Python

Introduction to Seaborn

In this lecture

- Introduction to Seaborn
- Comparing similar Matplotlib and Seaborn
  - plt.scatter() vs sns.scatter() & sns.regplot()
  - plt.boxplot() vs sns.boxplot()
  - plt.bar() & plt.hist() vs sns.barplot() & sns.countplot()
- Seaborn's unique plotting functions
  - joint_plot() & dist_plot() & pair_plot()
  - swarm_plot()
  - heat_map()
  - facet_grid()

Seaborn

Seaborn

- **Seaborn** is a Python visualisation library based on Matplotlib.
- Seaborn provides functions to create complex **statistical plots** with minimal code. Unlike Matplotlib, where all labels, colour, dimensions, size etc. need to be set.
- Seaborn has functions such as scatterplot(), barplot() and heatmap() which encapsulate common plotting patterns.
- More documentation available at: https://seaborn.pydata.org

Imports

Imports

In[ ]:

```
1 | import numpy as np
2 | import pandas as pd
3 | import matplotlib.pyplot as plt
4 | import seaborn as sns
```

Imports

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

Written in the first cell to reduce duplication across the following cells.

sns.set_style()

set_style

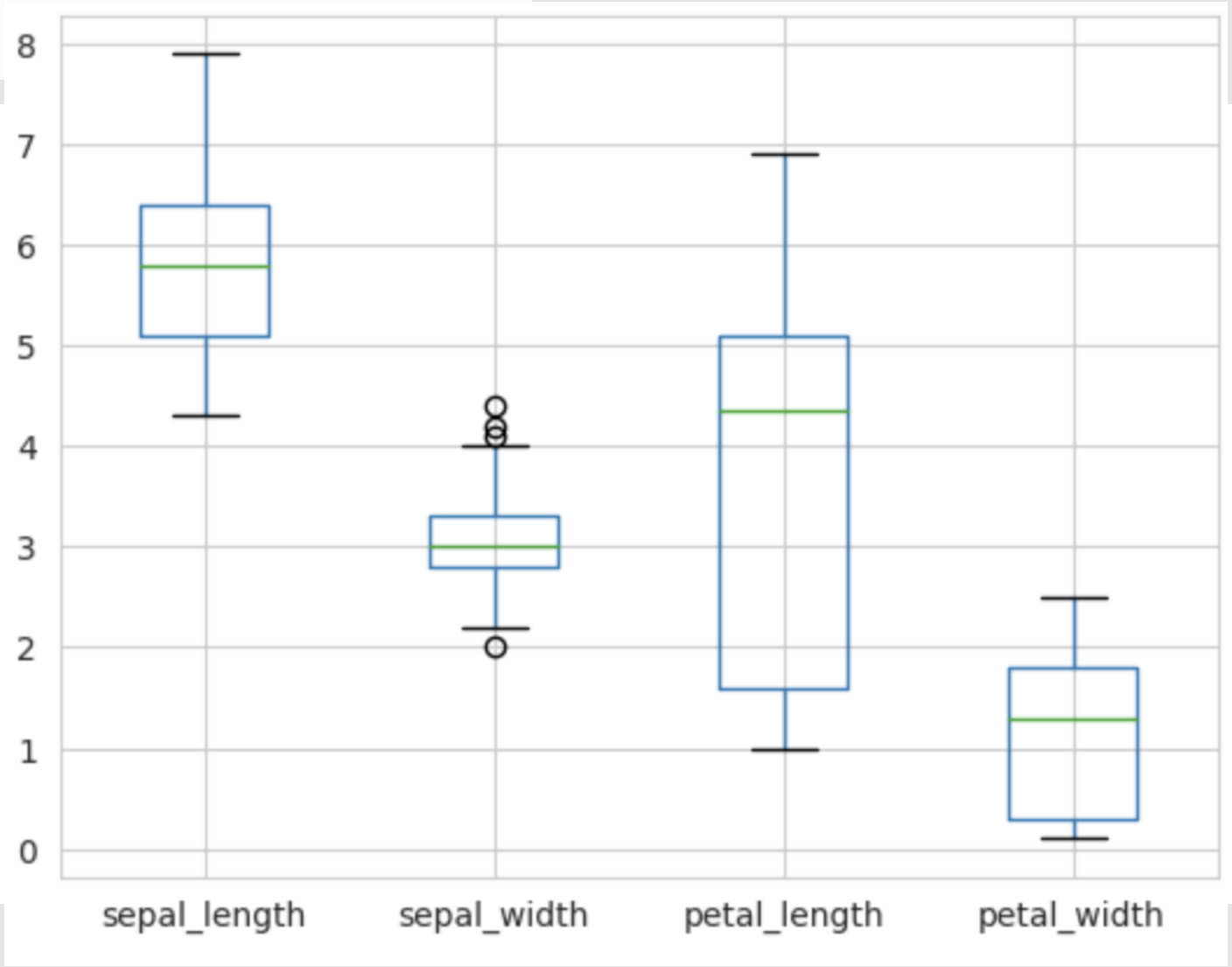
In[ ]:
```
1 | sns.set_style()
2 |
3 |
```

set_style

In[ ]:
```
1 | sns.set_style('whitegrid')
2 |
3 |
```

set_style

Out[]:

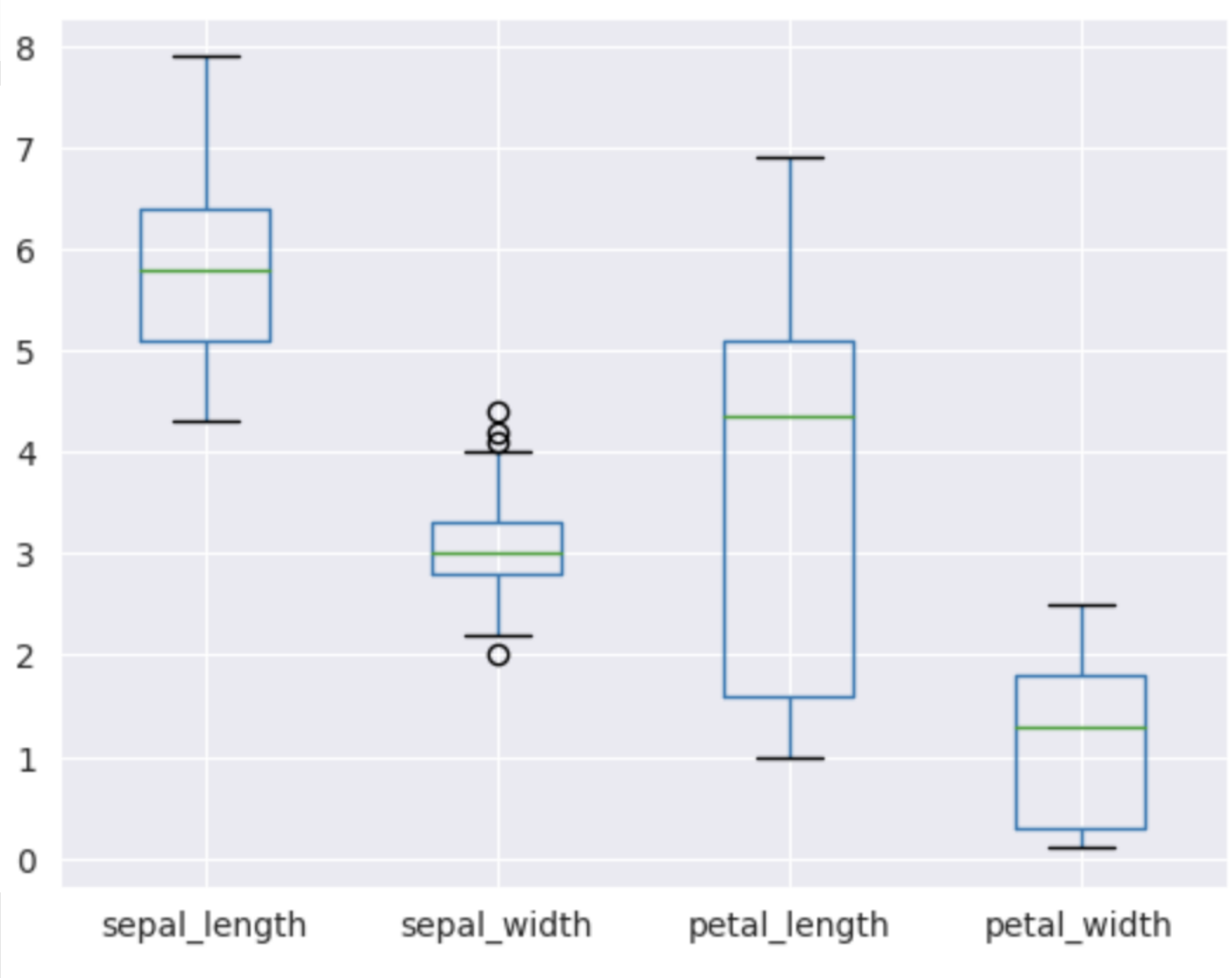


set_style

In[ ]:
```
1 | sns.set_style('darkgrid')
2 |
3 |
```

set_style

Out[]:



sns.load_dataset()

load_dataset()

In[ ]:
```
1 | sns.load_dataset()
2 |
3 |
```

load_dataset()

In[ ]:
```
1 | titanic = sns.load_dataset('titanic')
2 | titanic.head()
3 |
```

load_database()

Out[]:

	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

load_dataset()

In[ ]:
```
1 | iris = sns.load_dataset('iris')
2 | iris.head()
3 |
```

load_database()

Out[]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Mor

anagrams.csv	Rename messy anagrams dataset	4 years ago
anscombe.csv	Add anscombe dataset	10 years ago
attention.csv	Add attention dataset	10 years ago
brain_networks.csv	Add brain networks dataset	10 years ago
car_crashes.csv	Add 538 car crash dataset	9 years ago
dataset_names.txt	Add a file containing all available dataset names	last year
diamonds.csv	Add diamonds dataset	6 years ago
dots.csv	Add dots dataset	7 years ago
dowjones.csv	Add dowjones dataset	2 years ago
exercise.csv	Add exercise dataset	10 years ago
flights.csv	Add flights dataset	10 years ago
fmri.csv	Change sorting of events in fmri data	7 years ago
geyser.csv	Add geyser dataset	4 years ago
glue.csv	Add several new datasets	2 years ago
healthexp.csv	Remove one-off 2021 datapoint from healthexp datas...	2 years ago
iris.csv	Add iris dataset	10 years ago

https://github.com/mwaskom/seaborn-data

Local CSV

In[ ]:
```
1 | df = sns.load_dataset('../Nick/data.csv')
2 | df.head()
3 |
```

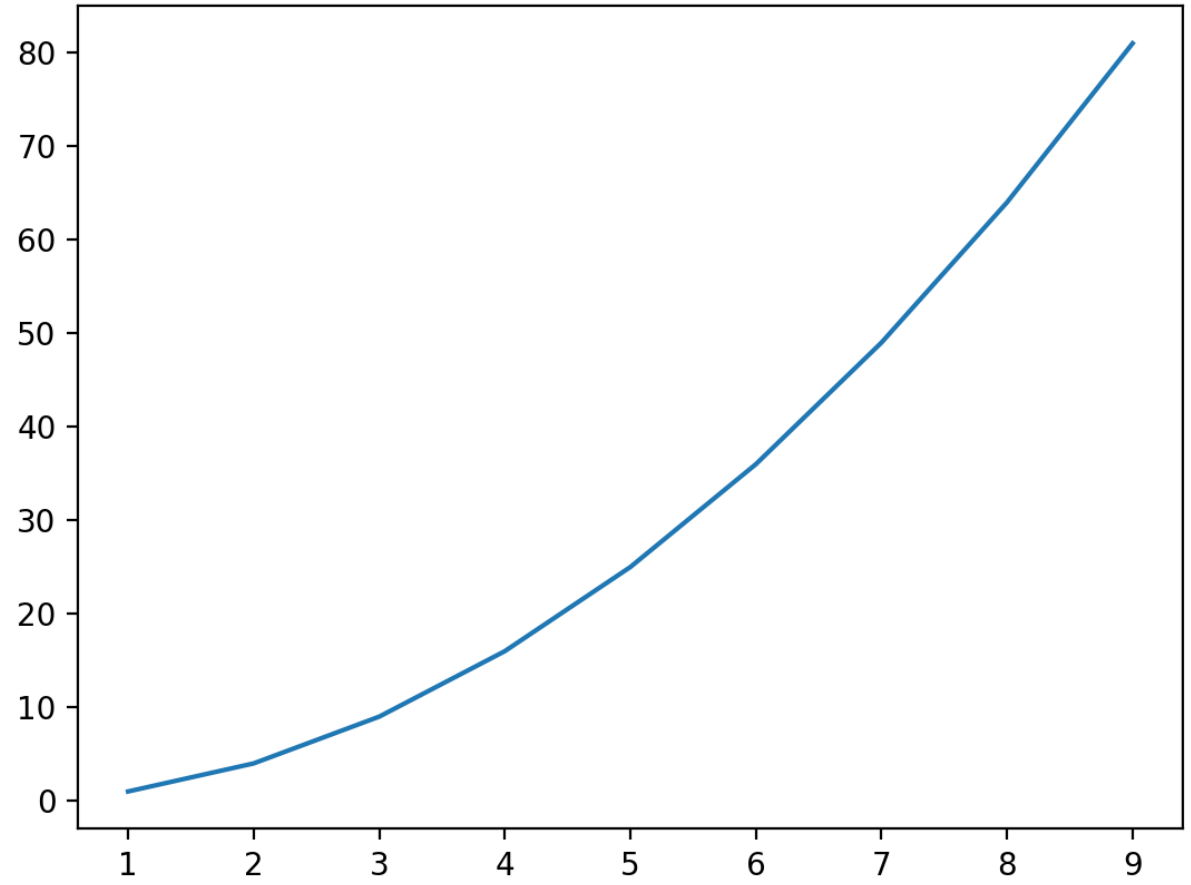
plt.plot()

Basic plt plot

In[ ]:
```
1 | x = np.arange(1, 10, 1)
2 | y = x**2
3 | plt.plot(x, y)
4 | plt.show()
```

Basic plot

Out[]:



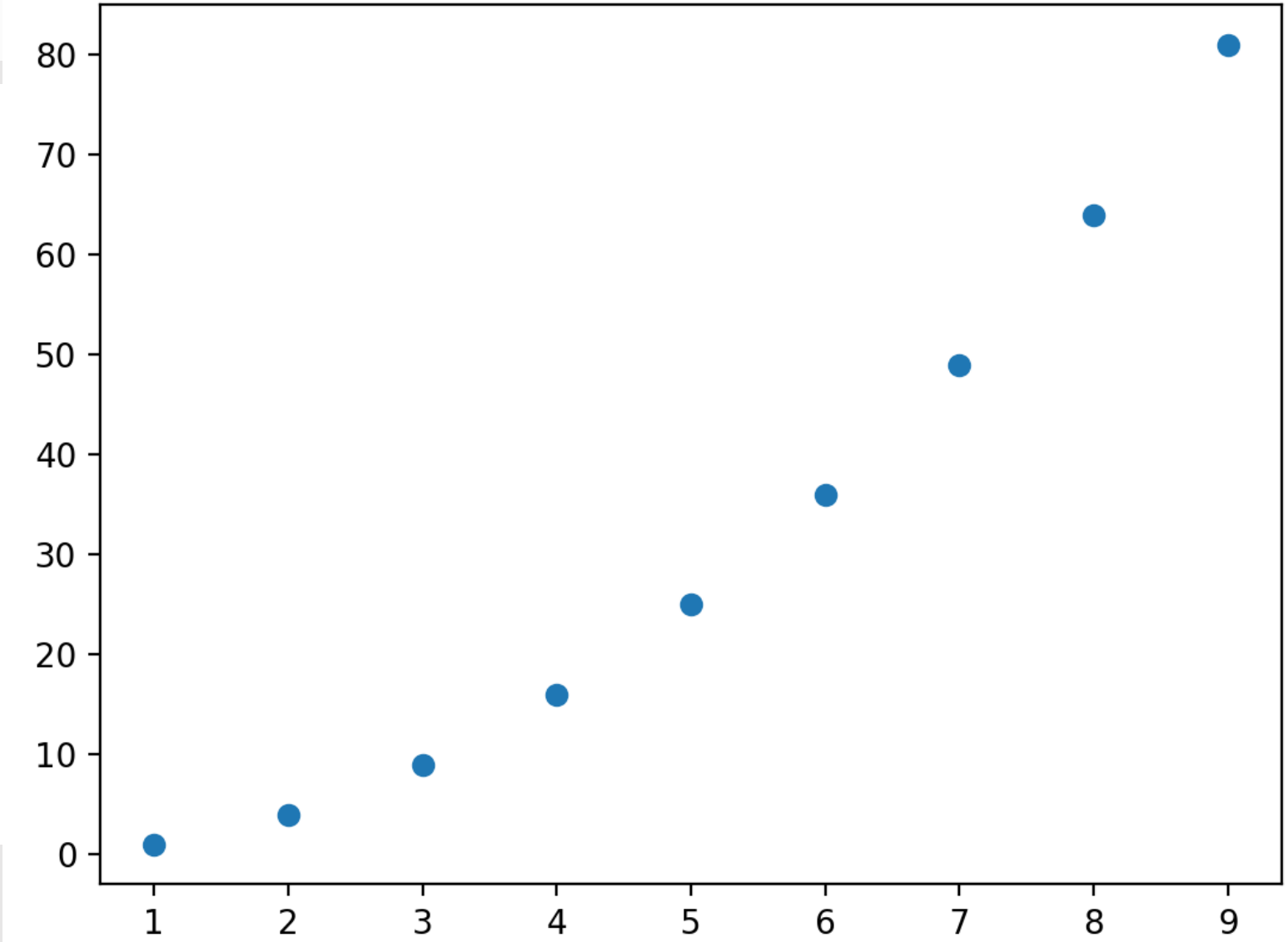
plt.scatter()

Basic Scatter plot

In[ ]:
```
1 | x = np.arange(1, 10, 1)
2 | y = x**2
3 | plt.scatter(x, y)
```

Scatter

Out[]:

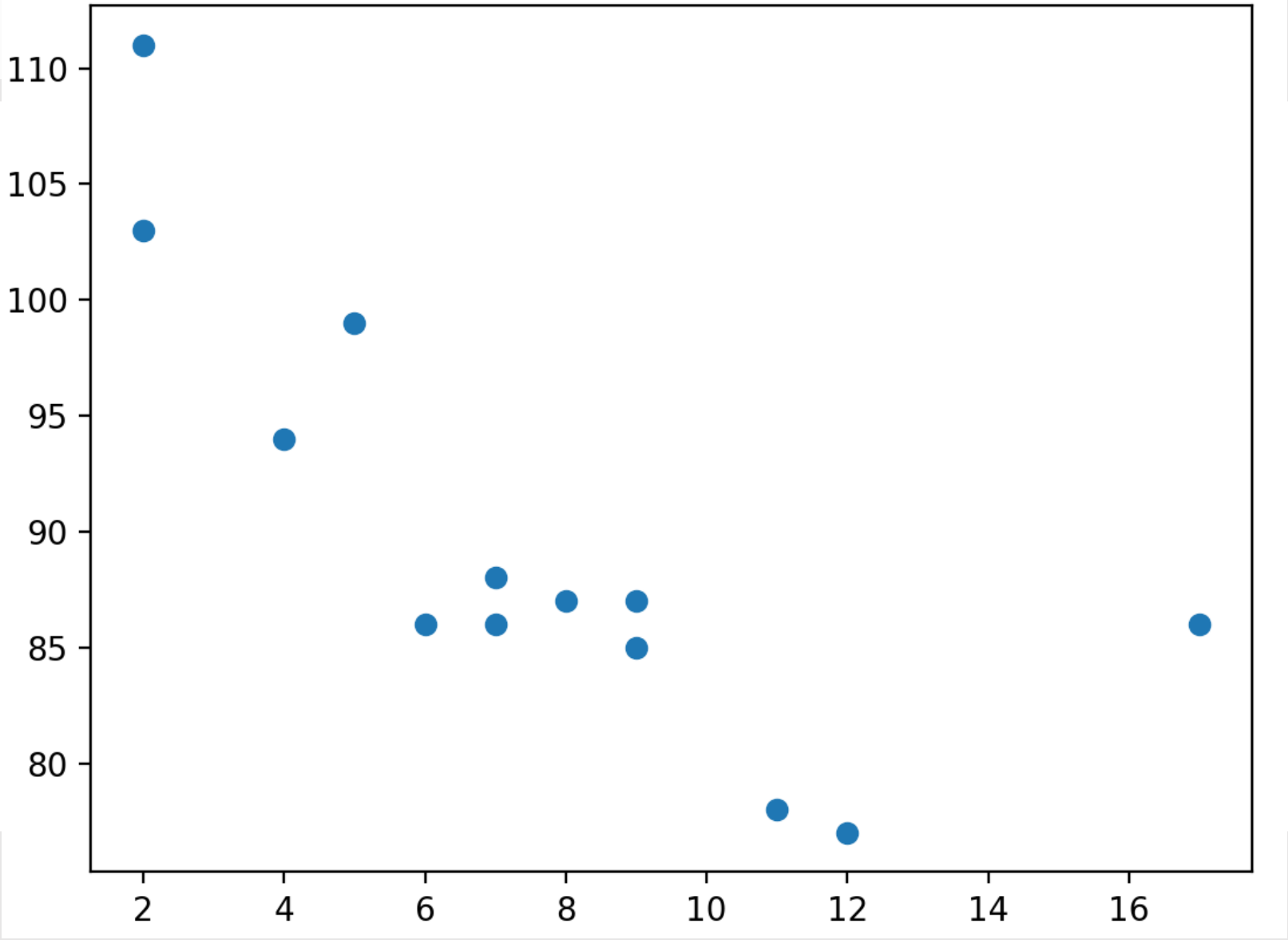


Plot a dataset

In[ ]:
```
1 | x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
2 | y = [99,86,87,88,111,86,103,87,94,78,77]
3 | plt.scatter(x, y)
```

