Data Visualization

02

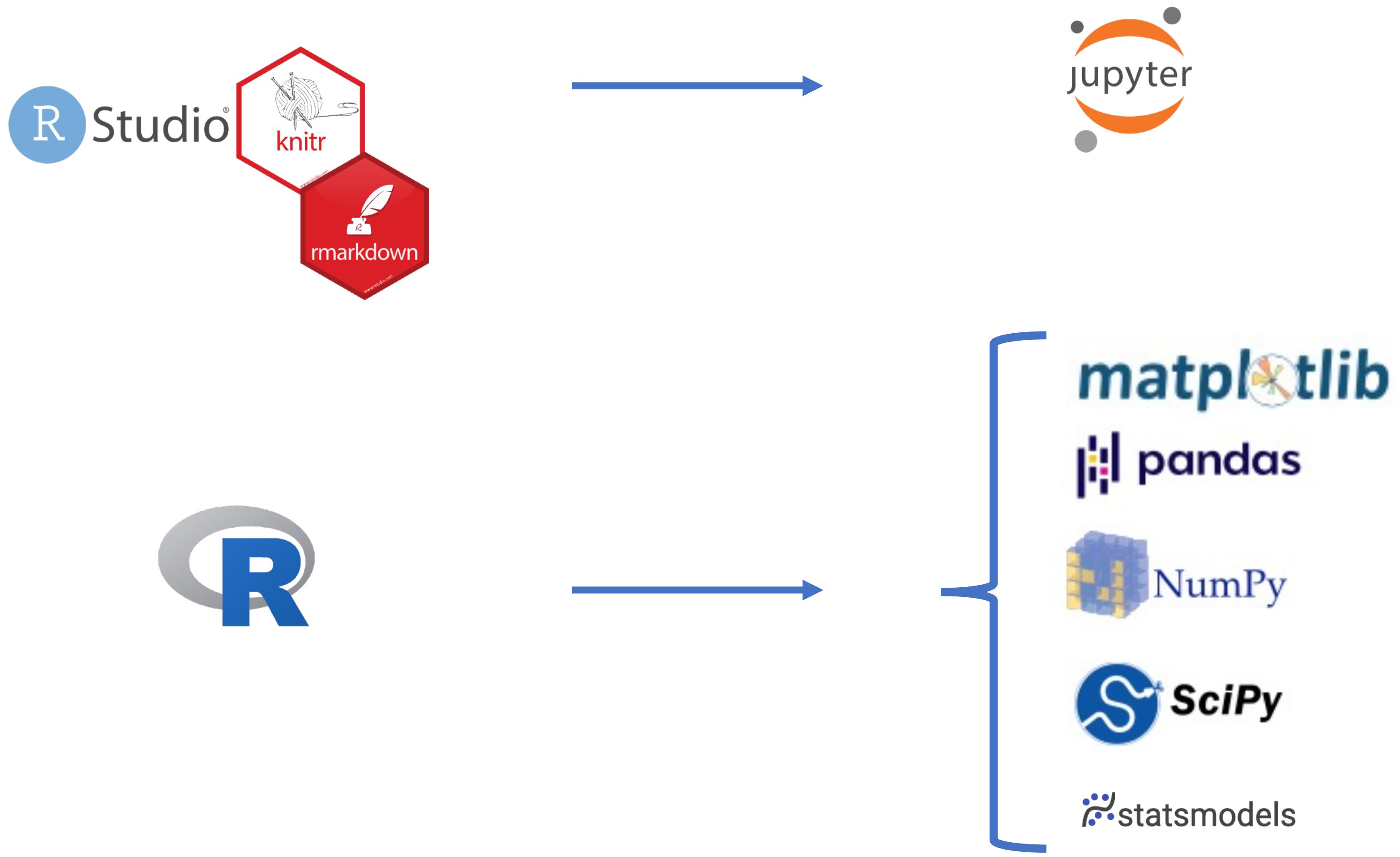
Prof. Dr. Phillipp Torkler

Technical Requirements

Python And R Packages In Comparison

TECHNISCHE HOCHSCHULE DEGGENDORF

R **Python**



Keeping Your Packages Organized

CONDA

https://docs.conda.io/en/latest/index.html

Package, dependency and environment management for any language—Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN, and more.

Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.

A package and environment management system helps you to manage your environments for different projects!

Reproducible Research / Product Development

Reproducible research is based upon the concept that research results and in particular the underlying data analysis of scientific results are published together with scientific publications. As a consequence, the raw data as well as the analysis are published and shared with the community so that findings and claims that are made in publications can be verified.

As data analysis is going to get more and more complex the need for reproducible research is increased.

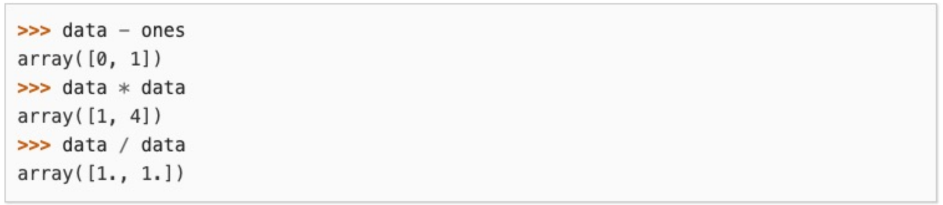
knitr/markdown as juptyer notebooks/lab serve the need to make data analysis easily accessible and to agree upon a standard format how data analysis shall be shared and documented.

