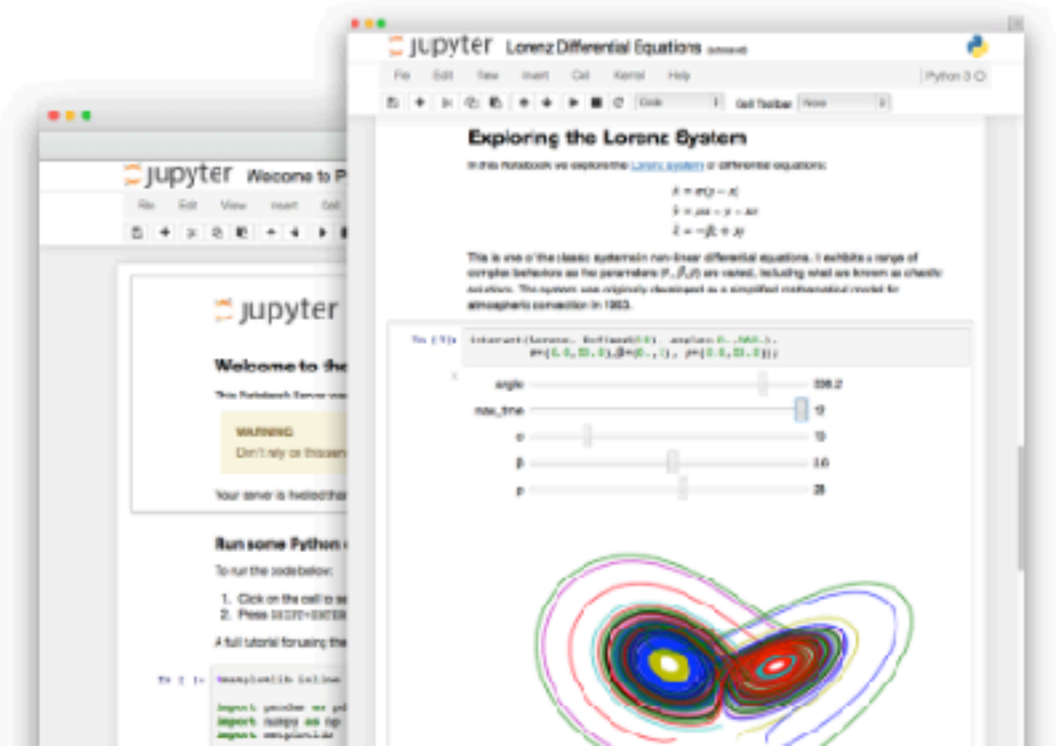
Linear Regression with Python

Tools

jupyter INSTALL PROJECT DOCUMENTATION BLOG DONATE

Jupyter Notebook

The Jupyter Notebook is a web-based interactive computing platform that allows users to author data- and code-driven narratives that combine live code, equations, narrative text, visualizations, interactive dashboards and other media.


jupyter Lorenz Differential Equations
File Edit View Insert Cell Kernel Help
Python 3
Cell Toolbar
jupyter Welcome to P
WARNING
angle
max_time