Scatter

Out[]:



Plot two datasets

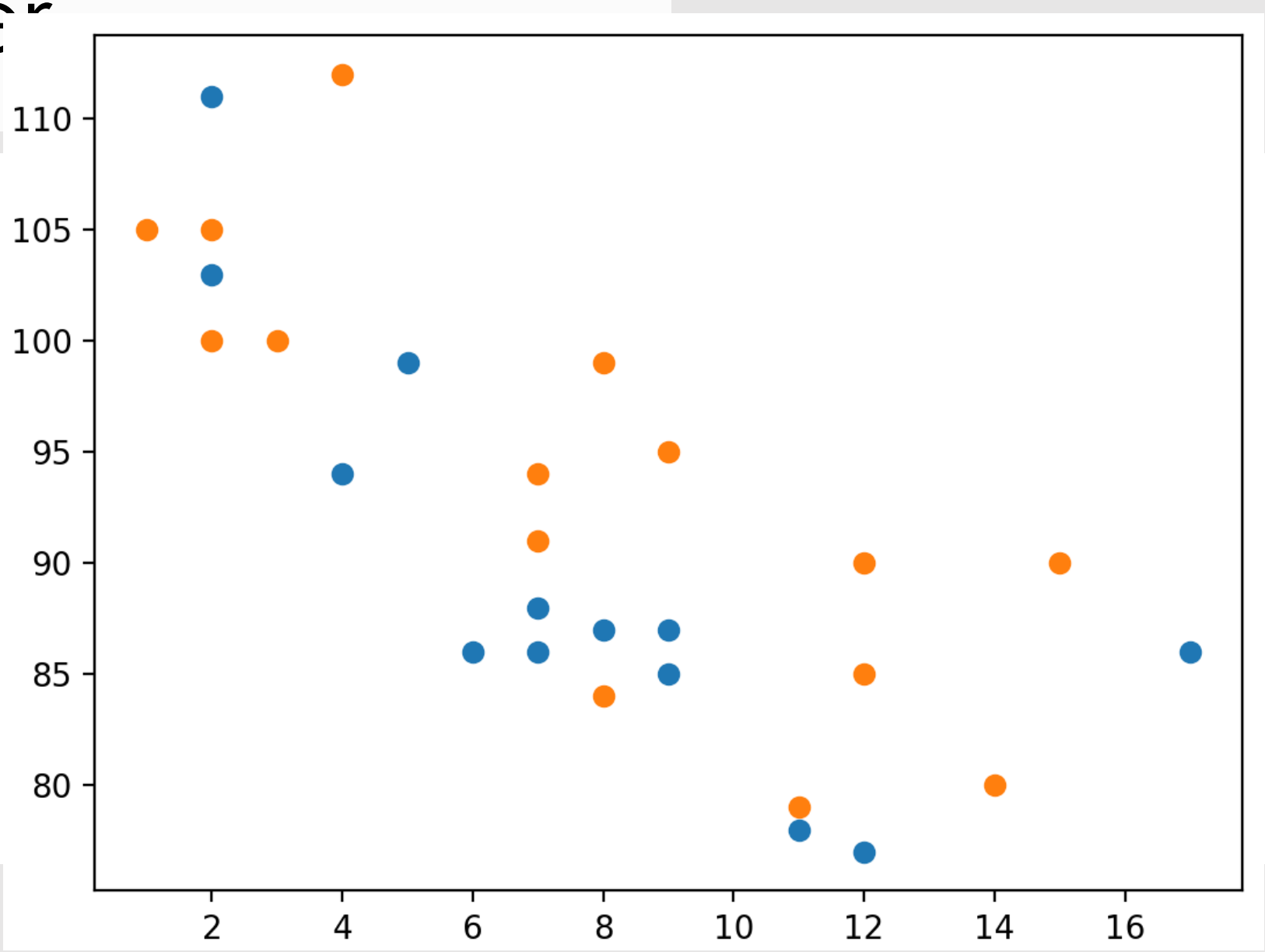
In[ ]:

```
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77]
plt.scatter(x, y)

x = [2,2,8,1,15,8,12,9,7,3,11,4,7]
y = [100,105,84,105,90,99,90,95,94,100]
plt.scatter(x, y)
```

Scatter

Out[]:



Line of Best Fit

In[ ]:

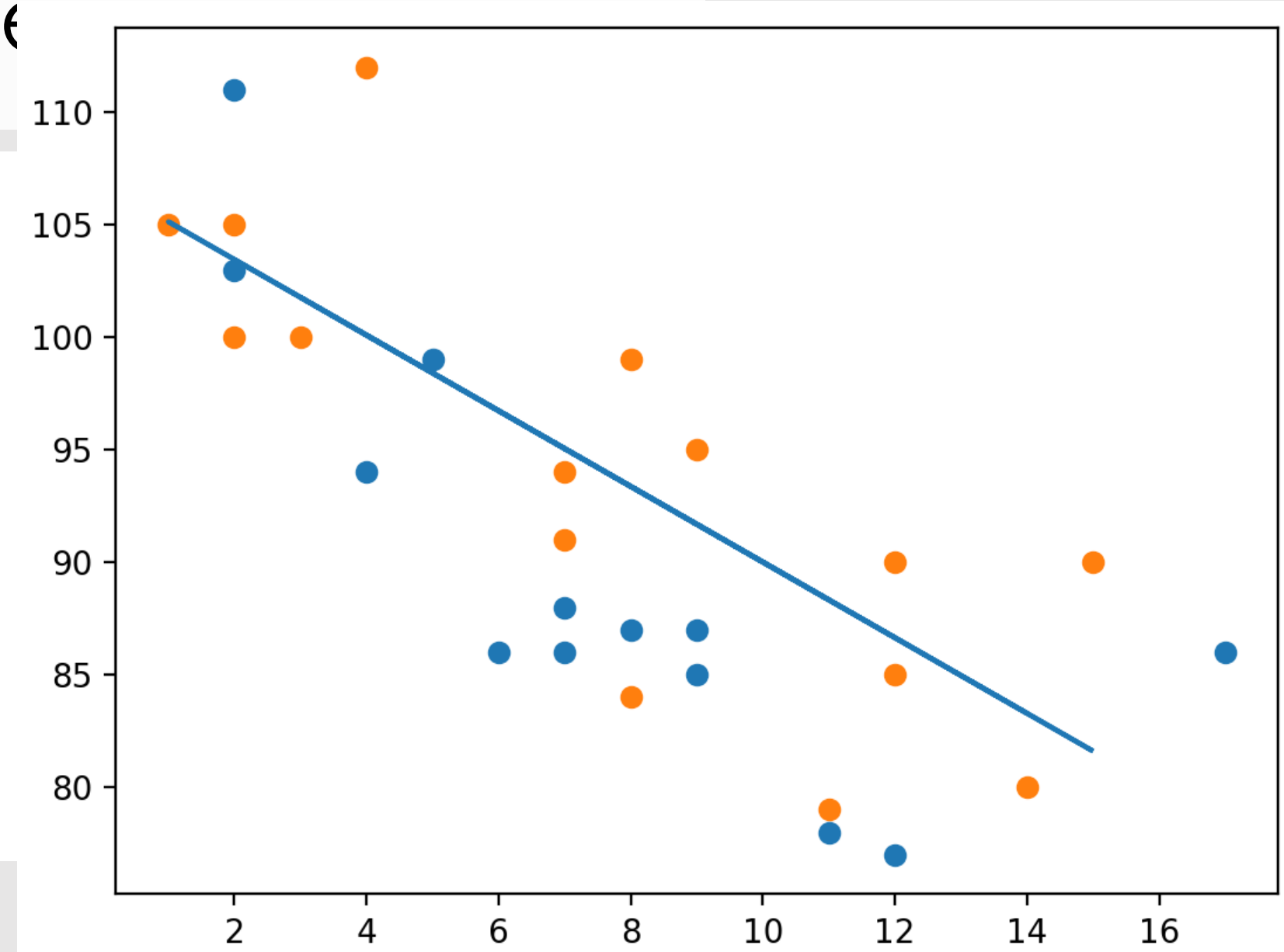
```
1 | a, b = np.polyfit(x, y, 1)
2 | plt.plot(x, a*x+b)
3 |
```

Line of Best Fit

In[ ]:
```
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77]
plt.scatter(x, y)
x = [2,2,8,1,15,8,12,9,7,3,11,4,7]
y = [100,105,84,105,90,99,90,95,94,100]
plt.scatter(x, y)
a, b = np.polyfit(x, y, 1)
plt.plot(x, a*x+b)
```

Scatte

Out[]:



sns.scatterplot()

Basic Scatter plot

In[ ]:
```
1 | x = np.arange(1, 10, 1)
2 | y = x**2
3 | sns.scatterplot(x, y)
```

Basic Scatter plot

In[ ]:
```
1 | x = np.arange(1, 10, 1)
2 | y = x**2
3 | sns.scatterplot(x, y)
```

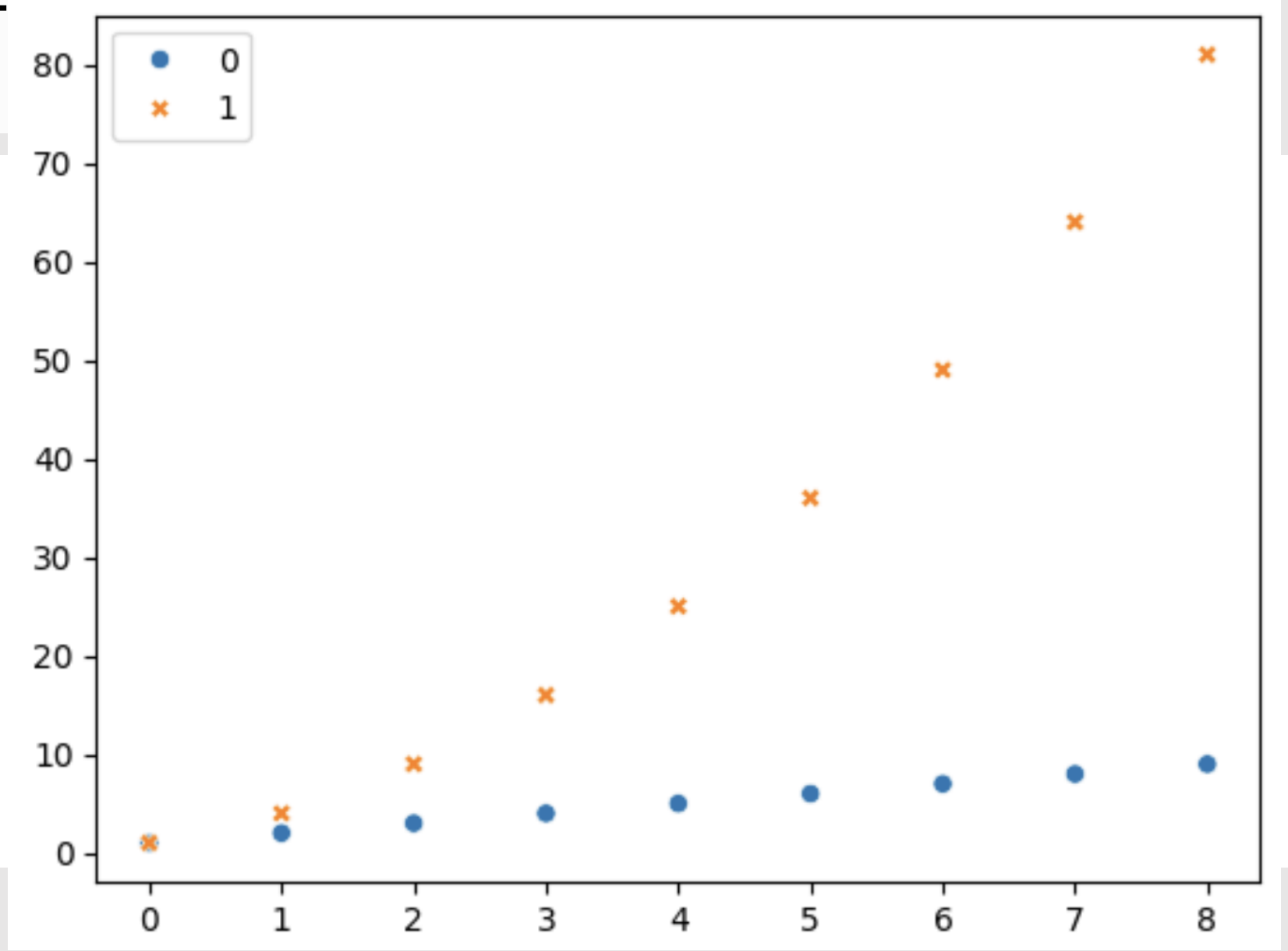
Out[]:
```
---------------------------------------------------------------
TypeError: scatterplot() takes from 0 to 1 positional arguments
but 2 were given
---------------------------------------------------------------
```

Basic Scatter plot

In[ ]:
```
1 | x = np.arange(1, 10, 1)
2 | y = x**2
3 | sns.scatterplot(data = (x, y))
```

Scatt

```
Out[]:
```

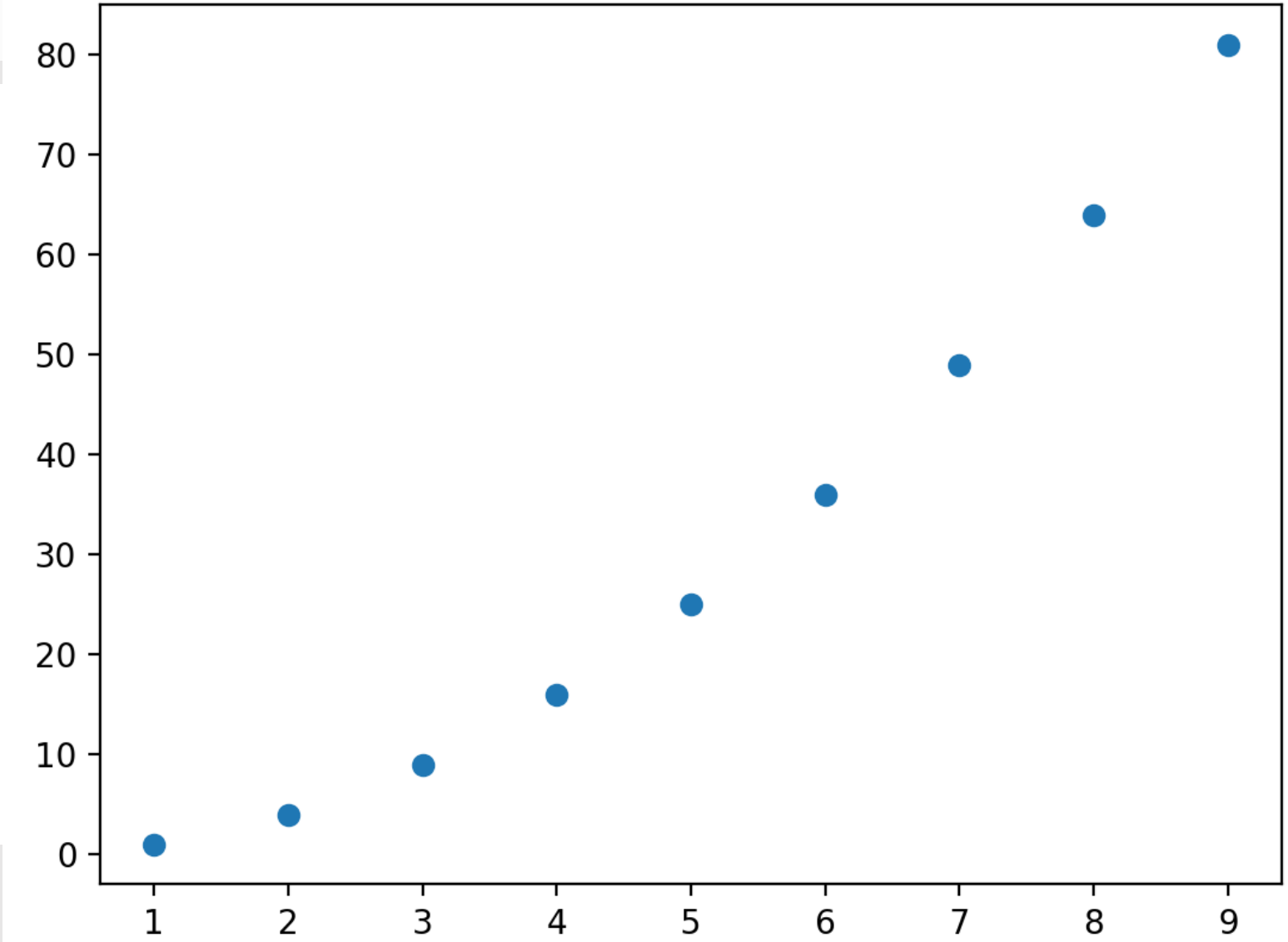


Basic Scatter plot

In[ ]:
```
1 | x = np.arange(1, 10, 1)
2 | y = x**2
3 | sns.scatterplot(x=x, y=y)
```

Scatter

Out[]:



Multiple datasets

In[ ]:

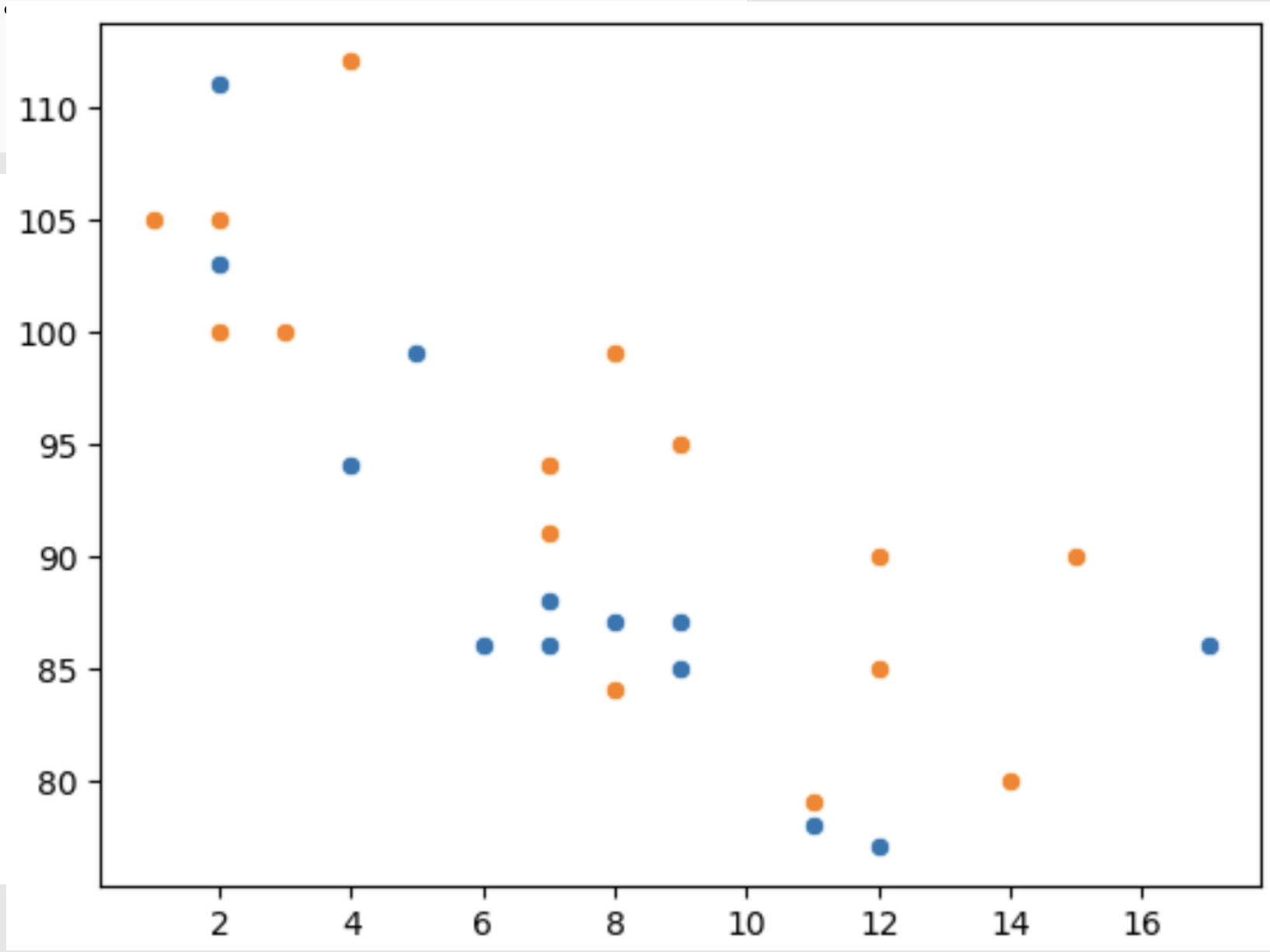
```
x1 = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y1 = [99,86,87,88,111,86,103,87,94,78]

x2 = [2,2,8,1,15,8,12,9,7,3,11,4,7]
y2 = [100,105,84,105,90,99,90,95,94,100]

sns.scatterplot(x=x1, y=y1)
sns.scatterplot(x=x2, y=y2)
```

