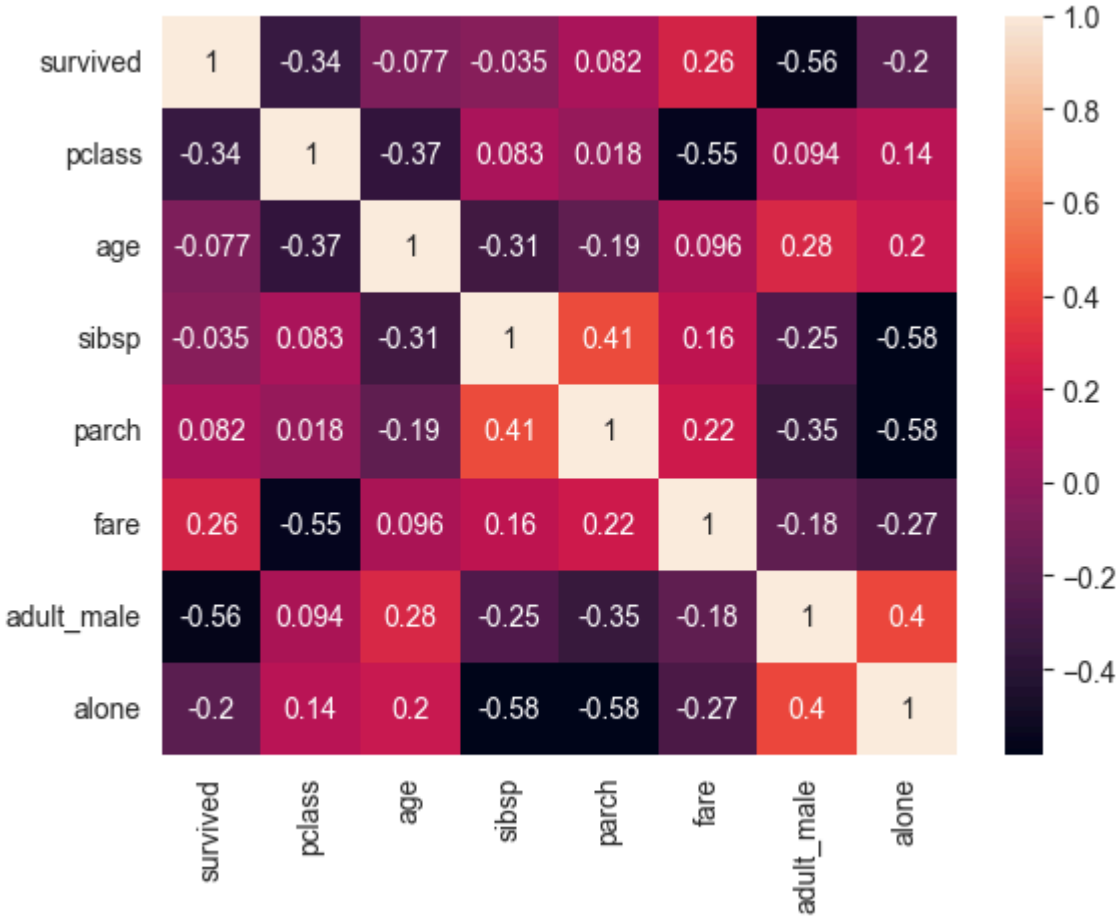
In [20]:
```
sns.heatmap(titanic.corr(numeric_only= True))
```

Out[20]:
```
<Axes: >
```



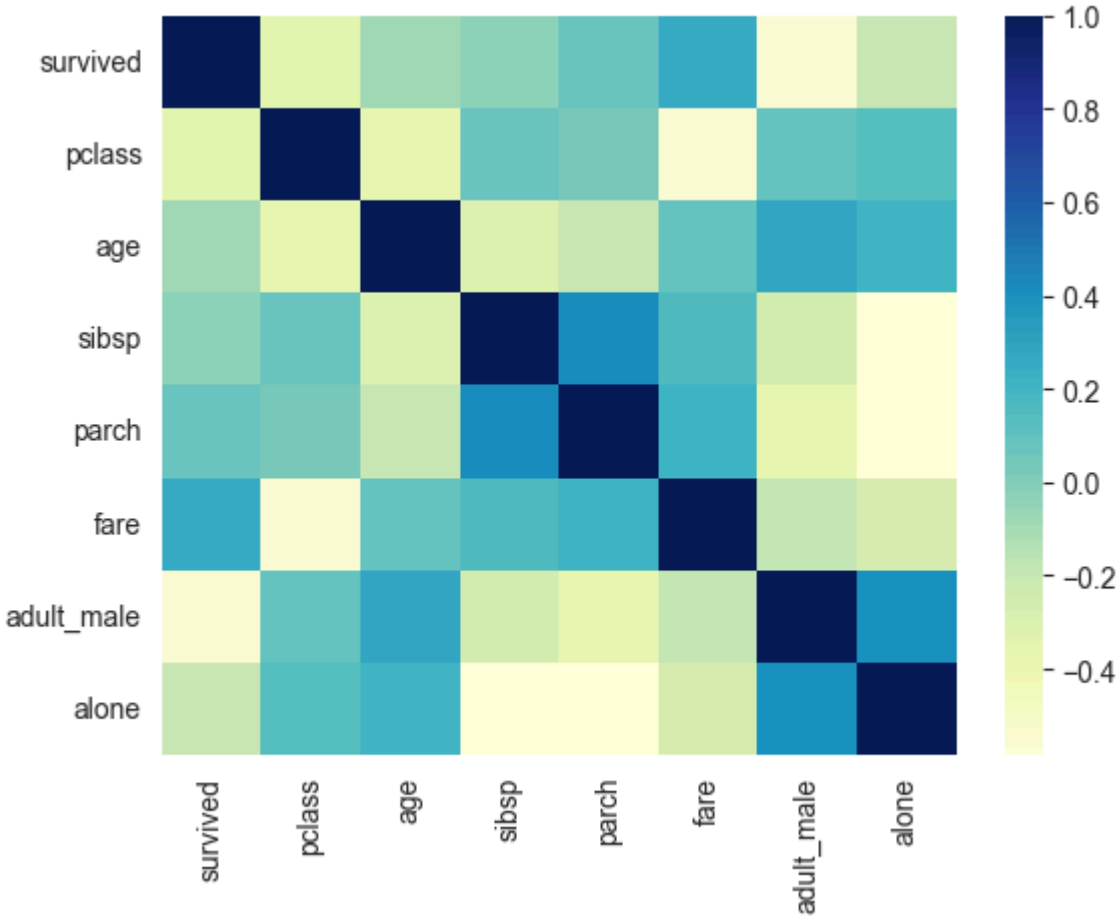
In [21]:
```
sns.heatmap(titanic.corr(numeric_only= True), annot = True)
```

