



Python for Data Science

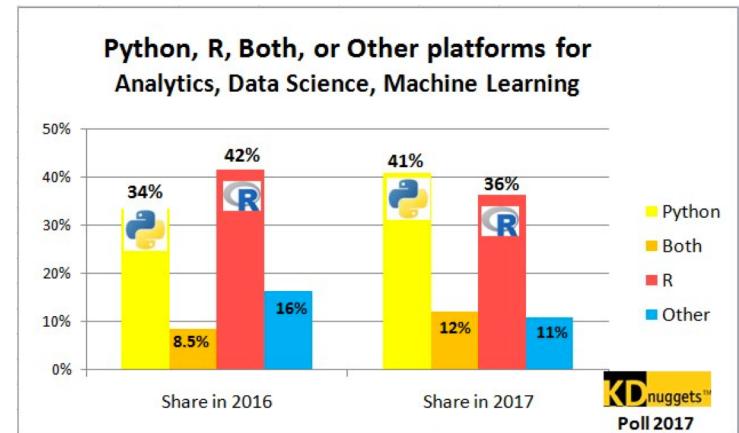
Jay Urbain, PhD

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/>

The screenshot shows the official website for SciPy.org. At the top, there's a blue header bar with the SciPy logo and the text "Sponsored By ENTHOUGHT". Below the header, there are five main navigation icons: "Install" (blue circle with a green arrow), "Getting Started" (yellow circle with a green "S"), "Documentation" (blue circle with a white book), "Report Bugs" (blue circle with a red bug), and "Blogs" (orange circle with a white RSS feed icon). A descriptive text block below these icons states: "SciPy (pronounced "Sigh Pie") is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:". To the right of this text is a sidebar with a list of links: "About SciPy", "Install", "Getting Started", "Documentation", "Bug Reports", "Topical Software", "Citing", "Cookbook", "SciPy Conferences", "Blogs", and "NumFOCUS". Further down the page, there are two rows of "CORE PACKAGES:" with their respective icons and descriptions: NumPy (Base N-dimensional array package), SciPy library (Fundamental library for scientific computing), Matplotlib (Comprehensive 2D Plotting), IPython (Enhanced Interactive Console), Sympy (Symbolic mathematics), and pandas (Data structures & analysis). At the bottom of the page is a blue button labeled "More information...".

SciPy.org

Sponsored By
ENTHOUGHT

Install Getting Started Documentation Report Bugs Blogs

SciPy (pronounced "Sigh Pie") is a 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
Fundamental library for scientific computing

Matplotlib
Comprehensive 2D Plotting

IPython
Enhanced Interactive Console

Sympy
Symbolic mathematics

pandas
Data structures & analysis

About SciPy
Install
Getting Started
Documentation
Bug Reports
Topical Software
Citing
Cookbook
SciPy Conferences
Blogs
NumFOCUS

CORE PACKAGES:
NumPy
SciPy library
Matplotlib
IPython
Sympy
Pandas

More information...

