PYTHON SEMINAR 2020

JENS HAHN

THEORETICAL BIOPHYSICS

TODAY

- I Recap pandas
- II Data visualisation
- III matplotlib
- IV Assignment

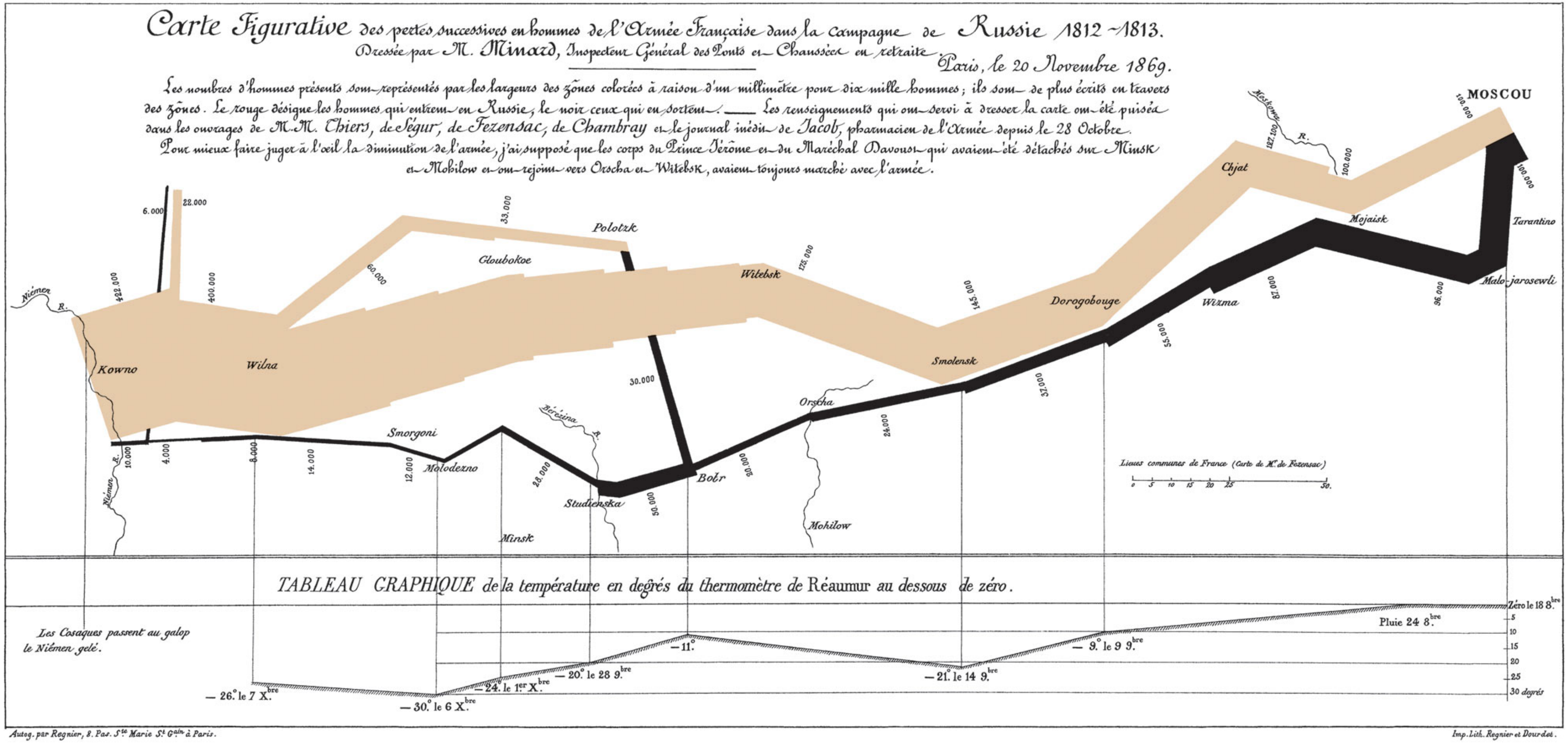
I. RECAP CLASSES

- What is actually in the data set?
- Which questions **can** be addressed?
- DataFrames (from language R)
- Accessing single entries: df.loc[index, column] / df.iloc[,j]

II. VISUALISATION

- Show prominent features of data
- Visualisations should be clear & easy to understand
- Most readers (also reviewers) only look at the figures
    - Make them nice & **never without axis labels**
    - **A good figure takes time!!**

II. VISUALISATION


Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812–1813.
Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.
Paris, le 20 Novembre 1869.
Les nombres d'hommes présents sont représentés par les largeurs des zônes colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zônes. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. —— Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M.M. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.
Pour mieux faire juger à l'oeil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davoust qui avaient été détachés sur Minsk et Mobilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.
MOSCOU
100.000
Moskowa
127.100
R.
100.000
Chjat
Mojaisk
Tarantino
Malo-jarosewli
96.000
87.000
Wizma
55.000
Dorogobouge
145.000
Smolensk
37.000
175.000
Witebsk
24.000
Orscha
Mohilow
20.000
Bobr
Polotzk
33.000
Gloubokoe
60.000
30.000
50.000
Studienska
Bérézina R.
28.000
Minsk
Molodezno
12.000
Smorgoni
14.000
Wilna
8.000
4.000
10.000
Kowno
Niémen R.
422.000
400.000
6.000
22.000
Lieues communes de France (Carte de M.r de Fezensac)
0 5 10 15 20 25 50.
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.
Les Cosaques passent au galop le Niémen gelé.
Zéro le 18 8.bre
Pluie 24 8.bre
— 9.° le 9 9.bre
— 21.° le 14 9.bre
— 11.°
— 20.° le 28 9.bre
— 24.° le 1.er X.bre
— 30.° le 6 X.bre
— 26.° le 7 X.bre
5
10
15
20
25
30 degrés
Autog. par Regnier, 8. Pas. S.te Marie S.t G.ain à Paris.
Imp. Lith. Regnier et Dourdet.

II. VISUALISATION

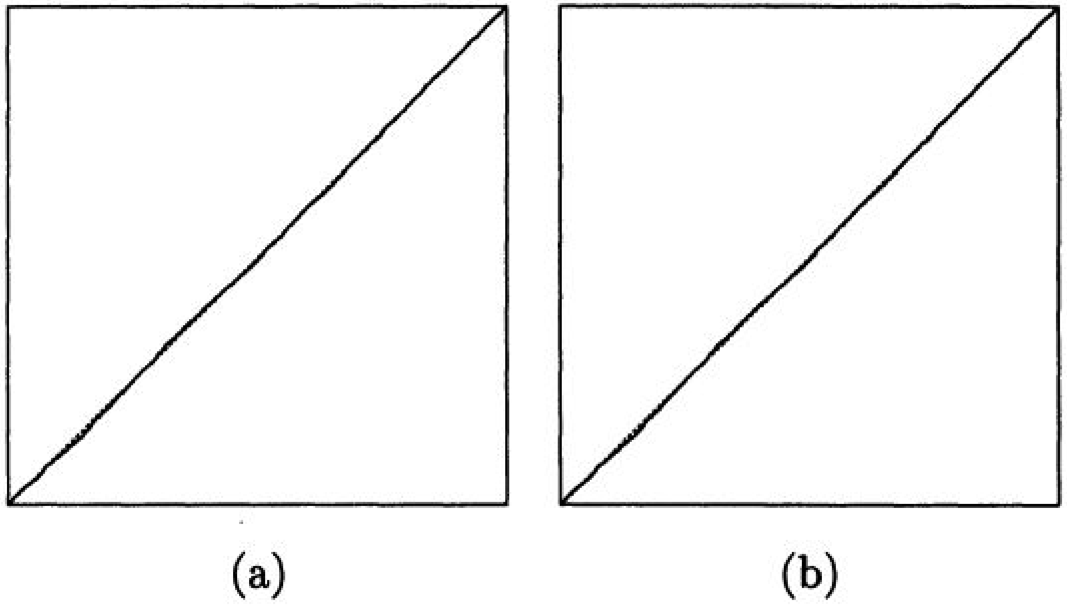
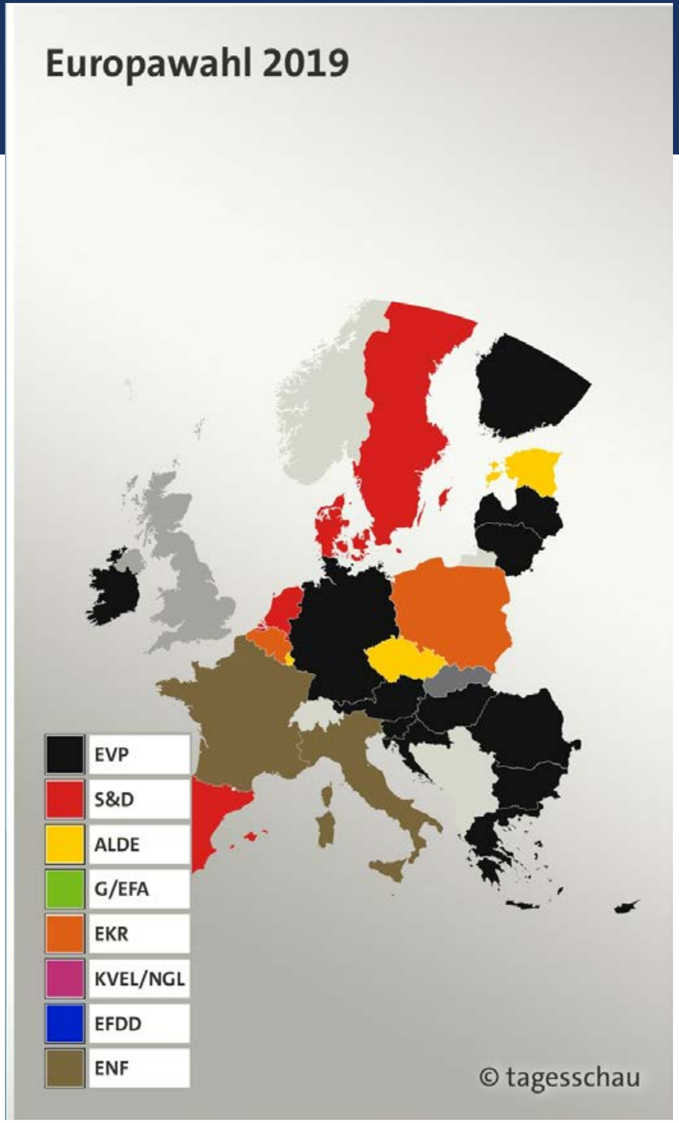


Figure 1. SRQ Plots of T_i/T_n (Vertical Axes) Against i/n (Horizontal Axes) for the Gibbs Sampler (a) and an Alternating Gibbs/Independence Sampler (b) for the Pump Failure Data Based on Runs of Length 5,000. Lines through the origin with unit slope are shown dashed; axis ranges are from 0 to 1 for all axes.