Mult

Out[]



sns.regplot()

Multiple datasets

In[ ]:

```
x1 = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y1 = [99,86,87,88,111,86,103,87,94,78]

x2 = [2,2,8,1,15,8,12,9,7,3,11,4,7]
y2 = [100,105,84,105,90,99,90,95,94,100]

sns.scatterplot(x=x1, y=y1)
sns.scatterplot(x=x2, y=y2)
```

Multiple datasets

In[ ]:

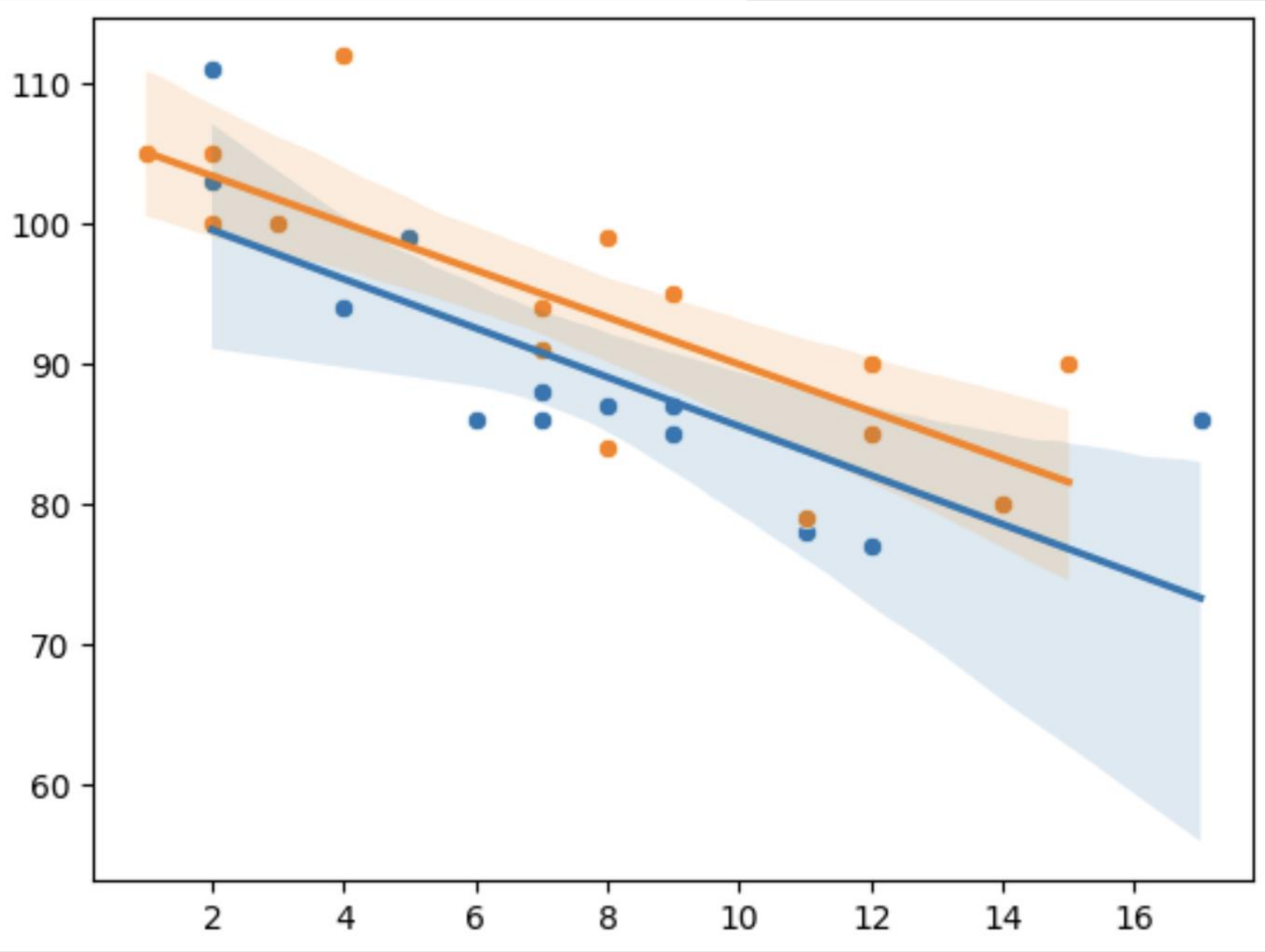
```
...
sns.scatterplot(x=x1, y=y1)
sns.scatterplot(x=x2, y=y2)

sns.regplot(x=x1, y=y1, scatter=False)
sns.regplot(x=x2, y=y2, scatter=False)

```

Mult

Out[]



Multiple datasets

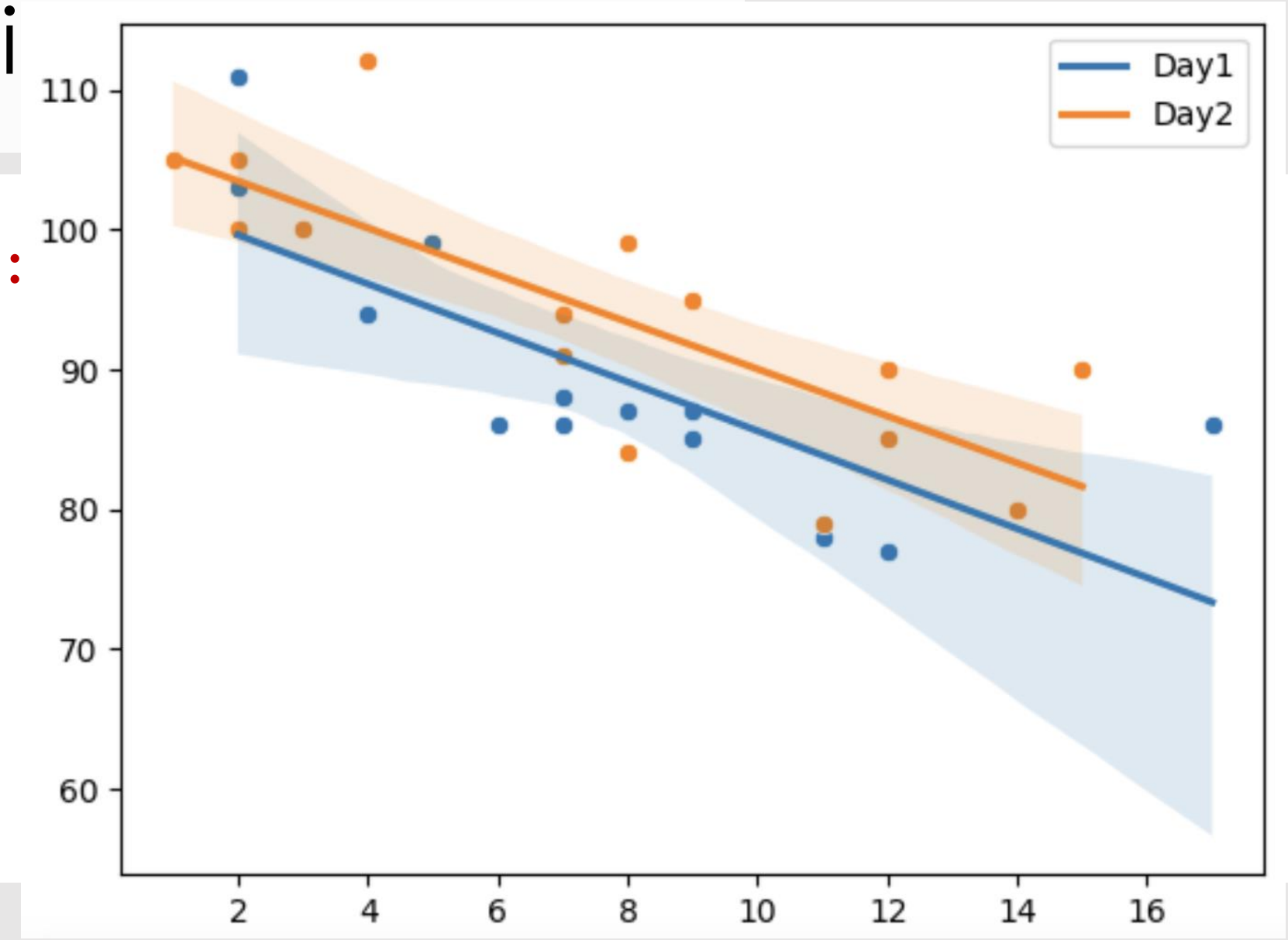
In[ ]:
```
...
sns.scatterplot(x=x1, y=y1)
sns.scatterplot(x=x2, y=y2)

sns.regplot(x=x1, y=y1, scatter=False)
sns.regplot(x=x2, y=y2, scatter=False)

plt.legend()
```

Multi

Out[]:



scatter = false

In[ ]:

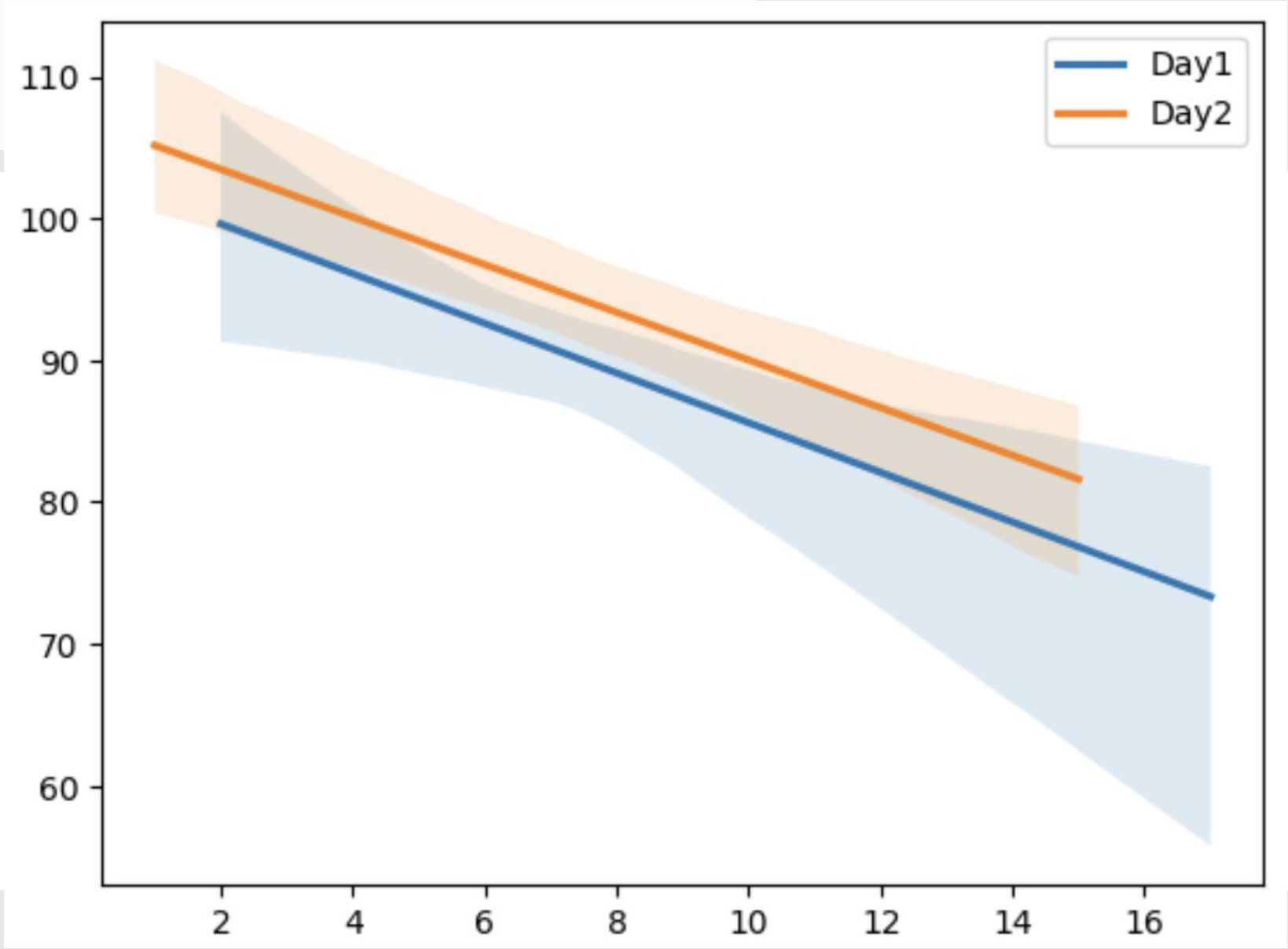
```
x1 = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y1 = [99,86,87,88,111,86,103,87,94,78]

x2 = [2,2,8,1,15,8,12,9,7,3,11,4,7]
y2 = [100,105,84,105,90,99,90,95,94,100]

sns.regplot(..., scatter=False, label…)
sns.regplot(..., scatter=False, label…)
plt.legend()
```

scat

Out[]



scatter = true

In[ ]:

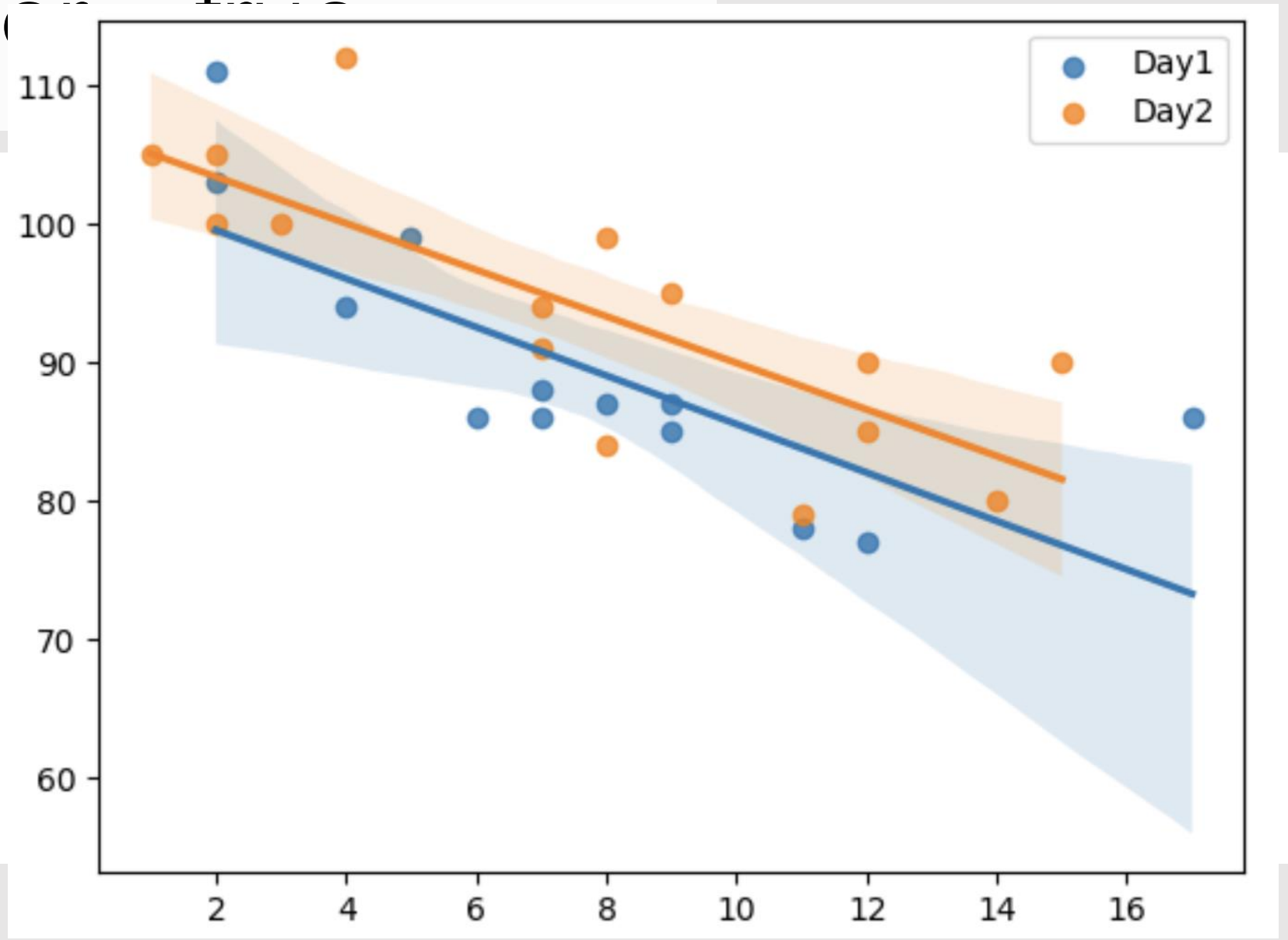
```
x1 = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y1 = [99,86,87,88,111,86,103,87,94,78]

x2 = [2,2,8,1,15,8,12,9,7,3,11,4,7]
y2 = [100,105,84,105,90,99,90,95,94,100]

sns.regplot(x=x1,y=y1,scatter=True, label…)
sns.regplot(x=x2,y=y2,scatter=True, label…)
plt.legend()
```

scatt

Out[]:



plt.boxplot()

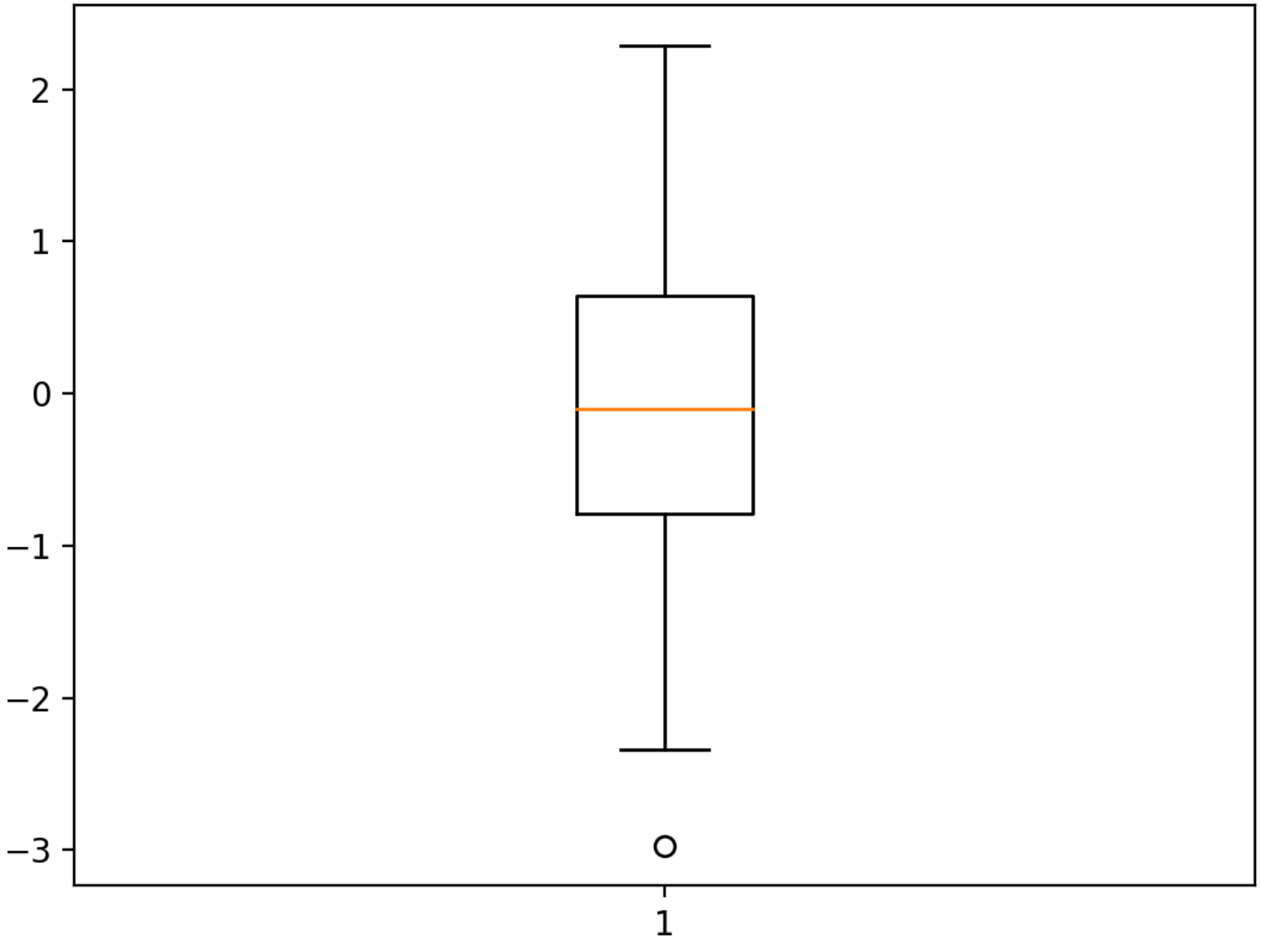
Box Plot

In[ ]:

```
1 | std = 1
2 | data = [np.random.normal(0, std, 100)]
3 | plt.boxplot(data, vert = True)
```