Guides Are Always Worth A Look! Again and Again

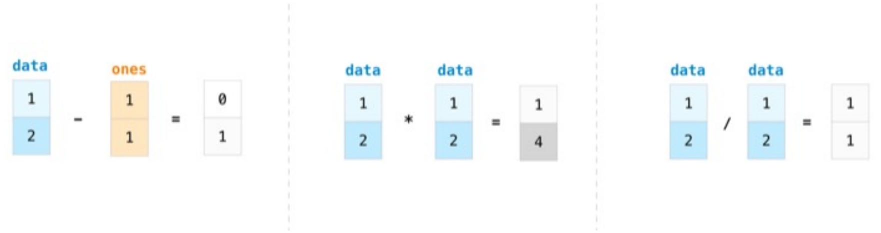
Both, the matplotlib as well as the NumPy guides are written very well. The NumPy guide also has visual representations of the commands that help to understand the behaviour. It is always good to go back to these guides if you forgot what a function really does!

https://numpy.org/doc/stable/user/absolute_beginners.html

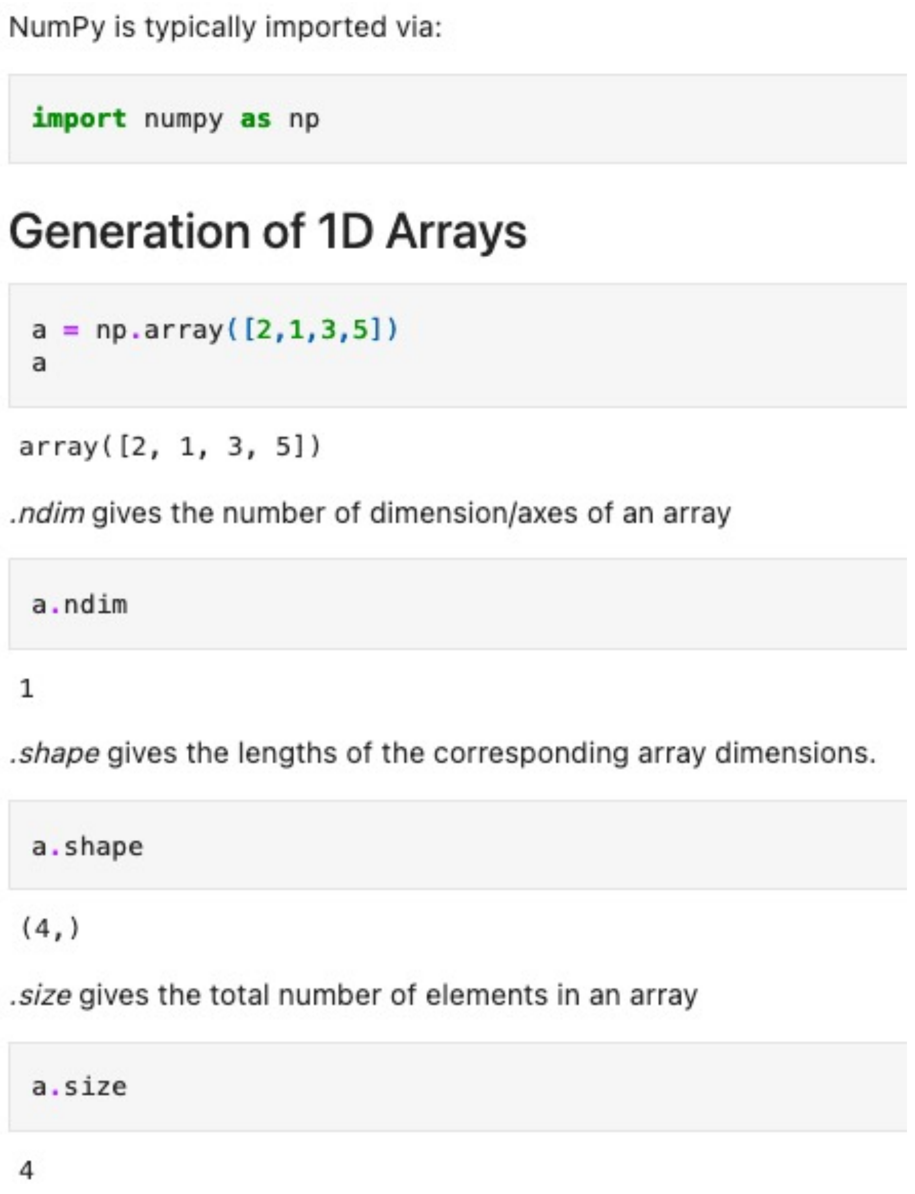
https://matplotlib.org/stable/tutorials/introductory/usage.html



```
>>> data - ones
array([0, 1])
>>> data * data
array([1, 4])
>>> data / data
array([1., 1.])
```



NumPy and Matplotlib



NumPy is typically imported via:

```
import numpy as np
```

Generation of 1D Arrays

```
a = np.array([2,1,3,5])
a
```

```
array([2, 1, 3, 5])
```

.ndim gives the number of dimension/axes of an array

```
a.ndim
```

```
1
```

.shape gives the lengths of the corresponding array dimensions.