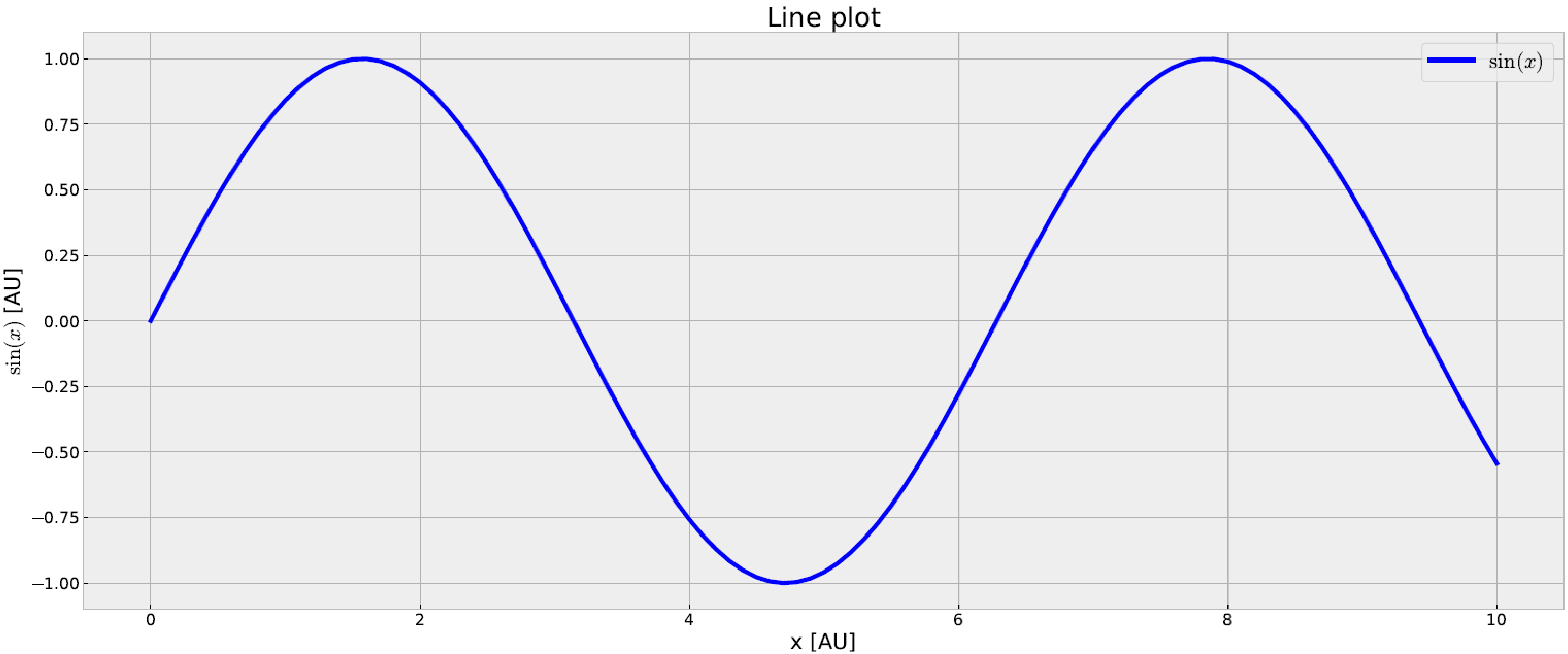
Mykland P, Tierney L, Yu B. Regeneration in Markov chain samplers.
Journal of the American Statistical Association. 1995 Mar 1;90(429):233-41.



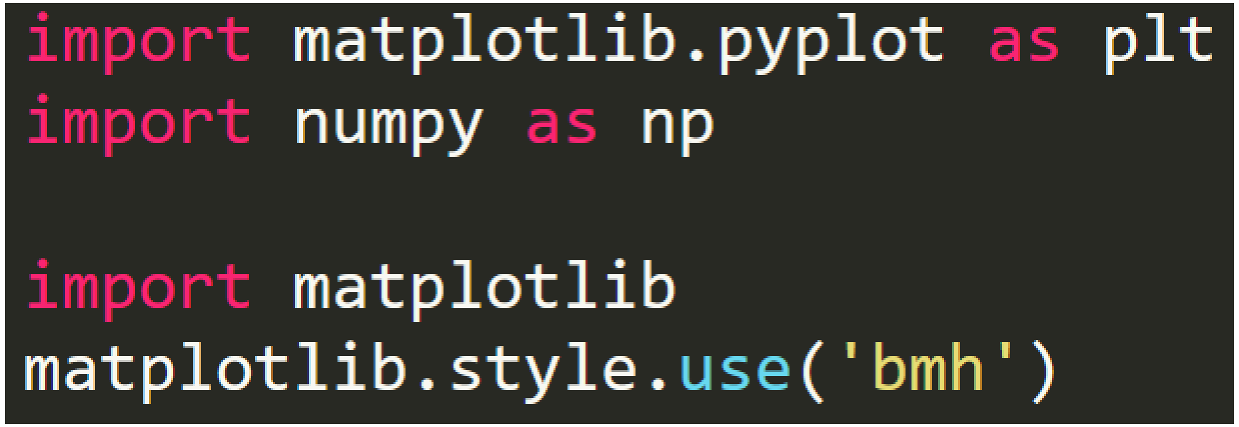
III. MATPLOTLIB

- Re-implementation of *MATLABs* plotting routine
- Huge library with numerous functions
- Many plotting libraries based on *matplotlib*
  - *see documentation: https://matplotlib.org/3.1.0/gallery/index.html*

III. LINE PLOT



III. LINE PLOT – IMPORTS

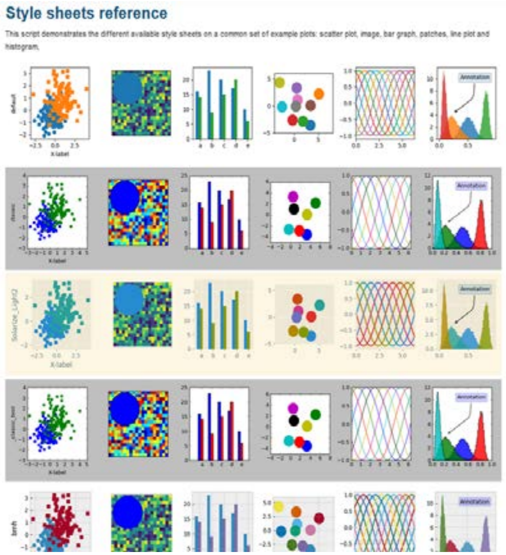


```
import matplotlib.pyplot as plt
import numpy as np

import matplotlib
matplotlib.style.use('bmh')
```

→ Import necessary packages

→ Only to change 'style' to *'bmh'*

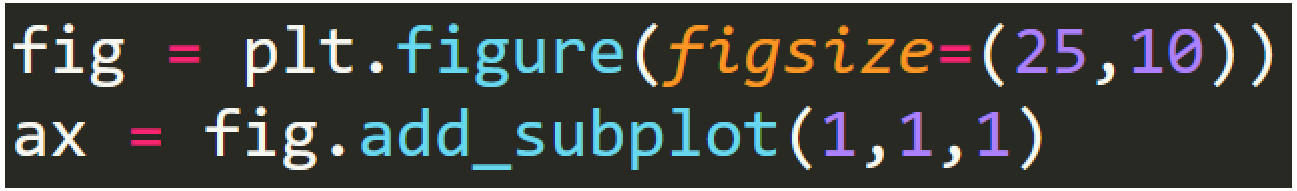


Style sheets reference

This script demonstrates the different available style sheets on a common set of example plots: scatter plot, image, bar graph, patches, line plot and histogram.

https://matplotlib.org/3.1.0/gallery/style_sheets/style_sheets_reference.html

III. LINE PLOT - SETUP CANVAS AND AXES

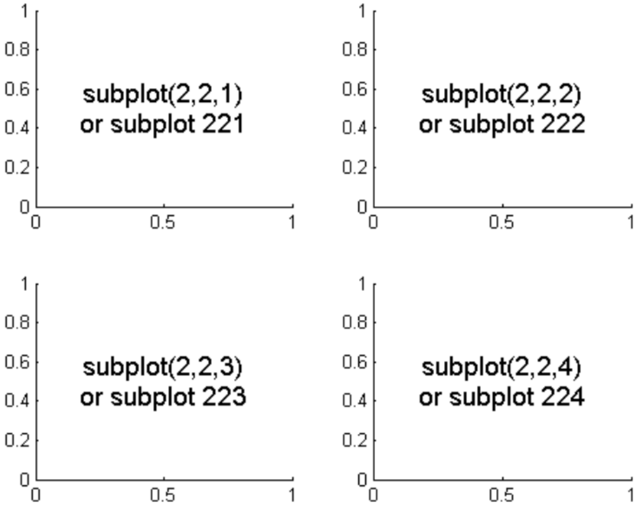


```
fig = plt.figure(figsize=(25,10))
ax = fig.add_subplot(1,1,1)
```

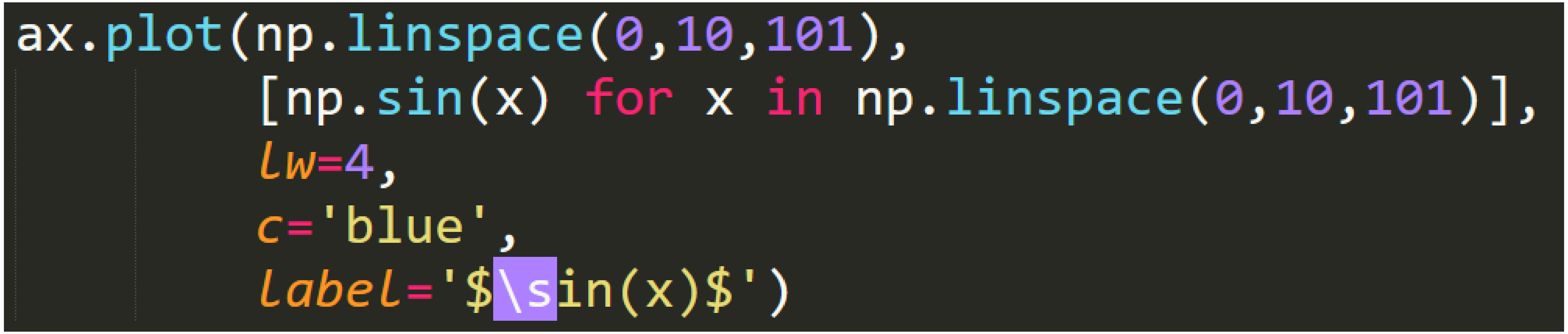
→ Create a figure or canvas object

→ Create an axes object

1. Figure size is extremely large for better scaling after export
2. Subplots are numbered **`[row, column, position]`**



III. LINE PLOT - PLOT THE DATA



```
ax.plot(np.linspace(0,10,101),
        [np.sin(x) for x in np.linspace(0,10,101)],
        lw=4,
        c='blue',
        label='$\sin(x)$')
```

→ x-values

→ y-values

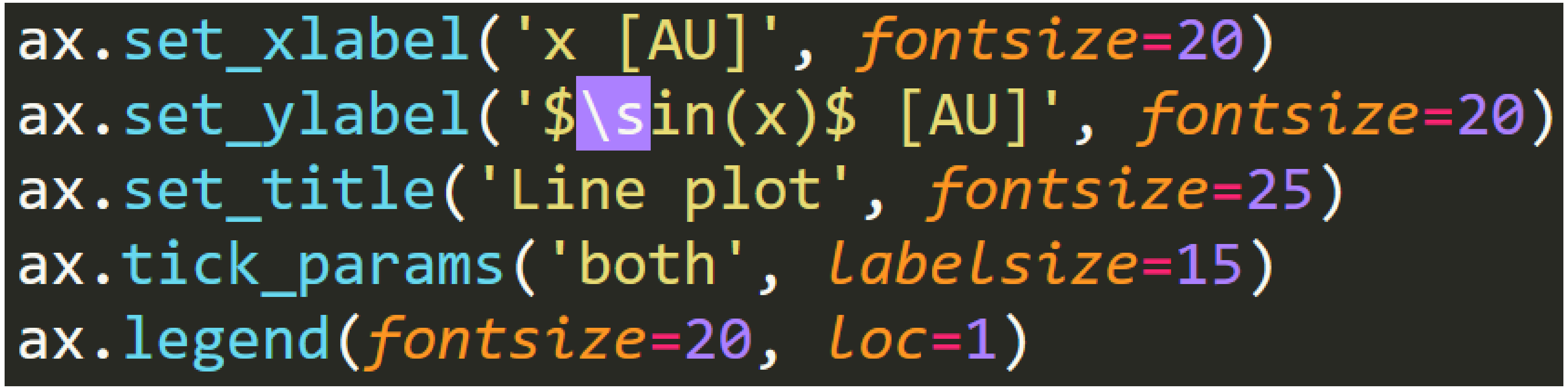
→ line-width

→ colour

→ legend

1. Use for continuous data (is it ever?)
2. *x* and *y* data have to have the same length
3. Different colour maps, *LaTeX* syntax

