Data Analysis in Python

Course for SEA/UAB 2017-2018

José María Lago Alonso

https://github.com/jmlago/DA_python

Structure of the Course

1. **Aerial perspective**
   Basic concepts of Data Analysis and Python. A global view of the environment...

2 (4.5) **Data Gathering**
   How to extract real and quality data from Internet...

3. **Exploratory Analysis and Cleaning**
   Now with real data, how to prepare this data, understand the data, and do visualizations...

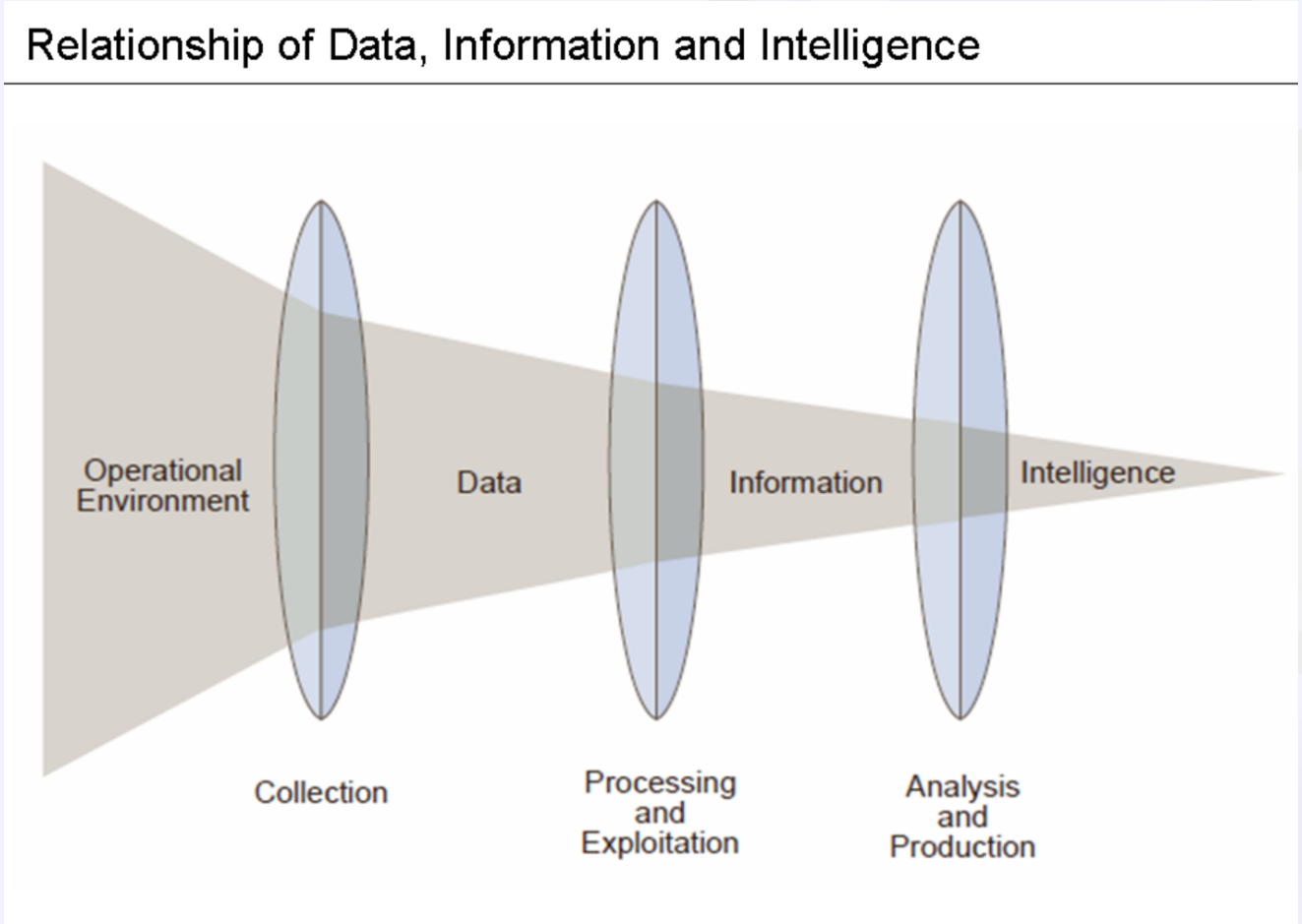
4. **Algorithm Selection**
   A tiny view of algorithms world and how to use them...