Box Plot

Out[]:



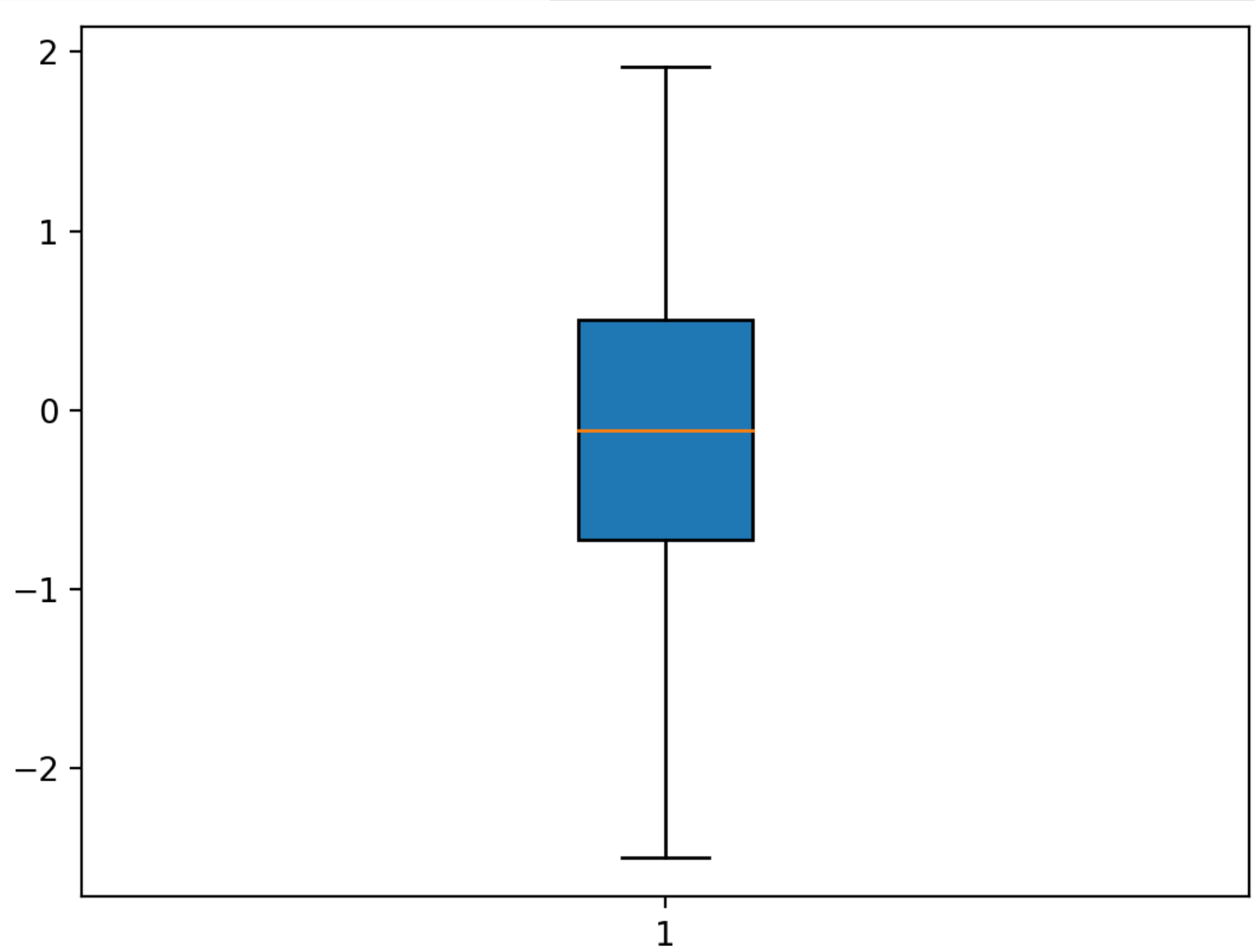
Box Plot

In[ ]:

```
1 | std = 1
2 | data = [np.random.normal(0, std, 100)]
3 | plt.boxplot(data,vert=True,patch_artist=True)
```

Box Plot

Out[]:



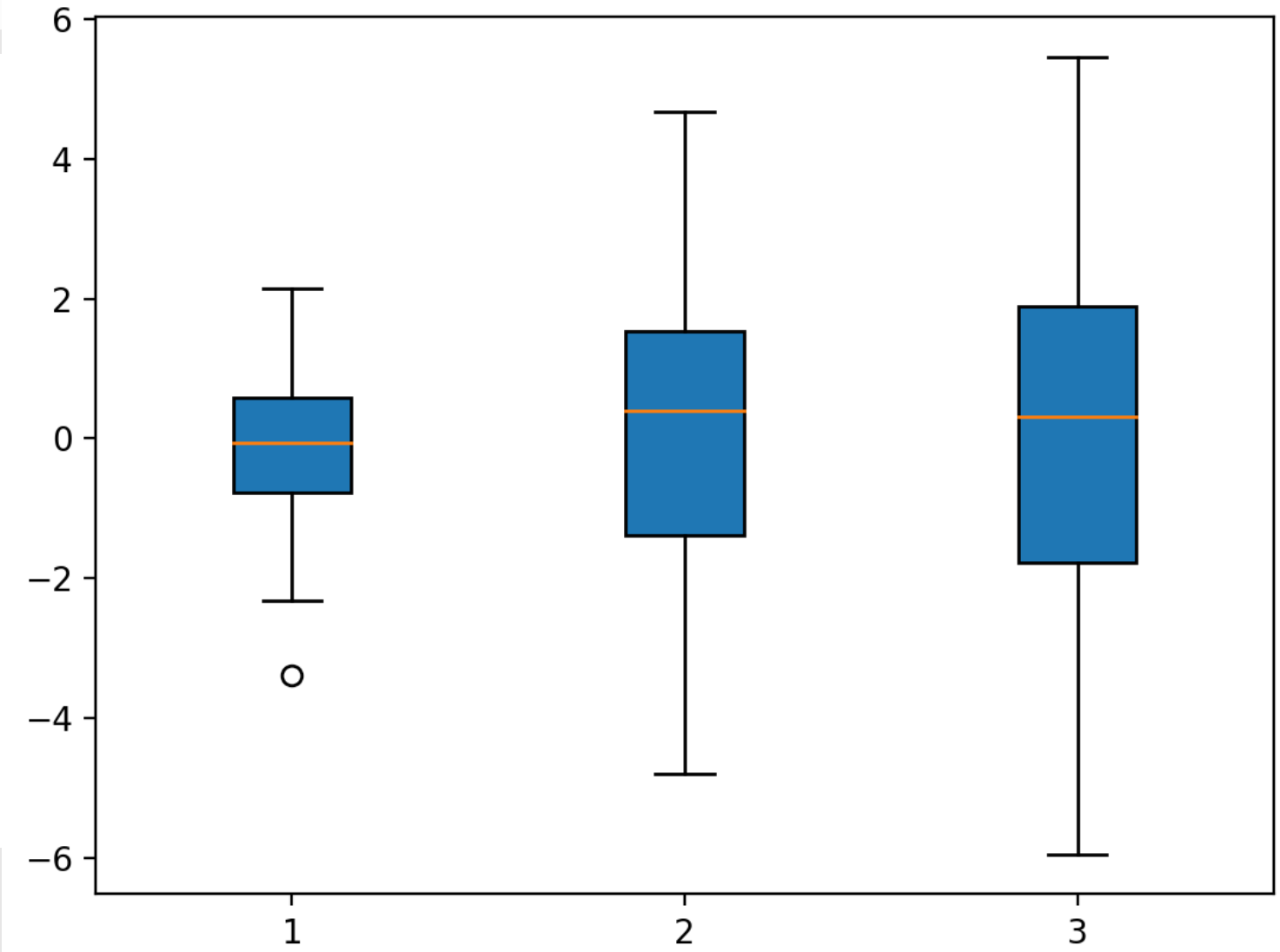
Box Plot

In[ ]:

```
std = 1
data = [np.random.normal(0, std, 100)
        for std in range(1,4)]
plt.boxplot(data,vert=True,patch_artist=True)
```

Box Plot

Out[]:



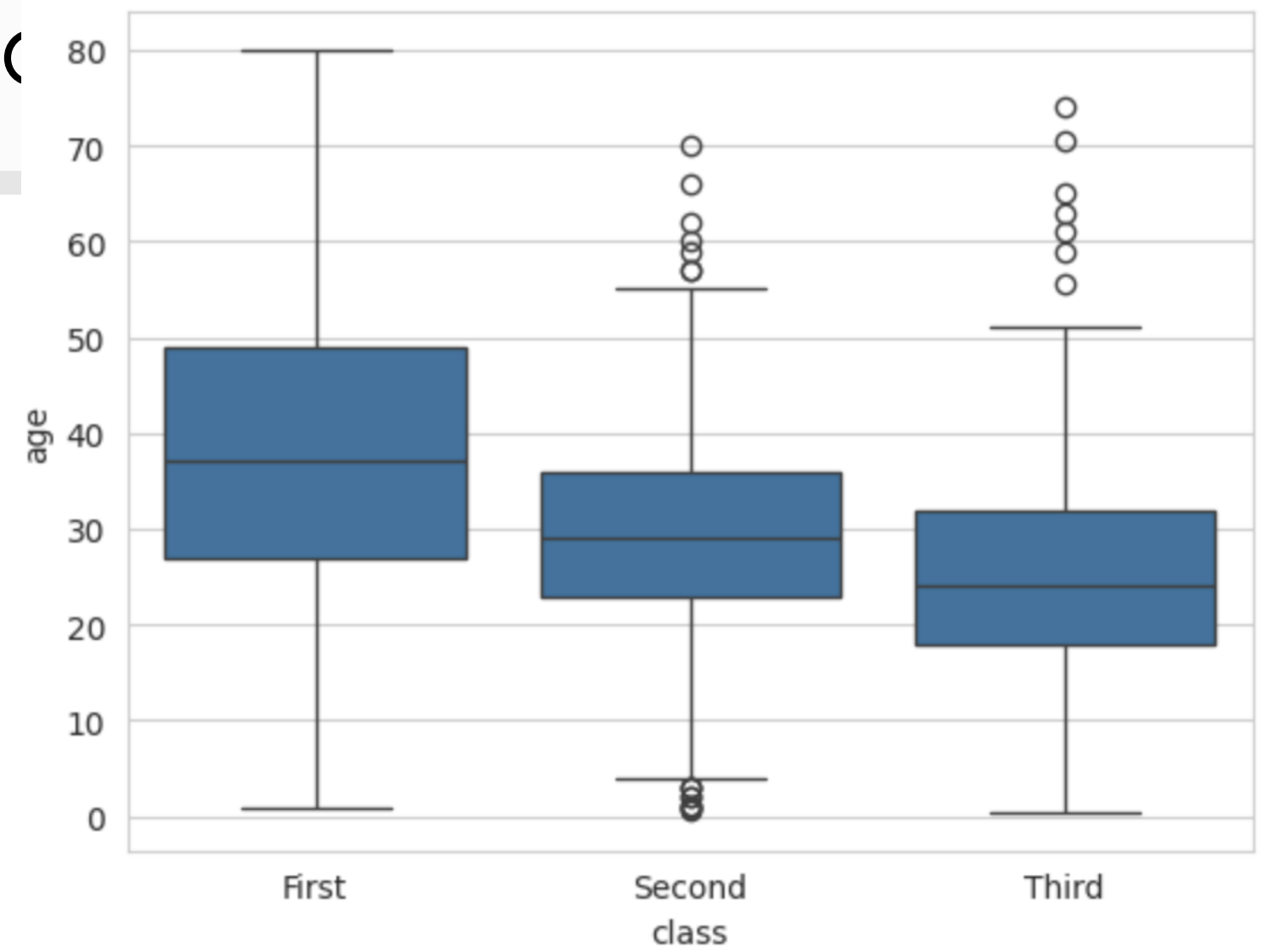
sns.boxplot()

boxplot()

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

boxplo

Out[]:

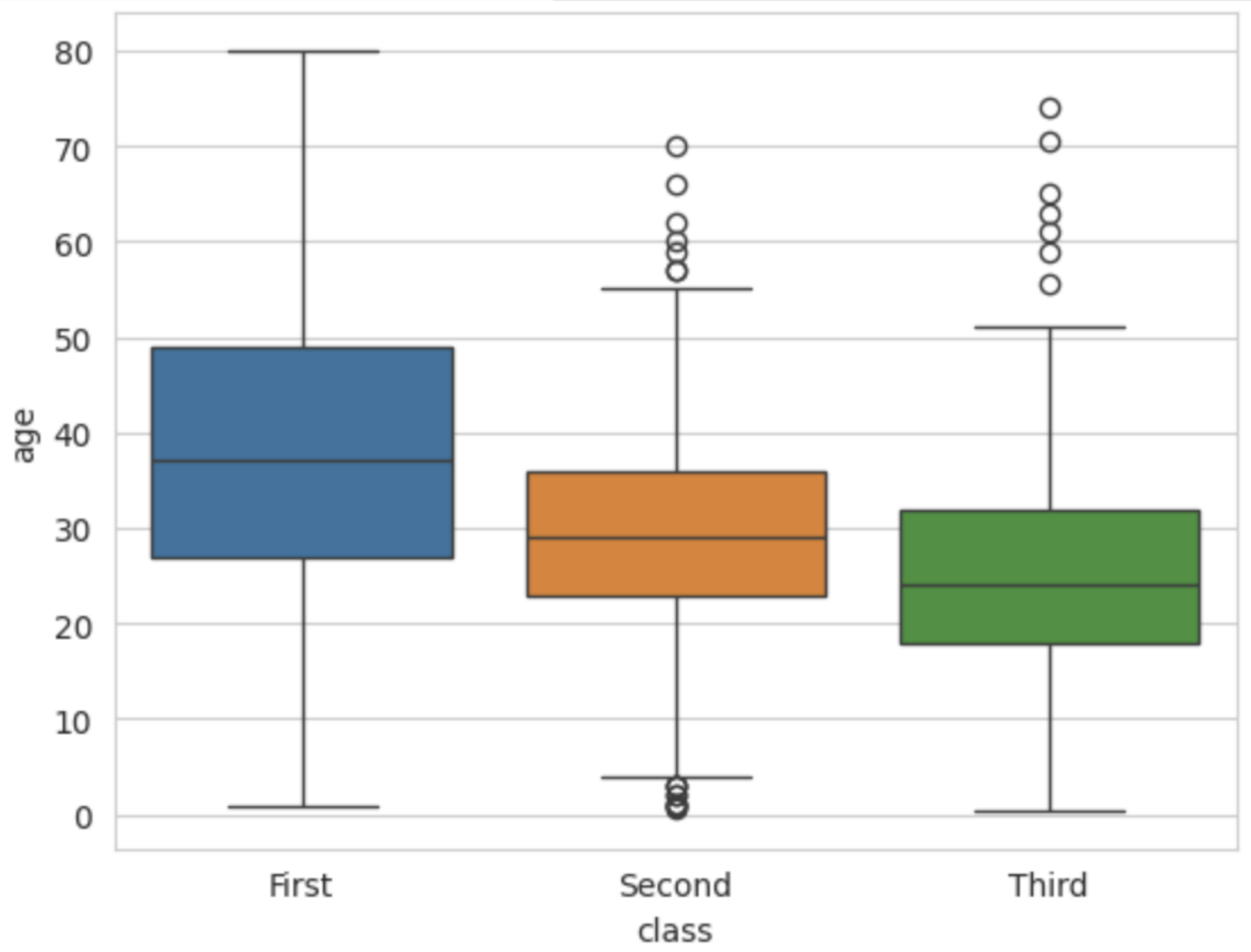


boxplot() hue

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

boxpl

Out[]:



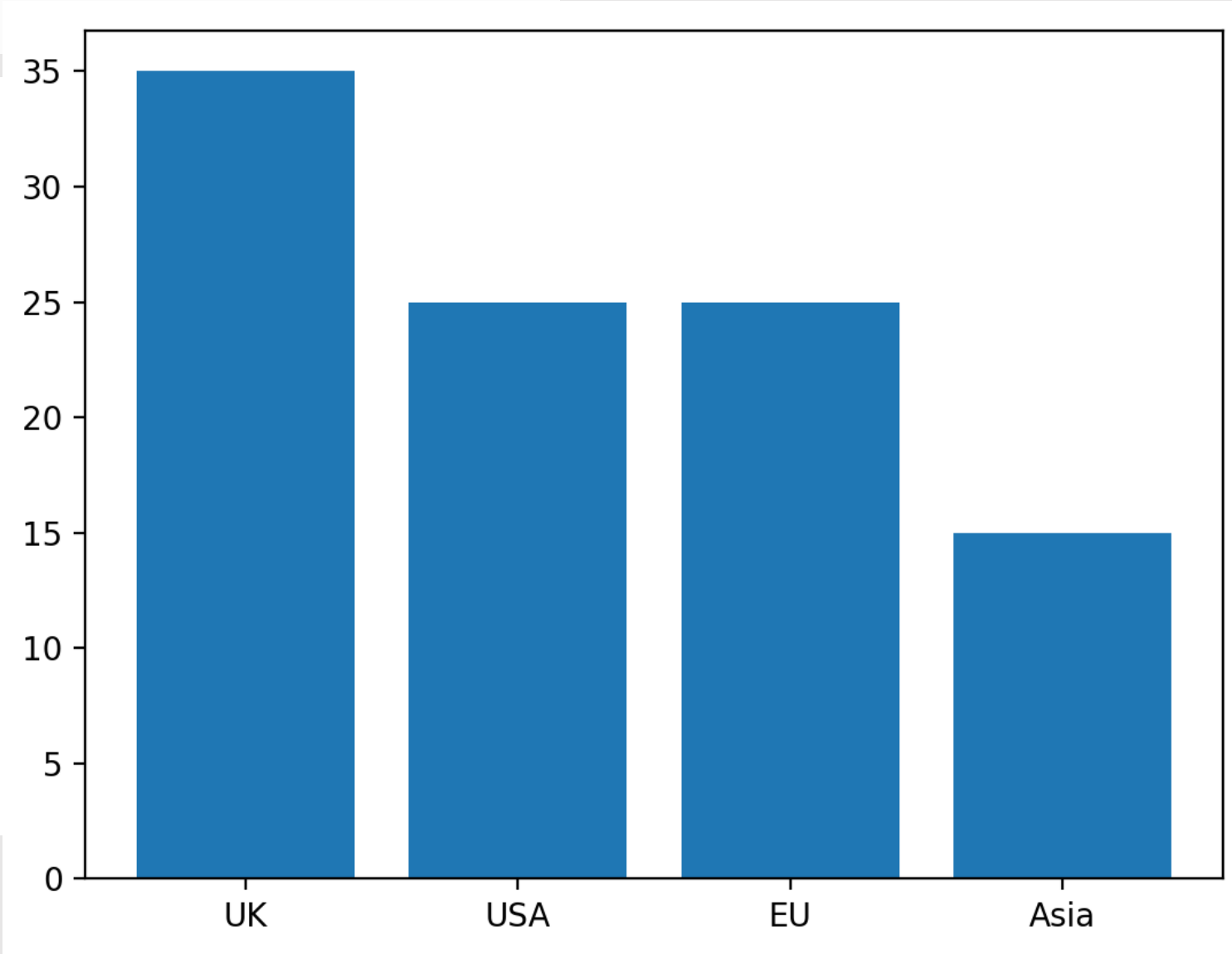
plt.bar()