III. LINE PLOT - LABELLING



```
ax.set_xlabel('x [AU]', fontsize=20)
ax.set_ylabel('$\sin(x)$ [AU]', fontsize=20)
ax.set_title('Line plot', fontsize=25)
ax.tick_params('both', labelsize=15)
ax.legend(fontsize=20, loc=1)
```

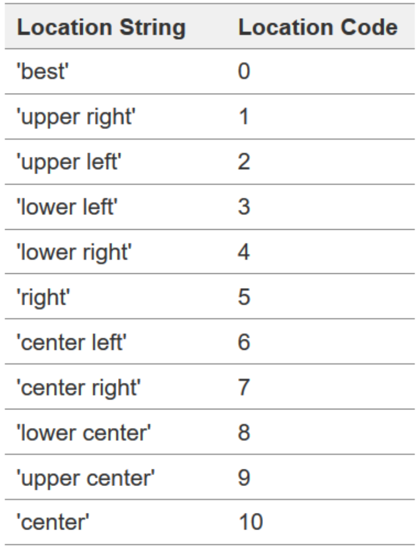
→ x-axis labels

→ y-axis labels

→ Figure title

→ Size of numbers at figure

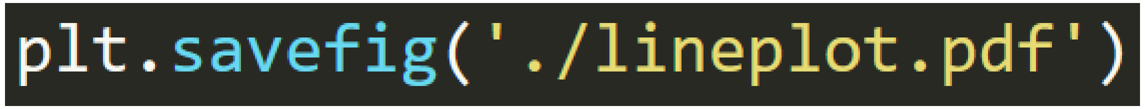
→ Create legend



Location String	Location Code
'best'	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8
'upper center'	9
'center'	10

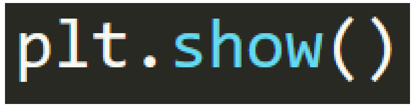
1. Font size has to be increased most of the time
2. Labels can be written using *LaTeX* syntax
3. ← Legend position is an integer

III. LINE PLOT - EXPORT



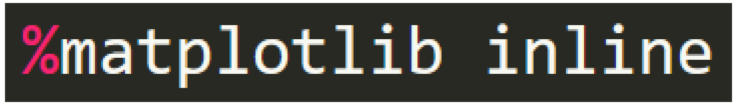
```python
plt.savefig('./lineplot.pdf')
```

→ Export as pdf (vector graphics!)



```python
plt.show()
```

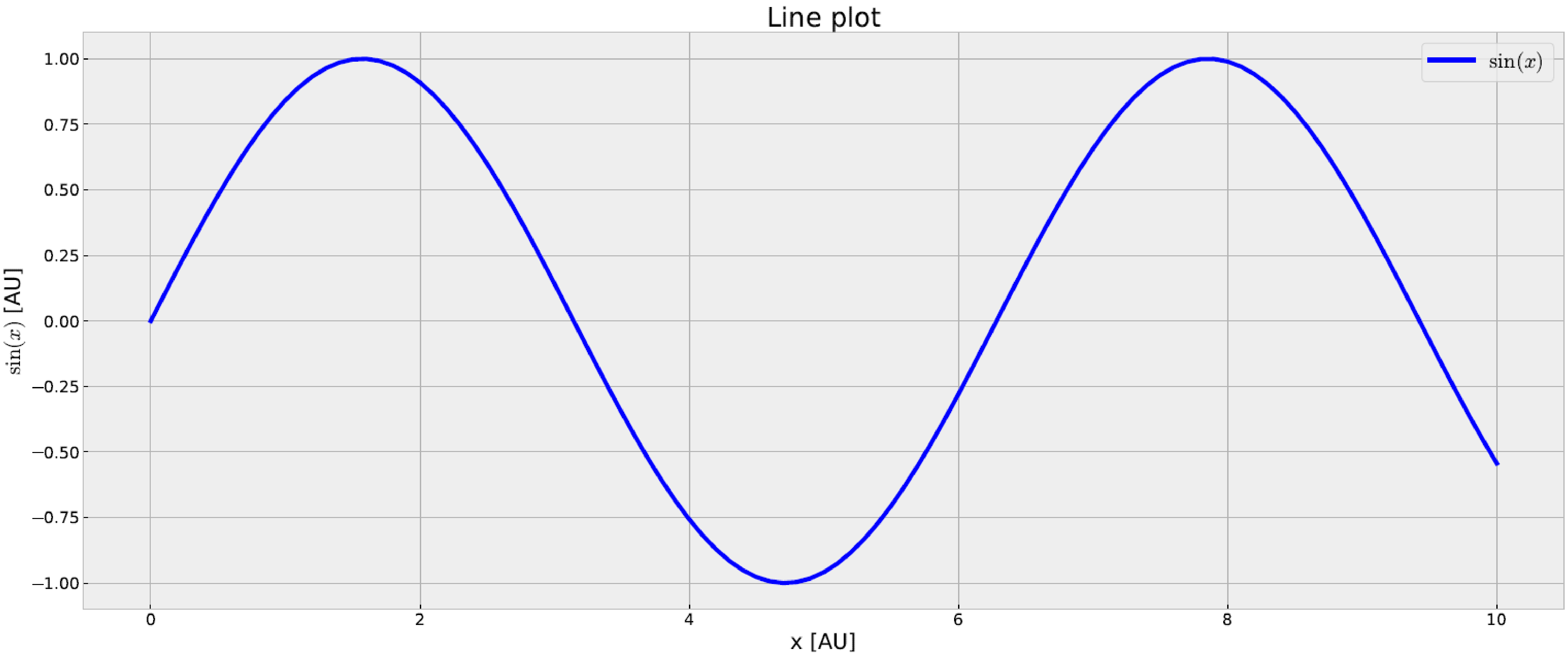
→ Show the figure (open a window)



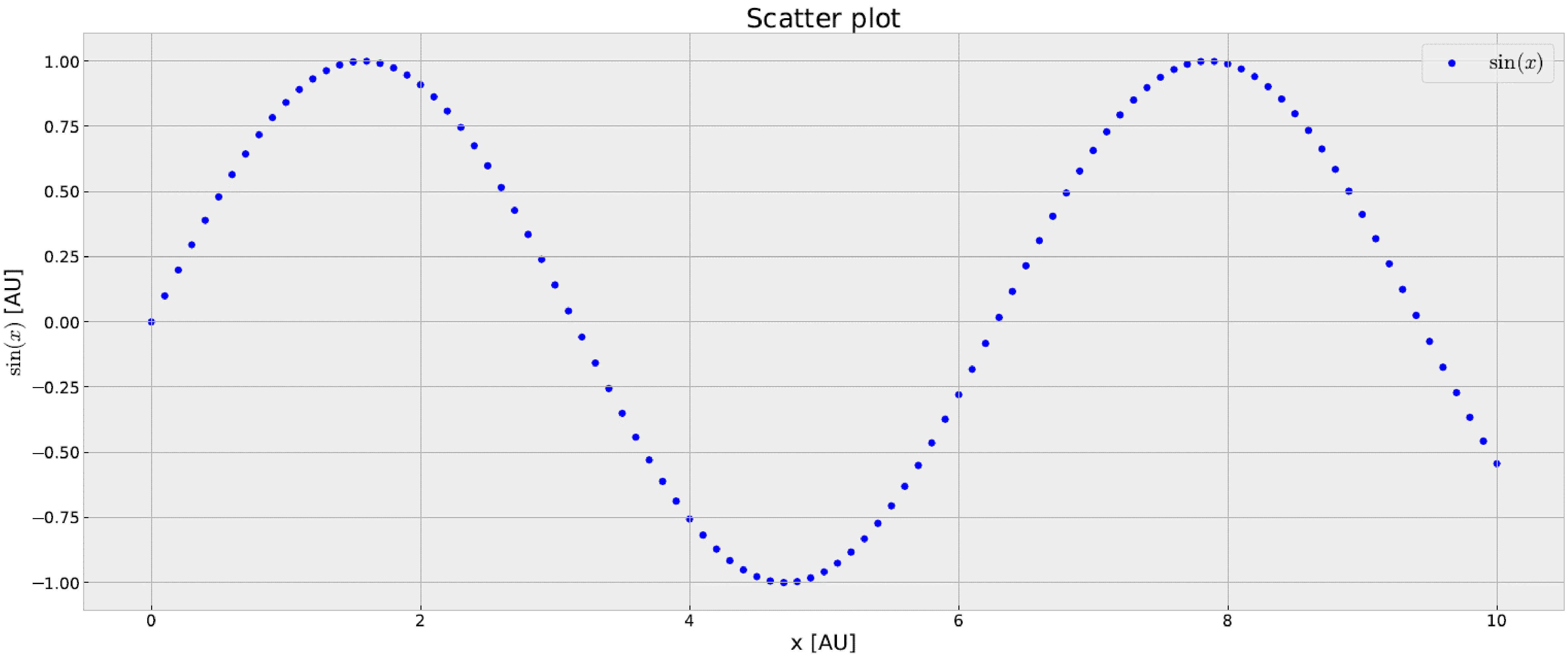
```python
%matplotlib inline
```

→ Show matplotlib figures in notebooks

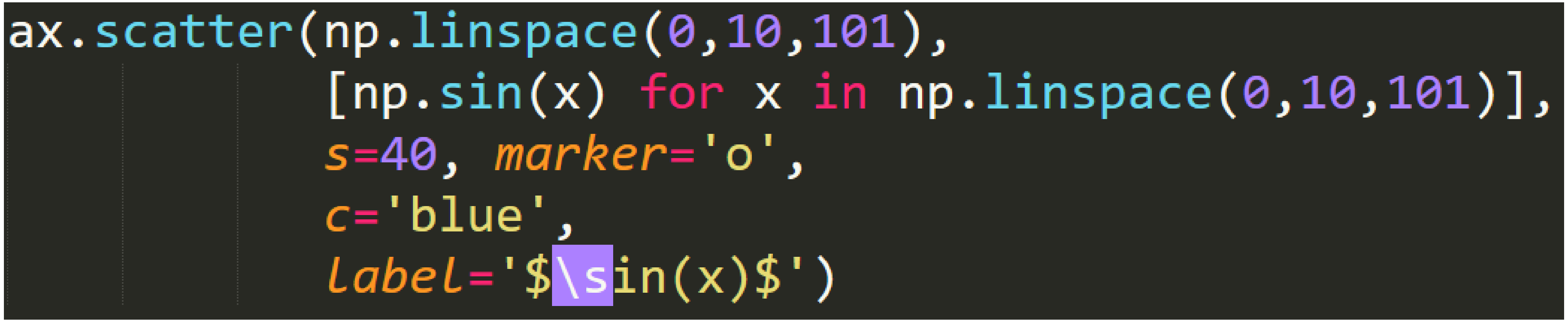
III. LINE PLOT



III. SCATTER PLOT



III. SCATTER PLOT - PLOT THE DATA



```
ax.scatter(np.linspace(0,10,101),
           [np.sin(x) for x in np.linspace(0,10,101)],
           s=40, marker='o',
           c='blue',
           label='$\sin(x)$')
```