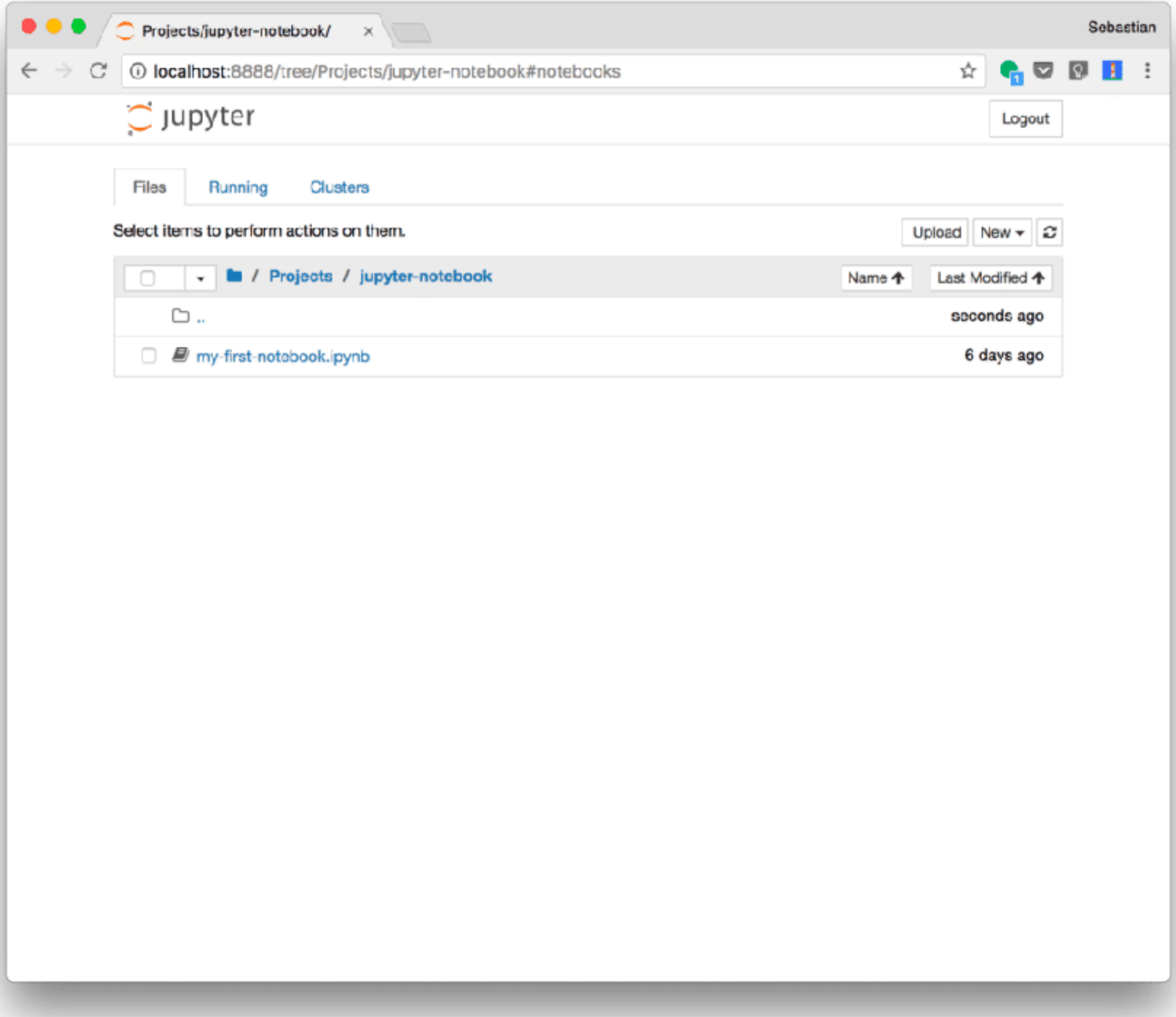
Exploring the Lorenz System

Welcome to the

Run some Python

To run the code below:

