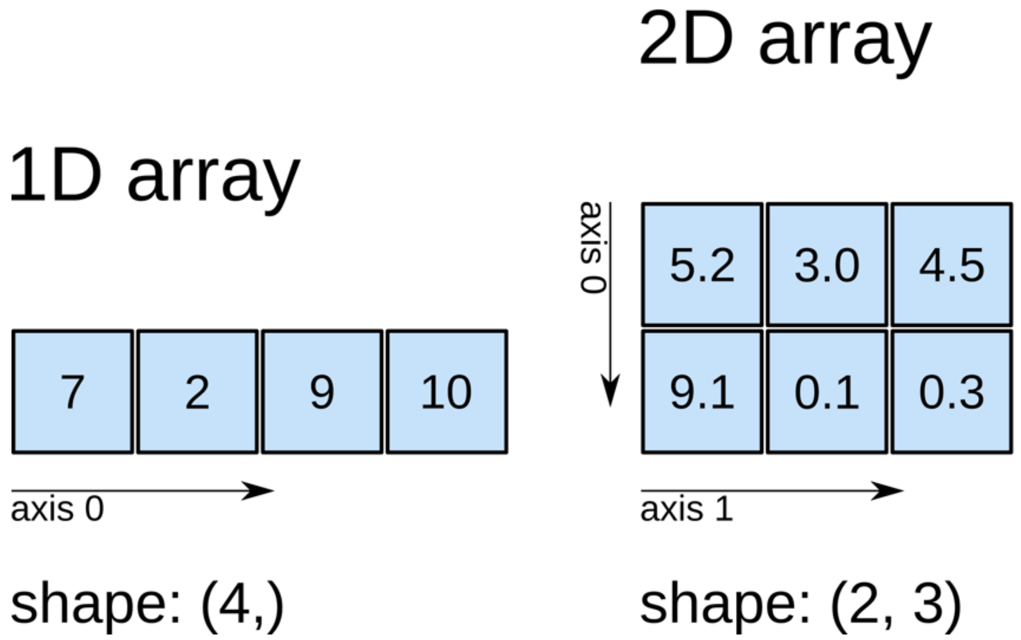
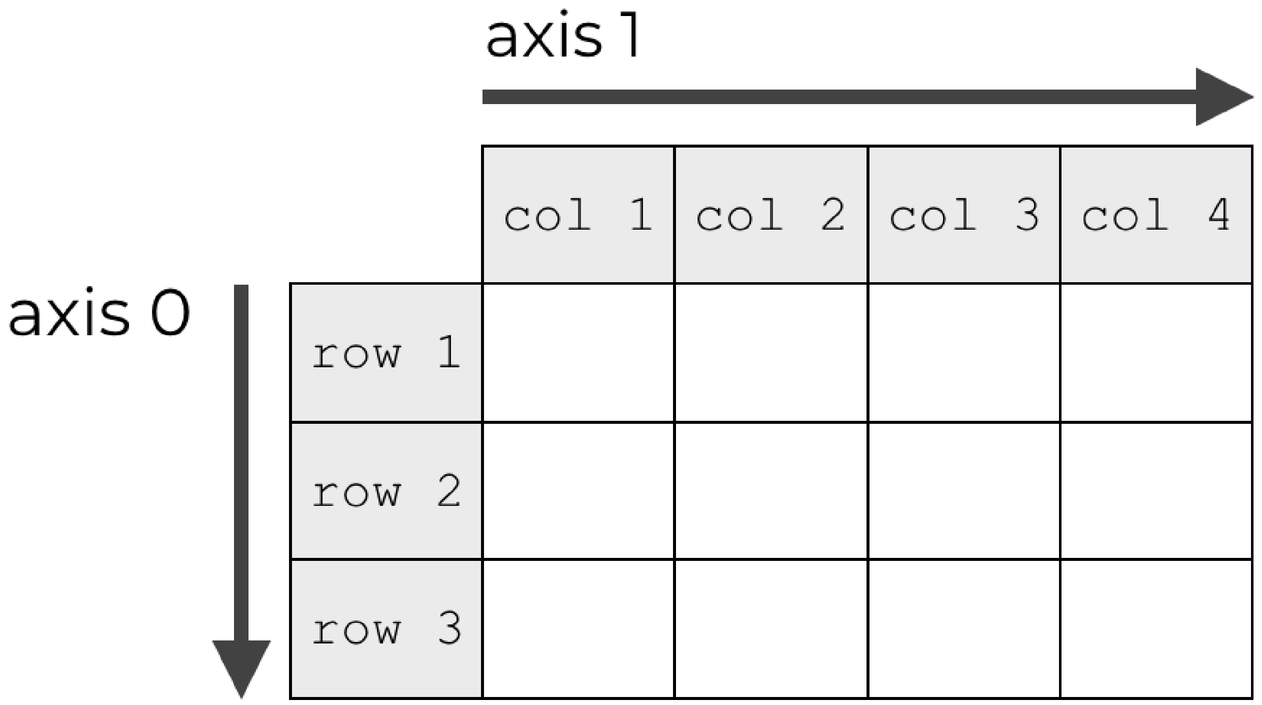
Numpy