→ x-values

→ y-values

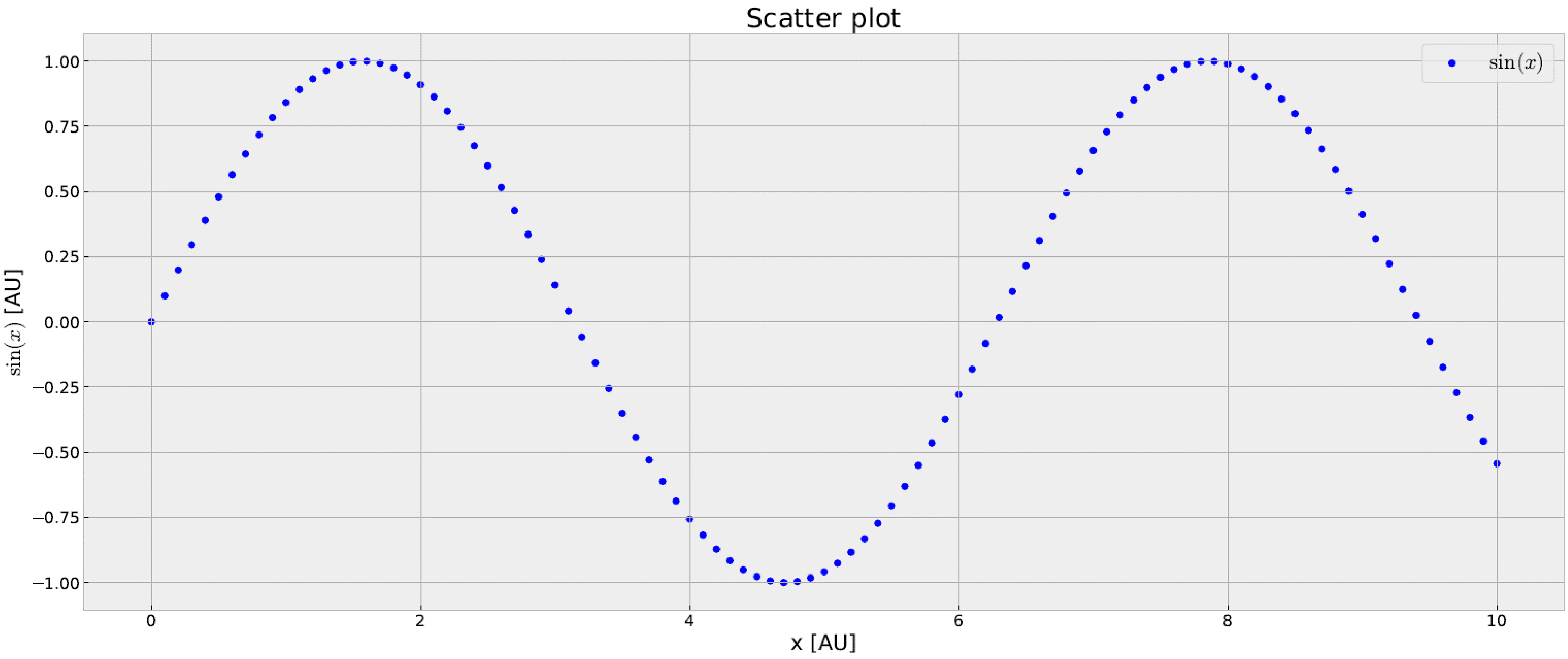
→ marker

→ colour

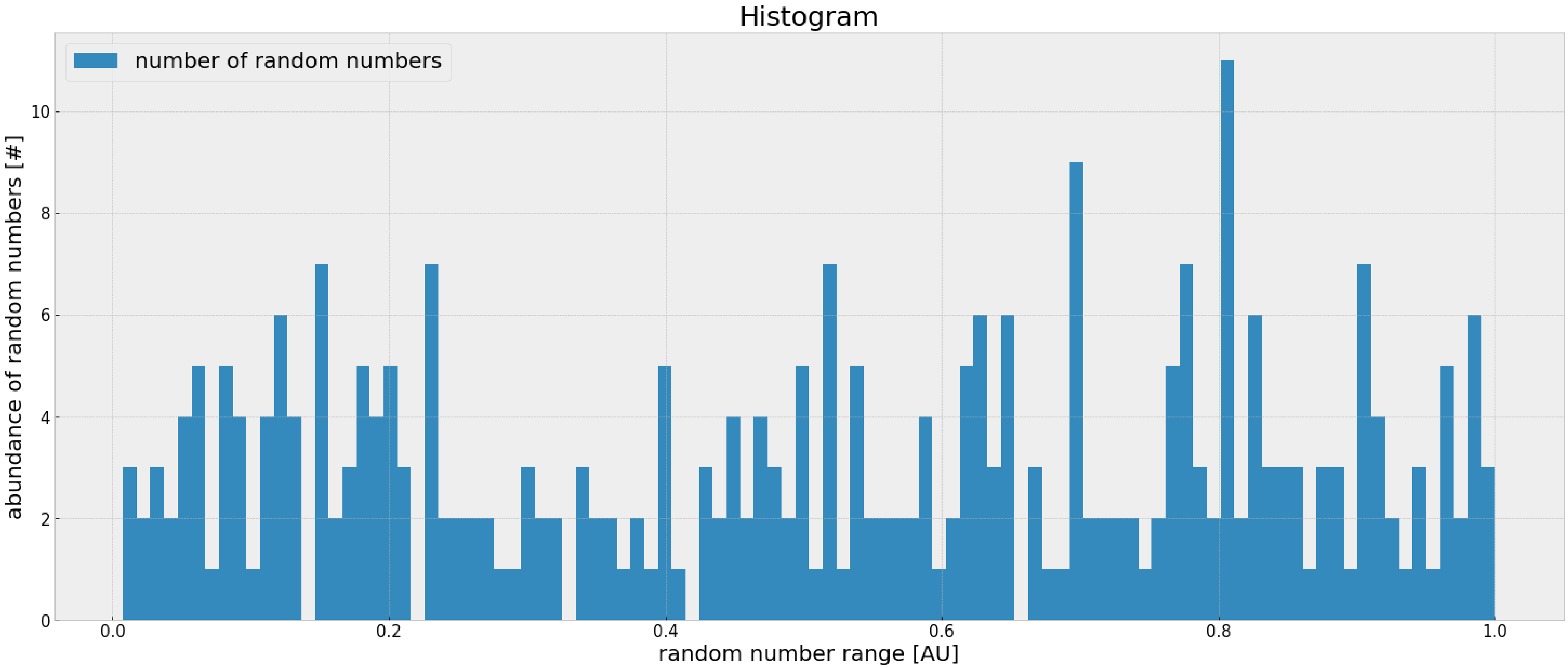
→ legend

1. Most common plot for measured data
2. *x* and *y* data have to have the same length
3. Different colour maps, *LaTeX* syntax

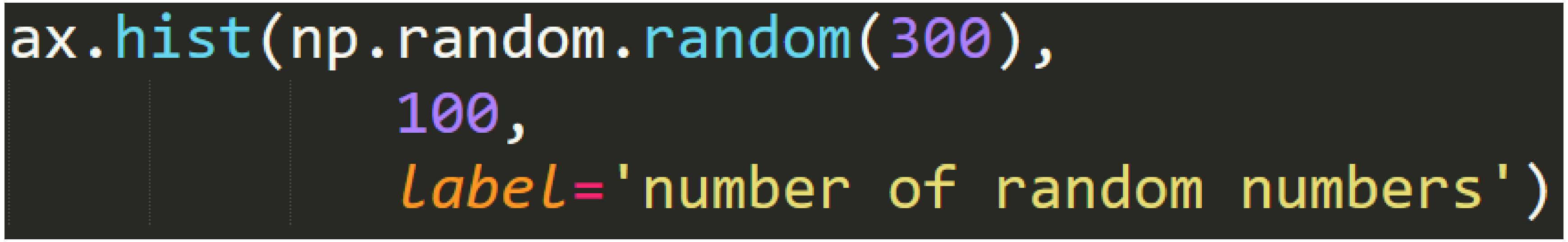
III. SCATTER PLOT



III. HISTOGRAM



III. HISTOGRAM - PLOT THE DATA



```python
ax.hist(np.random.random(300),
            100,
            label='number of random numbers')
```