5. **Performance Engineering**
   Usually, when we work with big amounts of data, we need to speed up our algorithms ...

6. **Deep Learning**
   A tiny view of the deep learning world...

What is Data Analysis?

Analysis of data is a process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, suggesting conclusions, and supporting decision-making.



What is Python?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.

- Interpreted:
  Compile and execute by any size blocks in any time.
- Object Oriented:
  Uses classes objects and attributes.
- High-Level:
  Can not allocate memory manually.
- Dynamic semantics:
  Python doesn't have static types.

Why use Python?

Strengths

- Glue language
- Simple and easy to learn
- Program modularity and code reuse
- Edit-test-debug really fast
- General purpose language
- Cross platform

Weaknesses

- Slower than compiled languages
- Python2 not compatible with Python3
- Lack of static types
- Can't free memory in usual way

Python vs Others

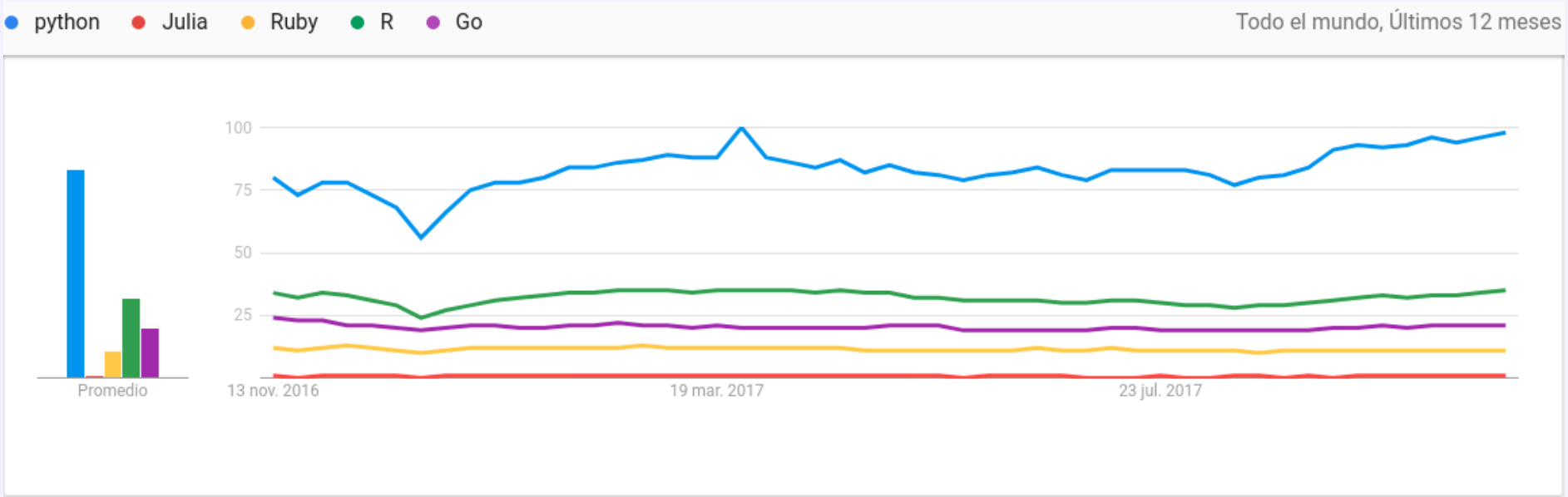


Figure: Python vs Julia vs R vs Go (Google Trends)

- **Community**
- Multi-purpose
- Easy to learn

- Juila is faster? And Go? (Numpy/Numba/Torch/CuPy...)
- Frameworks and wrappers

Conda

Anaconda

We are going to use Python3 in all this course !!!

Definition

Anaconda is an easy-to-install free package manager, environment manager, Python distribution, and collection of over 720 open source packages offering free community support.

Why?

Because Anaconda's packages are for data analytics, data science, and scientific computing.
Also Conda makes sure that all packages and environments works fine together.