Dimensions

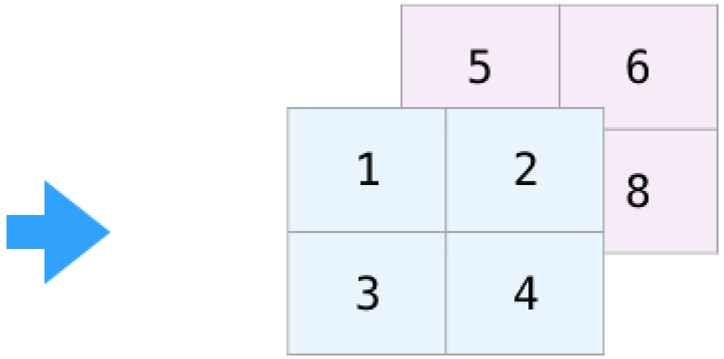

2D array
1D array
7
2
9
10
axis 0
shape: (4,)
axis 0
5.2
3.0
4.5
9.1
0.1
0.3
axis 1
shape: (2, 3)

axis



3-Dimensions

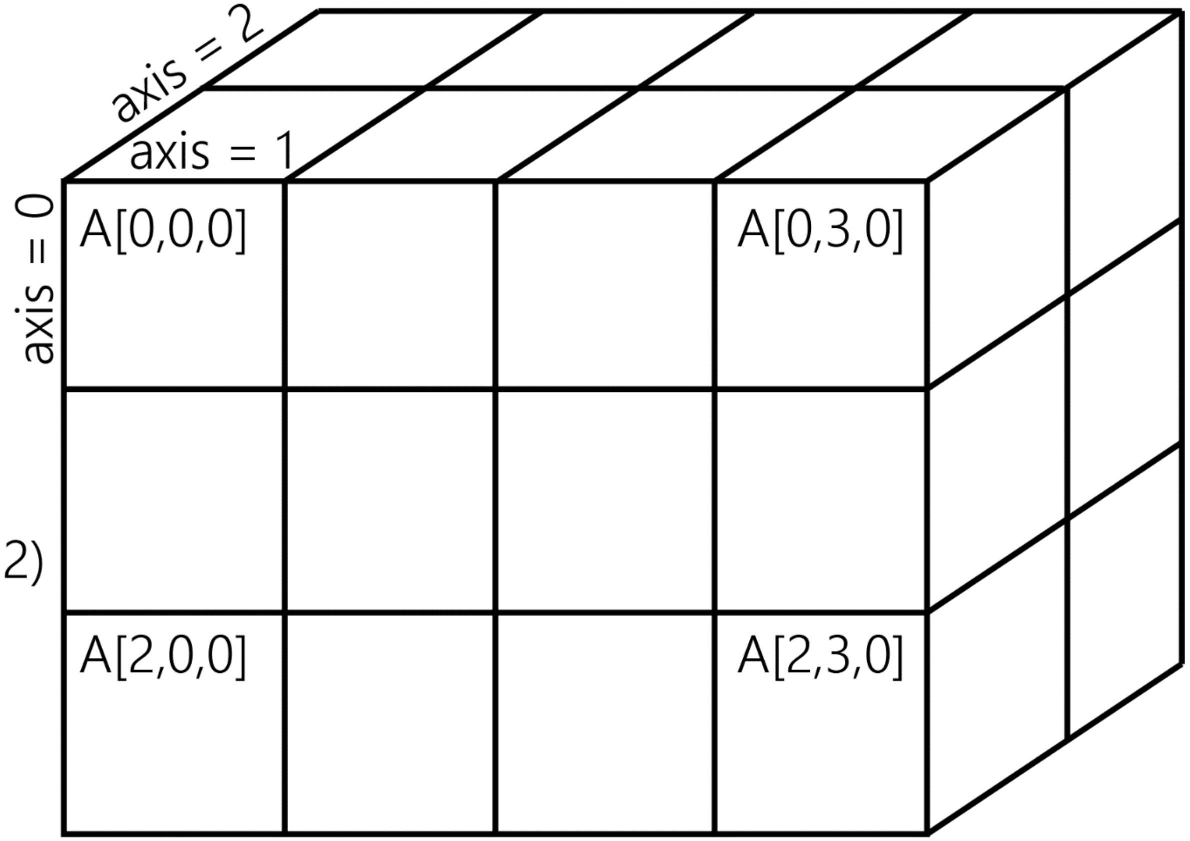
```
np.array([ [[1,2],[3,4]],
           [[5,6],[7,8]] ])
```