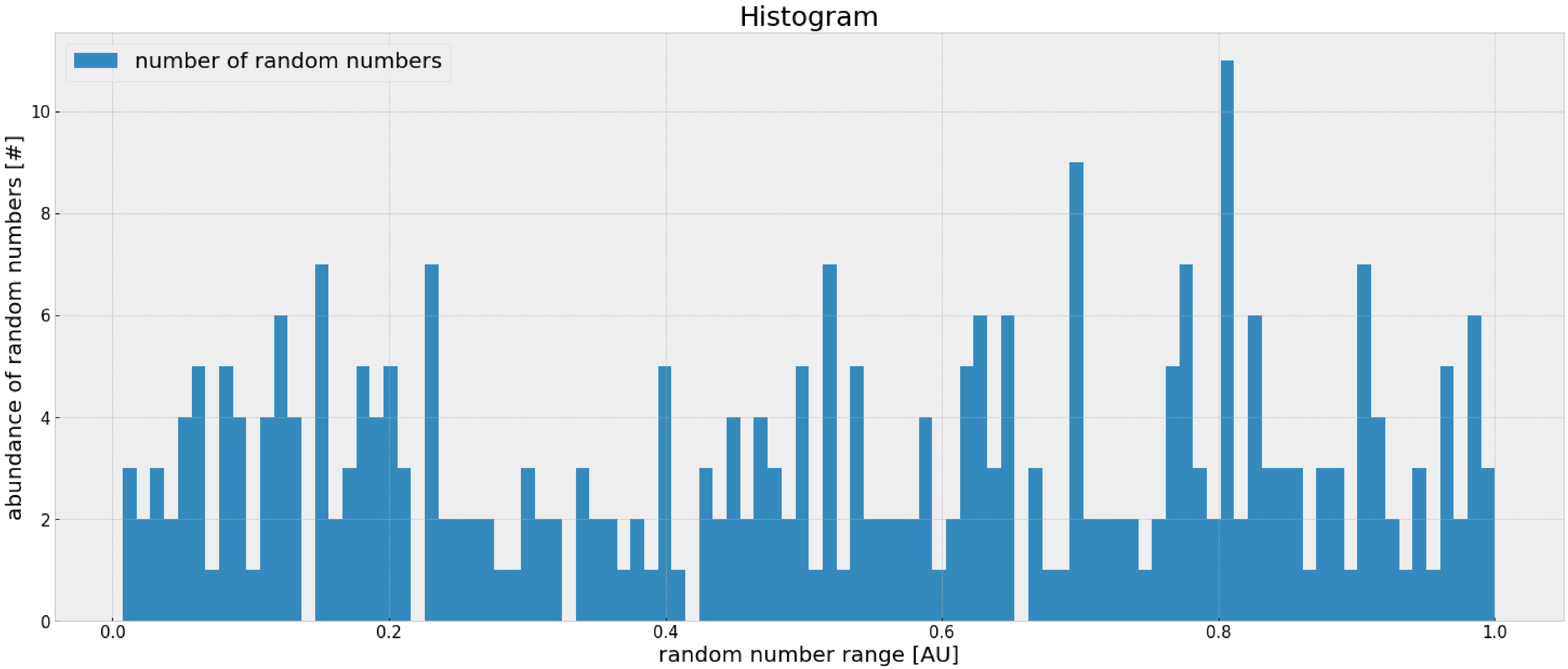
→ Data

→ Number of bins

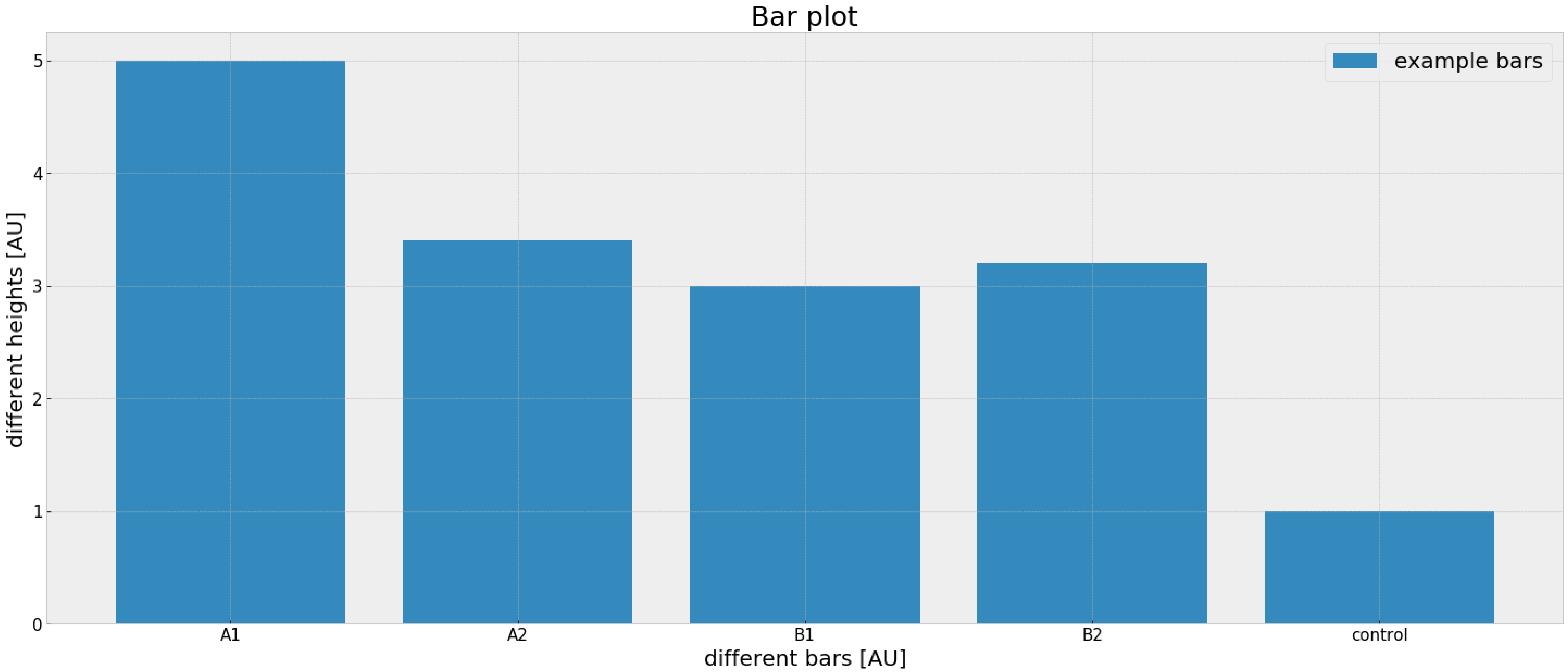
→ Legend label

1. List of numbers and number of bins (can be specified more)
2. Right number of bins is crucial for this visualisation!!

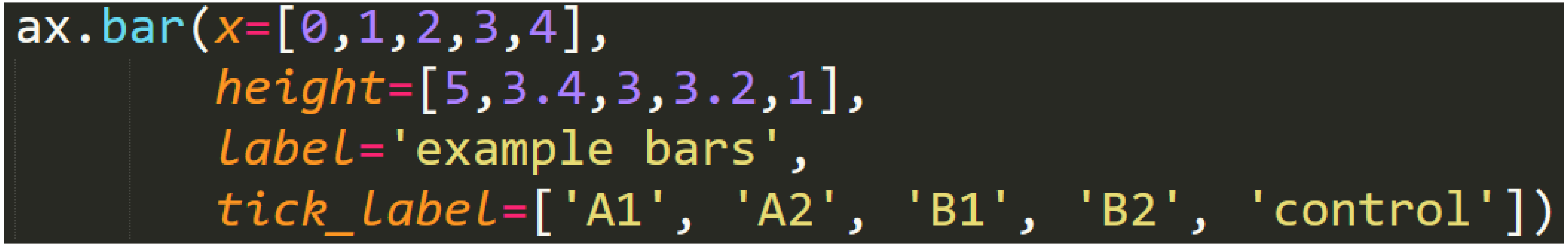
III. HISTOGRAM



III. BAR PLOT


Bar plot
example bars
different heights [AU]
0
1
2
3
4
5
A1
A2
B1
B2
control
different bars [AU]

III. BAR PLOT - PLOT THE DATA



```
ax.bar(x=[0,1,2,3,4],
       height=[5,3.4,3,3.2,1],
       label='example bars',
       tick_label=['A1', 'A2', 'B1', 'B2', 'control'])
```

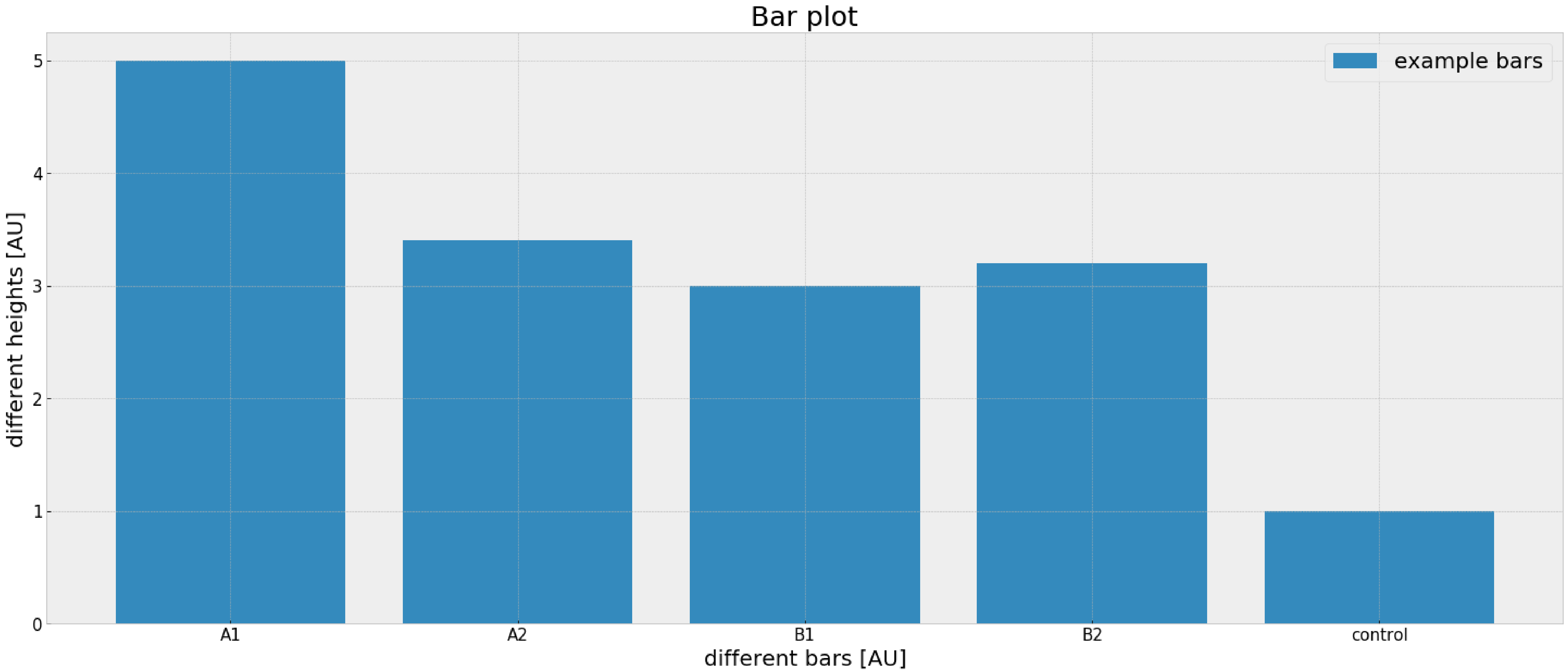
Position of bars

Height of bars

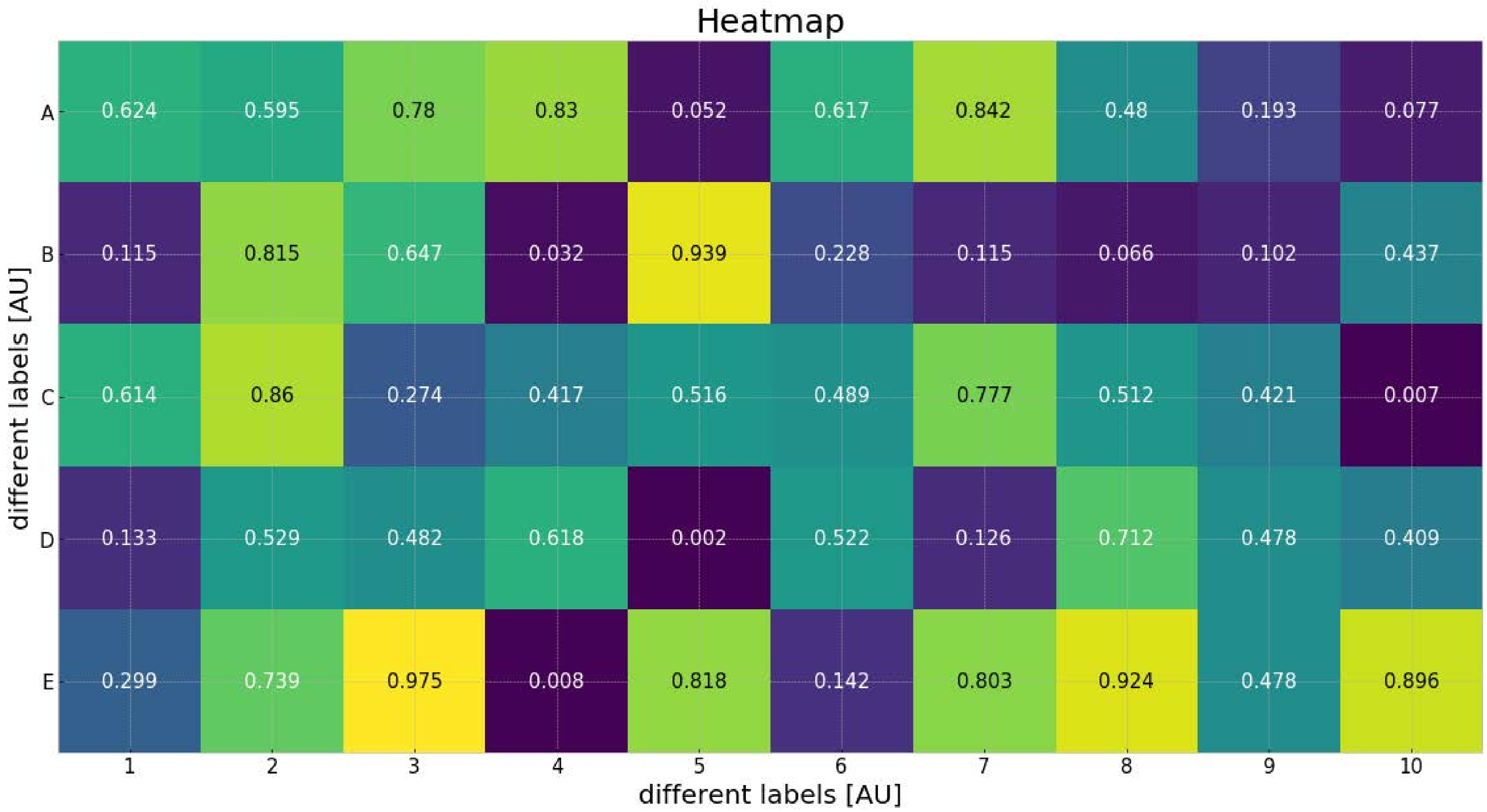
Labels on x axis

1. Set position, height, and label for every bar

III. BAR PLOT

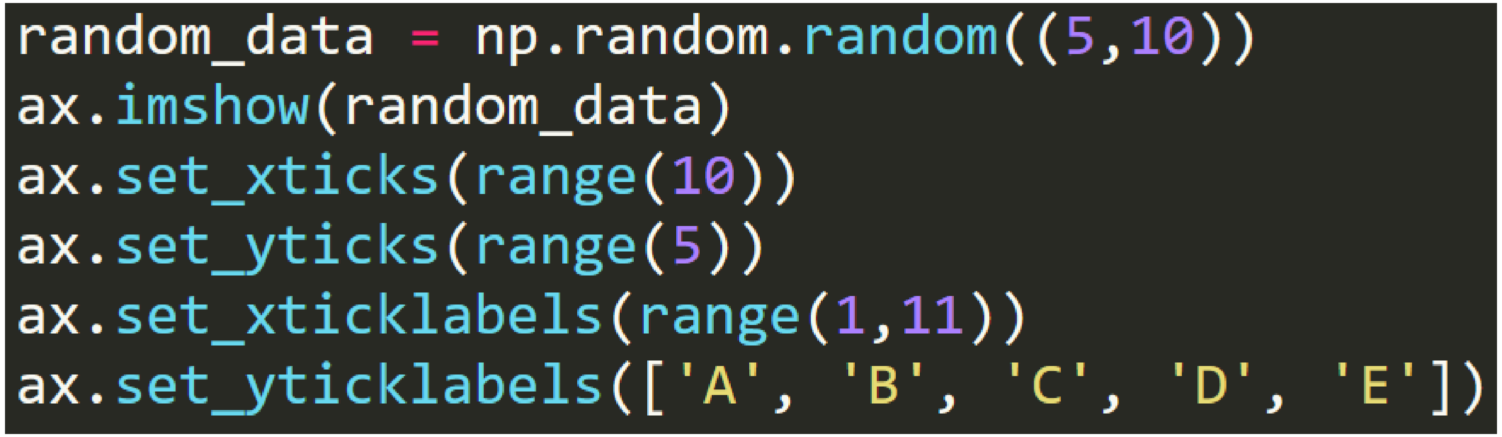

Bar plot
example bars
different heights [AU]
0
1
2
3
4
5
A1
A2
B1
B2
control
different bars [AU]