1. Click on the cell to s
2. Press

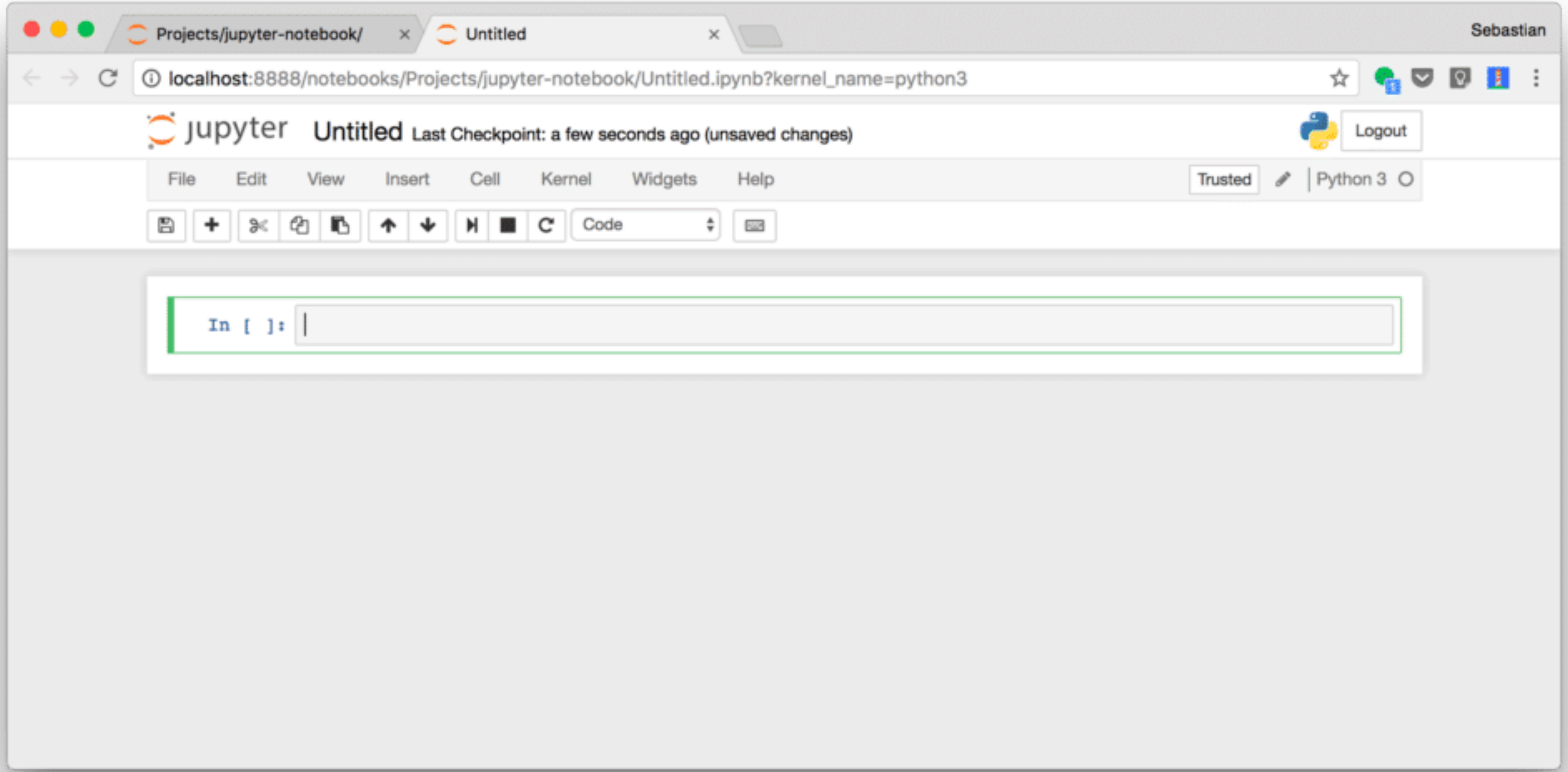


Projects/jupyter-notebook/
Sebastian
localhost:8888/tree/Projects/jupyter-notebook#notebooks
jupyter
Logout
Files
Running
Clusters
Upload
New
Name
Last Modified

Select items to perform actions on them.