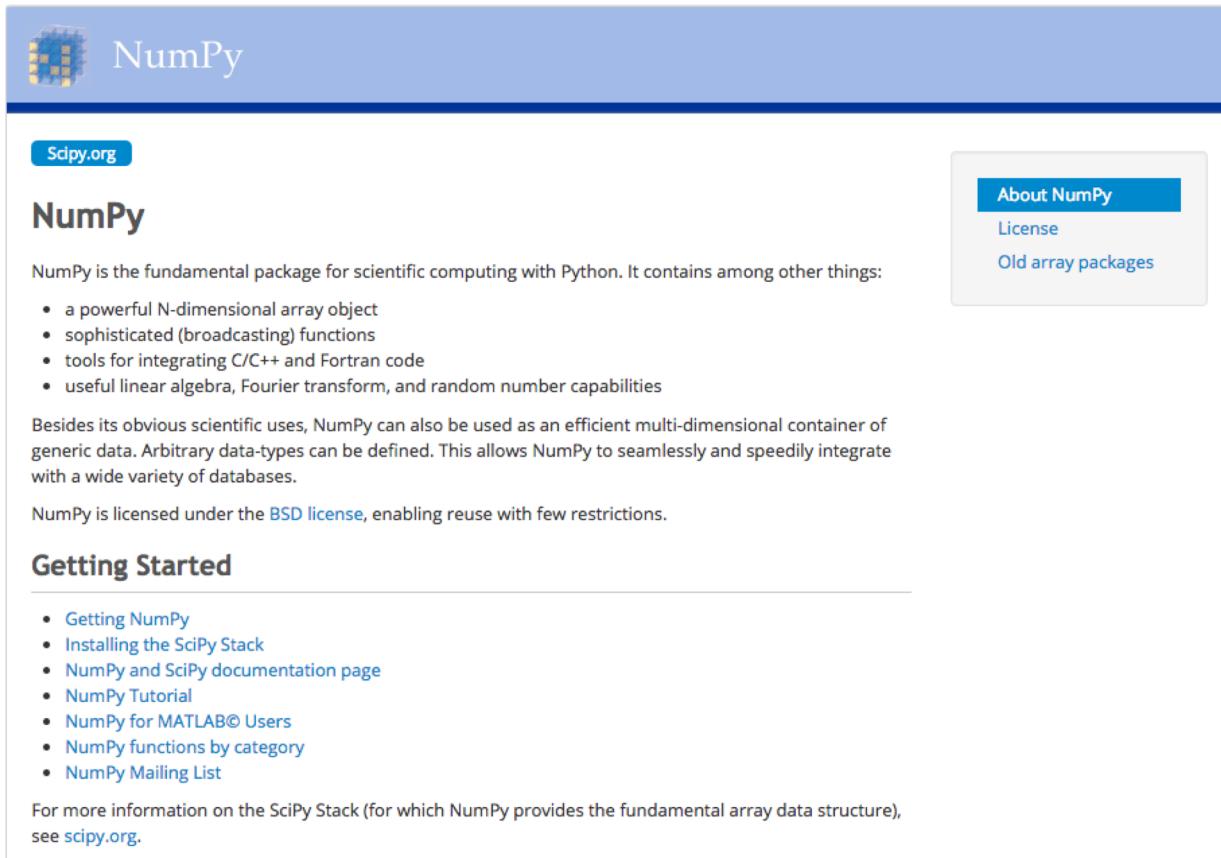
NumPy

- NumPy stands for *Numerical Python*.
 - 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/>

The screenshot shows the official NumPy website. At the top, there's a blue header bar with the NumPy logo (a 3x3 grid of colored squares) and the word "NumPy". Below the header, a navigation bar includes links for "Scipy.org" and "About NumPy". The main content area has a light gray background. It features a section titled "NumPy" with a brief introduction and a bulleted list of features. Another section discusses NumPy's use as a generic data container. A "Getting Started" sidebar on the right lists various resources like "Getting NumPy" and "NumPy Tutorial".

NumPy

Scipy.org

About NumPy

License

Old array packages

NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

NumPy is licensed under the [BSD license](#), enabling reuse with few restrictions.

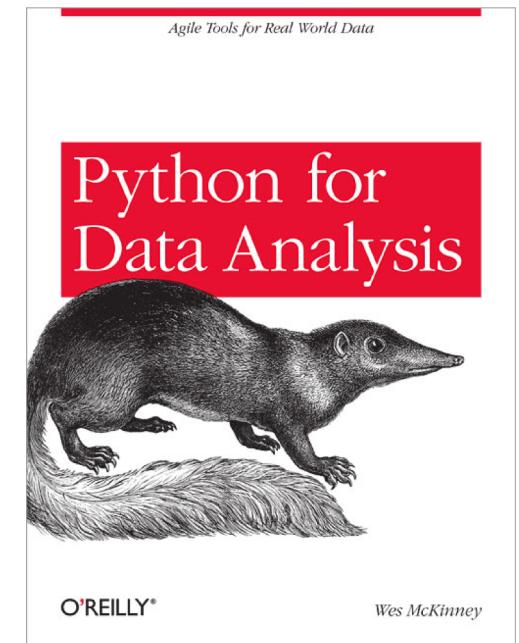
Getting Started

- [Getting NumPy](#)
- [Installing the SciPy Stack](#)
- [NumPy and SciPy documentation page](#)
- [NumPy Tutorial](#)
- [NumPy for MATLAB® Users](#)
- [NumPy functions by category](#)
- [NumPy Mailing List](#)

For more information on the SciPy Stack (for which NumPy provides the fundamental array data structure), see [scipy.org](#).