Out[21]: <Axes: >



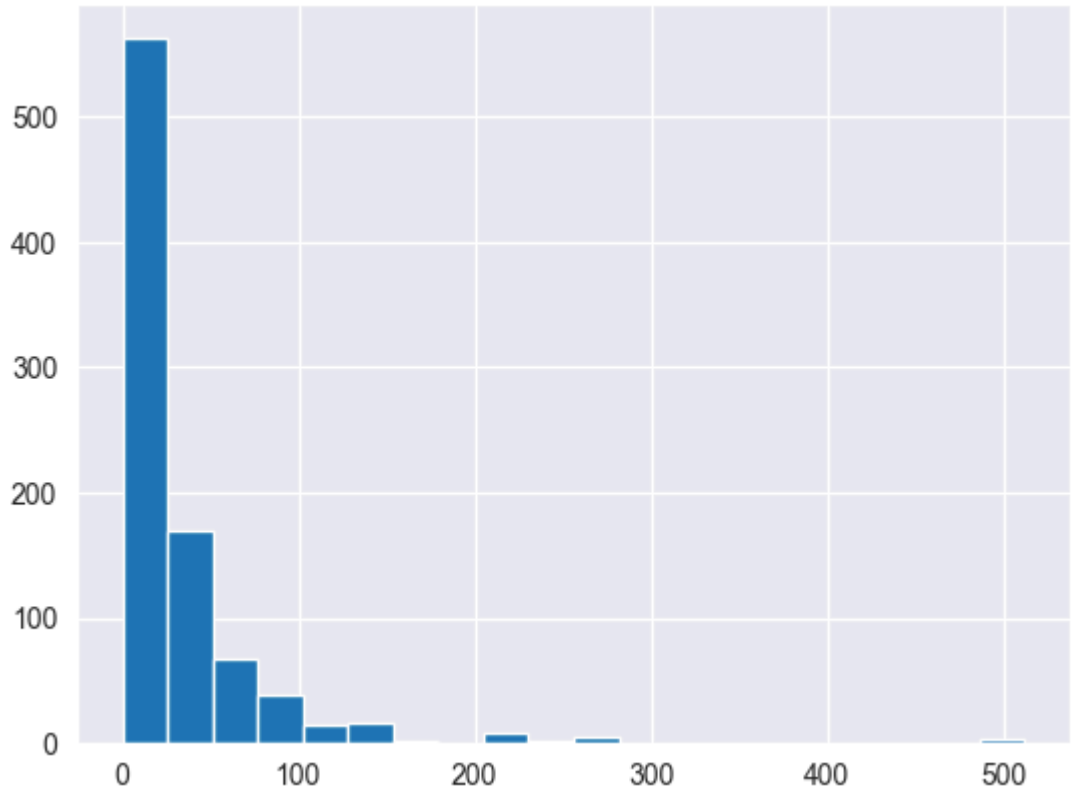
In [22]:
```
sns.heatmap(titanic.corr(numeric_only= True), cmap = 'YlGnBu')
```

Out[22]: <Axes: >



In [23]:
```
plt.hist(x = titanic['fare'], bins = 20)
```

Out[23]:
```
(array([562., 170.,  67.,  39.,  15.,  16.,   2.,   0.,   9.,   2.,   6.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   3.]),
 array([  0.     ,  25.61646,  51.23292,  76.84938, 102.46584, 128.0823 ,
        153.69876, 179.31522, 204.93168, 230.54814, 256.1646 , 281.78106,
        307.39752, 333.01398, 358.63044, 384.2469 , 409.86336, 435.47982,
        461.09628, 486.71274, 512.3292 ]),
 <BarContainer object of 20 artists>)
```


500
400
300
200
100
0
0
100
200
300
400
500