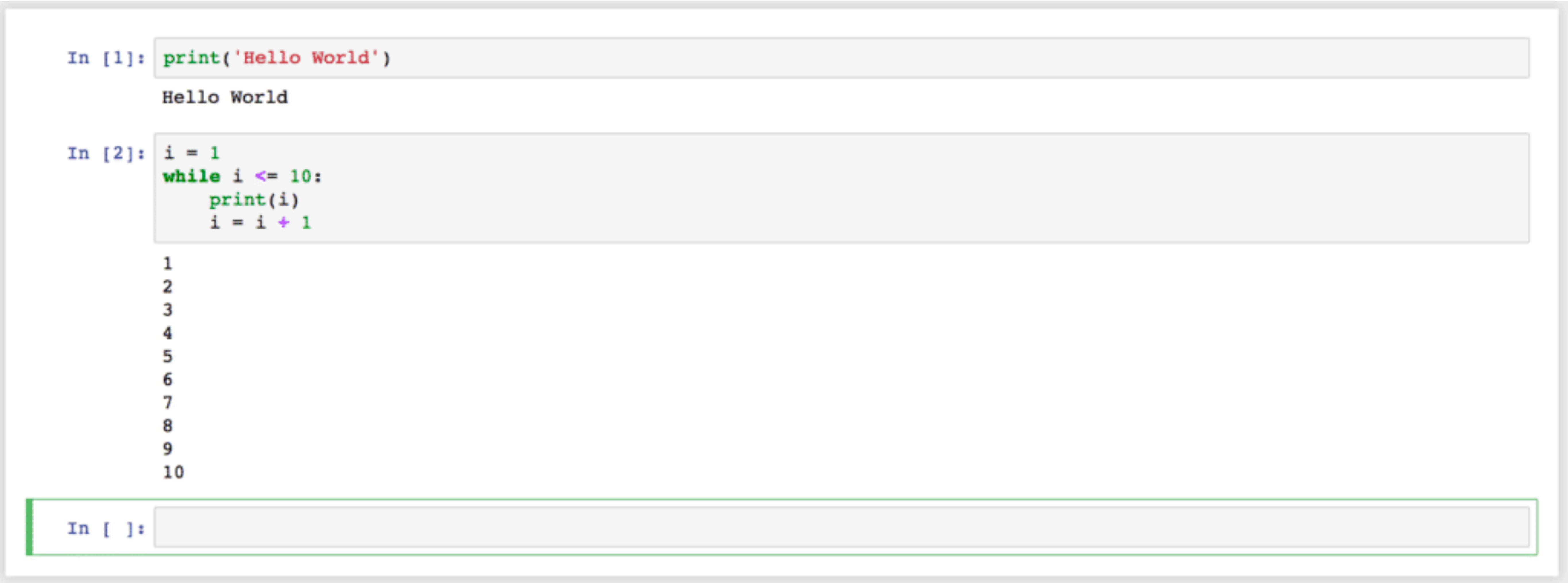
/ Projects / jupyter-notebook

..	seconds ago
my-first-notebook.ipynb	6 days ago



localhost:8888/notebooks/Projects/jupyter-notebook/Untitled.ipynb?kernel_name=python3

Untitled Last Checkpoint: a few seconds ago (unsaved changes)



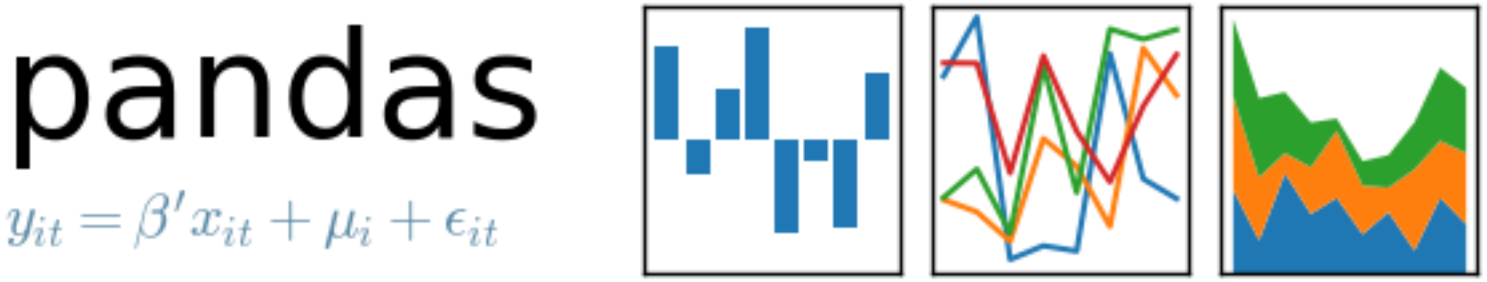
```
In [1]: print('Hello World')
```

```
Hello World
```

```
In [2]: i = 1
        while i <= 10:
            print(i)
            i = i + 1
```

```
1
2
3
4
5
6
7
8
9
10
```