Simple Bar

In[ ]:
```
1 | data = [35, 25, 25, 15]
2 | labels = ["UK", "USA", "EU", "Asia"]
3 |
4 | plt.bar(labels, data)
```

Bar graph

Out[]:



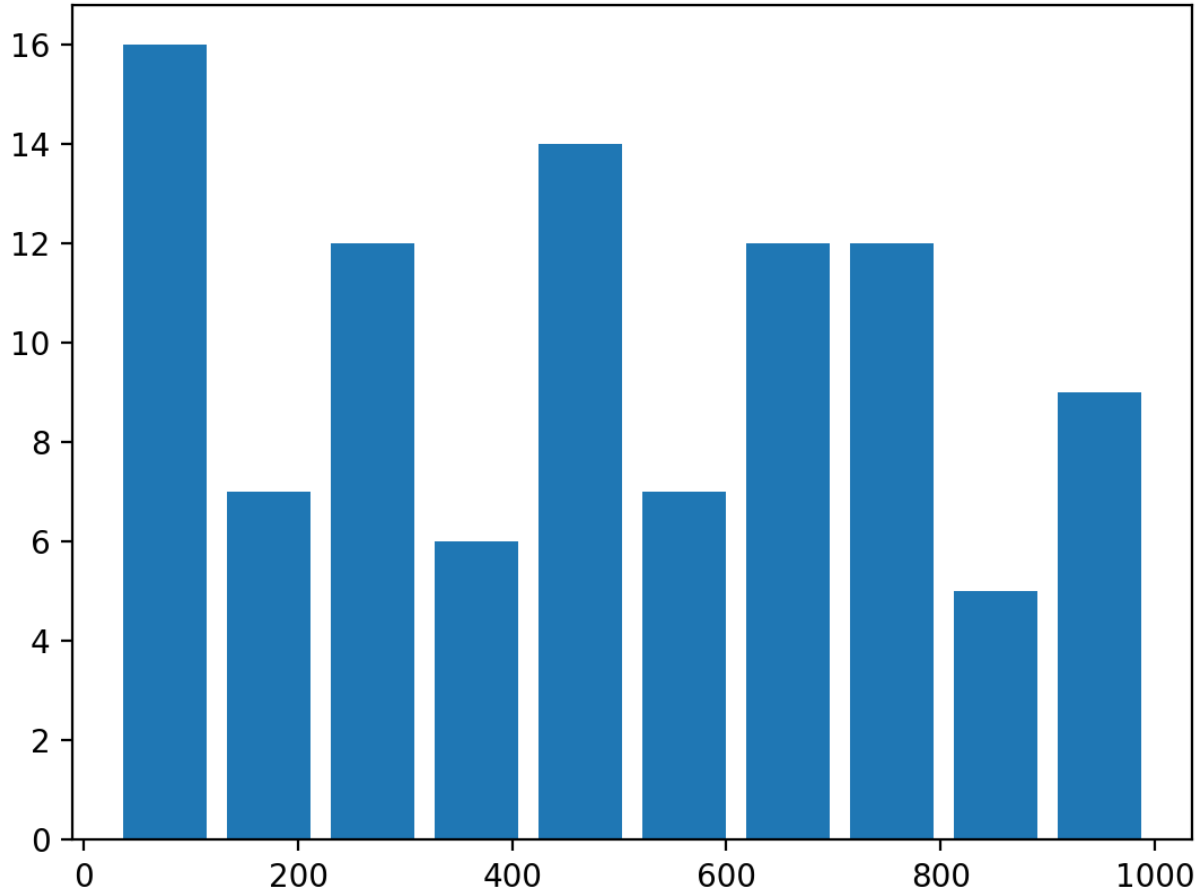
plt.hist()

Bins - intervals

In[ ]:
```
1 | from random import sample
2 | data = sample(range(1, 1000), 100)
3 | plt.hist(data, bins = 10, rwidth = 0.8)
```

10 bins

Out[]:

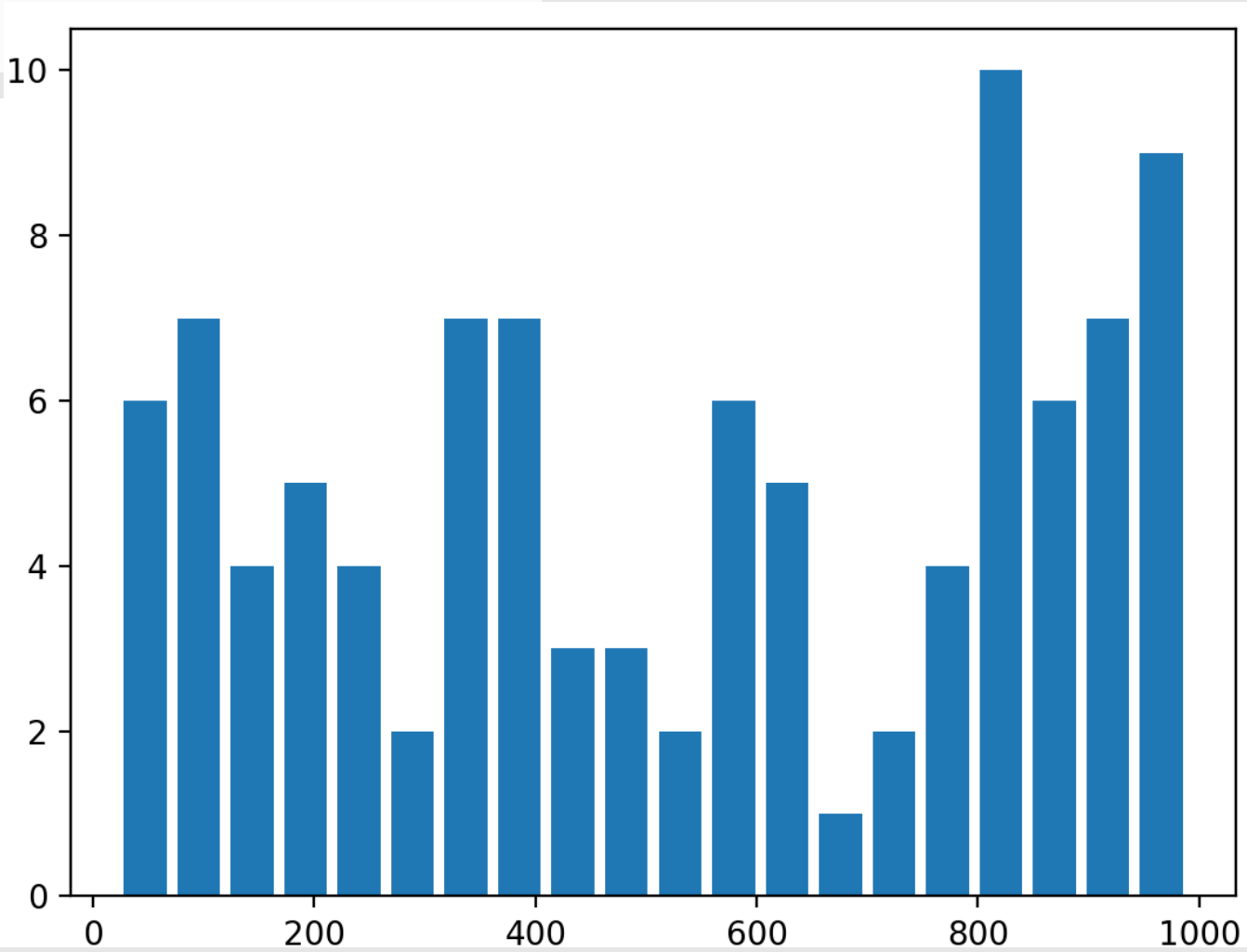


20 bins

In[ ]:
```
1 | from random import sample
2 | data = sample(range(1, 1000), 100)
3 | plt.hist(data, bins = 20, rwidth = 0.8)
```

20 bins

Out[]:



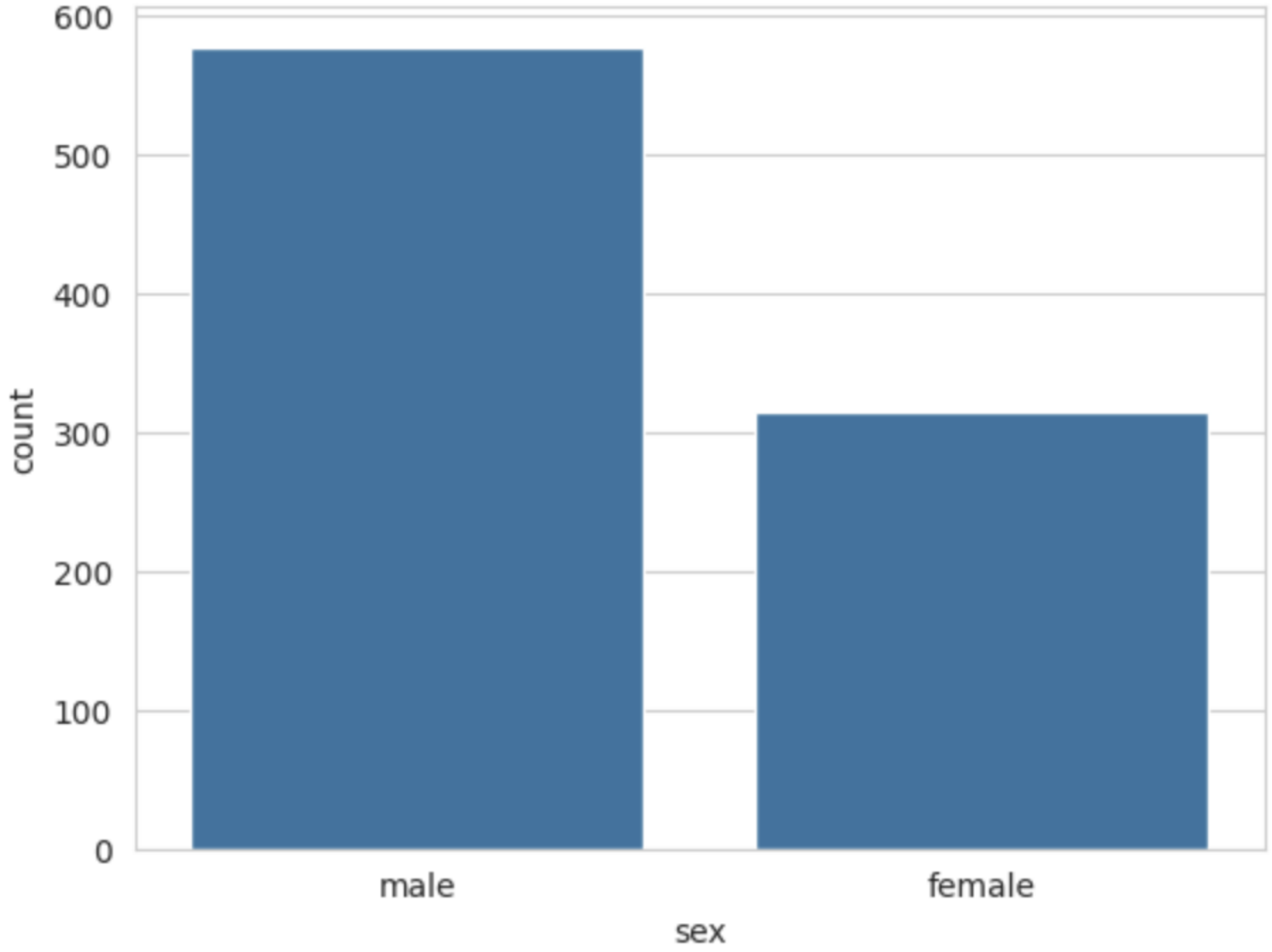
sns.countplot()

countplot()

In[ ]:
```
1 | sns.countplot(x='sex',
2 |               data=titanic)
3 |
```

count

Out[]:

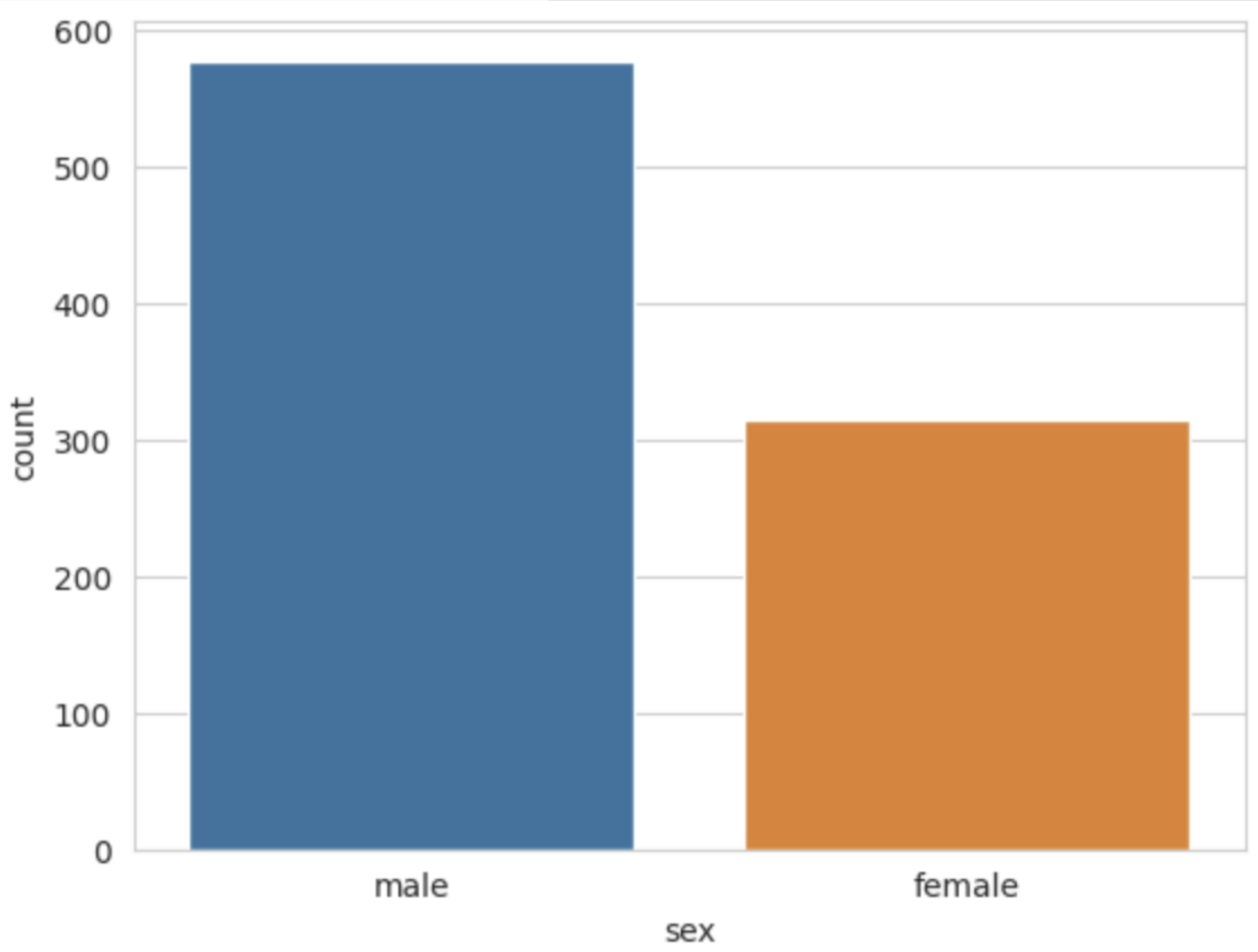


countplot() hue

In[ ]:
```
1 | sns.countplot(x='sex',
2 |               data=titanic,
3 |               hue='sex')
```

count

Out[]:



sns.barplot()

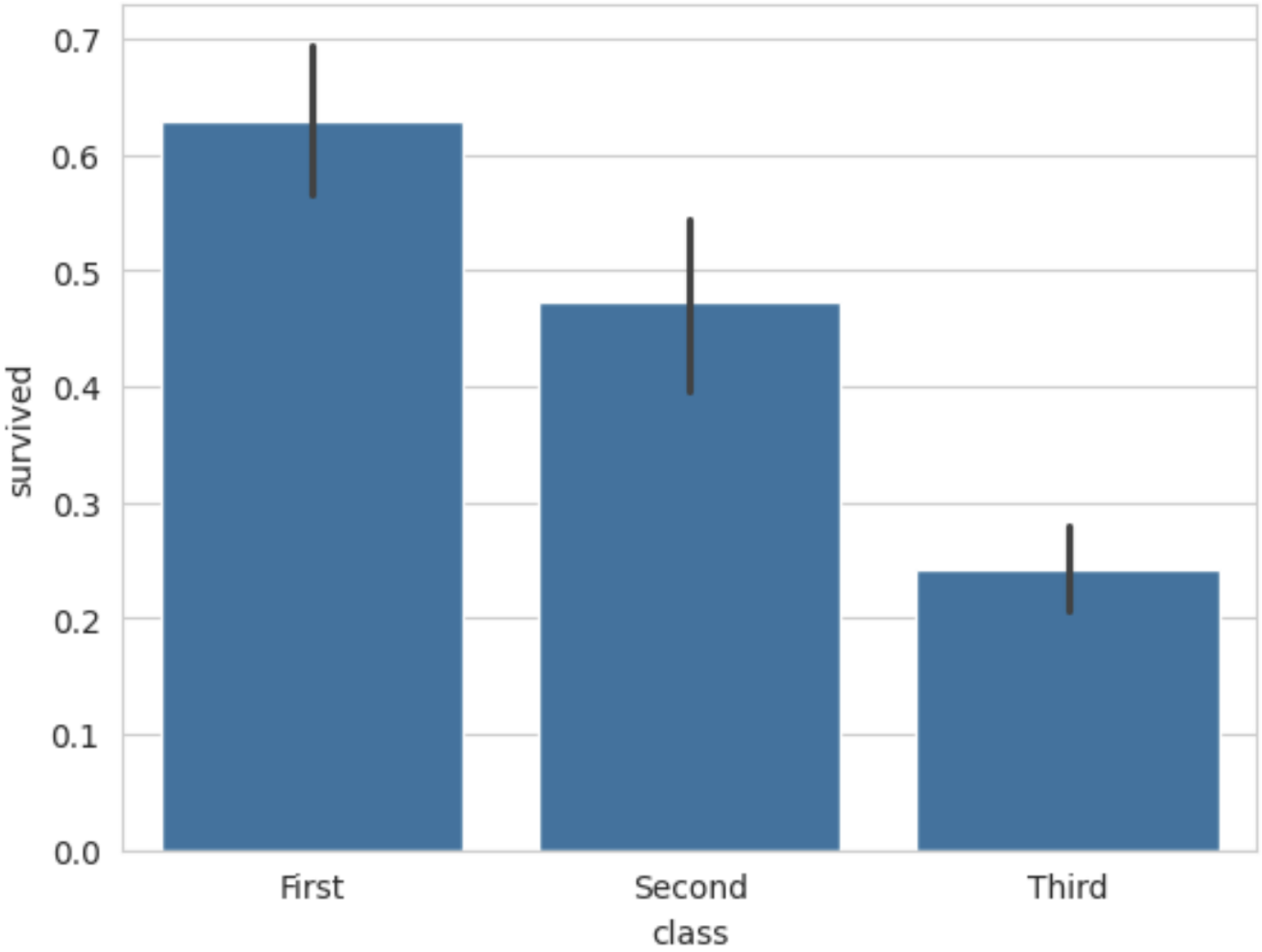
barplot()

In[ ]:

```
sns.barplot(x='class',
            y ='survived',
            data=titanic)
```

barp

Out[]

