Algorithms for Data Analysis

Julien Tissier

November 17, 2017

Overview

1. Introduction to Data Analysis
    - Definition
    - Tools and libraries
    - Numpy
    - Pandas
    - Matplotlib

What is data analysis?

Data

Pieces of information (measurement, values, facts...) that can be :

- structured (matrices, tabular data, RDBMS, time series...)
- unstructured (news articles, webpages, images/video...)

Data Analysis

Process of preparing, transforming and using models to find more information from data, as well as visualizing results.

How to analyze data?

There are usually two steps in data analysis. The first one is to find and **develop models** that can extract useful information from data (with languages like R or MATLAB). The second one is to **develop programs** that can be used in production systems (with languages like Java or C++).

With a growing popularity among scientists as well as the development of efficient libraries (numpy, pandas), **Python** became a great tool for data analysis. Python has many advantages :

- great for string/data processing
- can be used for both prototyping/production
- has a lot of existing libraries
- can easily integrate C/C++/FORTRAN legacy code
- easy to read/develop

Libraries

This course will be based on Python 3.5 (or above) and the following libraries :

- IPython (6.2+) : enhanced Python shell
- numpy (1.13+) : fast/efficient arrays and operations
- pandas (0.20+) : data structures (Series/DataFrame)
- matplotlib (2.1.0+) : plots and 2D visualization
- scipy (0.19+) : scientific algorithms
- scikit-learn (0.19+) : machine learning algorithms

Jupyter Notebook will also be used to give you samples of code, as they provide a more interactive way to learn and discover how these libraries work.

Numpy

Numpy (**Num**erical **Py**thon) is a high performance scientific computing library that can be used for matrices computations, Fourier transforms, linear algebra, statistical computations...

The main type of data in Numpy is the **ndarray** :

- n-dimensional array
- fixed size
- homogeneous datatypes
- similar to C arrays (continuous block of memory)

Why is Numpy efficient?

As a high level language, Python is slow to do any heavy computations, especially if very large arrays are involved. Numpy solves this problem thanks to the ndarray datatypes :

- efficient memory management (continuous block)
- use C loops instead of Python loops for computations on array
- vectorized operations (computations are done block by block, not element by element)
- rely on low-level routines for some operations (BLAS/LAPACK)

Numpy

Example

```
import numpy as np

a = np.array([1, 2, 3, 4])
b = np.array([6, 7, 8, 9])

c = a * b

d = np.array([[1, 2], [3, 4]])
```

Pandas

Pandas is a high-performance Python library used to work with data and analyze them. It contains many pre-implemented methods to read and parse data, as well as common statistical computations (mean, variance, correlation...).

Pandas has two main datatypes:

- Series : one-dimensional container. Indexes can be integers (like an array) or other objects (string, date...)
- DataFrame : tabular data, like a spreadsheet. It contains multiple rows and multiple columns. It can be seen as a collection of Series.

Pandas

Example (Series)

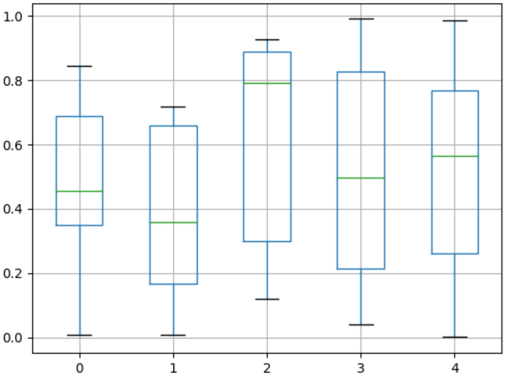
```
from pandas import Series

s1 = Series([4, 7, 9, -1])
s2 = Series([12, 3.2, "John"],
            index=["mark", "gpa", "name"])
```

Example (DataFrame)

```
from pandas import DataFrame
df = DataFrame({"key1": [1, 2, 3],
                "key2": [4, 5, 6]})
```

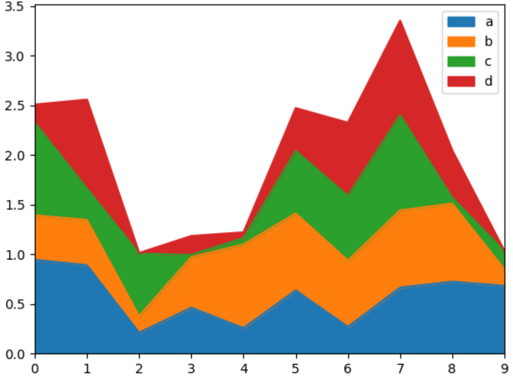
Matplotlib

Matplotlib is a plotting library used to visualize data and create graphics. Pandas directly uses matplotlib for representation.



Matplotlib

Matplotlib is a plotting library used to visualize data and create graphics. Pandas directly uses matplotlib for representation.


3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5
6
7
8
9
a
b
c
d