Python for Data Science

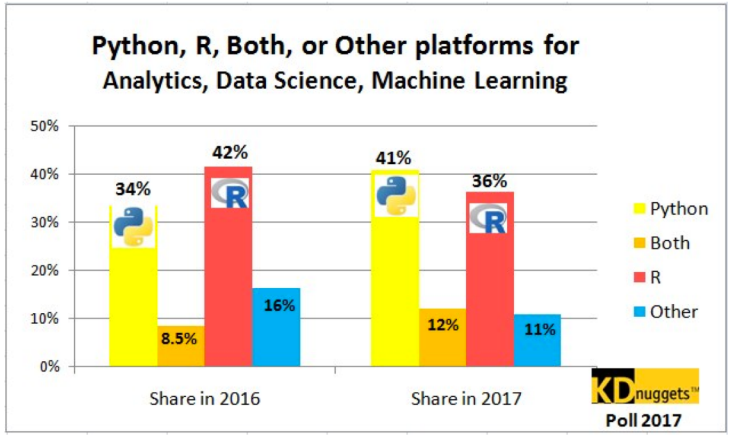
Jay Urbain, PhD

Topics

- Rationale for Python
- Python packages for data science
- Hands-on tutorials

Why Python?

- Lots of momentum in data science and machine learning.
- Easy to learn
  - Language of choice for 8 of 10 top US computer science programs (Philip Guo, CACM).
- Full featured
  - Not just a statistics language. Has full capabilities for data acquisition, cleaning, databases, high performance computing, etc.
- Powerful and efficient libraries:
  - NumPy, Pandas for data manipulation, processing, and analysis.
  - SciPy, StatsModels for statistics.
  - State of the art machine learning: SciKit-learn, TensorFlow, Keras, PyTorch.
- Can be used for end-to-end development
  - Data exploration, analysis, modeling, and deployment.



SciPy

- **Scientific Programming for** *Python.*
- Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:
  - NumPy – base N-dimensional array package
  - SciPy library – library for scientific computing
  - Matplotlib - plotting
  - iPython – enhanced interactive console
  - Sympy – symbolic mathematics
  - Pandas – data structures & analysis
  - StatsModels – statistics, like R syntax

https://www.scipy.org/

NumPy

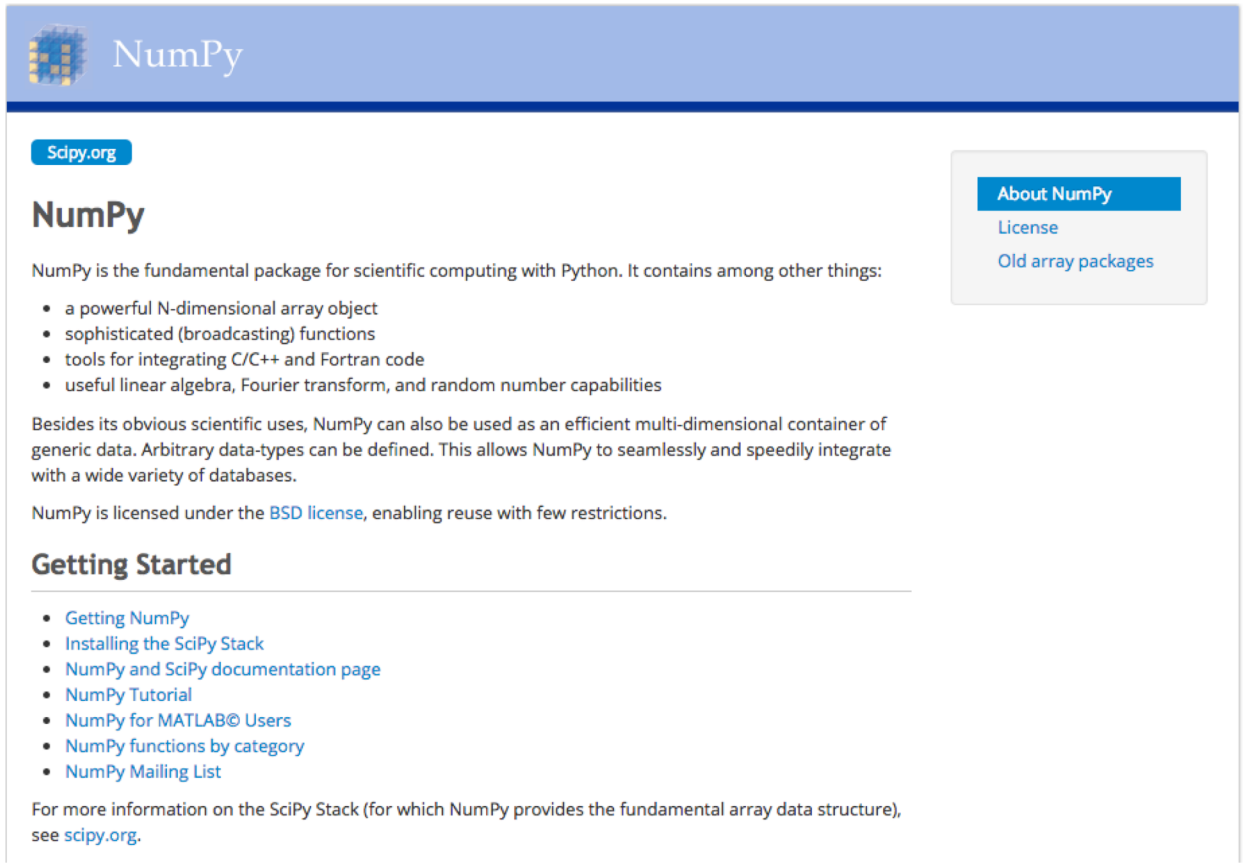
- **NumPy** stands for ***Num****erical* ***Py****thon.*
- Fundamental package for scientific computing in Python.
- Provides Python with an extensive math library capable of performing numerical computations effectively and efficiently.
- Underpins Pandas and many machine learning libraries.
- Vector based versus iterative processing