III. HEATMAP



	1	2	3	4	5	6	7	8	9	10
A	0.624	0.595	0.78	0.83	0.052	0.617	0.842	0.48	0.193	0.077
B	0.115	0.815	0.647	0.032	0.939	0.228	0.115	0.066	0.102	0.437
C	0.614	0.86	0.274	0.417	0.516	0.489	0.777	0.512	0.421	0.007
D	0.133	0.529	0.482	0.618	0.002	0.522	0.126	0.712	0.478	0.409
E	0.299	0.739	0.975	0.008	0.818	0.142	0.803	0.924	0.478	0.896

III. HEATMAP - PLOT THE DATA



```
random_data = np.random.random((5,10))
ax.imshow(random_data)
ax.set_xticks(range(10))
ax.set_yticks(range(5))
ax.set_xticklabels(range(1,11))
ax.set_yticklabels(['A', 'B', 'C', 'D', 'E'])
```

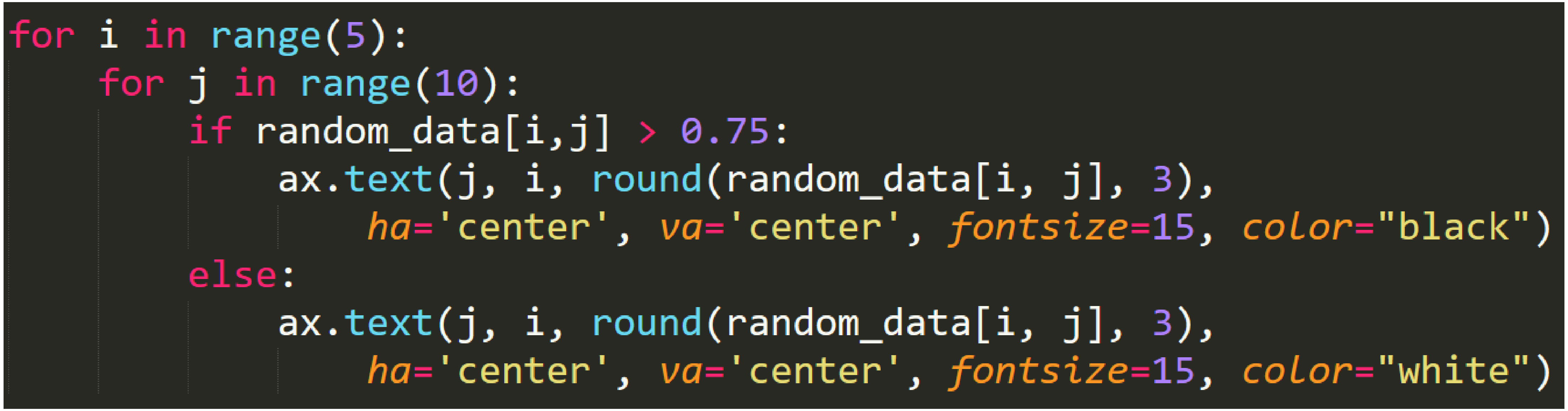
Data as matrix (array of arrays)

Position of x labels

Position of y labels

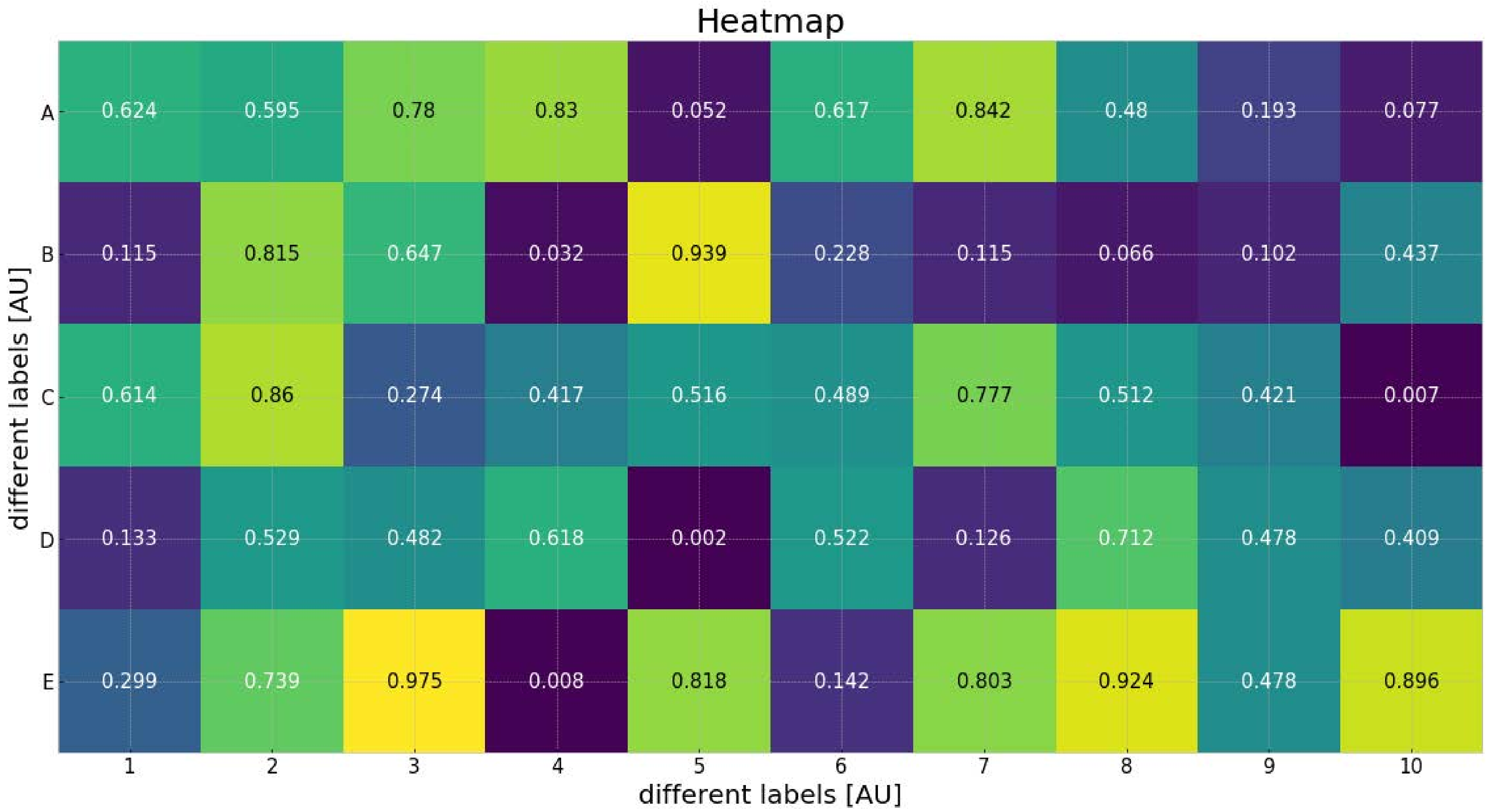
Labels on x axis

Labels on y axis



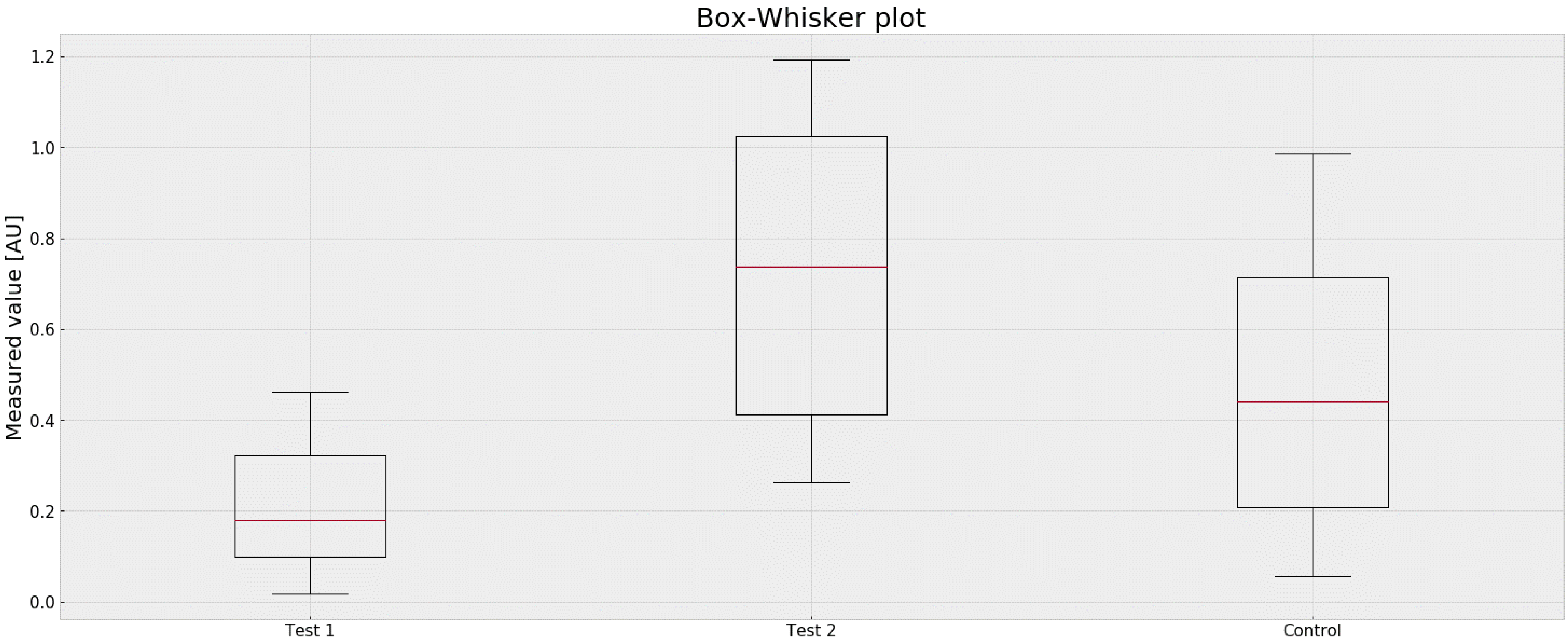
```
for i in range(5):
    for j in range(10):
        if random_data[i,j] > 0.75:
            ax.text(j, i, round(random_data[i, j], 3),
                ha='center', va='center', fontsize=15, color="black")
        else:
            ax.text(j, i, round(random_data[i, j], 3),
                ha='center', va='center', fontsize=15, color="white")
```

