Quarto Document

Sohyun Park

2024-10-04

Table of contents

1 Sections **1**

1.1 Colors . . . 1

1.2 Shapes . . . 2

1.3 Textures . . . 2

1.4 Equations . . . 2

2 Citations **2**

3 Cross References **2**

3.1 Plot . . . 3

3.2 Equation . . . 3

4 Callouts **3**

5 Placing Colorbars **4**

Quarto Document

1 Sections

1.1 Colors

- Red
- Green
- Blue

1.2 Shapes

- Square
- Circle
- Triangle

1.3 Textures

- Smooth
- Bumpy
- Fuzzy

1.4 Equations

Einstein’s theory of special relatively that expresses the equivalence of mass and energy:

$E = mc^2$

2 Citations

Mary says *Hydrophobicity is ubiquitous. Many aquatic and semi-aquatic plants, such as the lotus, utilise hydrophobicity in their self-cleaning mechanisms which reduce their chances of infection from harmful pathogens present in the bodies of water in which they grow [2]. Butterflies have been found to utilise hydrophobicity to ensure that rain droplets which fall on their wings roll off away from their bodies [3]. Proteins have been found to utilise localised hydrophobicity to evacuate surrounding water and enable ligand binding [4]. Detergents utilise the hydrophobicity driven self assembly of amphiphilic molecules into micelle structures to remove grease from clothes [5]. The influence of hydrophobicity is felt in every aspect of our lives.* (Coe, n.d.).

Coe, Mary Kathryn. n.d. “Hydrophobicity Across Length Scales: The Role of Surface Criticality.”

3 Cross References

See Figure 1 in Section 3.1 for a demonstration of a simple plot.

See Equation 1 to better understand standard deviation.

3.1 Plot

```
import matplotlib.pyplot as plt
plt.plot([1,23,2,4])
plt.show()
```

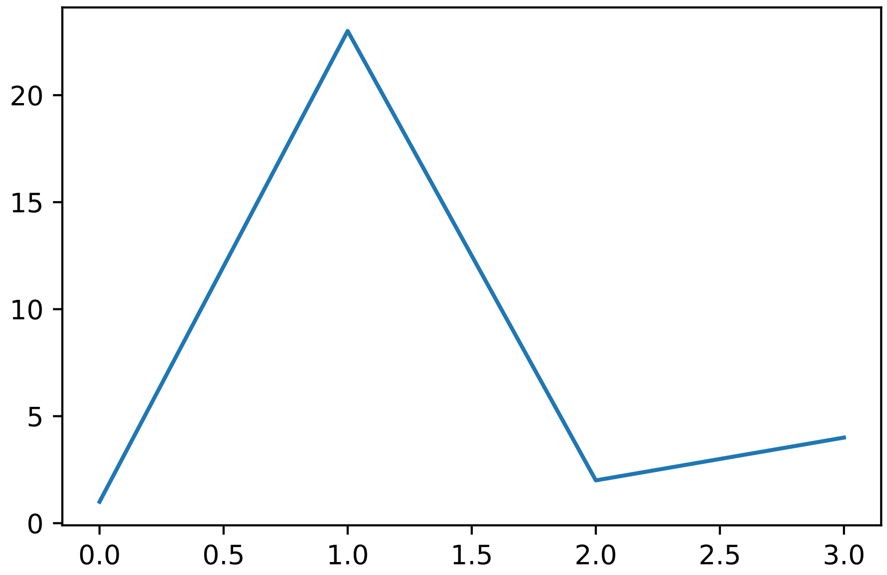


Figure 1: Simple Plot

3.2 Equation

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2} \qquad (1)$$

4 Callouts

ℹ Note

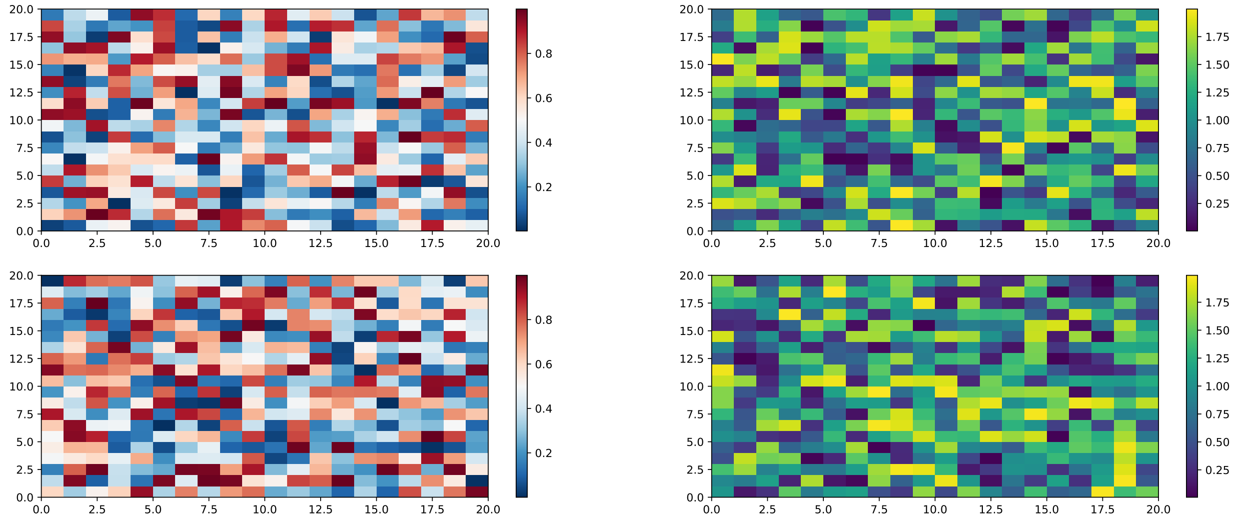
Note that there are five types of callouts, including: `note`, `tip`, `warning`, `caution`, and `important`.

5 Placing Colorbars

Colorbars indicate the quantitative extent of image data. Placing in a figure is non-trivial because room needs to be made for them. The simplest case is just attaching a colorbar to each axes:[1].

```
# #| column: screen-inset
import matplotlib.pyplot as plt
import numpy as np

fig, axs = plt.subplots(2, 2)
fig.set_size_inches(20, 8)
cmaps = ['RdBu_r', 'viridis']
for col in range(2):
    for row in range(2):
        ax = axs[row, col]
        pcm = ax.pcolormesh(
          np.random.random((20, 20)) * (col + 1),
          cmap=cmaps[col]
        )
        fig.colorbar(pcm, ax=ax)
plt.show()
```



[1] See the Matplotlib Gallery to explore colorbars further