Pandas

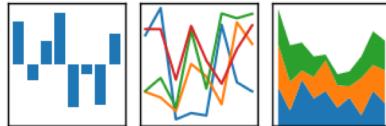
- **pandas** - Python Data Analysis Library
- Open-source data structures and data analysis tools for Python
- Key data structures:
 - Series
 - Dataframe (similar to R dataframe abstraction)
- Uses NumPy, Matplotlib
- Developed by Wes McKinney



<https://pandas.pydata.org/>

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



[home](#) // [about](#) // [get pandas](#) // [documentation](#) // [community](#) // [talks](#) // [donate](#)

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



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](#)

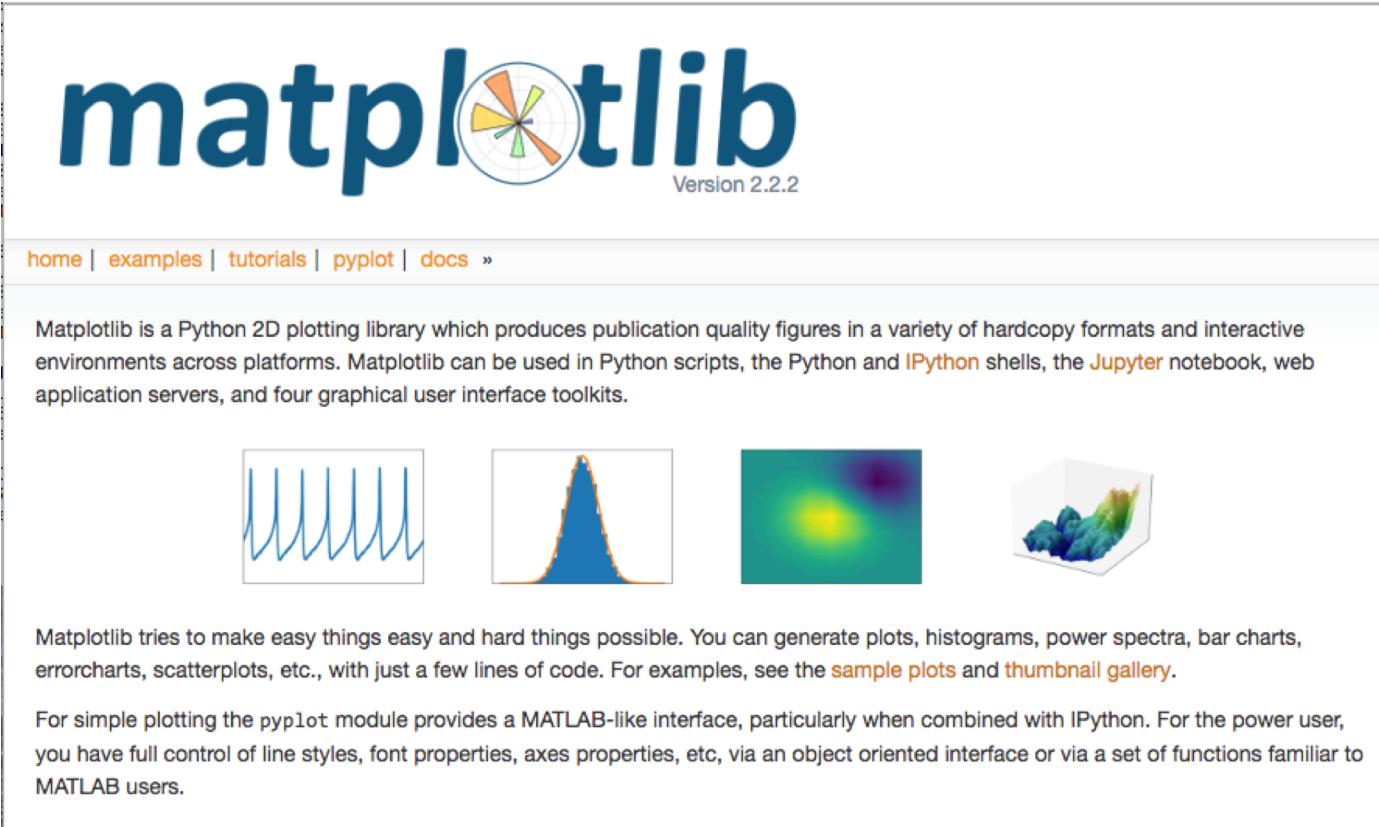
Pandas

- 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/>

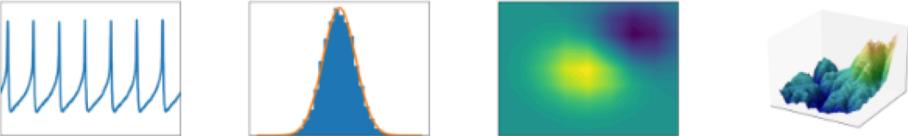


The screenshot shows the official Matplotlib website. At the top, the word "matplotlib" is written in a large, bold, blue sans-serif font. To the right of the word is a circular logo containing a multi-colored sunburst or radar-like pattern. Below the word and logo, the text "Version 2.2.2" is displayed. A horizontal navigation bar below the header contains links for "home", "examples", "tutorials", "pyplot", and "docs". The main content area begins with a paragraph describing Matplotlib as a Python 2D plotting library. It highlights its use in various environments like Python scripts, IPython shells, Jupyter notebooks, web servers, and graphical user interface toolkits. Below this text are four small square thumbnails showing different types of plots: a line plot with multiple oscillating curves, a histogram with a normal distribution curve overlaid, a 2D heatmap with a central peak, and a 3D surface plot.