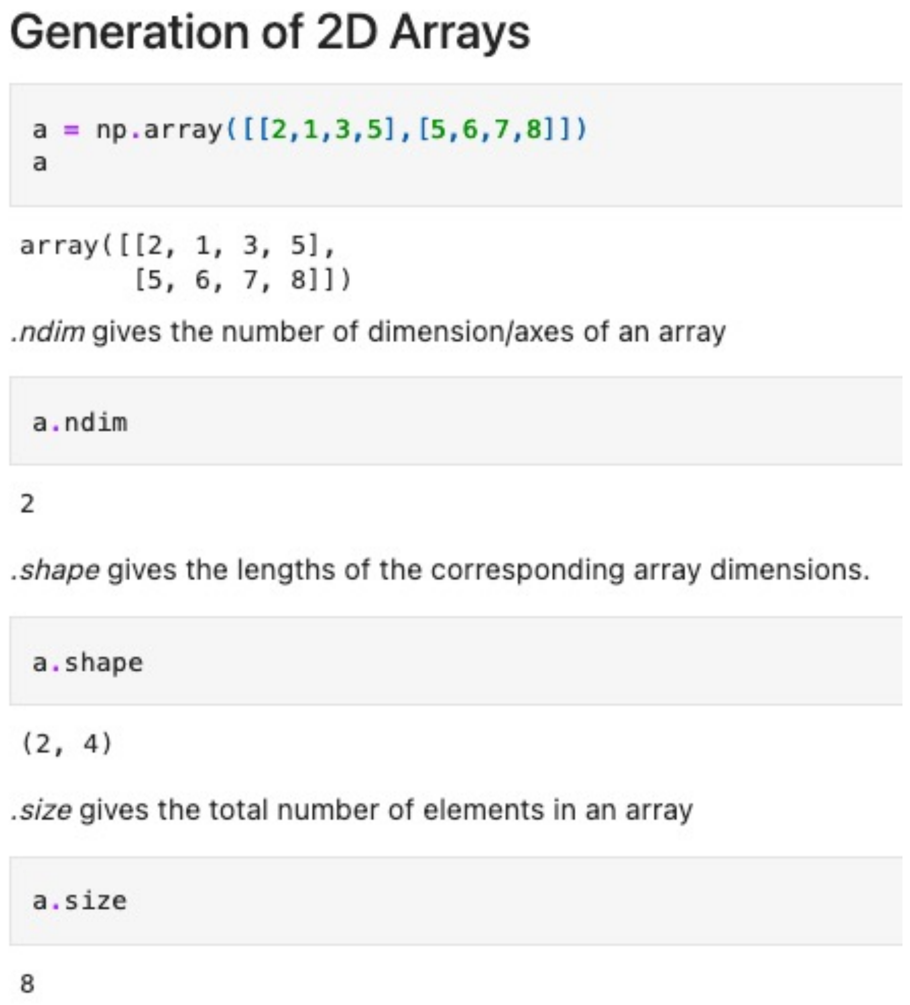
```
a.shape
```

```
(4,)
```

.size gives the total number of elements in an array

```
a.size
```

```
4
```



Generation of 2D Arrays

```
a = np.array([[2,1,3,5],[5,6,7,8]])
a
```

```
array([[2, 1, 3, 5],
       [5, 6, 7, 8]])
```

.ndim gives the number of dimension/axes of an array

```
a.ndim
```

```
2
```

.shape gives the lengths of the corresponding array dimensions.

```
a.shape
```

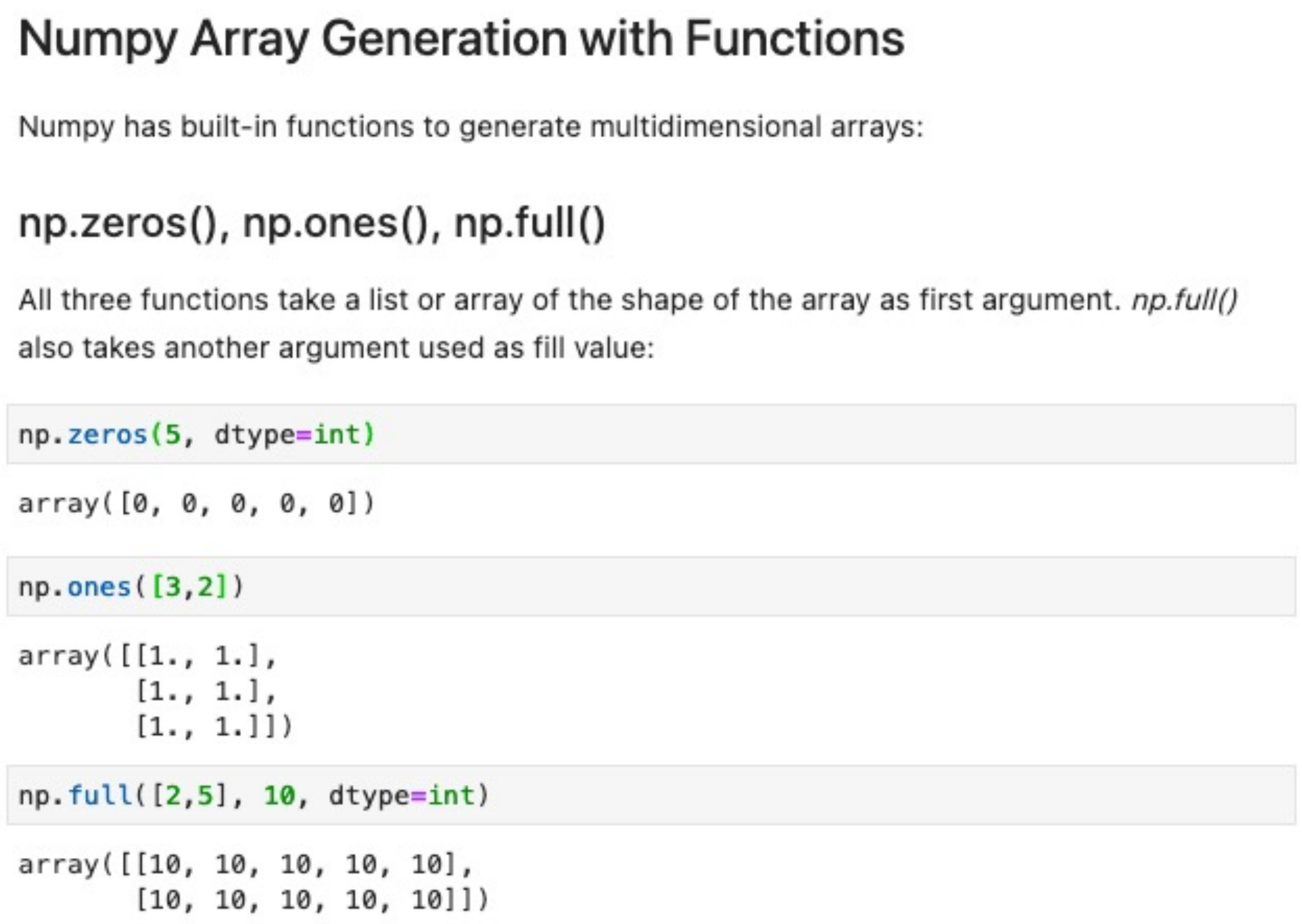
```
(2, 4)
```

.size gives the total number of elements in an array

```
a.size
```

```
8
```

NumPy and Matplotlib



Numpy Array Generation with Functions

Numpy has built-in functions to generate multidimensional arrays:

np.zeros(), np.ones(), np.full()