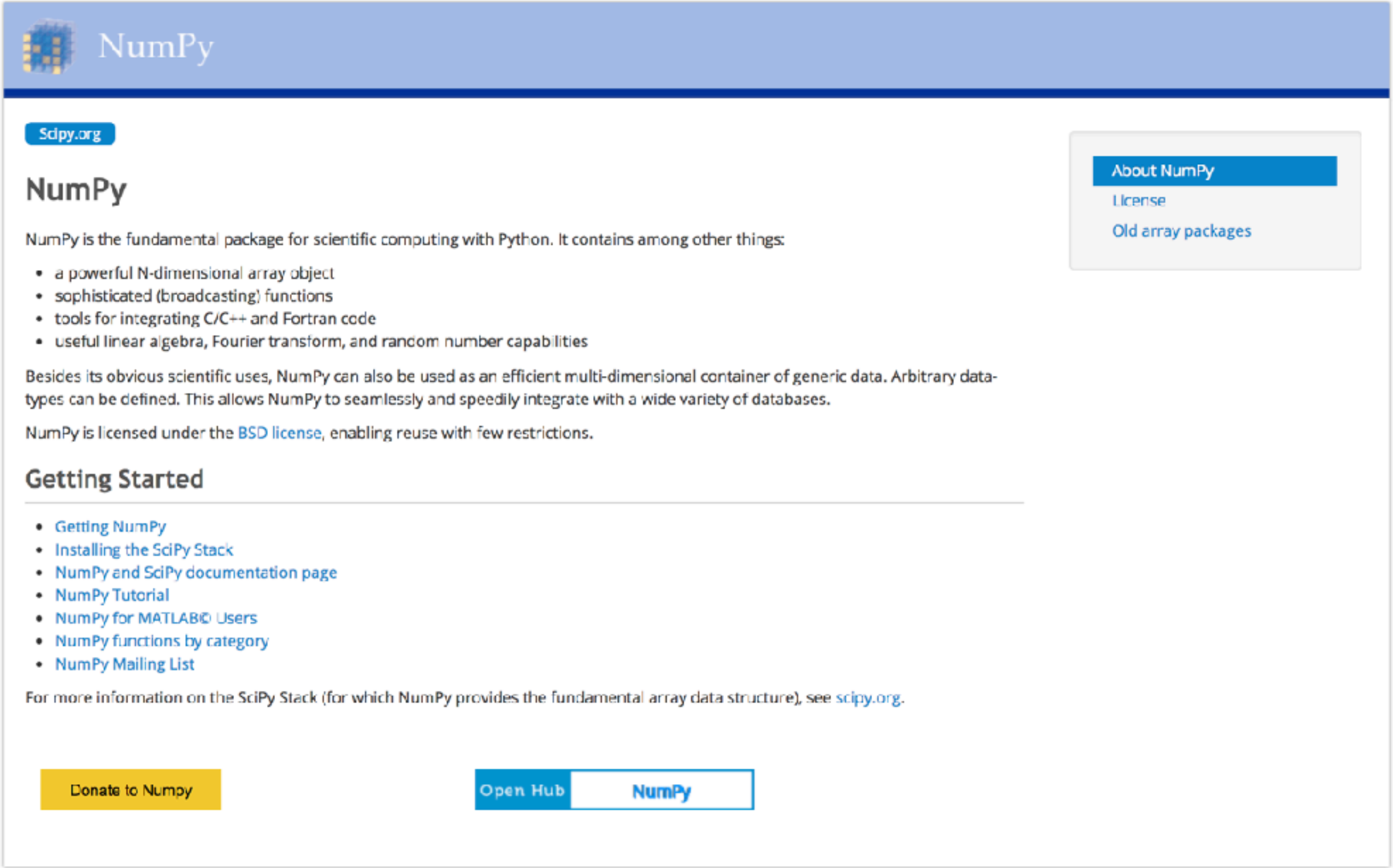
```
In [ ]:
```



pandas

- open source library
- high-performance, easy-to-use data structures and data analysis tools

https://pandas.pydata.org/



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.

