Lecture #3 | python review: variables, functions, numpy

SE377 Introduction to Big Data Analysis and Visualization (2017)

Prof. Min-gyu Cho

Today's Topic

- Variable
- List: list comprehension, `enumerate()`, `zip()`
- Functions
  - Review
  - Lambda functions
  - High order functions: `map()`, `filter()`, `reduce()`
- Introduction of python packages for data analysis

Variables

- Definition
  - A name that refers to an object (or a value)
  - A named storage in the memory for any type of object (or value)
- We can assign a value to a variable with '='

```
greeting = 'Hello, world!'
n = 42
pi = 3.14159265
t = [42, 1024, 23]
```

Another explanation of list comprehensions

- Example of set operations in mathematics

$$V = \{1, 2, 3, 4, 5\}$$
$$S = \{x^2 \mid x \in V\}$$
$$T = \{2x \mid x \in V, x \; mod \; 2 = 0\}$$

- List comprehension can be used to simply and conveniently describe the above concepts

```
v = [1, 2, 3, 4, 5]
s = [x**2 for x in v]
v = [2**i for i in v if x % 2 = 0]
```

Lambda functions (or lambda expressions)

- A simple function that returns the result of one expression can be written as a lambda function (or a lambda expression), i.e., an anonymous function
- Example

```
def add(x, y):
    return x + y

add_l = lambda x, y: x + y

print(add(42, 23), add_l(42, 23))
```

Higher order functions

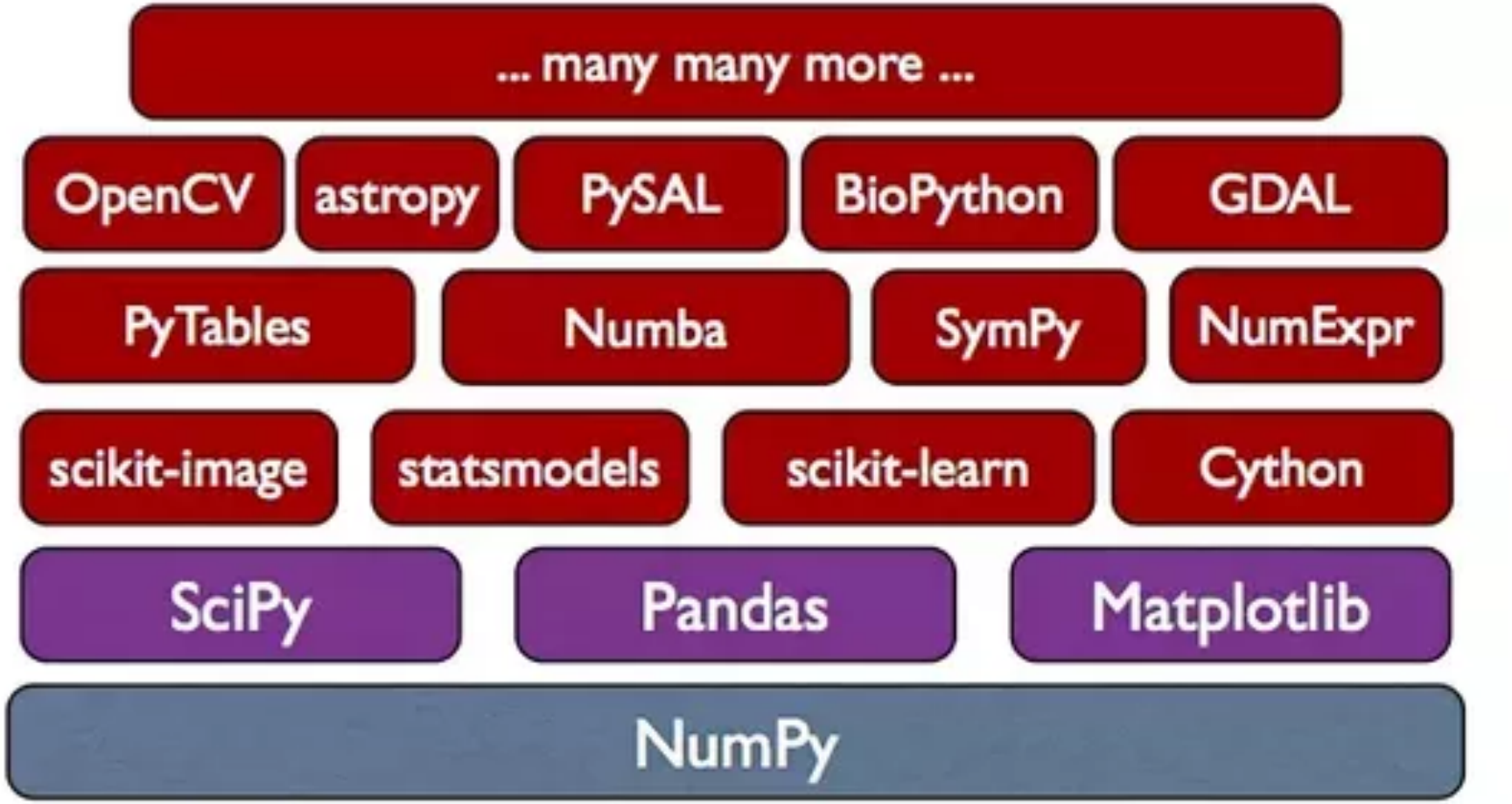
- A higher order functions does one of the following
  - Take one or more functions as arguments
  - Return a function as its list
  - c.f., all other functions are first order functions
- Why higher order functions?
  - In programming (esp. for data analytics), it is quite often to use the same structure of control flows with minor differences in operations
  - We separate functions for control flows and operations for code reusability, productivity and readability
- In functional programming paradigm, use of pure functions* and high order functions are strongly encouraged

* See https://en.wikipedia.org/wiki/Pure_function

Features of frequently used python packages

- **numpy**: large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays
- **scipy**: mathematical/statistical algorithms and convenience functions built on numpy to manipulating and visualizing data
- **pandas**: data structures (data frames) and operations for manipulating numerical tables and time series
- **scikit-learn**: machine learning libraries built on top of scipy
- **matplotlib**: 2D plotting library to produce a various forms of publication quality figures
- Usages of the above packages will be covered during the class; but only frequently used features will be covered. Students are expected & required to learn how to use those libraries if necessary.

Dependencies of python packages for data analysis



Source: https://www.quora.com/What-is-the-relationship-among-NumPy-SciPy-Pandas-and-Scikit-learn-and-when-should-I-use-each-one-of-them

numpy

- NumPy is the fundamental package for scientific computing with Python. It contains among other things:
  - a powerful N-dimensional array object
  - sophisticated (broadcasting) functions
  - tools for integrating C/C++ and Fortran code
  - useful linear algebra, Fourier transform, and random number capabilities
- Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Source: http://www.numpy.org

Reading list

- python 내장 함수: https://wikidocs.net/32
  - See `enumerate()` and `zip()`
- `map(), filter(), reduce()`
  - http://book.pythontips.com/en/latest/map_filter.html
- numpy
  - Tutorials
    - Scipy Lecture Notes: http://www.scipy-lectures.org
    - Numpy Tutorial: http://www.python-course.eu/numpy.php
  - Reference: https://docs.scipy.org/doc/numpy/reference/

ANY QUESTIONS?