All three functions take a list or array of the shape of the array as first argument. *np.full()* also takes another argument used as fill value:

```
np.zeros(5, dtype=int)
```

```
array([0, 0, 0, 0, 0])
```

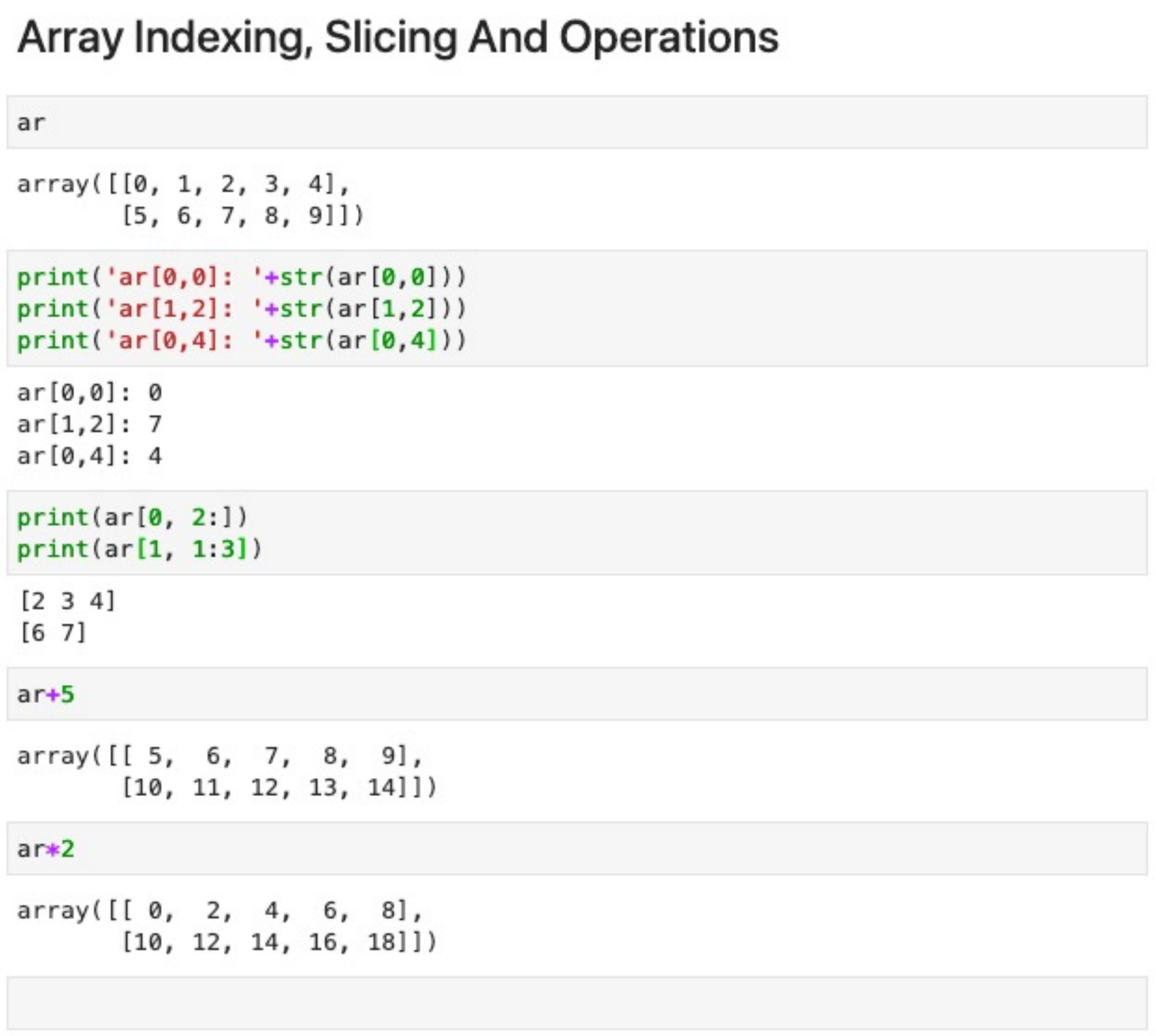
```
np.ones([3,2])
```

```
array([[1., 1.],
       [1., 1.],
       [1., 1.]])
```

```
np.full([2,5], 10, dtype=int)
```

```
array([[10, 10, 10, 10, 10],
       [10, 10, 10, 10, 10]])
```

NumPy and Matplotlib



Array Indexing, Slicing And Operations

```
ar
```

```
array([[0, 1, 2, 3, 4],
       [5, 6, 7, 8, 9]])
```

```
print('ar[0,0]: '+str(ar[0,0]))
print('ar[1,2]: '+str(ar[1,2]))
print('ar[0,4]: '+str(ar[0,4]))
```

```
ar[0,0]: 0
ar[1,2]: 7
ar[0,4]: 4
```

```
print(ar[0, 2:])
print(ar[1, 1:3])
```

```
[2 3 4]
[6 7]
```

```
ar+5
```