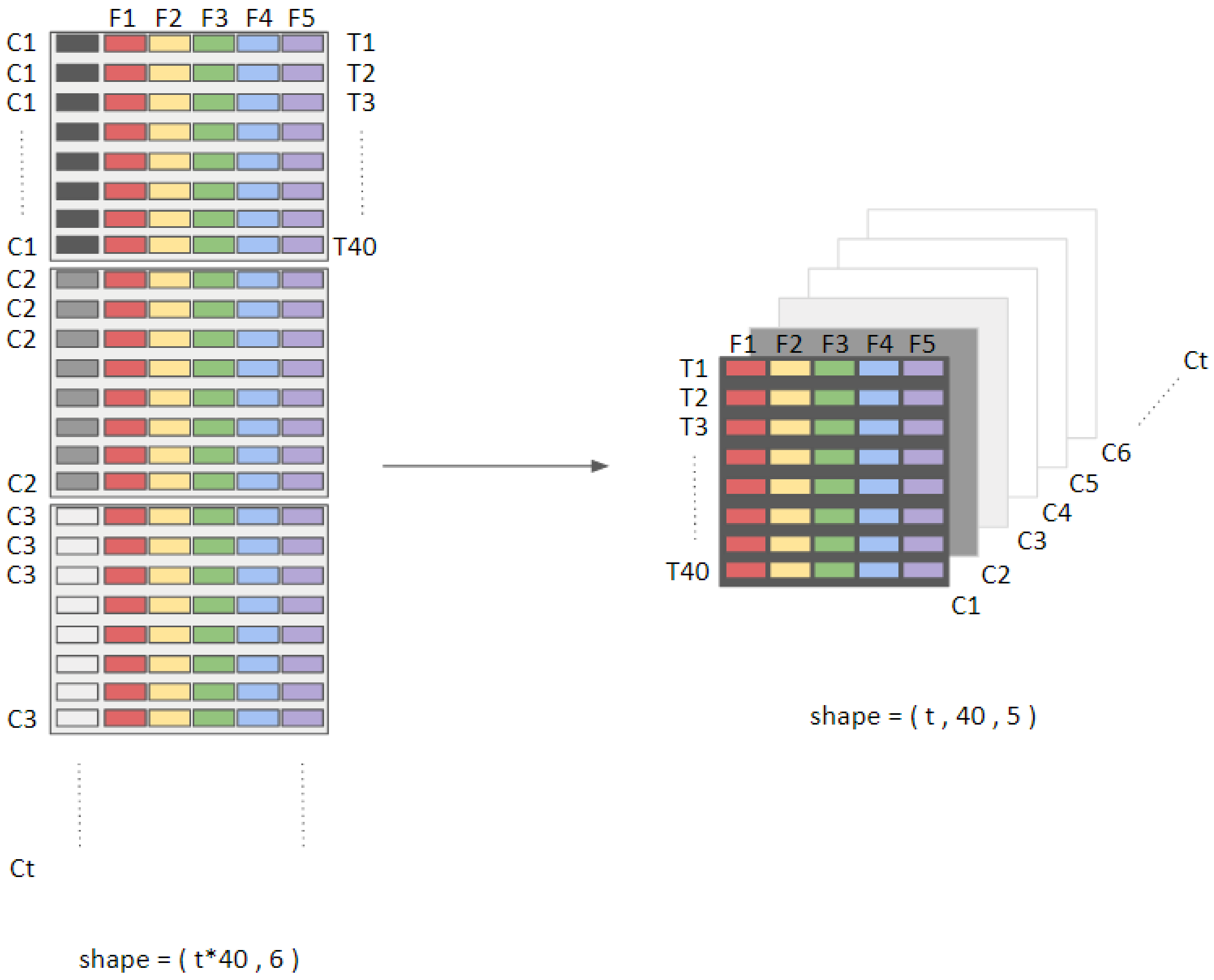
```
[[[1 2]
  [3 4]]

 [[5 6]
  [7 8]]]
```

