

Python for Scientific Computing^{*}

1 Why Python for Scientific Computing

- For beginners there really is no difference between version two and three of *Python*.
- *Python* is the most popular coding language for teaching introductory computer science courses at top-ranked U.S. departments. Numerous online courses, lecture notes, and tutorials are readily available online.
- In the private sector, most recent results from *CodeEval* point in the same direction.
- *Python* is used heavily used by tech companies (e.g. Google, Dropbox, etc.) and in the financial sector (e.g. AQR).
- *Python* is so simple to learn, a lot if books explicitly target kids.

2 SciPy Stack

- *SciPy Library*, a collection of numerical algorithms and domain-specific toolboxes, including signal processing, optimization, statistics and much more
- *NumPy*, the fundamental package for numerical computation. It defines the numerical array and matrix types and basic operations on them
- *matplotlib*, a mature and popular plotting package, that provides publication-quality 2D plotting as well as rudimentary 3D plotting
- *pandas*, providing high-performance, easy to use data structures

^{*}For further information or questions and suggestions, please contact us at info@policy-lab.org.

- *SymPy*, for symbolic mathematics and computer algebra
- *IPython*, a rich interactive interface, letting you quickly process data and test ideas
- *nose*, a framework for testing Python code.

Depending on your particular specialization, these packages might be of additional interest to you.

- *statsmodels*, a *Python* module that allows users to explore data, estimate statistical models, and perform statistical tests.

statsmodels, together with *pandas*, is a potential replacement for the *R*, just use *rpy2* to call *R* functions directly from *Python*. All these packages are included in the *Anaconda Distribution*.

3 Basic Example

- The *IPython* notebook works in your web browser, allowing you to document your computation in an easily reproducible form. See a notebook for ? as an example here.

3.1 Data Visualization

- See the *matplotlib Thumbnail Gallery* for many and much more elaborate examples of data visualization.

3.2 Statistical Analysis

- We will fit an *Ordinary Least Squares (OLS)* model using *statsmodels*. See the online documentation for a full list of the library's capabilities.

4 Integrated Development Environment

- For simple analysis the IPython Notebook or even the command line is sufficient. However, for more involved scientific programming. I found the use of an IDE very useful.
- An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.

If we have time, we can get going on the Getting Started Guide for Students (<http://bit.ly/1WDDJny>) together.