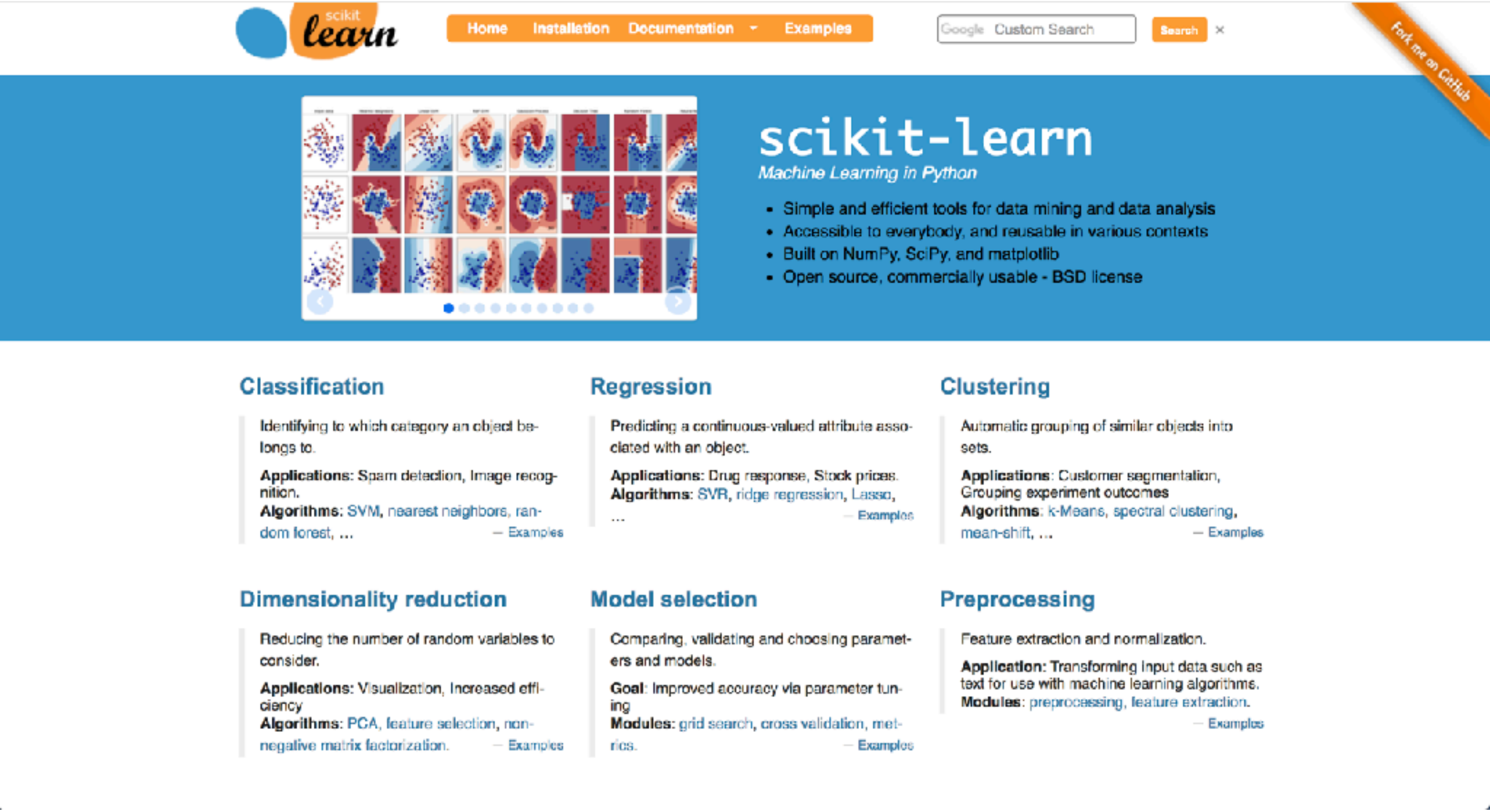
NumPy is licensed under the BSD license, enabling reuse with few restrictions.

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.

http://www.numpy.org/