III. HEATMAP

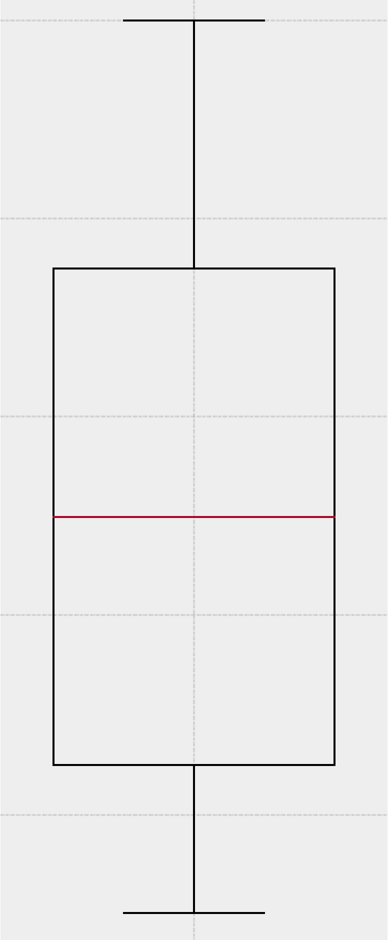


	1	2	3	4	5	6	7	8	9	10
A	0.624	0.595	0.78	0.83	0.052	0.617	0.842	0.48	0.193	0.077
B	0.115	0.815	0.647	0.032	0.939	0.228	0.115	0.066	0.102	0.437
C	0.614	0.86	0.274	0.417	0.516	0.489	0.777	0.512	0.421	0.007
D	0.133	0.529	0.482	0.618	0.002	0.522	0.126	0.712	0.478	0.409
E	0.299	0.739	0.975	0.008	0.818	0.142	0.803	0.924	0.478	0.896

III. BOX-WHISKER PLOT



III. BOX-WHISKER PLOT



Highest value in the data
(in range of 1.5 x IQR)

IQR = Interquartile range

75th percentile

Median

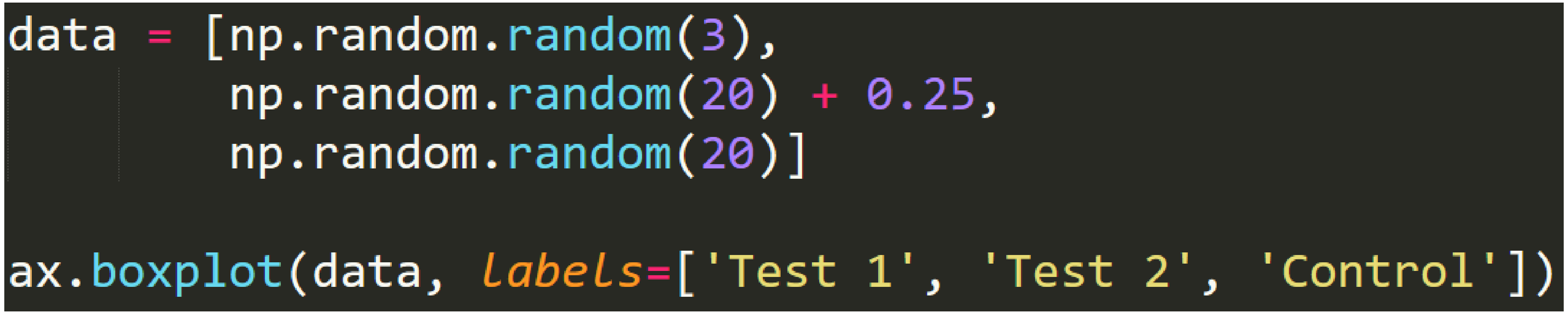
25th percentile

Lowest value in the data
(in range of 1.5 x IQR)

Box-Whisker Plot

1. Good to get important values of a distribution
2. Shows outliers
3. Size of distribution should always be given!!

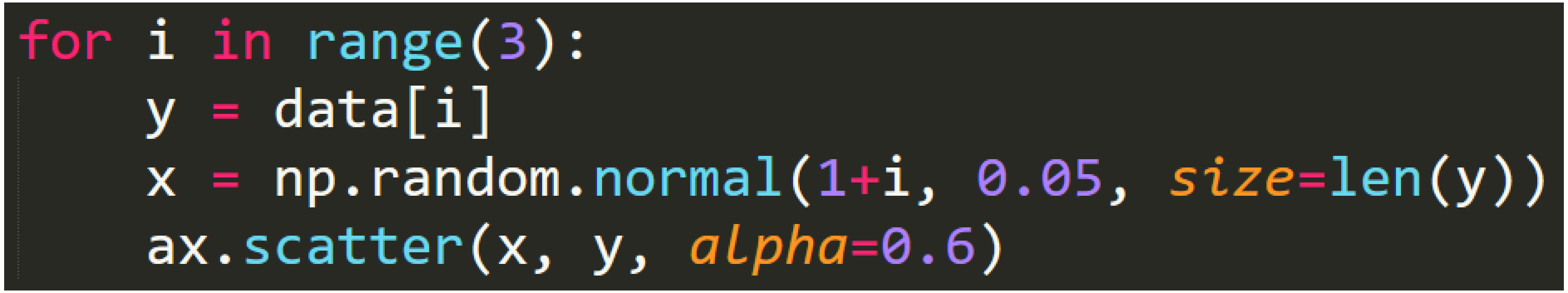
III. BOX-WHISKER PLOT - PLOT THE DATA



```
data = [np.random.random(3),
        np.random.random(20) + 0.25,
        np.random.random(20)]

ax.boxplot(data, labels=['Test 1', 'Test 2', 'Control'])
```

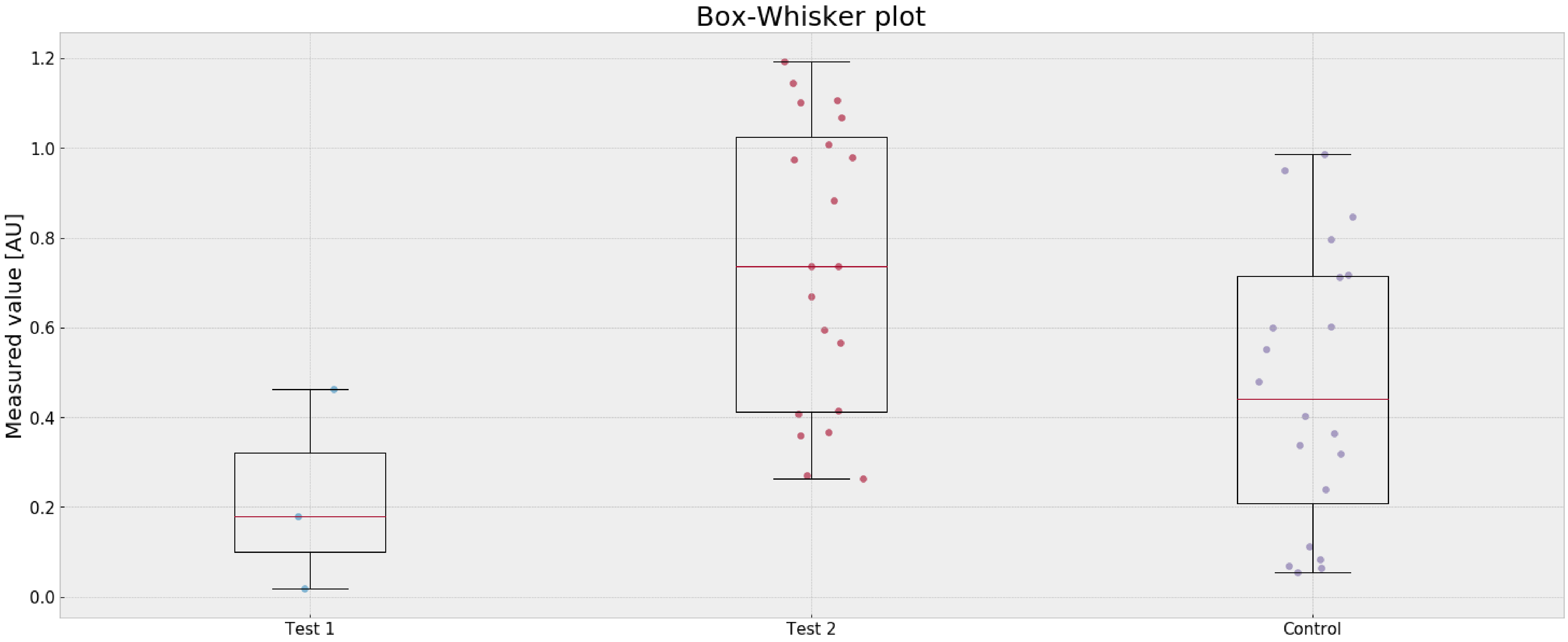
Data as matrix
(array of arrays)



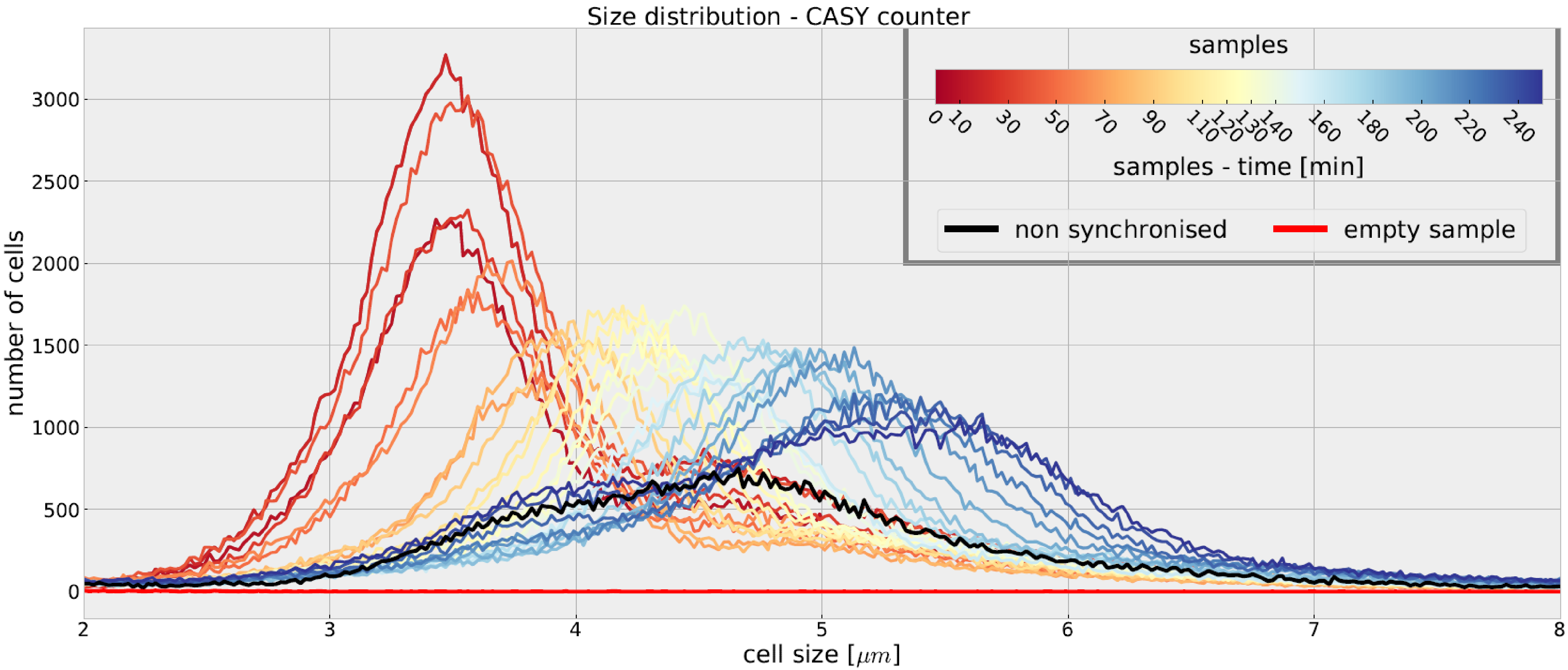
```
for i in range(3):
    y = data[i]
    x = np.random.normal(1+i, 0.05, size=len(y))
    ax.scatter(x, y, alpha=0.6)
```