3
(2, 2, 2)
(Row,Col,Depth/layers)

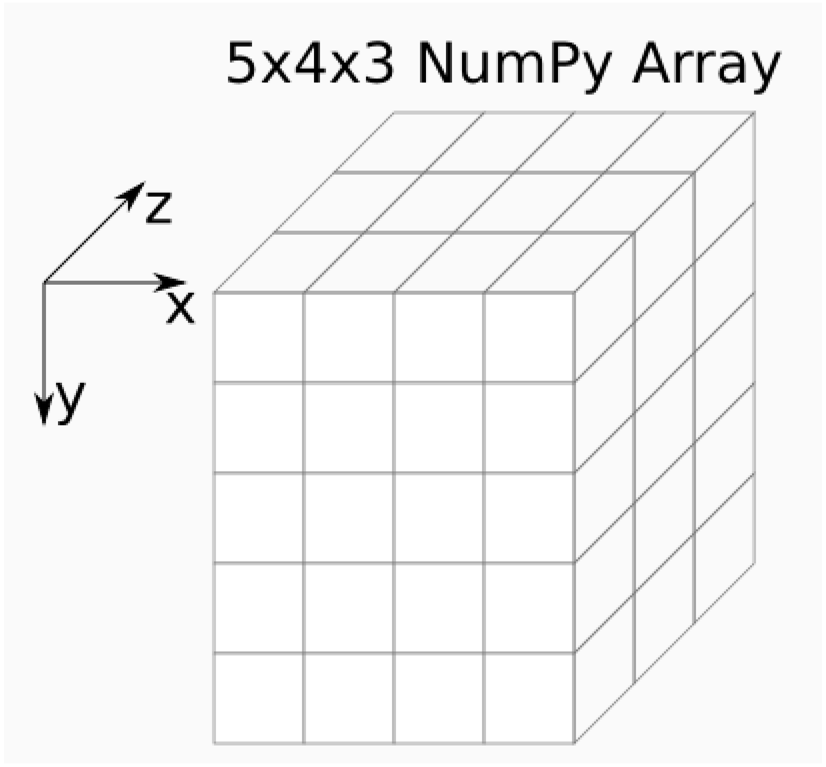
Numpy -3d

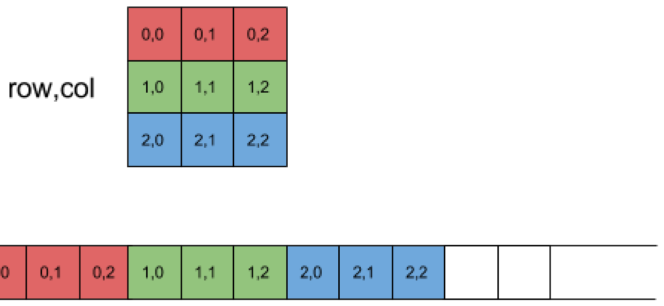


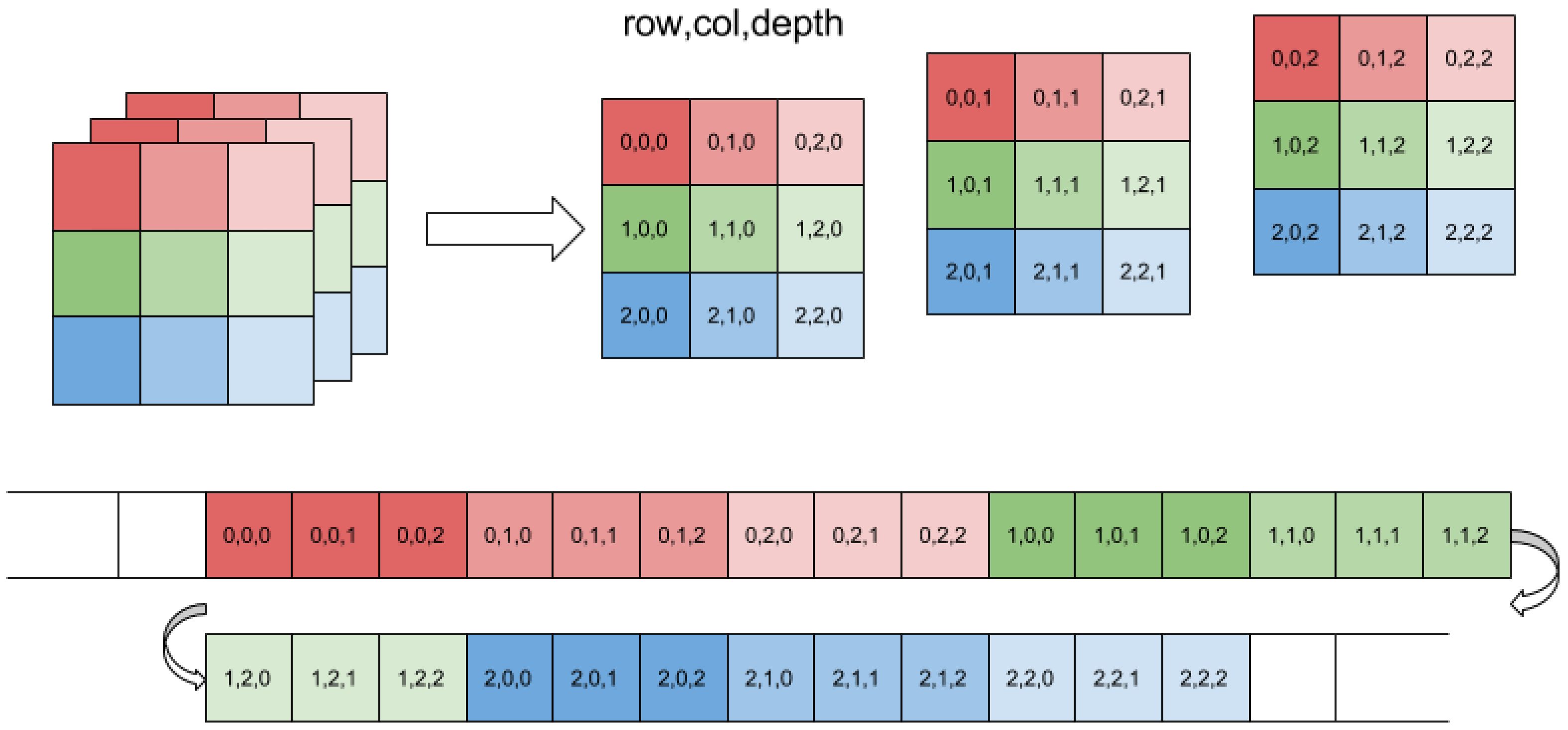
ndarray
ndim = 3
shape = (3, 4, 2)


F1 F2 F3 F4 F5
C1 T1
C1 T2
C1 T3
C1 T40
C2
C2
C2
C2
C3
C3
C3
C3
Ct
shape = (t*40 , 6)
F1 F2 F3 F4 F5
T1
T2
T3
T40
C1
C2
C3
C4
C5
C6
Ct
shape = (t , 40 , 5)

Numpy 3d




row,col
0,0 0,1 0,2
1,0 1,1 1,2
2,0 2,1 2,2
0,0 0,1 0,2 1,0 1,1 1,2 2,0 2,1 2,2


row,col,depth
0,0,0 0,1,0 0,2,0
1,0,0 1,1,0 1,2,0
2,0,0 2,1,0 2,2,0
0,0,1 0,1,1 0,2,1
1,0,1 1,1,1 1,2,1
2,0,1 2,1,1 2,2,1
0,0,2 0,1,2 0,2,2
1,0,2 1,1,2 1,2,2
2,0,2 2,1,2 2,2,2
0,0,0 0,0,1 0,0,2 0,1,0 0,1,1 0,1,2 0,2,0 0,2,1 0,2,2 1,0,0 1,0,1 1,0,2 1,1,0 1,1,1 1,1,2
1,2,0 1,2,1 1,2,2 2,0,0 2,0,1 2,0,2 2,1,0 2,1,1 2,1,2 2,2,0 2,2,1 2,2,2

3-D Data

- Panel data can be represented in three dimensions. Data that tracks attributes of a cohort (group) of individuals over time could be structured as (**respondents, dates, attributes**). The 1979 National Longitudinal Survey of Youth follows **12,686 respondents over 27 years**.
- Assuming that you have **~500 directly** asked or derived data points per individual, per year, this data would have shape (12686, 27, 500) for a total of 177,604,000 data points.

4-D data

- Color-image data for multiple images is typically stored in four dimensions. Each image is a three-dimensional array of (height, width, channels), where the channels are usually red, green, and blue (RGB) values. A collection of images is then just (image_number, height, width, channels). One thousand 256x256 RGB images would have shape (1000, 256, 256, 3). (An extended representation is RGBA, where the A–alpha–denotes the level of opacity.)