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

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

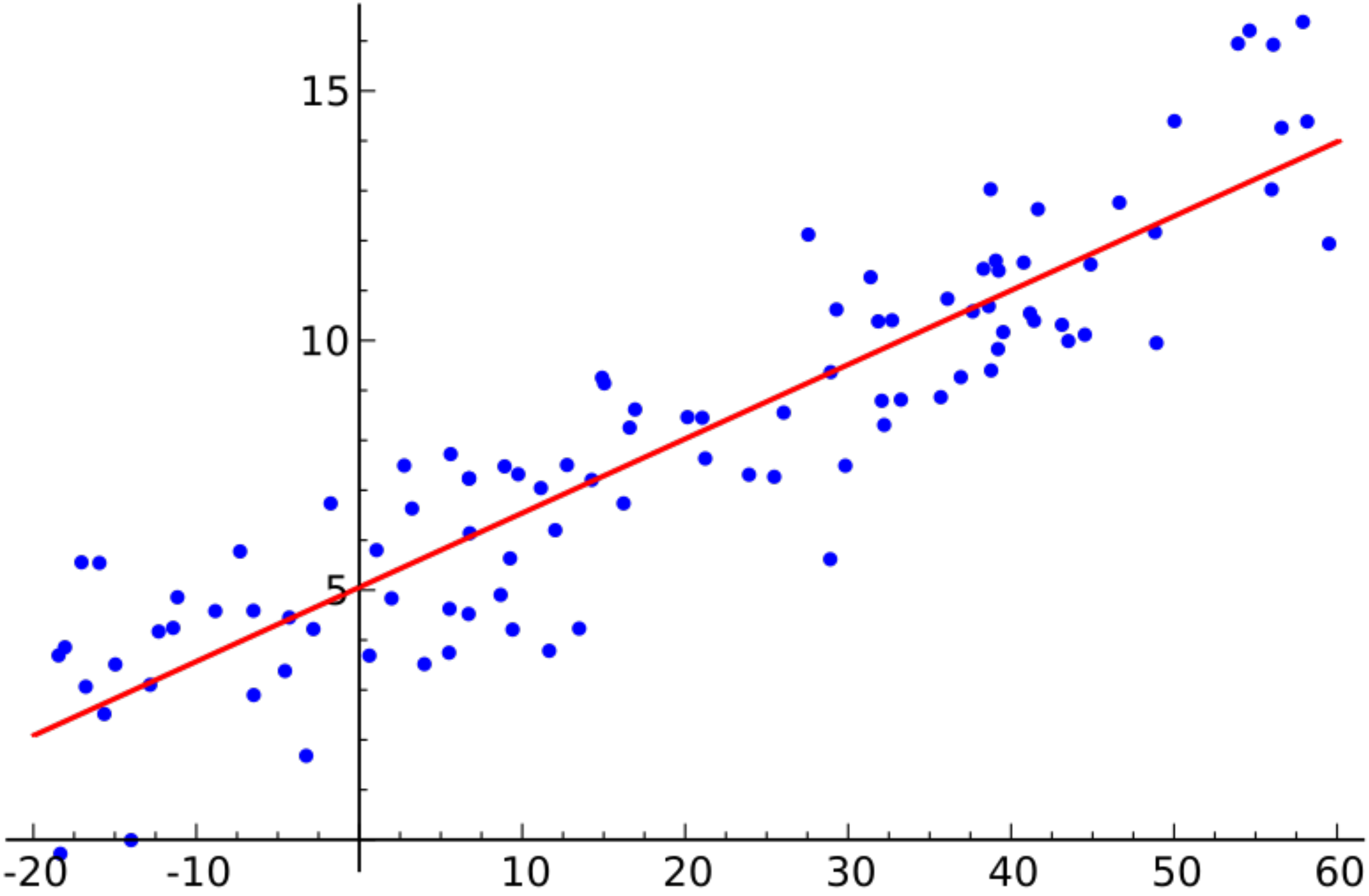
Feature extraction and normalization.