- **NumPy Manual**
- **NumPy User Guide**
- **NumPy Reference**
- **Scipy Lectures**

Why NumPy

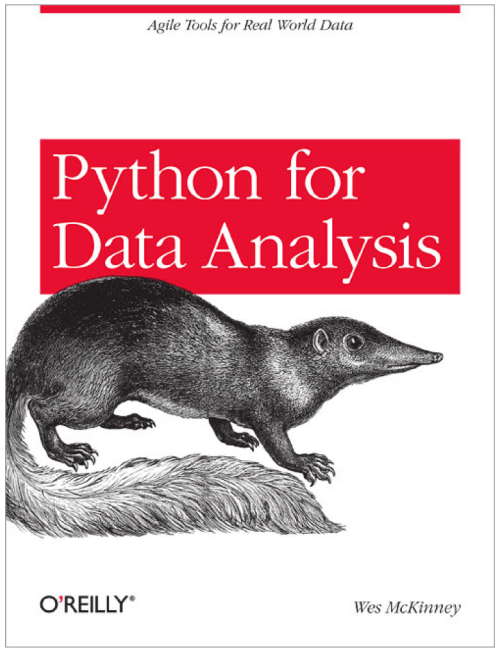
- NumPy has a number of key features that give it great advantages over Python lists.
- One such feature is speed. When performing operations on large arrays NumPy can often perform several orders of magnitude faster than Python lists.
- This speed comes from the nature of NumPy arrays being memory-efficient and from optimized algorithms used by NumPy for doing arithmetic, statistical, and linear algebra operations.
- NumPy is that it has multidimensional array data structures that can represent vectors and matrices.
- Machine learning algorithms rely on matrix operations.
  - For example, when training a Neural Network, you often have to carry out many matrix multiplications. NumPy is optimized for matrix operations and it allows us to do Linear Algebra operations effectively and efficiently, making it very suitable for solving machine learning problems.
- NumPy has become so popular that a lot of Python packages, such as Pandas, are built on top of NumPy.

http://www.numpy.org/

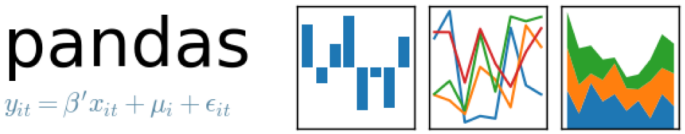


Pandas

- **pandas** - Python Data Analysis Library
- Open-source data structures and data analysis tools tools for Python
- Key data structures:
  - Series
  - Dataframe (similar to R dataframe abstraction)
- Uses NumPy, MatplotLib
- Developed by Wes McKinney



https://pandas.pydata.org/



Python Data Analysis Library

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

pandas is a NumFOCUS sponsored project. This will help ensure the success of development of *pandas* as a world-class open-source project, and makes it possible to donate to the project.

A Fiscally Sponsored Project of
NUMFOCUS
OPEN CODE = BETTER SCIENCE

v0.23.1 Final (June 12, 2018)

This is a minor bug-fix release in the 0.23.x series and includes some regression fixes, bug fixes, and performance improvements. We recommend that all users upgrade to this version.