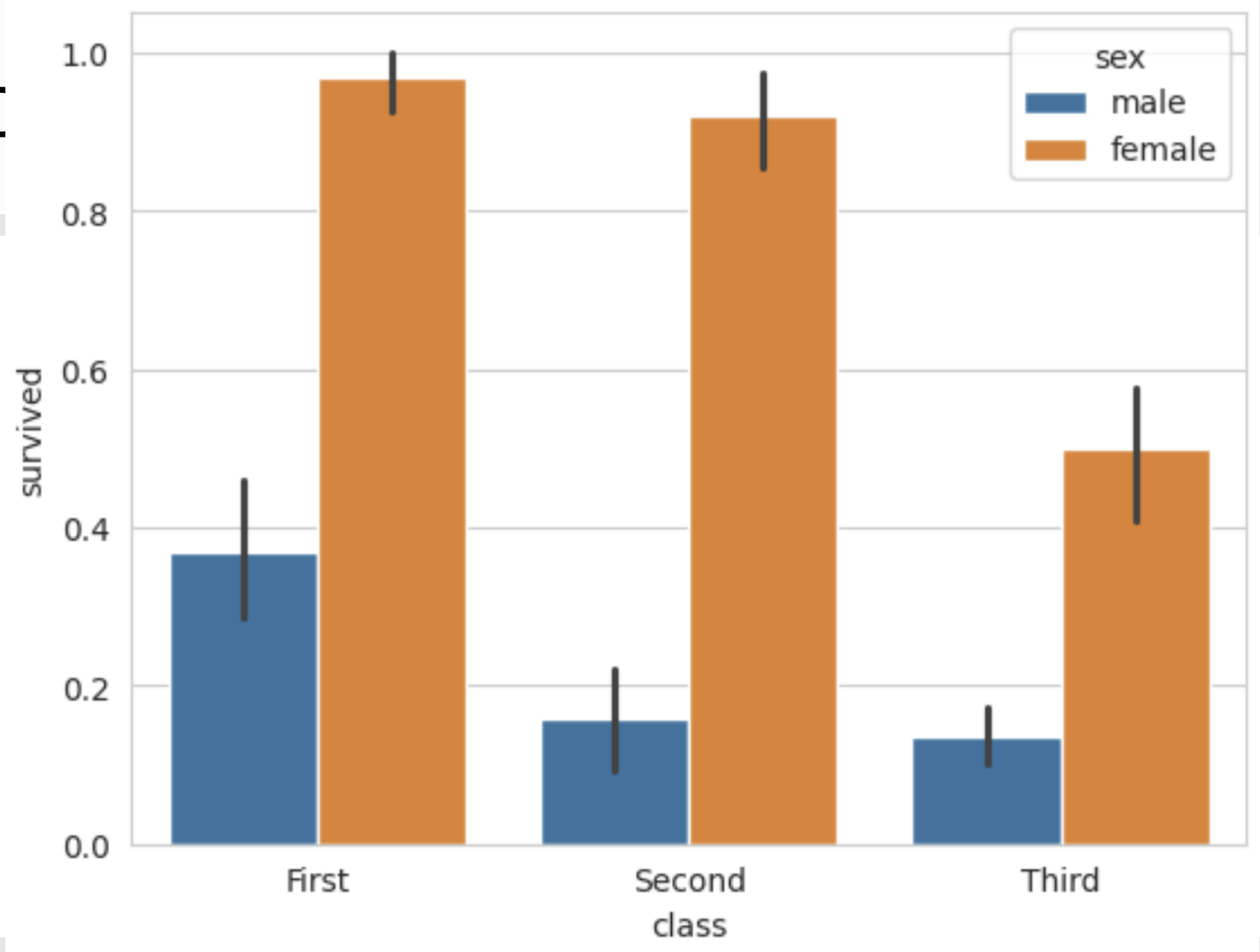


barplot()

In[ ]:
```
sns.barplot(x='class',
            y ='survived',
            data=titanic,
            hue ='sex')
```

barp

Out[]


1.0
0.8
0.6
0.4
0.2
0.0
survived
sex
male
female
First
Second
Third
class

sns.catplot()

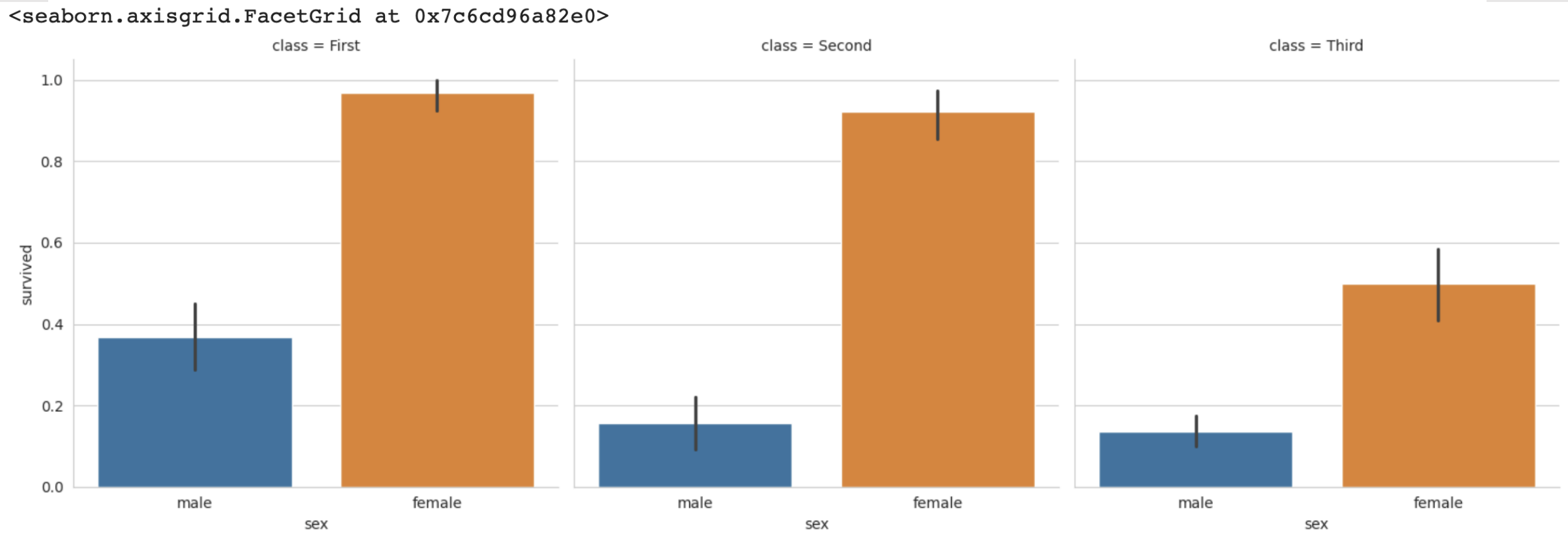
catplot()

In[ ]:

```
sns.catplot(x='sex',
            y ='survived',
            data=titanic,
            col='class',
            hue ='sex’)
```

catplot()

```
<seaborn.axisgrid.FacetGrid at 0x7c6cd96a82e0>
```



catplot()

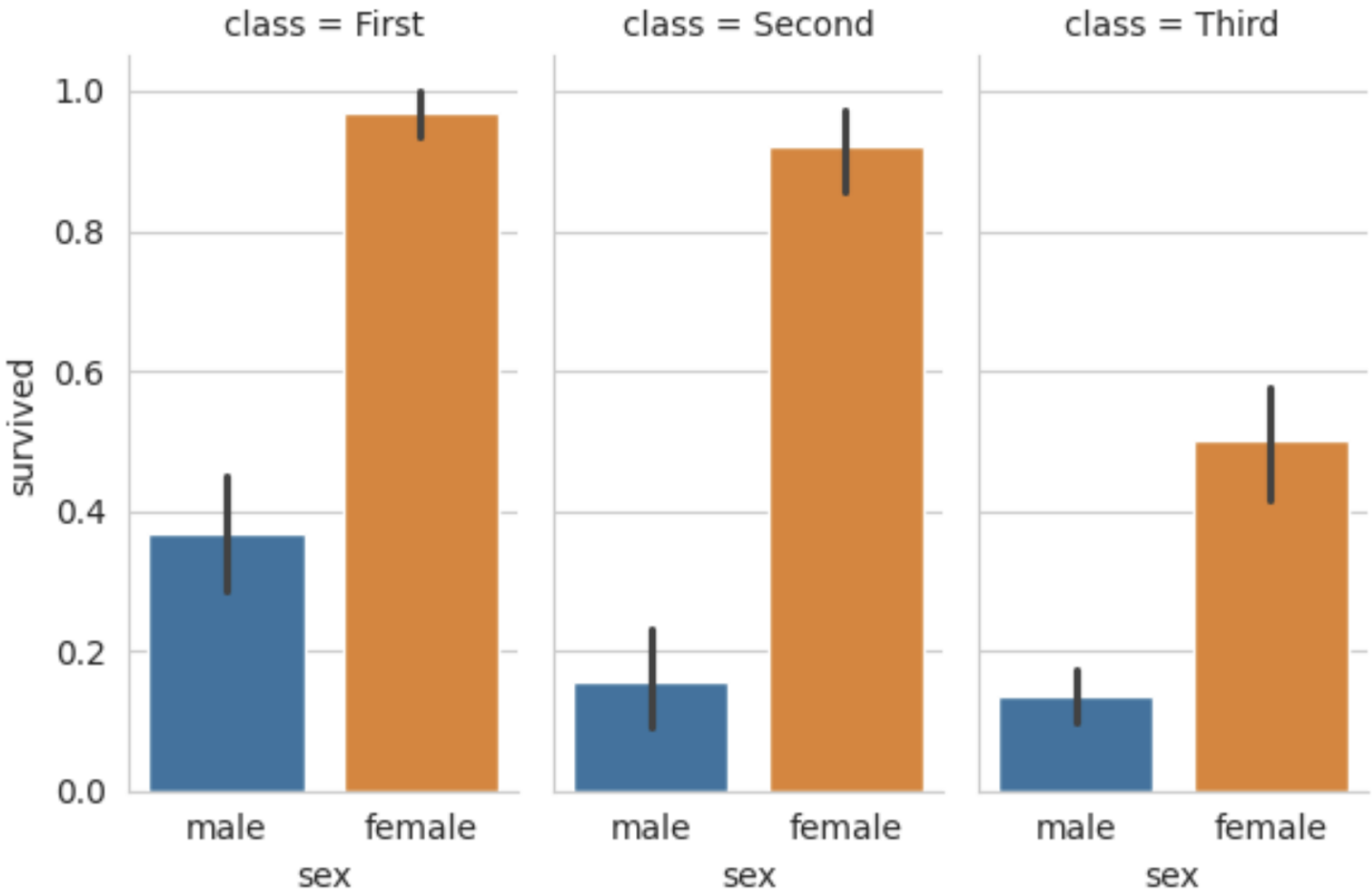
In[ ]:

```
sns.catplot(x='sex',
                y ='survived',
                data=titanic,
                col='class',
                hue ='sex',
            height=4, aspect=0.5)
```

catplot()

Out[]

```
<seaborn.axisgrid.FacetGrid at 0x7c6cd89bfd60>
```



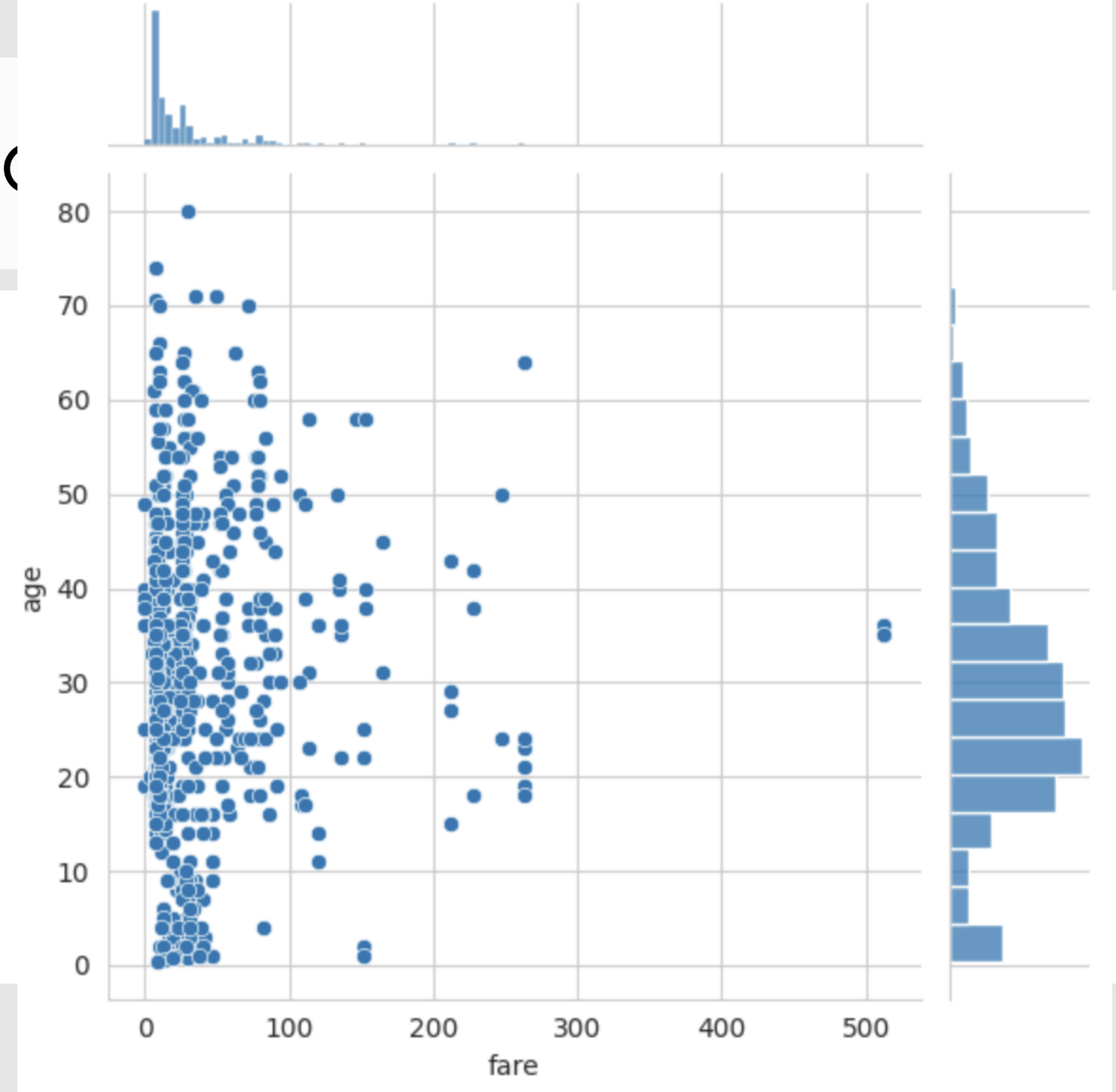
sns.jointplot()

jointplot()

In[ ]:
```
1 | sns.jointplot(x='fare',
2 |               y='age',
3 |               data=titanic)
```

jointplo

Out[]:

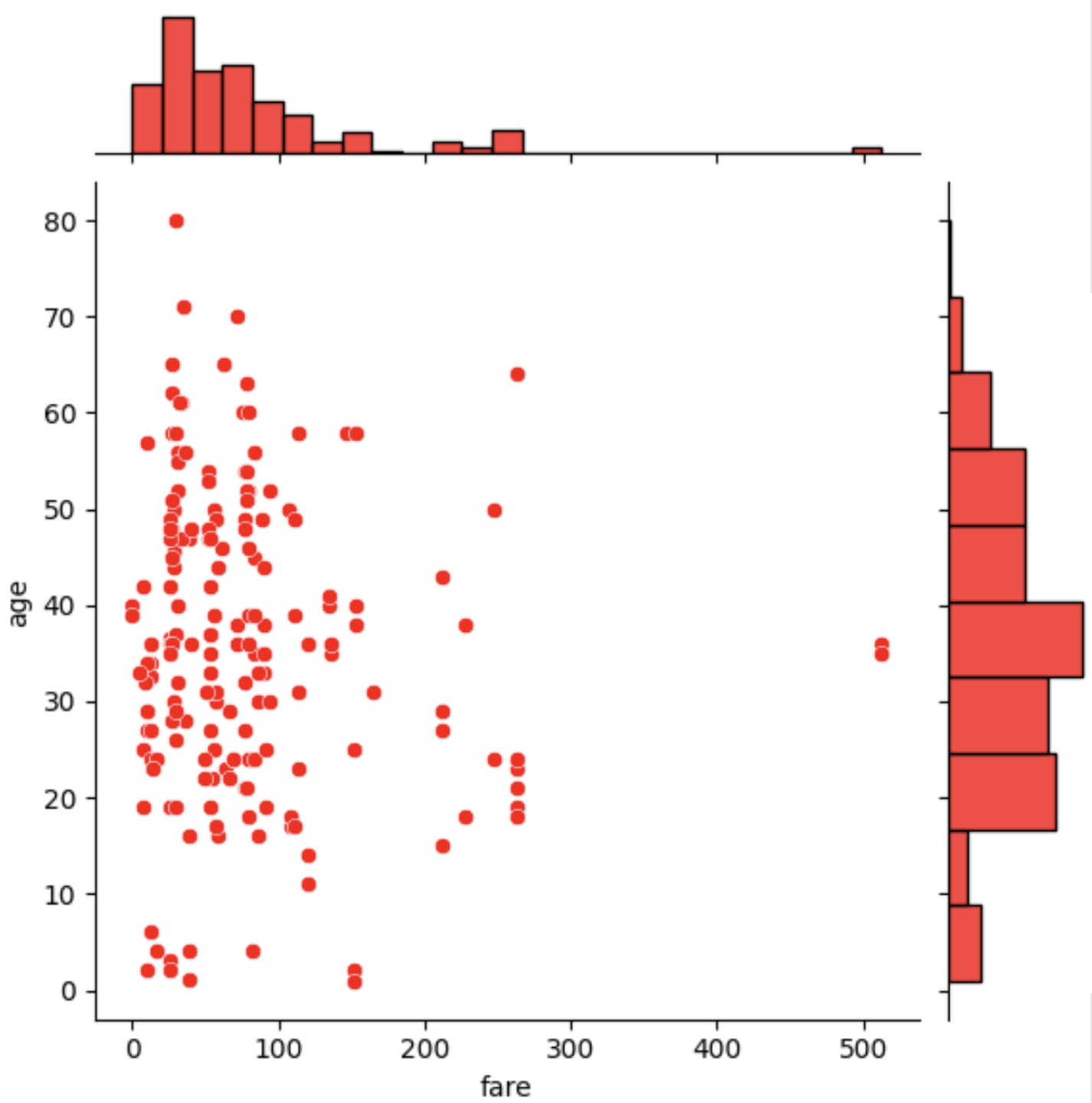


jointplot()

In[ ]:
```
sns.jointplot(x='fare',
              y='age',
              data=titanic,
              color='red')
```

jointp

Out[]:



sns.displot()

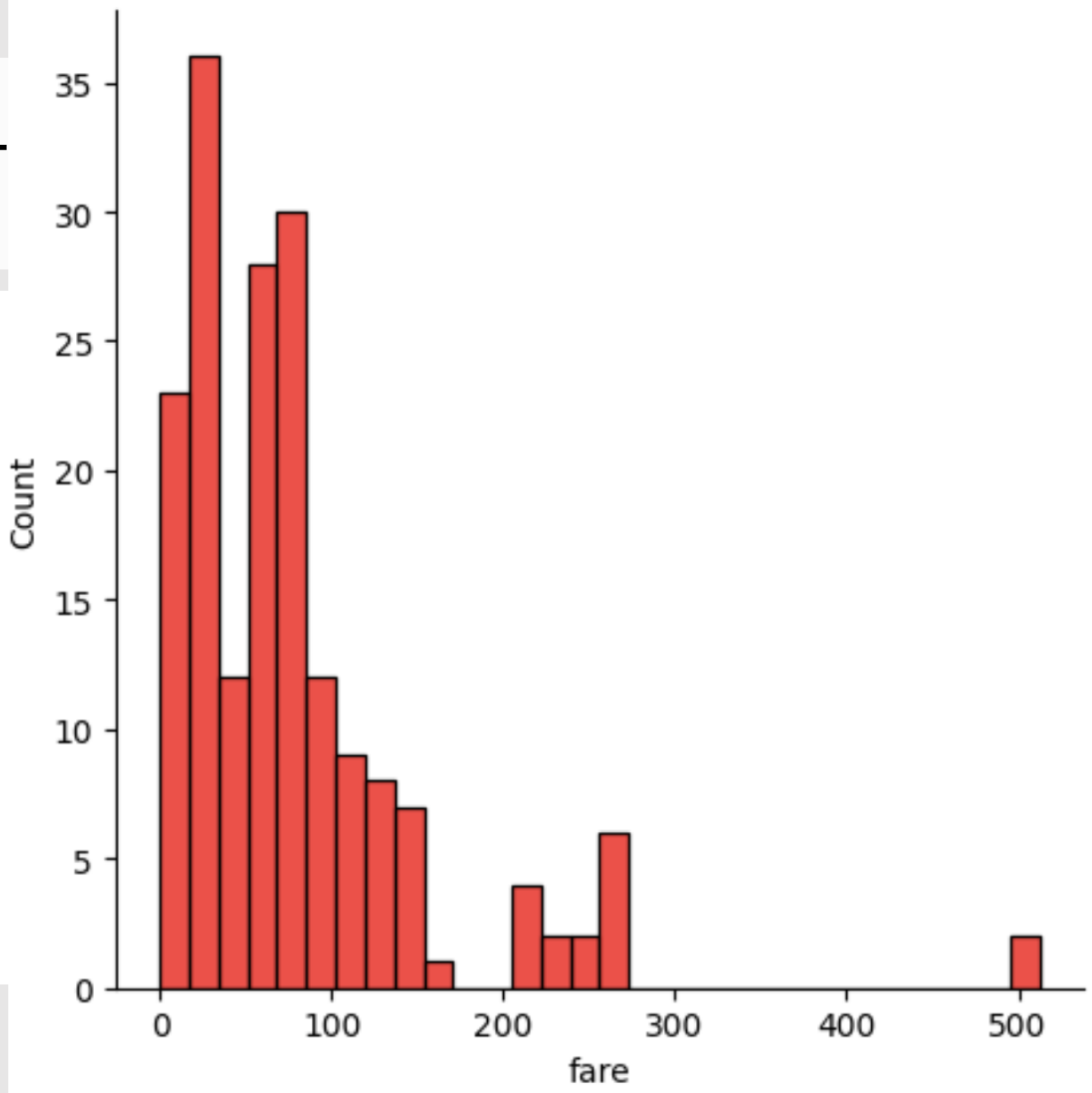
displot()

