final review/overview

Ben Bolker

03 December 2019

## all topics

* simple variable types
* arithmetic and logical operators
* repr() (print representation)
* logical expressions
* indexing and slicing (strings, lists, arrays, data frames …)
* string methods (.lower(), .upper(), .replace(), .isalpha()
* lists, list operators (+=), list methods
* **mutability**
* **conditionals and flow control**
  + if, for, while (break)
  + nested loops
* functions
* modules
* tuples, tuple methods
* files
  + opening and closing, .closed
  + .read(), .readlines(), next, StopIteration
  + .strip(), .split(), type conversion
* sets (**non-ordered, unique**): .add, .remove, …
* dictionaries
  + indexing (not by number unless keys are numeric)
  + .keys(), .values(), .items(), for
  + inversion
* random numbers (random or numpy.random)
  + random.seed()
  + .choice, .uniform, .randrange
  + Monte Carlo methods/simulations
  + use np.mean or np.sum on a bool array to count fraction or total
* numpy
  + arrays
    - defining with dtype
    - .shape
    - zeros(), ones(), eye(), identity, reshape(), flatten(), arange(), linspace(), copy(), fill()
    - operators, indexing, slicing, selections by logical
    - vectorized and non-vectorized operators (np.sin vs math.sin)
    - operations over axes: sum, mean, min, max, newaxis
    - np.logical.[and,not,or]
* numerics
  + underflow (too close to zero)
  + overflow (integer and float)
  + loss of precision (small number + large number)
  + nan
* matplotlib
  + .plot (uses index as x-variable if no x provided: draws lines by default)
  + fig, ax = plt.subplots()
  + .scatter (draws points by default)
  + .hist (histogram)
  + .bar (barplot)
  + set\_xlabel, set\_xticklabels, suptitle (recognize)
  + label, legend
  + imshow (image)
* error handling
  + raise
  + try/except (pass)
  + ValueError (inappropriate value), NameError (undefined symbol), IndexError (incorrect indexing), TypeError (inappropriate type)
* pandas
  + DataFrame and Series
  + indexing: .loc and .iloc; indexing columns d[["key1","key2"]]; extracting columns as d.key1
  + read\_csv(), .to\_csv()
  + operations across rows/columns
  + .groupby, .aggregate (collapse by group: MC only)