Australian National University

COMP1730/COMP6730
Programming for Scientists

NumPy special

Lecture outline

* Recap of arrays
* Programming problems

NumPy Arrays

* (Assuming `import numpy as np`.)
* `np.ndarray` is sequence type, and can also represent *n*-dimensional arrays.
  - `len(A)` is the size of the first dimension.
  - Indexing an *n*-d array returns an $(n-1)$-d array.
  - `A.shape` is a sequence of the size in each dimension.
* All values in an array must be of the same type.
  - Typically numbers (integers, floating point or complex) or Booleans, but can be any type.

Generalised indexing

- If *L* is an array of `bool` of the same size as *A*, *A*[*L*] returns an array with the elemnts of *A* where *L* is `True` (does not preserve shape).
- If *I* is an array of integers, *A*[*I*] returns an array with the elemnts of *A* at indices *I* (does not preserve shape).
- If *A* is a 2-d array,
  - *A*`[i,j]` is element at `i`, `j` (like *A*`[i][j]`).
  - *A*`[i,:]` is row `i` (same as *A*`[i]`).
  - *A*`[:,j]` is column `j`.
  - `:` can be *start*`:`*end*.

Operations and functions

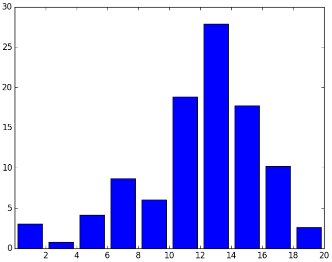
- Arithmetic (`+`,`-`,`*`,`/`,`**`,`//`,`%`), comparison (`==`,`!=`,`<`,`>`,`<=`,`>=`) and logical (`&`,`|`) operators work element-wise on arrays of same size, or array and value.
- Math functions provided by NumPy also work element-wise on arrays.
- `np.min`, `np.max`, `np.sum`, `np.mean`, `np.std`, `np.median` work on arrays.

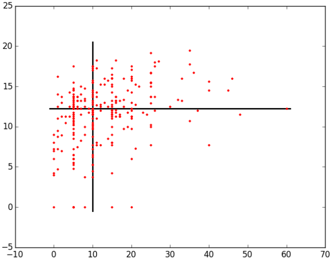
Copying and reshaping

* Most indexing/slicing operations on arrays do *not* copy, but return a “view” into the array.
* `np.copy(A)` copies array *A*.
* `np.reshape(A, shape)` returns a copy of the elements in *A* arranged into *shape* (size must match).
* `np.concatenate((A, B), axis = i)` returns a new array with *A* and *B* concatenated along dimension *i* (sizes must be equal in all other dimensions).

Data Analysis

- Plotting distributions and histograms.
- Mean and median.
- Scatterplots.
- Counting and statistical testing.





Systems of Linear Equations

$$\begin{array}{rcrcrcr} x_1 & + & x_2 & + & x_3 & = & 5 \\ -x_1 & + & 2x_2 & & & = & 9 \\ 4x_1 & + & x_2 & - & x_3 & = & -2 \end{array}$$

- In matrix form ($A \times x = b$):

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & 2 & 0 \\ 4 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 9 \\ -2 \end{bmatrix}$$

- Substitution (Gauss' elimination)
- Cramer's rule.
- `np.linalg.solve`.

Gauss' elimination

* Form [A b] and reduce to triangular:

$$\begin{bmatrix} 1 & 1 & 1 & 5 \\ -1 & 2 & 0 & 9 \\ 4 & 1 & -1 & -2 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 5 \\ 0 & 3 & 1 & 14 \\ 0 & 0 & -4 & -8 \end{bmatrix}$$

* Solve $x_n, x_{n-1}, \ldots, x_1$:

$$\begin{aligned} x_3 &= & -8/-4 &= 2 \\ x_2 &= & (14 - 1 \times x_3)/3 &= 4 \\ x_1 &= & 5 - [1\ 1] \times \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} &= -1 \end{aligned}$$

https://en.wikipedia.org/wiki/Gaussian_elimination

Cramer's rule

* $x_i = \frac{|A_i|}{|A|}$ where
  * $|\cdot|$ is the matrix determinant;
  * A_i is like A with ith column replaced by b.

$$\begin{vmatrix} 5 & 1 & 1 \\ 9 & 2 & 0 \\ -2 & 1 & -1 \end{vmatrix} = 12 \qquad \begin{vmatrix} 1 & 1 & 1 \\ -1 & 2 & 0 \\ 4 & 1 & -1 \end{vmatrix} = -12$$

$$x_1 = |A_1|/|A| = -1$$

https://en.wikipedia.org/wiki/Cramer's_rule