

Data Analytics with Python

Maximilian Blesch (Slides by Tobias Raabe)

Table of Contents

Why Python?

- ▶ **open source** (you are able to review the source code)
- ▶ **easy to learn** (you are able to write your own code)
- ▶ **general-purpose language** (you are able to perform all actions ranging from creating folders to analyzing data)
- ▶ **glue language** (your are able to implement a variety of other programming language into a project like R, C, Julia, etc.)
- ▶ **increasing popularity among the economics and econometrics community**
- ▶ **fast growing**

Projections of future traffic for major programming languages

Future traffic is predicted with an STL model, along with an 80% prediction interval.

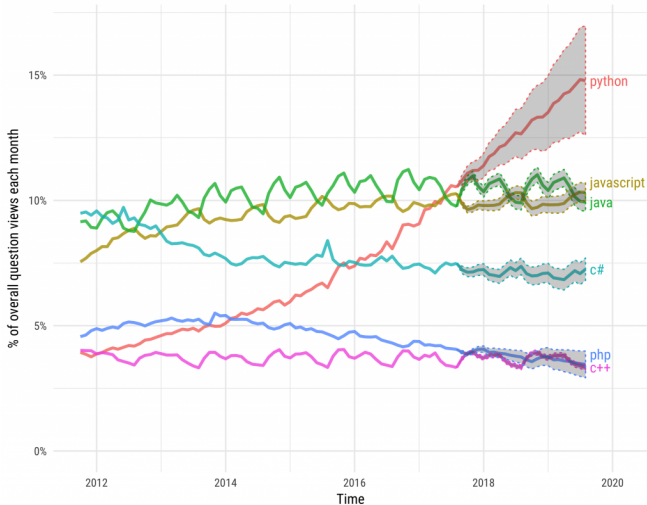


Figure: source: <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>

Stack Overflow Traffic to Questions About Selected Python Packages

Based on visits to Stack Overflow questions from World Bank high-income countries

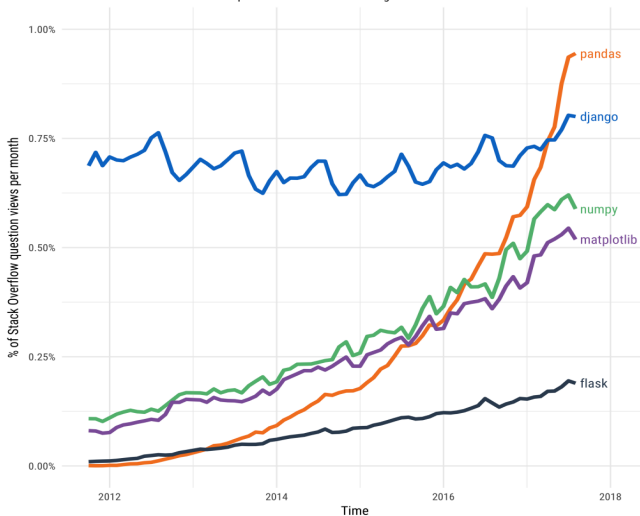


Figure: source: <https://stackoverflow.blog/2017/09/14/python-growing-quickly/>

Visits to Python by industry

Based on visits to Stack Overflow questions from the US/UK in January-August 2017.
The denominator in each is the total traffic from that industry.

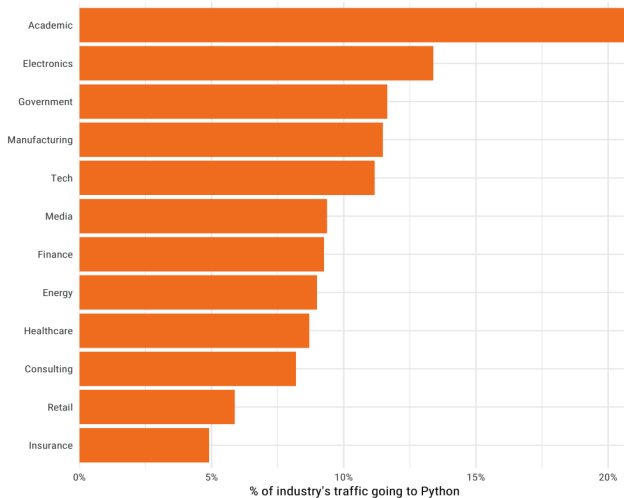


Figure: source: <https://stackoverflow.blog/2017/09/14/python-growing-quickly/>

Why not R, Stata

- R
 - ▶ major inspiration for most scientific Python packages
 - ▶ [Tidyverse](#) is a collection of incredible powerful data analysis tools
 - ▶ (my opinion: quirky syntax)

- Stata
 - ▶ proprietary and closed code base
 - ▶ useful for quick analysis
 - ▶ (my opinion: quirky syntax, hard to manage bigger projects, how to ensure reproducibility)

Scientific Computing Tools for Python

Packages¹

NumPy fundamental package for numerical operations

SciPy collection of numerical algorithms, statistics, optimizations, etc.

Matplotlib plotting library

pandas provides high-performance and easy-to-use data structures

scikit-learn collection of algorithms and tools for machine learning

Jupyter powerful IDE (integrated development environment) which combines python and markdown

Anaconda an installer for a preconfigured python environment containing the scientific stack and many other useful libraries

Setup

1. download the files required for the tutorial from [here](#), unzip and place them into a folder in your user directory.
2. download the installer for Python 3.6 from <https://www.anaconda.com/download/> and run it
 - ▶ *If you are asked whether Anaconda and its paths should be added to your system's PATH or not, choose the option to add them*
3. start the Jupyter notebook in one of two ways
 - 3.1 use terminal, shell, cmd, powershell to navigate to your project's folder and enter `jupyter notebook`
 - 3.2 start Jupyter via the Anaconda Navigator (installed with Anaconda)
4. make sure that you can navigate to the tutorial folder inside the opened tab in your browser
5. (optional) Start a new notebook by clicking on New in the top right corner and select Python 3

Tutorials

- ▶ Zed A. Shaw - Learn Python the Hard Way - General Python Tutorial
- ▶ Patrick Triest - Exploring US Policing Data using Python

Documentation

- ▶ [stackoverflow](#) - World's largest developer community

Others

- ▶ [Anaconda Distribution](#) delivers Python with a pre-compiled stack of scientific packages
- ▶ [Jake VanderPlas - Python Data Science Handbook](#) is inspiration for this tutorial
- ▶ [Wes McKinney - Python for Data Analysis](#) is book from the developer of pandas
- ▶ [Python Weekly](#) is a weekly newsletter which covers all aspects of Python but also includes links to tutorials, etc.
- ▶ [Kaggle](#) is a data science and machine learning community with tutorials, competitions, etc.
- ▶ [Templates for Reproducible Research Projects in Economics](#) by Hans-Martin von Gaudecker
- ▶ [Cookiecutter - Data Science](#) is a template for the structure of a research project