The release can be installed with conda from conda-forge or the default channel:

VERSIONS

Release
0.23.1 - June 2018
download // docs // pdf

Development
0.24.0 - 2018
github // docs

Previous Releases
0.23.0 - download // docs // pdf
0.22.0 - download // docs // pdf
0.21.1 - download // docs // pdf
0.21.0 - download // docs // pdf
0.20.3 - download // docs // pdf
0.19.2 - download // docs // pdf
0.18.1 - download // docs // pdf
0.17.1 - download // docs // pdf
0.16.2 - download // docs // pdf
0.15.2 - download // docs // pdf
0.14.1 - download // docs // pdf
0.13.1 - download // docs // pdf
0.12.0 - download // docs // pdf

Pandas

- A fast and efficient **DataFrame** object for data manipulation and indexing.
- Tools for **reading and writing data** between in-memory data structures.
- Handling of **missing data**.
- Flexible **reshaping** and pivoting of data sets.
- Label-based **slicing**, **fancy indexing**, and **subsetting** of large data sets.
- Aggregating or transforming data with a powerful **group by** engine.
- High performance **merging and joining** of data sets.
- **Hierarchical axis indexing** provides a way of working with high-dimensional data in a lower-dimensional data structure.
- **Time series**-functionality.
- Highly **optimized for performance**, with critical code paths written in Cython or C.

MatplotLib

- Python 2D plotting library.
- Produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.
- Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

https://matplotlib.org/

matplotlib

Version 2.2.2

home | examples | tutorials | pyplot | docs »

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

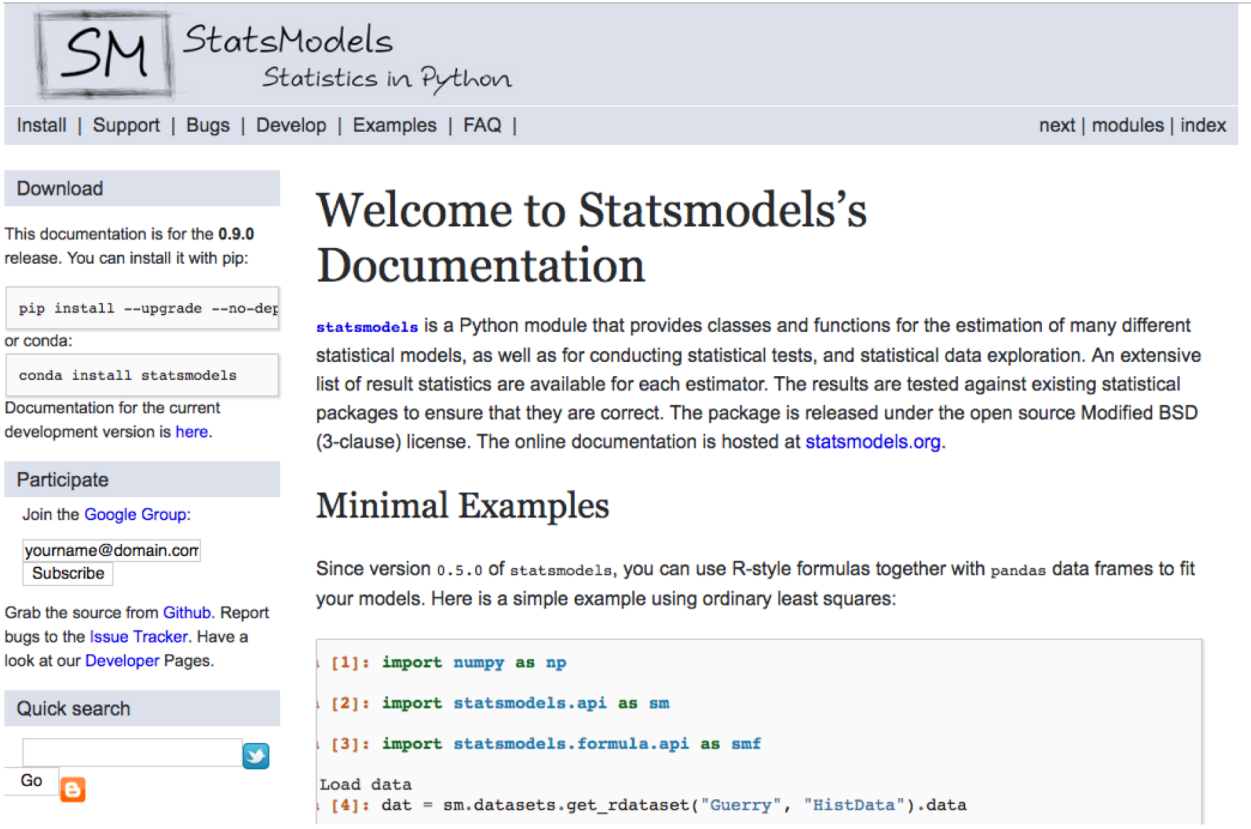
Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the `pyplot` module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

