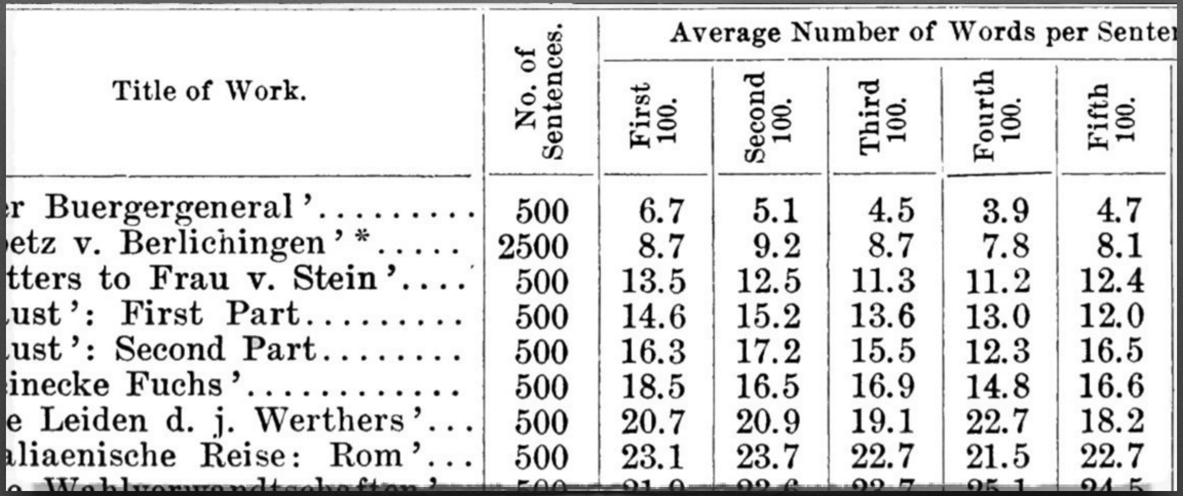


Title of Work.	No. of Sentences.	Average Number of Words per Sente				
		First 100.	Second 100.	Third 100.	Fourth 100.	Fifth 100.
r Buergergeneral'.........	500	6.7	5.1	4.5	3.9	4.7
etz v. Berlichingen'*.....	2500	8.7	9.2	8.7	7.8	8.1
tters to Frau v. Stein'....	500	13.5	12.5	11.3	11.2	12.4
ust': First Part.........	500	14.6	15.2	13.6	13.0	12.0
ust': Second Part........	500	16.3	17.2	15.5	12.3	16.5
inecke Fuchs'............	500	18.5	16.5	16.9	14.8	16.6
e Leiden d. j. Werthers'...	500	20.7	20.9	19.1	22.7	18.2
aliaenische Reise: Rom'...	500	23.1	23.7	22.7	21.5	22.7
e Wahlverwandtschaften'	500	21.9	23.6	23.7	25.1	24.5

Introduction to Data Analysis with Python

Prof. David Losada, Universidade de Santiago de Compostela

Email: david.losada@usc.es Universitat de Barcelona

Marcos Fernández, Investigador predotural, University of Santiago de Compostela

Email: marcosfernandez.pichel@usc.es

Why Python?

- Python is a modern, open source, **object-oriented programming** language, created by a Dutch programmer, Guido van Rossum, in 1991.
- Officially, it is an **interpreted scripting language**. There are several implementations, being the most used the one implemented in C.
- It offers the power and flexibility of lower level (i.e. compiled) languages, without the steep learning curve, and without most of the associated debugging pitfalls.
- The language is very **clean and readable**, and it is available for almost every modern computing platform.
- It is widely used: Google, DropBox, Deutsche Borse, NASA, etc.

Why Python?

In Data Science, **programming is mostly about discovering solutions by writing code and using a language to express them**.