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>

The screenshot shows the StatsModels documentation website. At the top left is the logo 'SM' for StatsModels. To its right is the title 'StatsModels' and the subtitle 'Statistics in Python'. Below the title is a navigation bar with links: 'Install | Support | Bugs | Develop | Examples | FAQ |'. On the far right of the navigation bar are links for 'next | modules | index'. The main content area has a grey header 'Download'. Below it, text says 'This documentation is for the 0.9.0 release. You can install it with pip:' followed by code snippets for pip and conda installation. It also mentions documentation for the current development version. A 'Participate' section includes links to a Google Group and a GitHub account. Another section encourages grabbing the source from GitHub and reporting bugs. A 'Quick search' bar is at the bottom left, and a 'Go' button with a magnifying glass icon is to its right.

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](https://www.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.

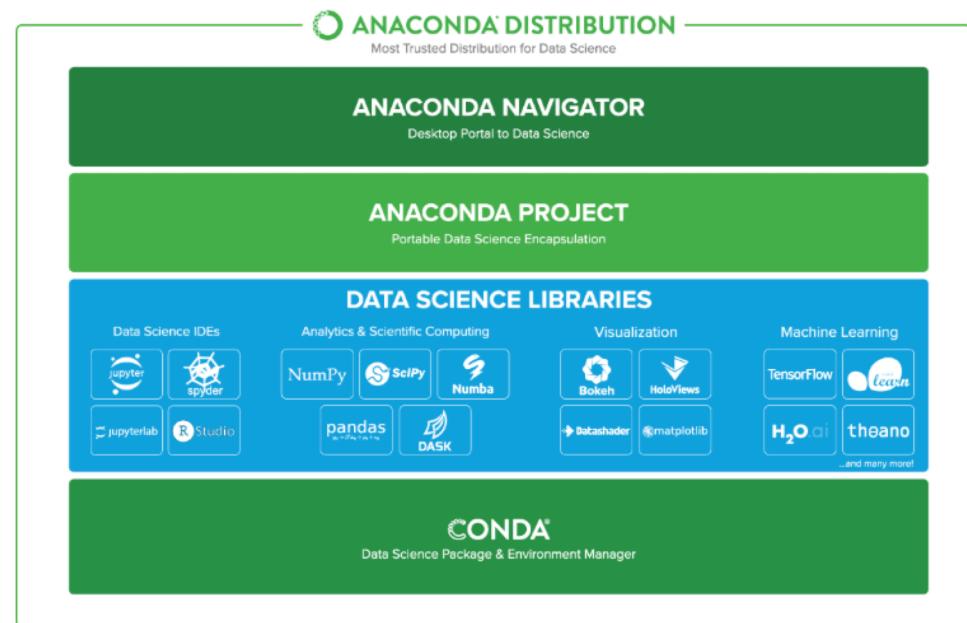
Application: Transforming input data such as text for use with machine learning algorithms.

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.