Installation

- Go to: https://www.continuum.io/downloads
- Choose your OS
- Download the installer
- Follow the steps on the webpage

Warning

For Linux users, you may use:

```
bash Anaconda3-5.0.1-Linux-x86_64.sh
```

To install Anaconda.

Anaconda tips

For Windows users, you should open `AnacondaPrompt` terminal. For Mac and Linux users, you can open the regular terminal.

- List available pythons:

```
conda search "^python$"
source ~/.bashrc #(Linux users)
```

- Create and activate the environment:

```
conda create --name my_env python=3
source activate my_env
```

- Install packages:

```
conda install some_package
pip install some_package
```

- Remove environment:

```
conda remove --name my_env --all
```

Alternative Installations

Linux

Linux

```
sudo apt-get install python3-pip python3-dev
 ↪  python-virtualenv
pip3 install --upgrade pip3
cd /usr/local/share
sudo mkdir virtualenvs
sudo chown -R root:sudo virtualenvs/
sudo chmod -R g+w virtualenvs/
virtualenv --system-site-packages -p python3
 ↪  /usr/local/share/virtualenvs/v1
source /usr/local/share/virtualenvs/v1/bin/activate
pip3 install -r requirements.txt
```

Alternative Installations

Mac and Windows

Mac

Same script that in Linux, but changing the `sudo apt-get install` for `brew install`

Windows

- Download Python3 from `https://www.python.org/downloads/windows/`
- open the cmd and type `python get-pip.py`
- After that, execute:
  - `pip install virtualenv`
  - `pip install virtualenvwrapper-powershell`
  - `mkdir '~\.virtualenvs'`
  - `Import-Module virtualenvwrapper`
- To see the lists of commands that we can use, just type: `Get-Command *virtualenv*`

Spyder

Integrated Development Enviroment

Definition

Scientific PYthon Development EnviRonment

- Similar to RStudio and MATLAB IDE's
- IDE for Science
- Exploratory
- Easy debugging

Other IDEs

Figure: Atom + Hydrogen

Figure: Jupyter

Figure: Eclipse + PyDev

All IDEs have advantages and disadvantages, you will need to choose what suits you most. In this course we will use SPYDER for simplicity, but each IDE has different purposes.

Verify that all is working OK

- Open SPYDER
- Try to execute the next code:

```
import pandas as pd
import os
import torch
import scipy as sp

print("Everything is working OK!!!")
```

Type the code, select all the code and press CTRL+INTRO

Python Basics 1

Libraries

In Python we need to set the modules that we are going to use at the beginning of the script. We do in the following way:

```
import somelibrary as somename
somename.somefunctioninthelib()
```

Generic Data Types

Python really has dynamic semantics so a variable is somehow dynamic type, for instance:

```
a = [1,2,3] #--> list of numbers
a = a[1] #--> position in a list
a = "abcd" #--> string
...
```

Python Basics 2

Functions

We can use functions very easy because of the dynamic semantics. Also in Python the most important thing is **INDENTATION** this is how we determine the loops and the range of the functions.
Beautiful and readable code.

```
def somefunction(param1,param2):
    a = param1*param2
    return a
```

Python Basics 3

Classes

Here is the OO part.

Easy example:

```
class Complex:
    def __init__(self, realpart, imagpart):
        self.r = realpart
        self.i = imagpart

x = Complex(3.0, -4.5)

x.r, x.i
(3.0, -4.5)
```

Let's practice a little bit !!!

Open the `practice0.py`

After this open `oo.py`

Python Basics 4

Reserved words

```
if __name__ == "__main__":
        print("Hello main")
########
class C2(C1):
        def __len__(self,...):
```

Relative paths and modules

What is a python module? What is `__init__.py`?

```
from ..outer.inner import foo
```

```
hello/
    __init__.py
    params.py
    bye/
        __init__.py
        params2.py
```

Exercise: Try to import in `params2.py` some variable defined in `params.py` without executing the main script.

Help!

- `StackOverFlow` and `How To "do something" in Python`
- Python documentation
- External libraries documentation

Warning

Be careful of `StackOverFlow` because it's easy to copy and paste the code but if you don't understand what are you doing you'll have several problems.

Libraries that we need

- Pandas
- BeautifulSoup
- Matplotlib
- NumPy
- SciPy
- Scikit-learn
- Plotly
- Pytorch
- Numba

And many others...