StatsModels

- Python module that provides classes and functions for the estimation of many different statistical models.
- Conducting statistical tests, and statistical data exploration.
- Provides result statistics for each estimator.

https://www.statsmodels.org/stable/index.html



StatsModels
Statistics in Python

Install | Support | Bugs | Develop | Examples | FAQ | next | modules | index

Download

This documentation is for the **0.9.0** release. You can install it with pip:

```
pip install --upgrade --no-dep
```

or conda:

```
conda install statsmodels
```

Documentation for the current development version is here.

Participate

Join the Google Group:

yourname@domain.com
Subscribe

Grab the source from Github. Report bugs to the Issue Tracker. Have a look at our Developer Pages.

Quick search

Go

Welcome to Statsmodels's Documentation

statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. An extensive list of result statistics are available for each estimator. The results are tested against existing statistical packages to ensure that they are correct. The package is released under the open source Modified BSD (3-clause) license. The online documentation is hosted at statsmodels.org.

Minimal Examples

Since version `0.5.0` of `statsmodels`, you can use R-style formulas together with `pandas` data frames to fit your models. Here is a simple example using ordinary least squares:

```
[1]: import numpy as np

[2]: import statsmodels.api as sm

[3]: import statsmodels.formula.api as smf

Load data
[4]: dat = sm.datasets.get_rdataset("Guerry", "HistData").data
```

scikit-learn

- Machine learning in Python
- Tools for data mining and data analysis.
- Classification, regression, dimensionality reduction, model selection.
- Relatively easy to use, and reusable in various contexts.
- Built on NumPy, SciPy, and matplotlib.
- Extensive examples and support.
- Does not include deep learning.

http://scikit-learn.org/stable/index.html

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, … — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, … — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, … — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

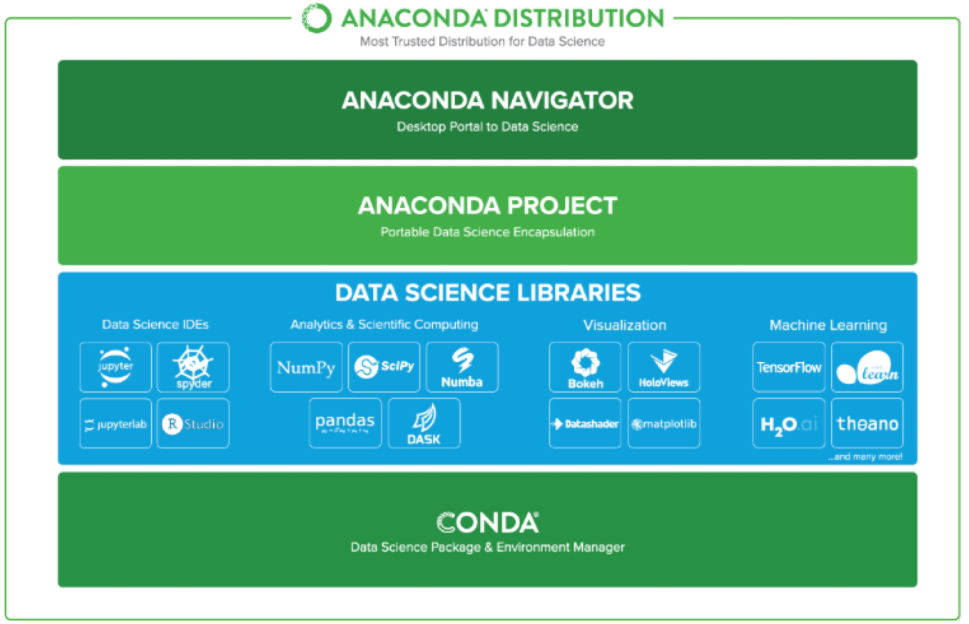
Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

Anaconda

- Most popular Python distribution for data science
- Includes most packages already discussed
- Very efficient and flexible package management
- https://www.anaconda.com/distribution/



Students TODO:

- Start numpy notebooks.

p, Lambda, and List Comprehensions