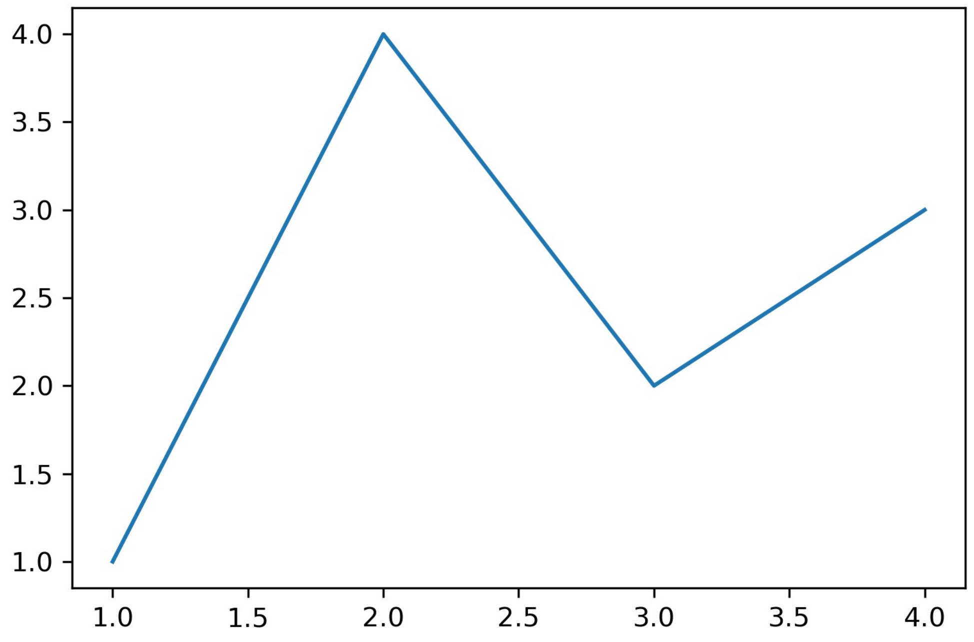
```
array([[ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14]])
```

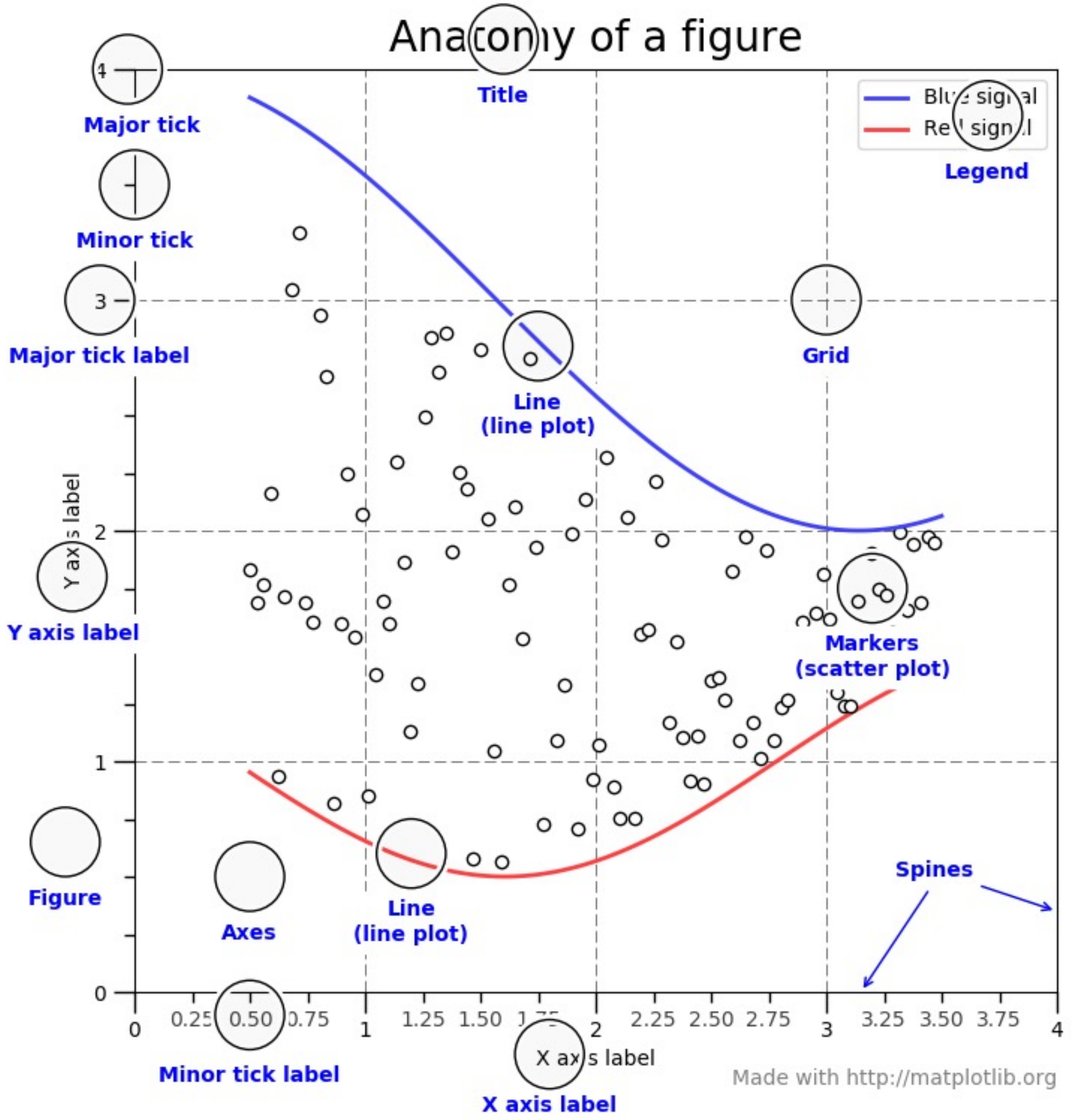
```
ar*2
```

```
array([[ 0,  2,  4,  6,  8],
       [10, 12, 14, 16, 18]])
```