Plot data points
inside the box plot

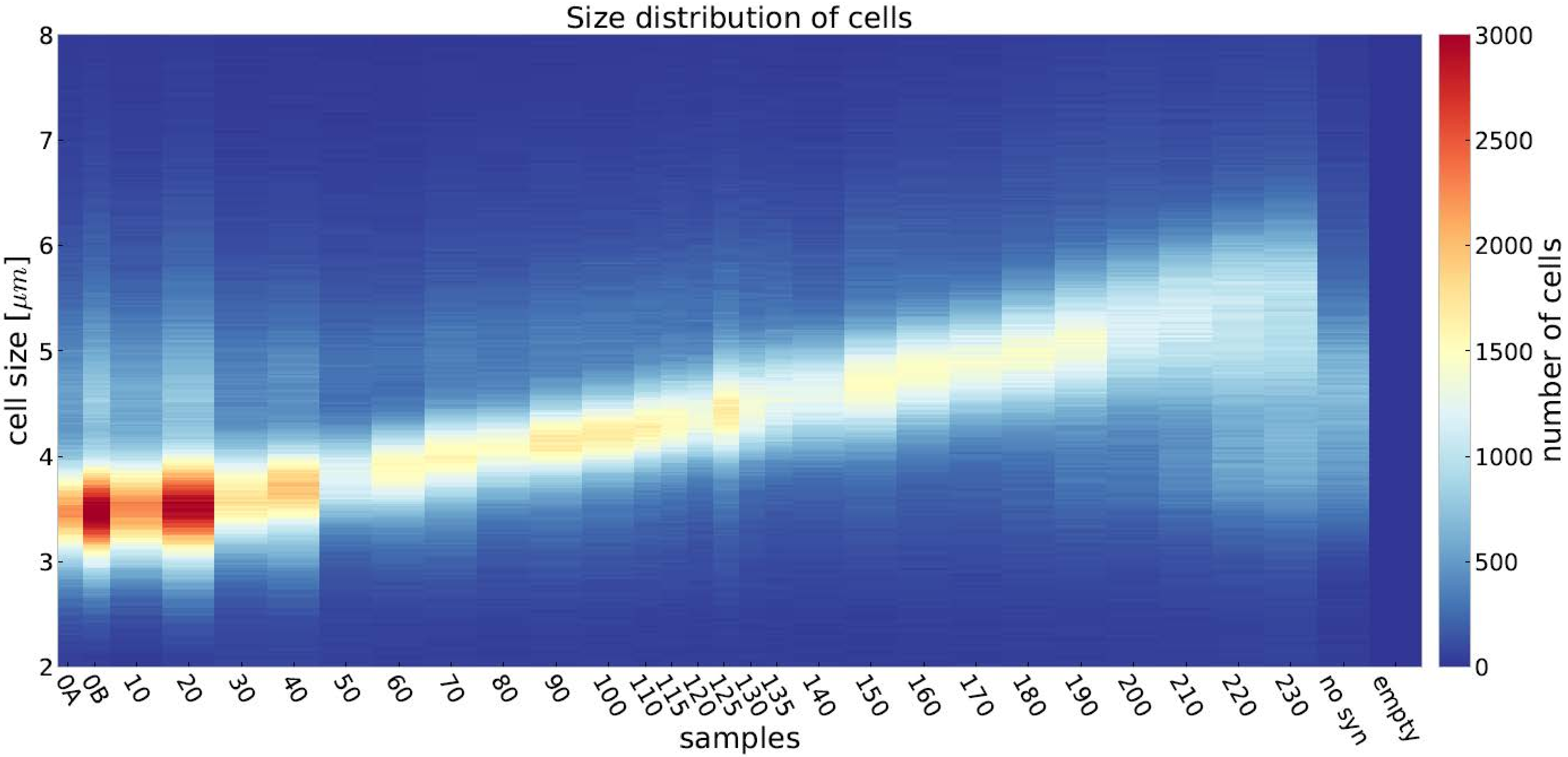
III. BOX-WHISKER PLOT



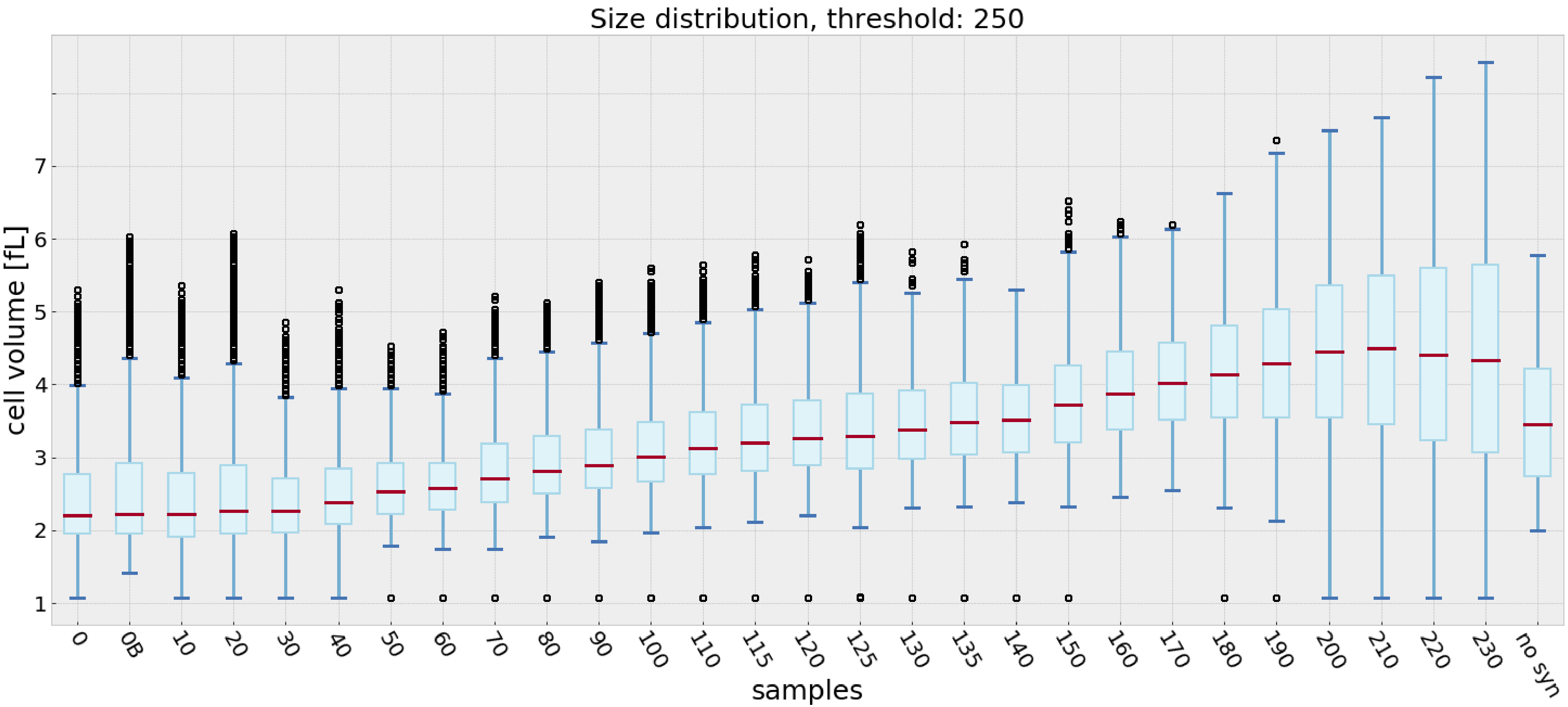
III. MATPLOTLIB EXAMPLES



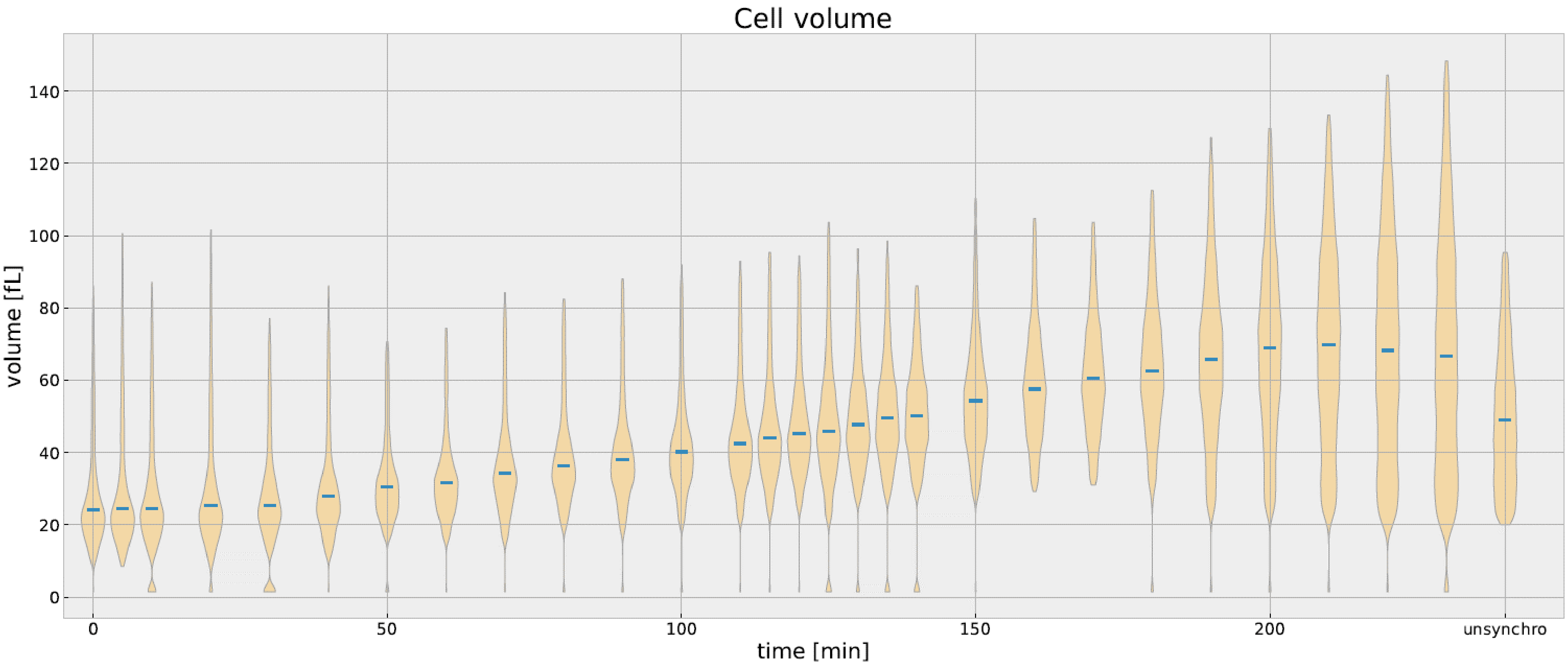
III. MATPLOTLIB EXAMPLES



III. MATPLOTLIB EXAMPLES



III. MATPLOTLIB EXAMPLES



IV. ASSIGNMENT

Plot your data set

V. FURTHER READING

Data visualisation

- Data visualisations catalogue

https://datavizcatalogue.com/index.html

- Worst plots of all times

https://www.biostat.wisc.edu/~kbroman/topten_worstgraphs/

Visualisation in Python

- Python visualisation packages – Dan Saber

https://dsaber.com/2016/10/02/a-dramatic-tour-through-pythons-data-visualization-landscape-including-ggplot-and-altair/

- matplotlib documentation

https://matplotlib.org/