In[ ]:

```
sns.displot(titanic['fare'],
                    bins=30,
                    kde=False)
                    color='red')
```

displot

Out[]:



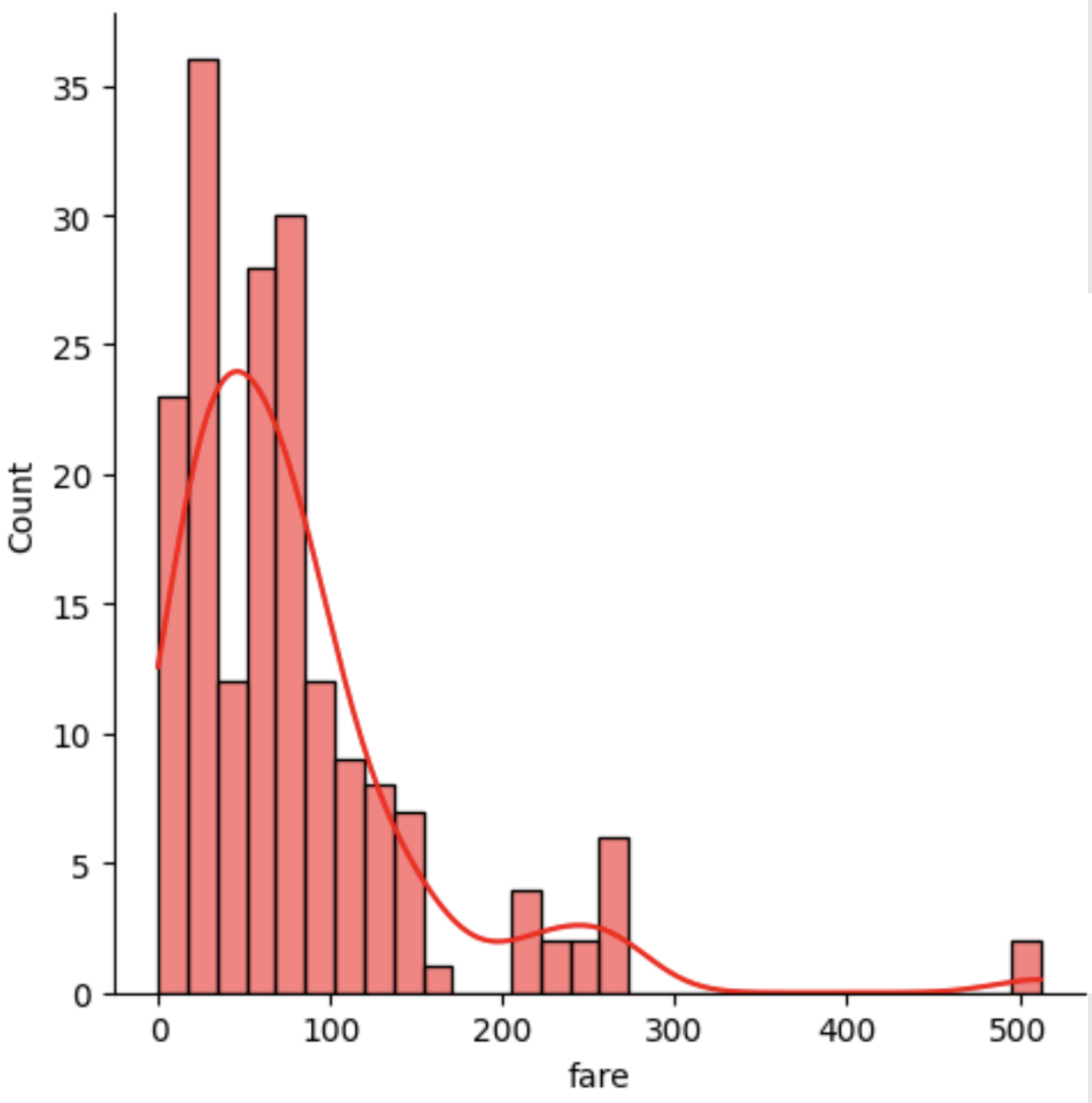
displot()

In[ ]:

```
sns.displot(titanic['fare'],
                    bins=30,
                    kde=True)
                    color='red')
```

displot

Out[]:



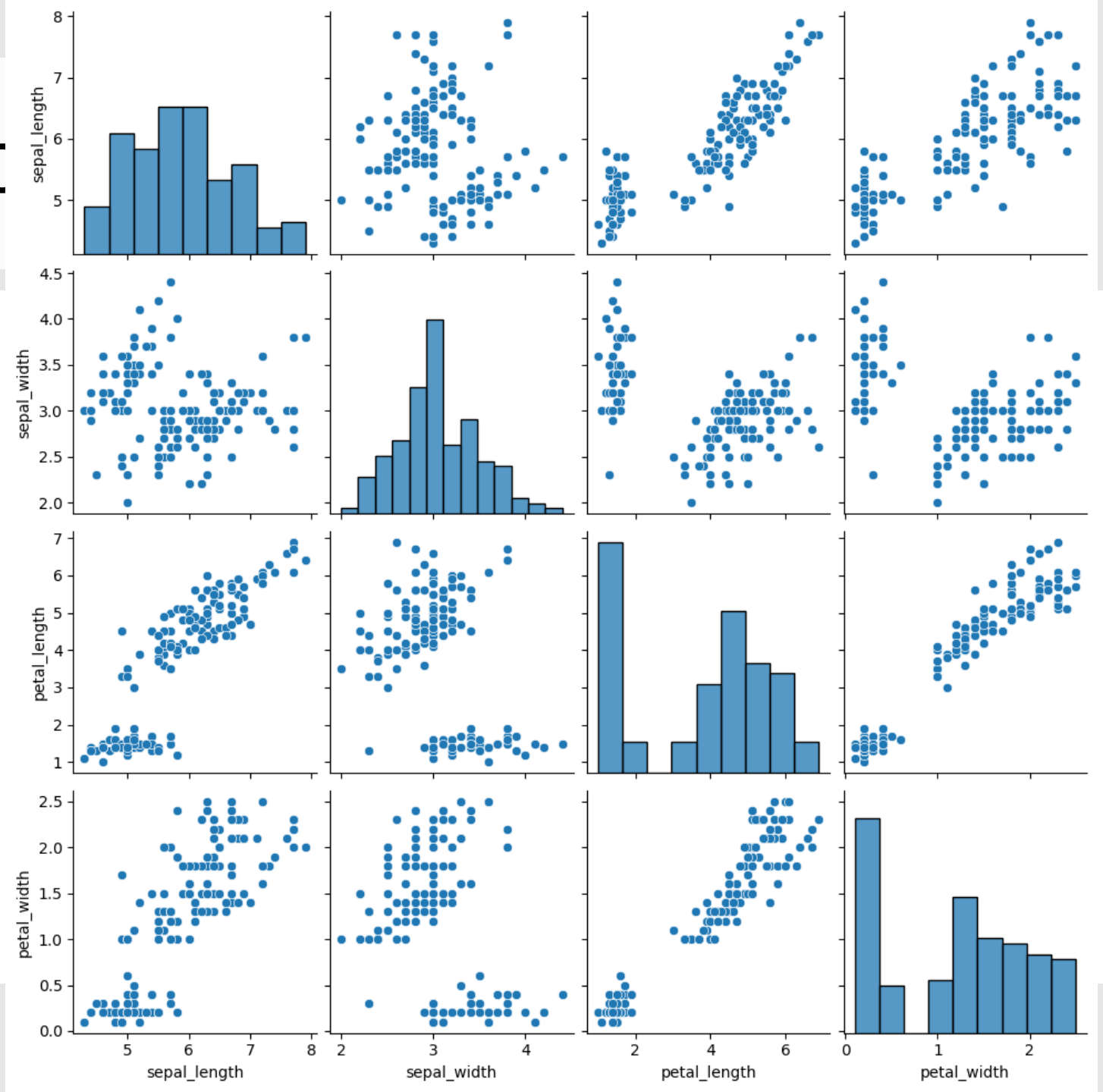
sns.pairplot()

pairplot()

In[ ]:
```
1 | sns.pairplot(iris, dropna=False)
2 |
3 |
4 |
```

pairplo

Out[]:


sepal_length
sepal_width
petal_length
petal_width
sepal_length
sepal_width
petal_length
petal_width

pairplot()

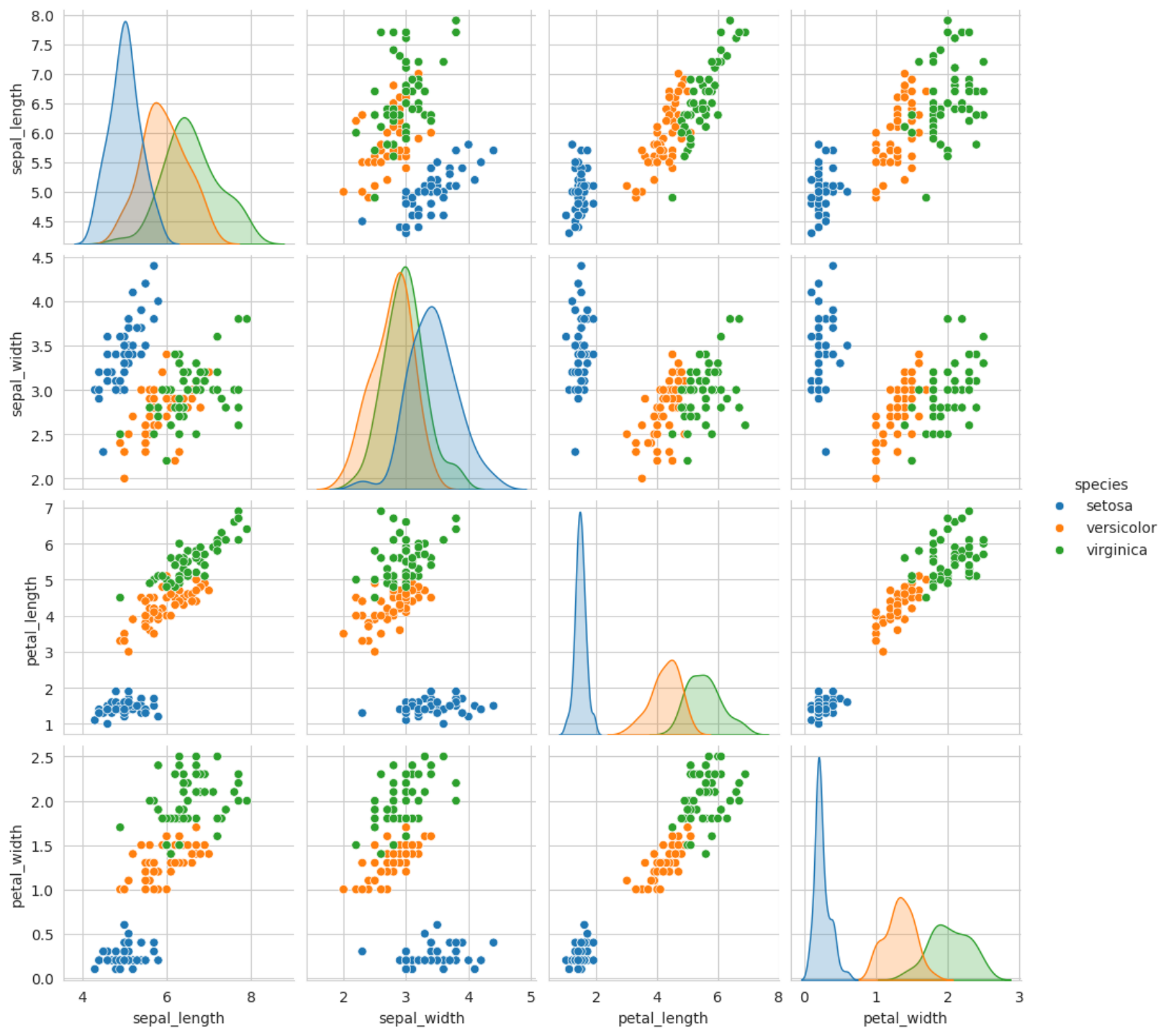
In[ ]:

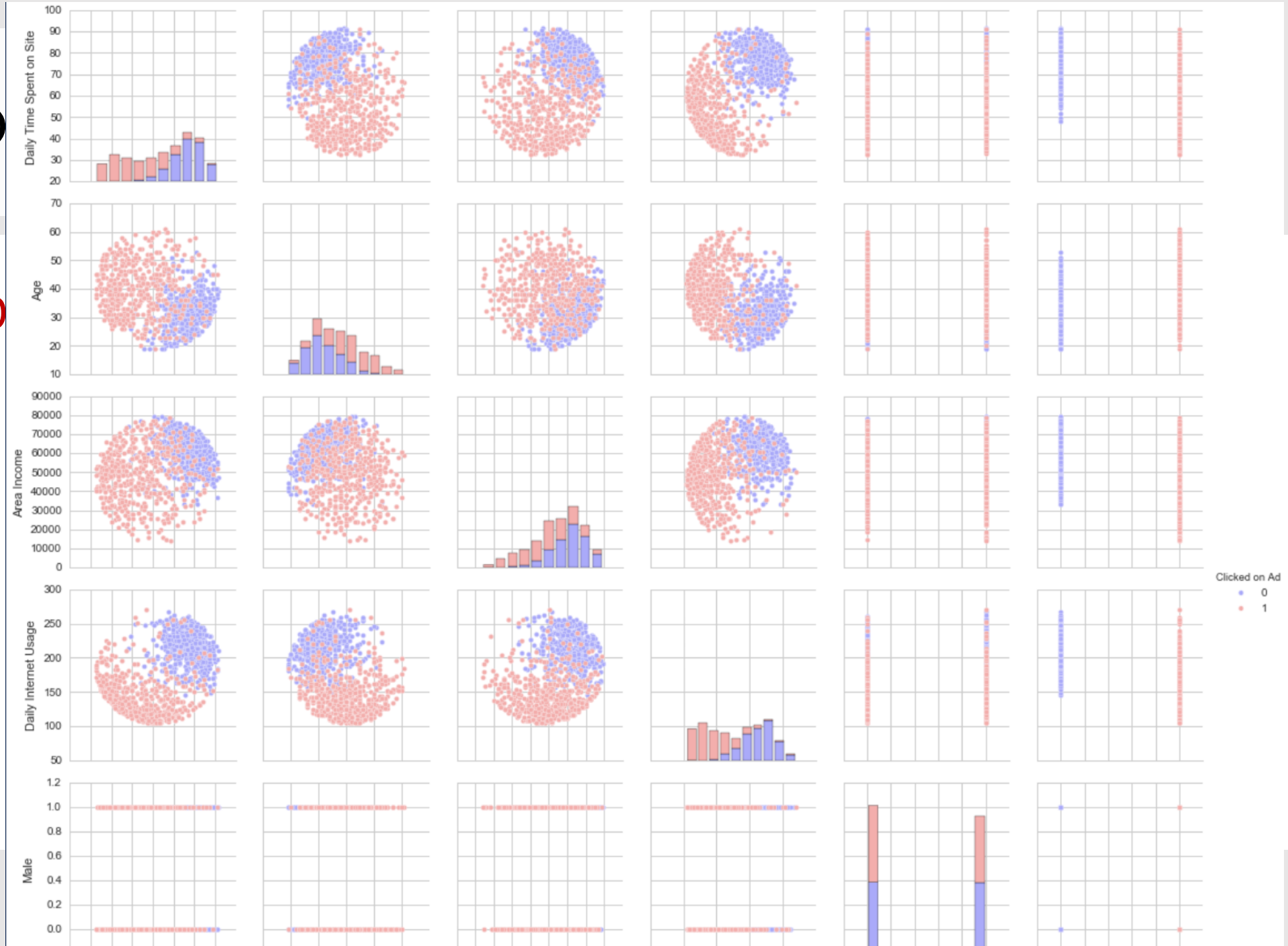
```
sns.pairplot(iris, dropna=False,
                   hue='species')

```

pairp

Out[]:


sepal_length
sepal_width
petal_length
petal_width
species
setosa
versicolor
virginica


Daily Time Spent on Site
Age
Area Income
Daily Internet Usage
Male
Clicked on Ad
0
1

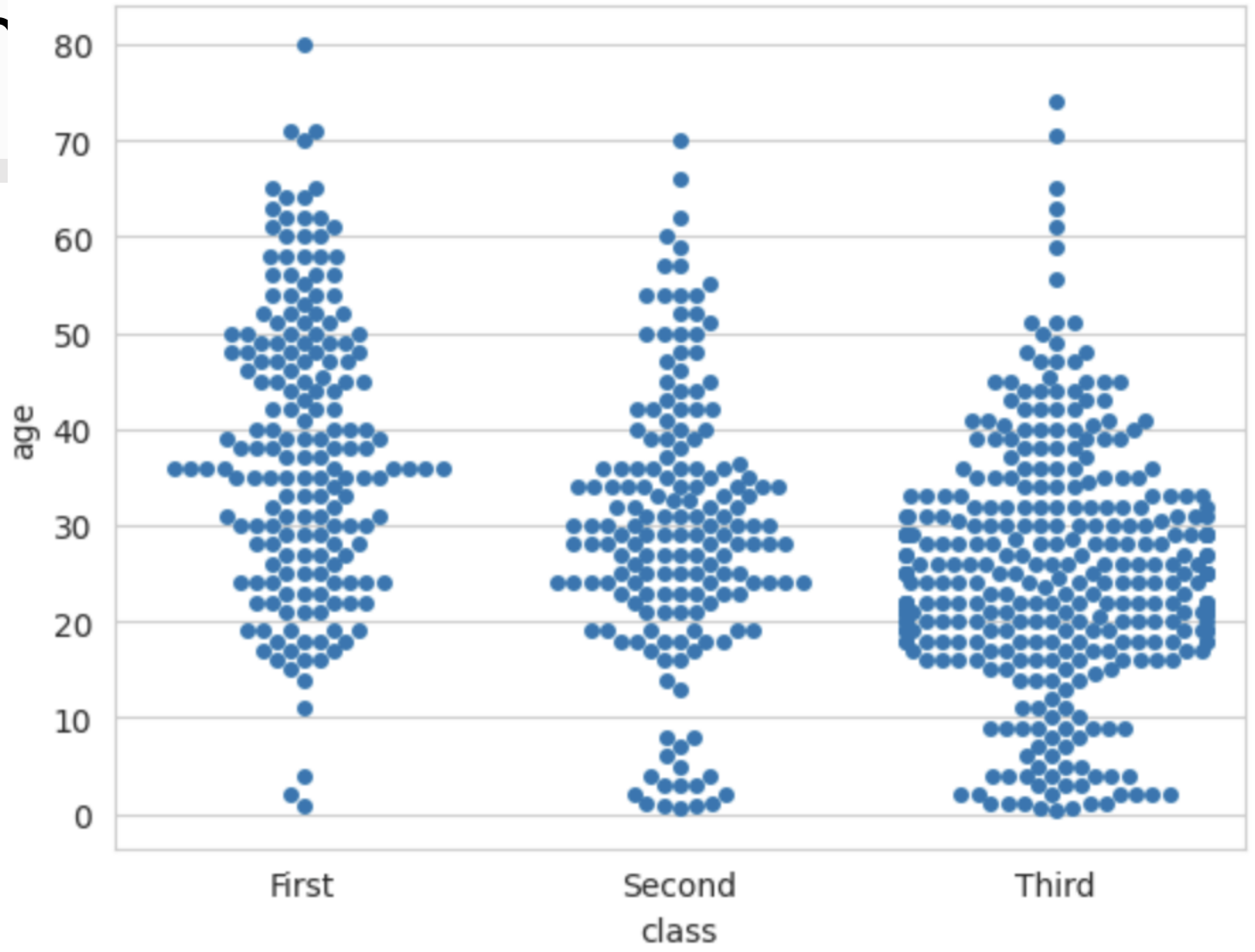
sns.swarmplot()

swarmplot()