Python offers a number of advantages to data scientists:

- Powerful and easy to use.
- Anything that can be coded in C, FORTRAN, or Java can be done in Python, almost always in fewer lines of code, and with fewer debugging headaches.
- Its standard library is extremely rich, including modules for string manipulation, regular expressions, file compression, mathematics, scientific programming, profiling and debugging.
- Unnecessary language constructs, such as END statements and brackets are absent, making the code efficient and easy to read.
- Python is object-oriented, which allows data structures to be abstracted in a natural way.

Why Python?

- It is interactive
  - Python may be run interactively on the command line. Commands may entered serially followed by the Return key.
  - This is useful for exploratory programming.
- It is extensible
  - Python is often referred to as a “glue” language, meaning that it is a useful in a mixed-language environment. C or FORTRAN code can be compiled directly into Python programs.
  - Sometimes, it can be slow relative to compiled languages. In many cases this performance deficit is due to a short loop of code that runs thousands or millions of times. Such bottlenecks may be easily removed by coding a function in C or Cython that can be compiled into a Python module.

Why Python?

- There is a vast body of Python third party modules:
  - **NumPy**: Numerical Python (NumPy) is a set of extensions that provides the ability to specify and manipulate array data structures.
  - **SciPy**: An open source library of scientific tools for Python, SciPy supplements the NumPy module. SciPy includes modules for graphics and plotting, optimization, integration, special functions, signal and image processing, genetic algorithms, ODE solvers, and others.
  - **Matplotlib**: Matplotlib is a python 2D plotting library which produces publication-quality figures.
  - **pandas**: A module that provides high-performance, easy-to-use data structures and data analysis tools. In particular, the DataFrame class is useful for spreadsheet-like representation and mannipulation of data.
  - **IPython**: An enhanced Python shell, designed to increase the efficiency and usability of coding, testing and debugging Python.

Why Python?

- Strong points:
  - Viable alternative to SAS/Matlab/R/....
  - Uptake in the financial sector.
  - Upcoming generation of programmers with Python as a first language.
  - Python communities in HPC/Scientific Computing.

- Weak points
  - No (public) great success story.
  - Weak support.

What is IPython?

I is for interactive

I is for interactive

In this class, we will use **IPython notebooks** for teaching and programming assignments.

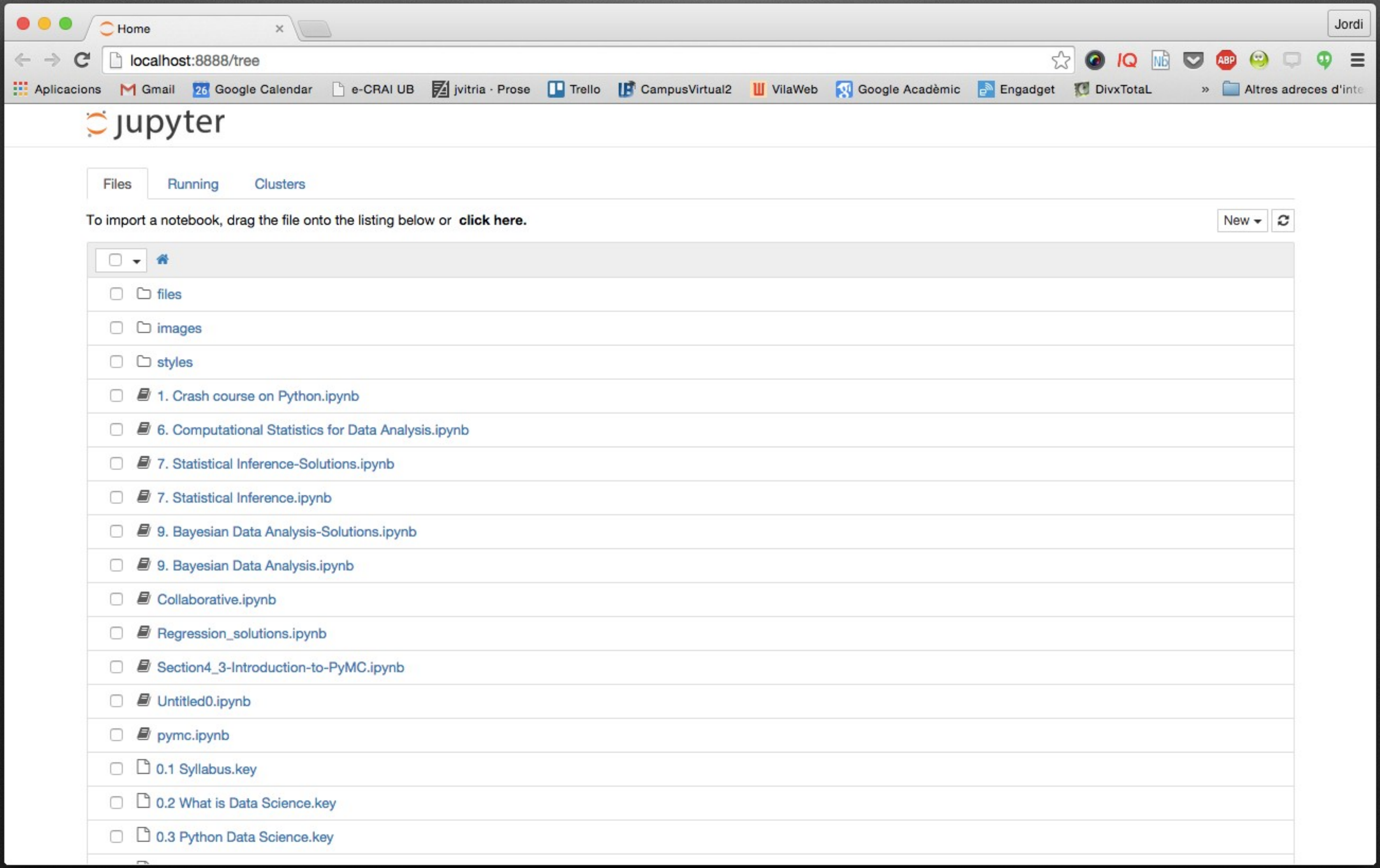
An IPython notebook lets you write and execute Python code in your web browser.

IPython notebooks make it very easy to tinker with code and execute it in bits and pieces; for this reason IPython notebooks are widely used in scientific computing.

What is IPython?

I is for interactive

I is for interactive



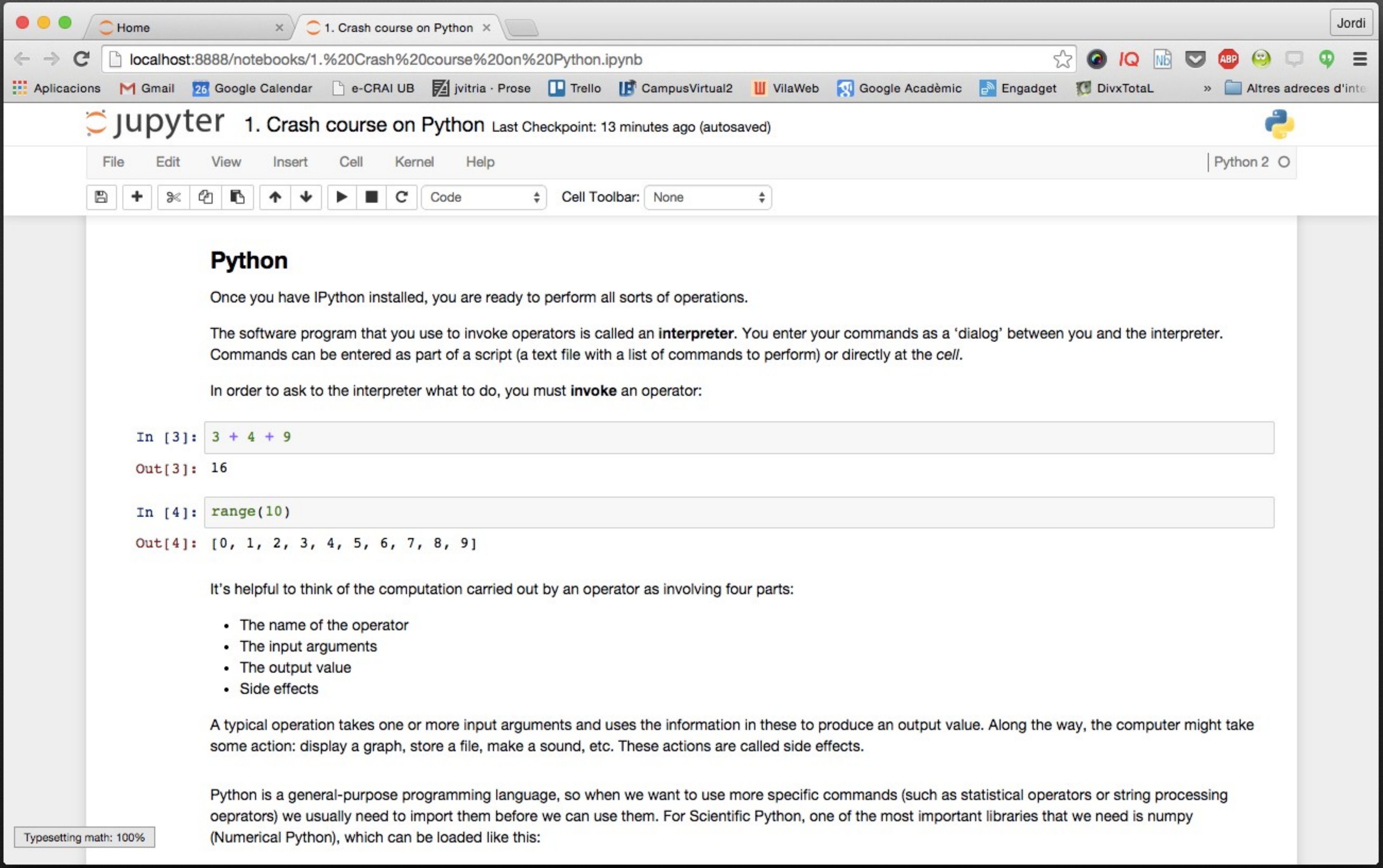
To import a notebook, drag the file onto the listing below or **click here.**

- files
- images
- styles
- 1. Crash course on Python.ipynb
- 6. Computational Statistics for Data Analysis.ipynb
- 7. Statistical Inference-Solutions.ipynb
- 7. Statistical Inference.ipynb
- 9. Bayesian Data Analysis-Solutions.ipynb
- 9. Bayesian Data Analysis.ipynb
- Collaborative.ipynb
- Regression_solutions.ipynb
- Section4_3-Introduction-to-PyMC.ipynb
- Untitled0.ipynb
- pymc.ipynb
- 0.1 Syllabus.key
- 0.2 What is Data Science.key
- 0.3 Python Data Science.key

What is IPython?

I is for interactive

I is for interactive


Home
1. Crash course on Python
localhost:8888/notebooks/1.%20Crash%20course%20on%20Python.ipynb
jupyter
1. Crash course on Python Last Checkpoint: 13 minutes ago (autosaved)
File Edit View Insert Cell Kernel Help
Python 2
Code
Cell Toolbar: None
Typesetting math: 100%

Python

Once you have IPython installed, you are ready to perform all sorts of operations.

The software program that you use to invoke operators is called an **interpreter**. You enter your commands as a 'dialog' between you and the interpreter. Commands can be entered as part of a script (a text file with a list of commands to perform) or directly at the *cell*.

In order to ask to the interpreter what to do, you must **invoke** an operator:

```
In [3]: 3 + 4 + 9
Out[3]: 16

In [4]: range(10)
Out[4]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

It's helpful to think of the computation carried out by an operator as involving four parts:

- The name of the operator
- The input arguments
- The output value
- Side effects

A typical operation takes one or more input arguments and uses the information in these to produce an output value. Along the way, the computer might take some action: display a graph, store a file, make a sound, etc. These actions are called side effects.

Python is a general-purpose programming language, so when we want to use more specific commands (such as statistical operators or string processing oeprators) we usually need to import them before we can use them. For Scientific Python, one of the most important libraries that we need is numpy (Numerical Python), which can be loaded like this: