Numpy for the practical

Numpy is a library to process n-dimensional arrays : vectors, matrices, and arrays of more dimensions.

Read the official documentation

- quickstart
- guide for the absolute beginner

This is just a short list with the methods we will need for the practical.

- load it with

  `import numpy as np`
- make a 2d array from a list

  `np.array([1,2,3,4], dtype=np.int) # integers, 1 dimension, length 4`

  `X = np.array([[1,2,3,4],[5,6,7,8]]) # floats, 2 rows, 4 columns`
- array dimensions

  `len(X)` num elements in first dimension (rows)

  `X.shape` , `np.shape(X)` returns tuple (2,4)
- indexing and slicing, if `X` a matrix:

  `X[:,3]` 4th column

  `X[0]` first row

  `X[-1]` last row

  `X[1::2]` second, fourth... last even row

  `v[v>0]` vector of positive elements only

- randomly reorder the elements of a vector, rows of a matrix `X`

  `X = np.random.permutation(X)`, also

  `idx = np.random.permutation(len(X))` and then `X = X[idx]`
- 20 random values $3 \leq y \leq 44$ (uniform distribution)

  `Y = (3 + 44*np.random.rand(20)).astype(np.int)`
- draw samples with/without replacement, if replace True drawn samples can be repeated

  `np.random.choice(Y, 10, replace=True)`

- histogram : how many occurrences of each value in `Y`

  `counts, bins = np.histogram(Y, bins=range(3,44+2))`
- extrema

  `(np.max(Y), np.max(Y))` also `(Y.min(), Y.max())`
- index of maximum value

  `i = np.argmax(Y)` so that `Y[i] == Y.max()`
- vector with the different values (no repeated values), sorted

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

- point-wise operations

  `np.log(X)`, `np.log2(X)`, `X**2`, `np.sqrt(X)`, `np.abs(X)`
- arithmetic operations

  `Y = 3*X`, `Z = X + Y`, `X / Y`
- sum, mean, variance of elements

  ```
  z = np.sum(X)
  ```

  ```
  np.mean(X)
  ```

  ```
  np.var(X)
  ```
- sort elements, if a matrix along each row

  ```
  np.sort(X, axis=0)
  ```
- `np.Inf` the largest positive number