In[ ]:
```
1 | sns.swarmplot(x='class',
2 |               y='age',
3 |               data=titanic)
```

swarm

Out[]:



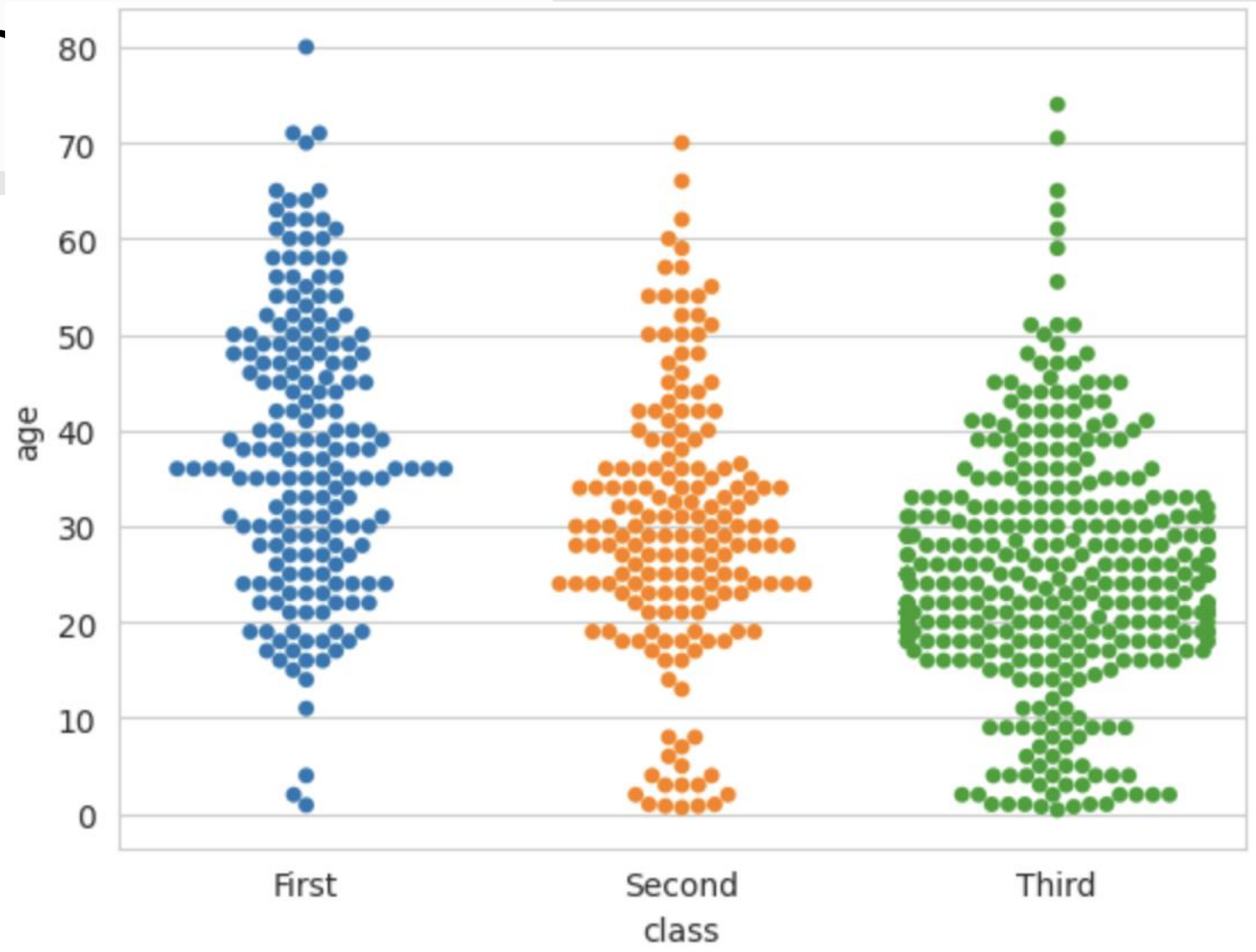
swarmplot() hue

In[ ]:

```
sns.swarmplot(x='class',
              y='age',
              data=titanic,
              hue='class')
```

swarm

Out[]:



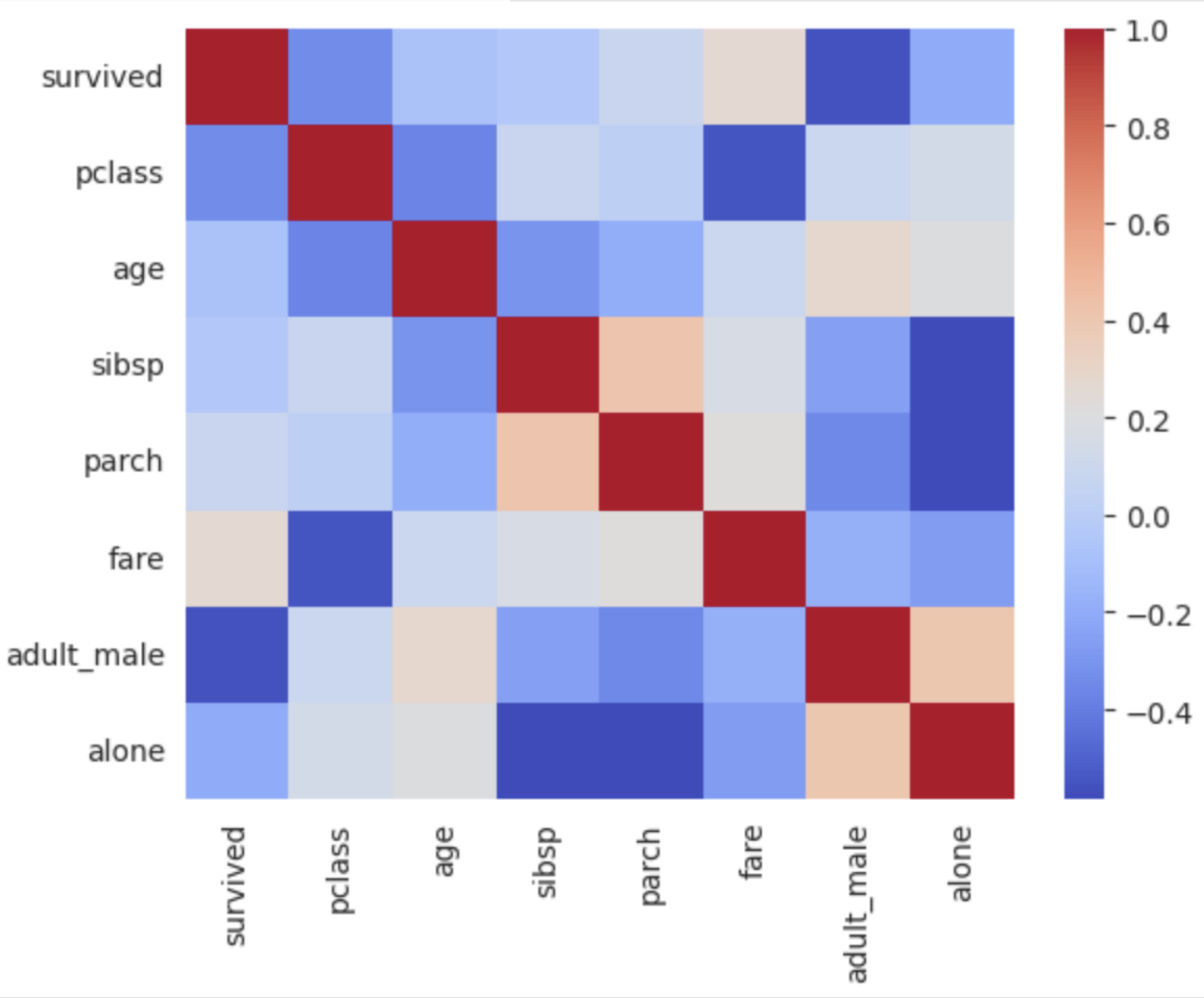
sns.heatmap()

heatmap()

In[ ]:
```
1 | sns.heatmap(
2 |       titanic.corr(numeric_only=True),
3 |       cmap='coolwarm')
4 |
```

heatm

Out[]:


survived
pclass
age
sibsp
parch
fare
adult_male
alone
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

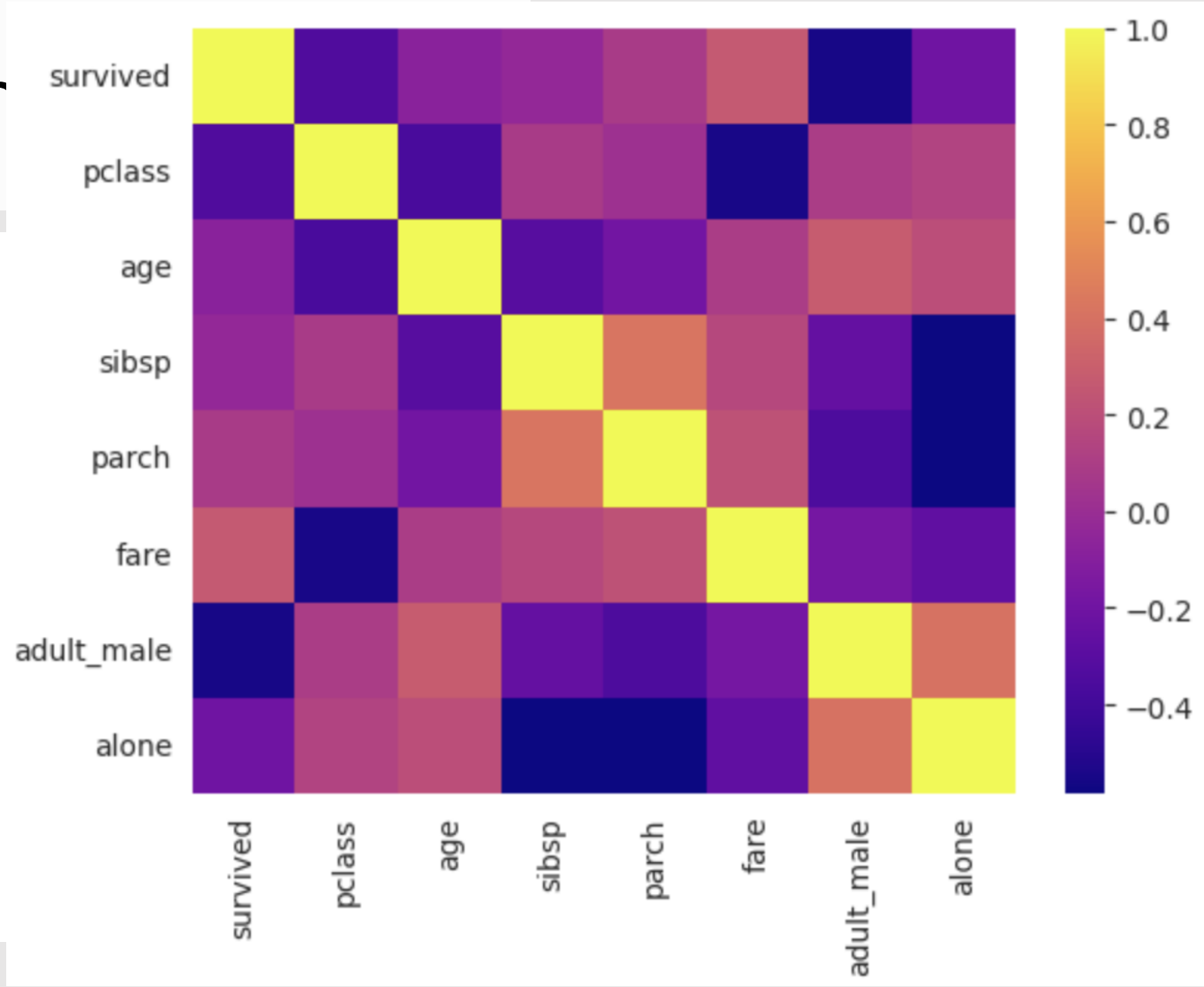
heatmap()

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

```

heatr

Out[]:


survived
pclass
age
sibsp
parch
fare
adult_male
alone
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

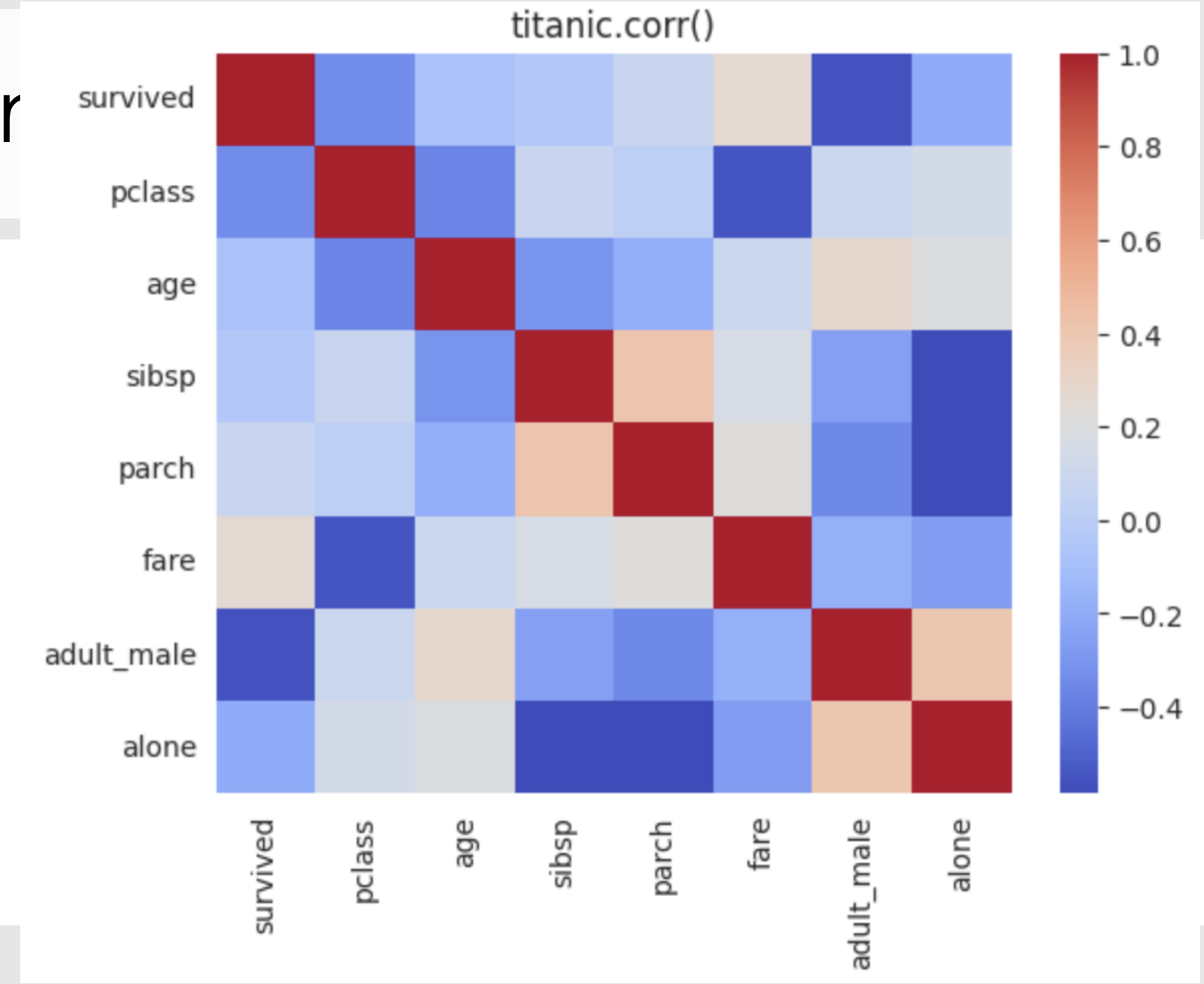
heatmap() title

In[ ]:

```
1 | sns.heatmap(
2 |       titanic.corr(numeric_only=True),
3 |       cmap='coolwarm')
4 | plt.title('titanic.corr()')
```

heatr

Out[]:


titanic.corr()
survived
pclass
age
sibsp
parch
fare
adult_male
alone
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

FacetGrid()

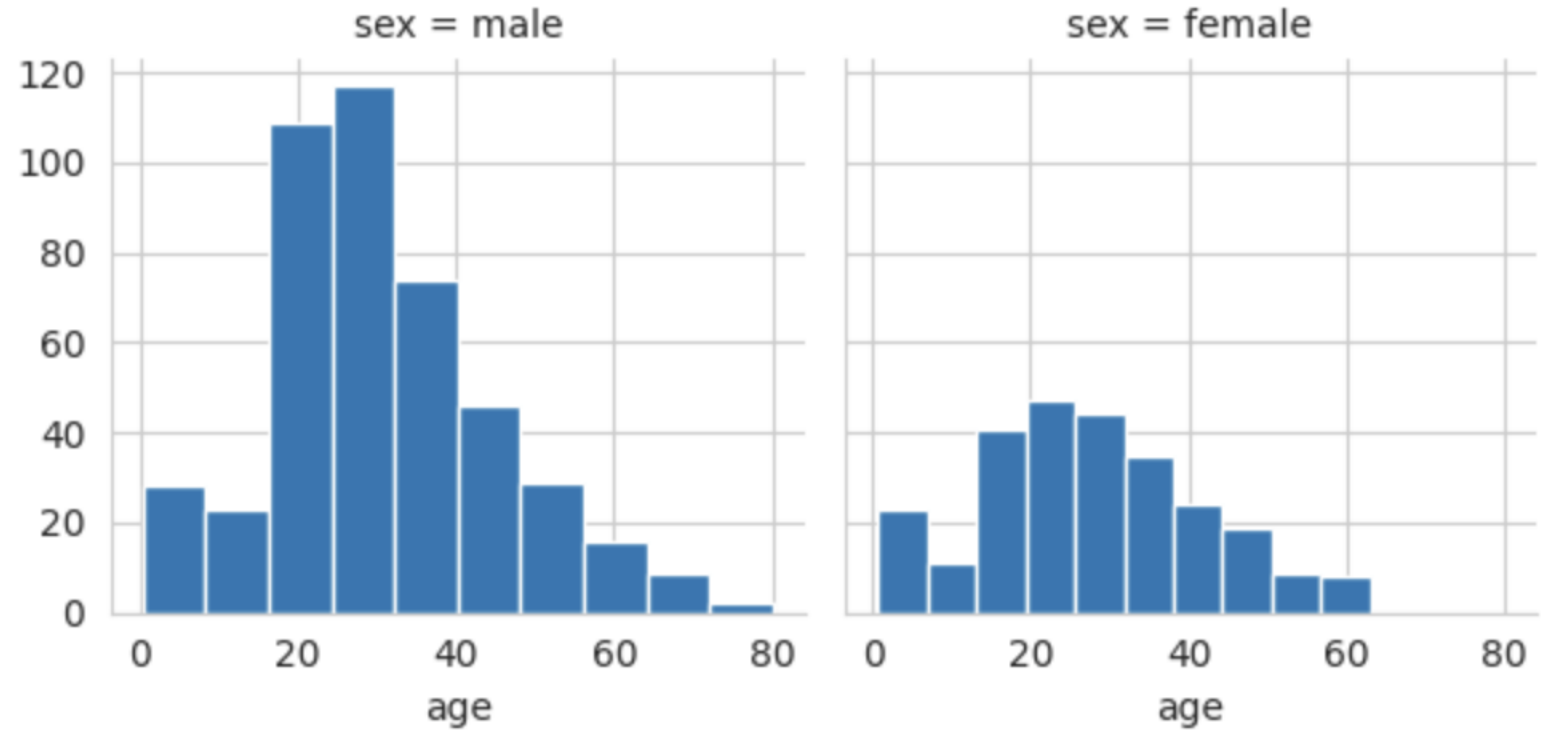
FacetGrid()

In[ ]:
```
g = sns.FacetGrid(
                data=titanic,
                col='sex')
g.map(plt.hist,'age')
```

FacetGrid()

Out[]:

```
<seaborn.axisgrid.FacetGrid at 0x7ceb05191f90>
```



FacetGrid()

In[ ]:
```
g = sns.FacetGrid(data=titanic,col='sex',
                  row='class', hue='sex',
                  margin_titles=True)
g.map(plt.hist,'age')
```

FacetGrid(

Out[]: