

# Python for Scientific Computing

## Why *Python* for Scientific Computing

- 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 (<http://www.fullstackpython.com/best-python-resources.html>).
- In the private sector, most recent results from *CodeEval* (<https://www.codeeval.com>) point in the same direction.

## *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
- *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* (<https://www.continuum.io/why-anaconda>).

## Basic Example

- The *IPython* notebook works in your web browser, allowing you to document your computation in an easily reproducible form.

## Data Visualization

- See the *matplotlib Thumbnail Gallery* (<http://matplotlib.org/gallery.html>) for many and much more elaborate examples of data visualization.

## Statistical Analysis

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

## 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.