Matplotlib

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

fig, ax = plt.subplots()  # Create a figure containing a single axes.
ax.plot([1, 2, 3, 4], [1, 4, 2, 3])  # Plot some data on the axes.
```





matplotlib user guide:

https://matplotlib.org/stable/tutorials/introductory/usage.html

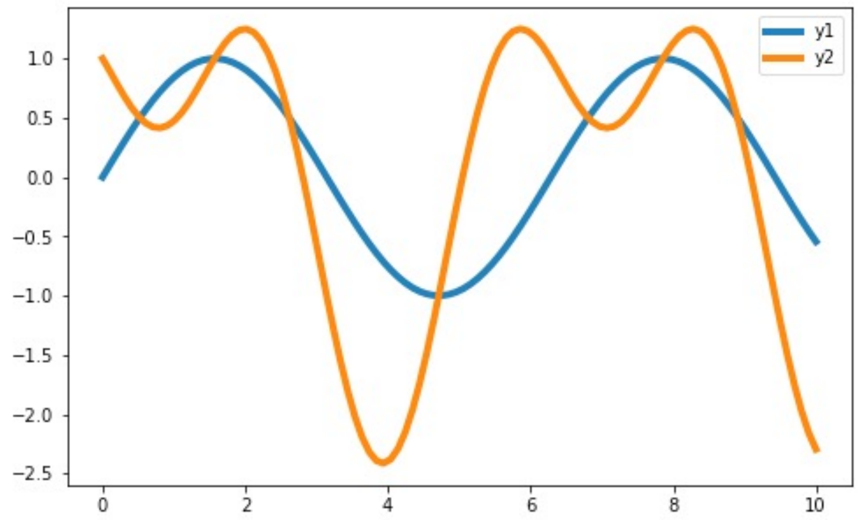
Matplotlib

```
x = np.linspace(0,10,100)
print(np.round(x[0:10],2))
```

```
[0.   0.1  0.2  0.3  0.4  0.51 0.61 0.71 0.81 0.91]
```

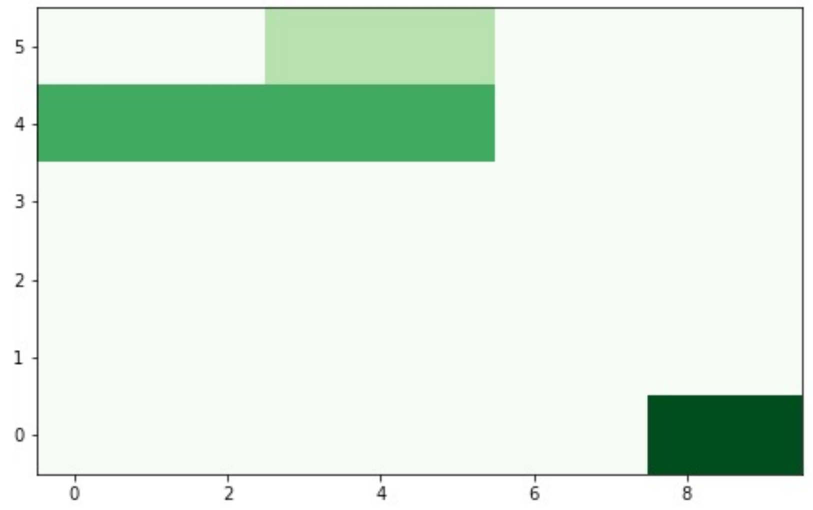
```
y1 = np.sin(x)
y2 = np.sin(x)-np.sin(2*x)+np.cos(x)

fig, ax = plt.subplots()

fig.set_size_inches(8,5)

ax.plot(x,y1, linewidth=4, label='y1')
ax.plot(x, y2, linewidth=4, label='y2')

ax.legend()
```

```
<matplotlib.legend.Legend at 0x7fcd68103af0>
```



```
Z = np.full([6,10], 0, dtype=int)
Z[0,0:3] = np.array([0,0,0])
Z[0,3:6] = np.array([1,1,1])
Z[1,0:6] = np.array([2,2,2,2,2,2])
Z[5,8:10] = np.array([3,3])

x = np.arange(0, Z.shape[1], 1)  # len = 10
y = np.arange(0, Z.shape[0], 1)  # len = 6

fig, ax = plt.subplots()
fig.set_size_inches(8,5)

ax.pcolormesh(x, y[::-1], Z, shading='nearest', cmap='Greens', zorder=0)

print(Z)
```

```
[[0 0 0 1 1 1 0 0 0 0]
 [2 2 2 2 2 2 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 3 3]]
```



Programming Exercise

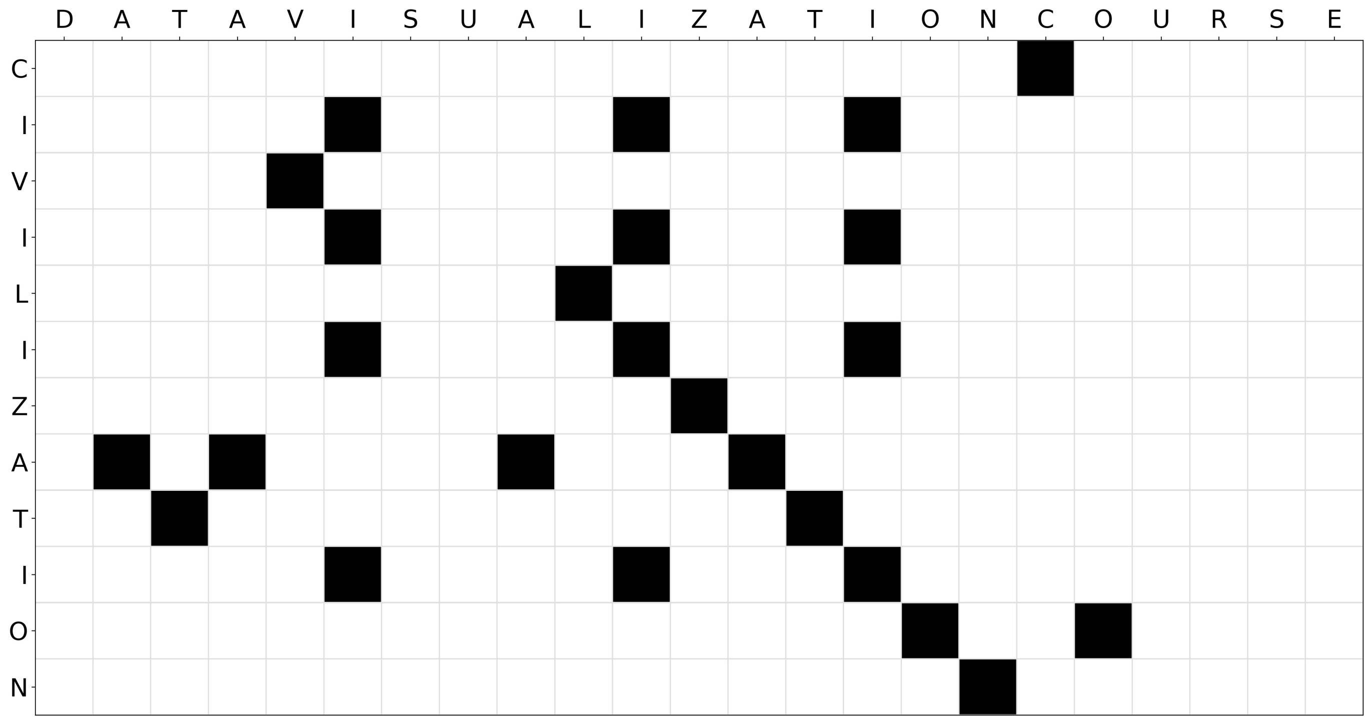
- Create A Dot Plot From Scratch Using Python/matplotlib

A Dot Plot Visualizes Sequence Similarity

A simple and visual way of comparing two sequences s_1 and s_2 of length n and m with each other is a dot plot. The sequences s_1 and s_2 span a $n \times m$ matrix M where a dot is plotted for $M_{i,j}$ if $s_1[i] = s_2[j]$.

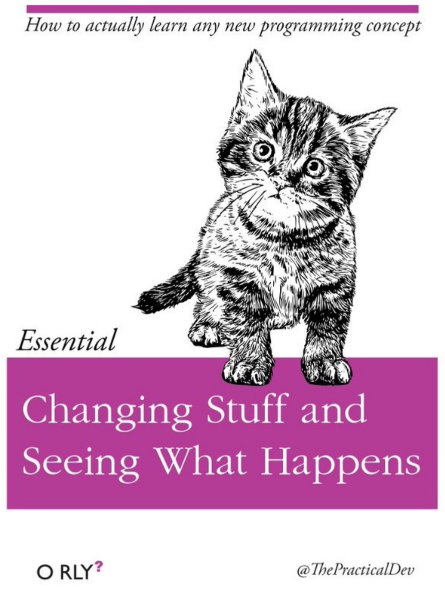
$$M_{i,j} = \begin{cases} 1, \text{if } s_1[i] = s_2\,[j] \\ 0, \ \text{if } s_1[i] \neq s_2[j] \end{cases}$$

The figure below shows a dot plot for $s_1 = CIVILIZATION$ and $s_2 = DATAVISUALIZATIONCOURSE$.



Matplotlib user guide:

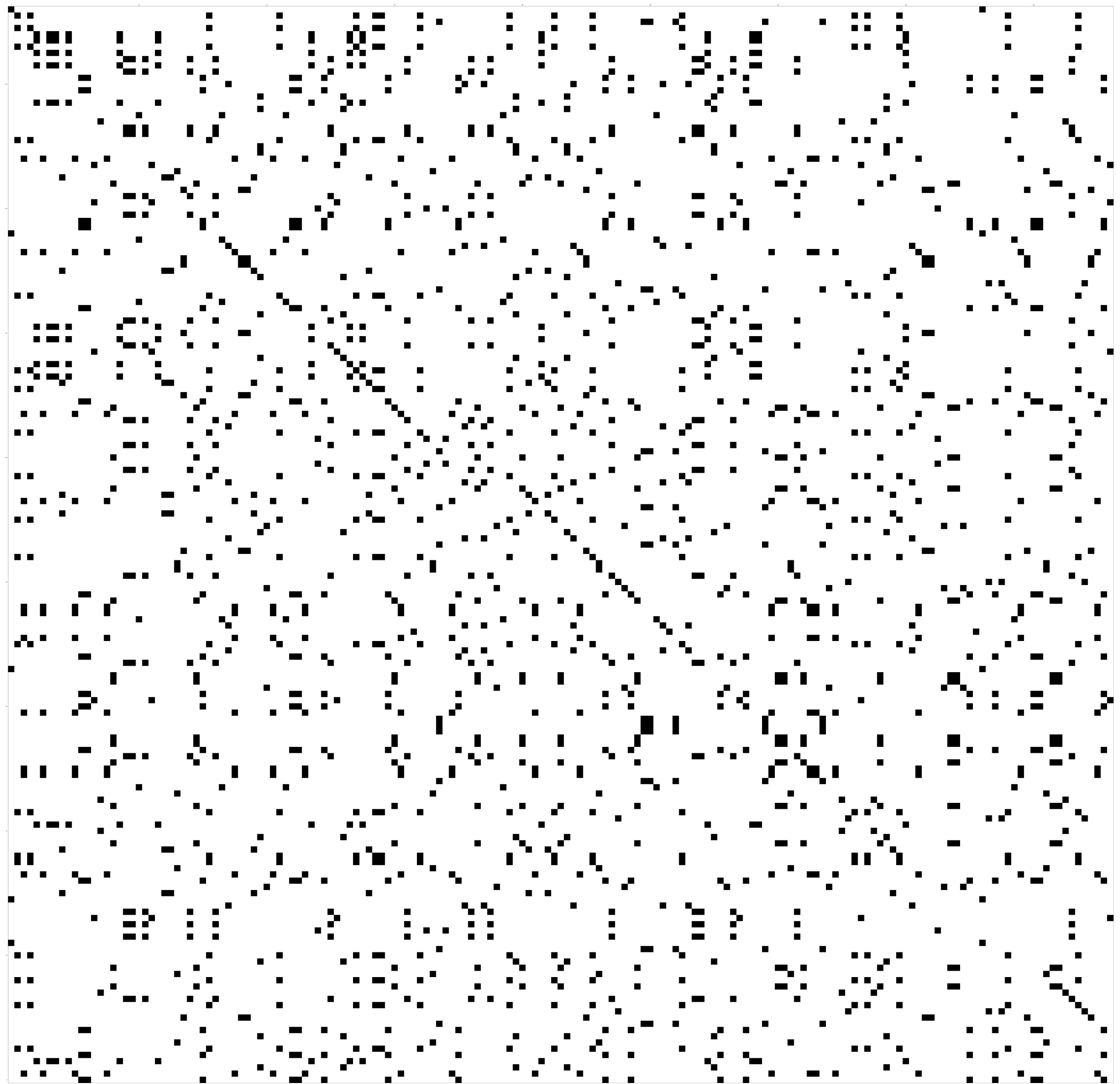
https://matplotlib.org/stable/tutorials/introductory/usage.html

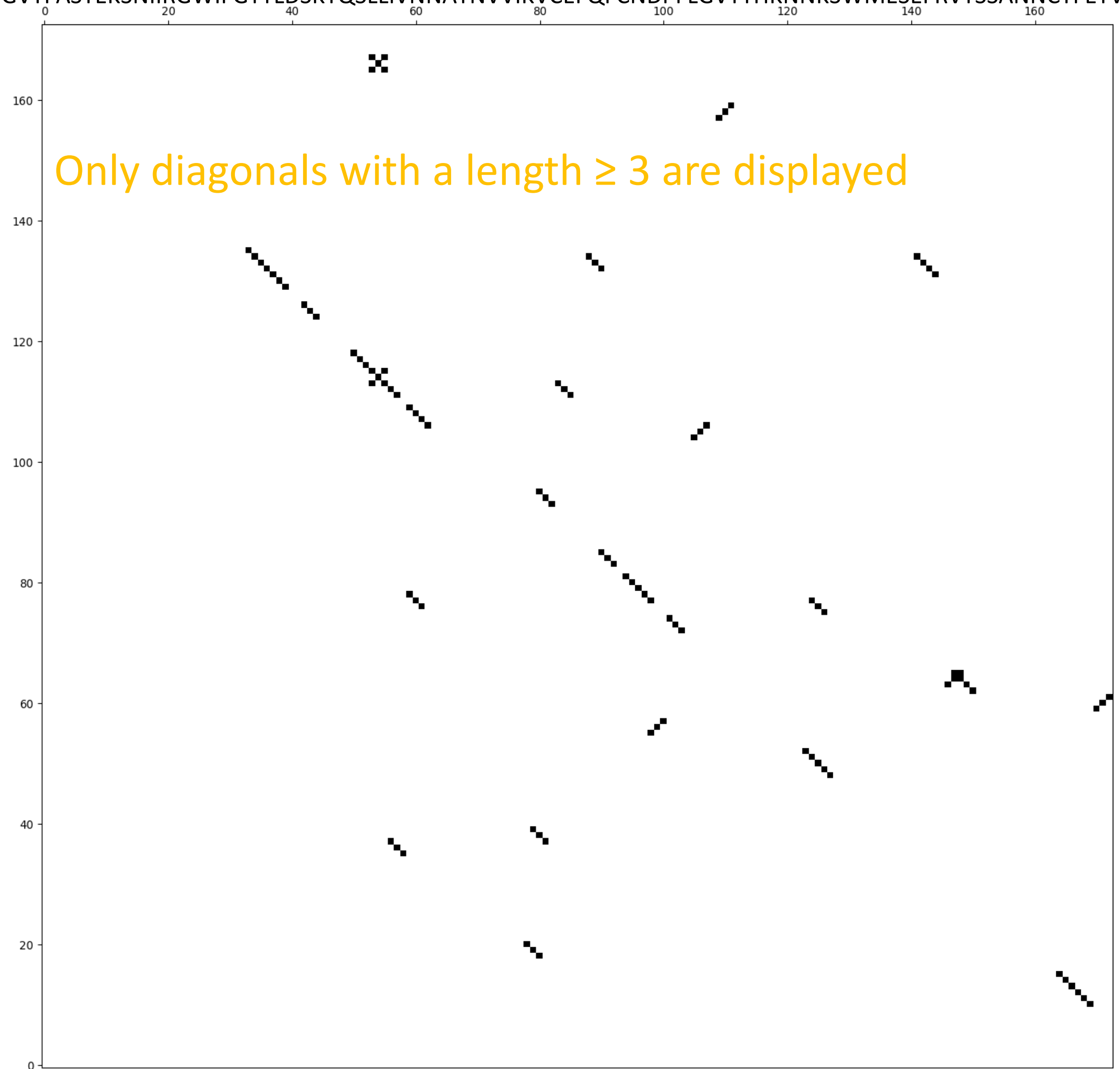

How to actually learn any new programming concept
Essential
Changing Stuff and Seeing What Happens
O RLY?
@ThePracticalDev

https://twitter.com/ThePracticalDev/

Dot Plots Without Length Threshold Are Overcrowded

s1 = 'MFIFLLFLTLTSGSDLDRCTTFDDVQAPNYTQHTSSMRGVYYPDEIFRSDTLYLTQDLFLPFYSNVTGFHTINHTFGNPVIPFKDGIYFAATEKSNVVRGWVFGSTMNNKSQSVIIINNSTNVVIRACNFELCDNPFFAVSKPMGTQTHTMIFDNAFNCTFEYISDAFSLDVS'
s2 = 'MFVFLVLLPLVSSQCVNLTTRTQLPPAYTNSFTRGVYYPDKVFRSSVLHSTQDLFLPFFSNVTWFHAIHVSGTNGTKRFDNPVLPFNDGVYFASTEKSNIIRGWIFGTTLDSKTQSLLIVNNATNVVIKVCEFQFCNDPFLGVYYHKNNKSWMESEFRVYSSANNCTFEYVSQ'





Reading Exercise

1. Describe the impact of graphical elements to the ability to assess the relative magnitude between elements.

2. Why do we need to know about the impact of graphical elements?

Notes

Notes