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

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

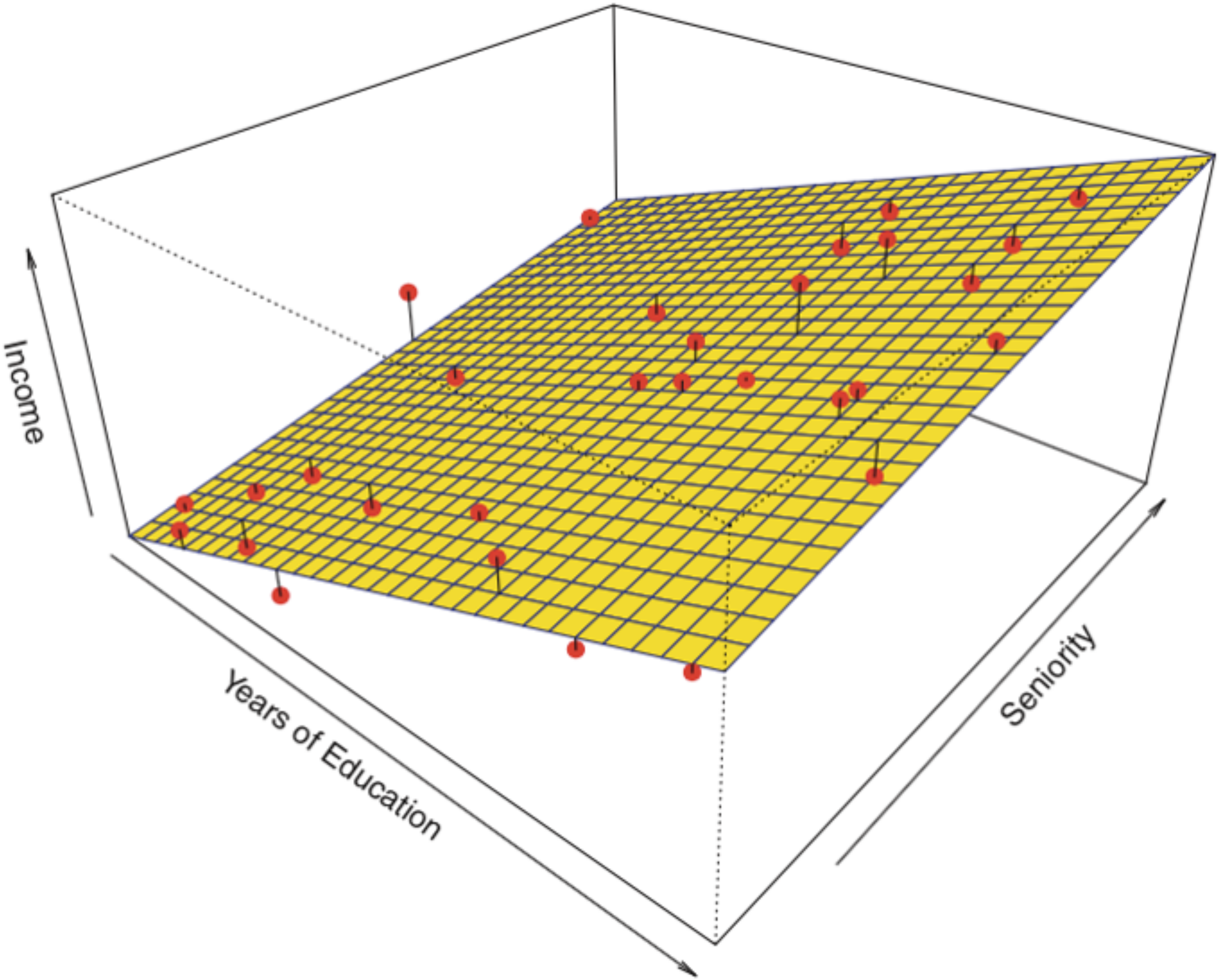
Linear Regression

$y = a\,x + b$



https://en.wikipedia.org/wiki/Linear_regression

$y = a\,x + b\,y + c$



http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R5_Correlation-Regression/R5_Correlation-Regression4.html

General formulation:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + ... + w_D x_D$$

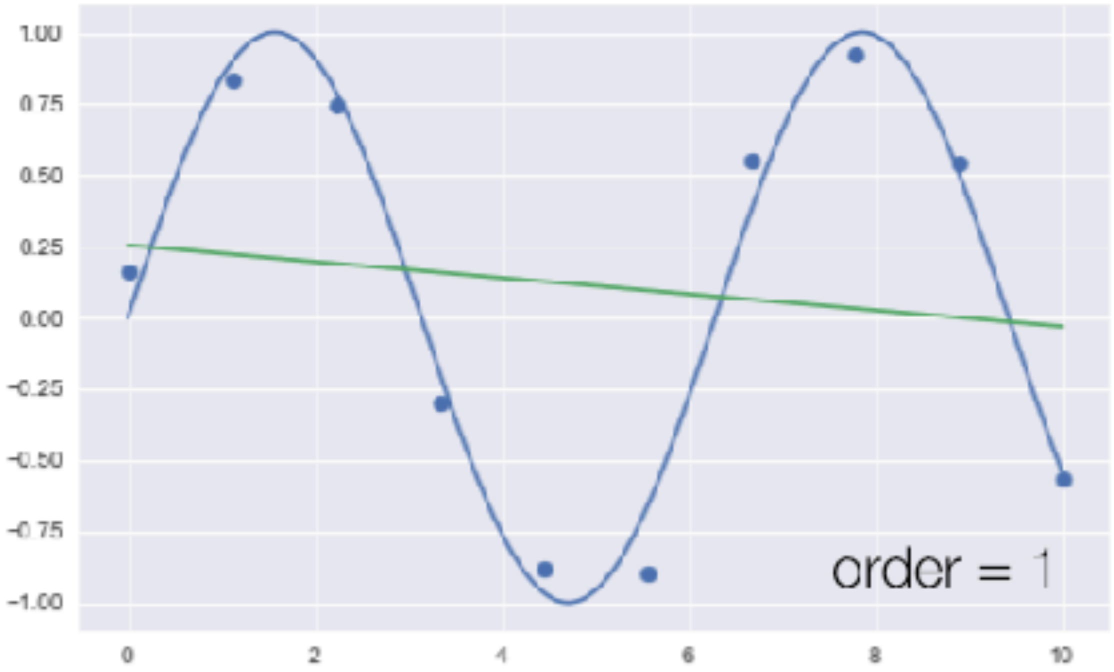
General formulation:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + ... + w_D x_D$$

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j x_j$$

General formulation:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + ... + w_D x_D$$



?

The key property of this model is that it is a linear function of the parameters !

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j x_j$$

Linear in x and w

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \boxed{\phi_j(\mathbf{x})}$$

Linear in w and phi(x)

The key property of this model is that it is a linear function of the parameters !

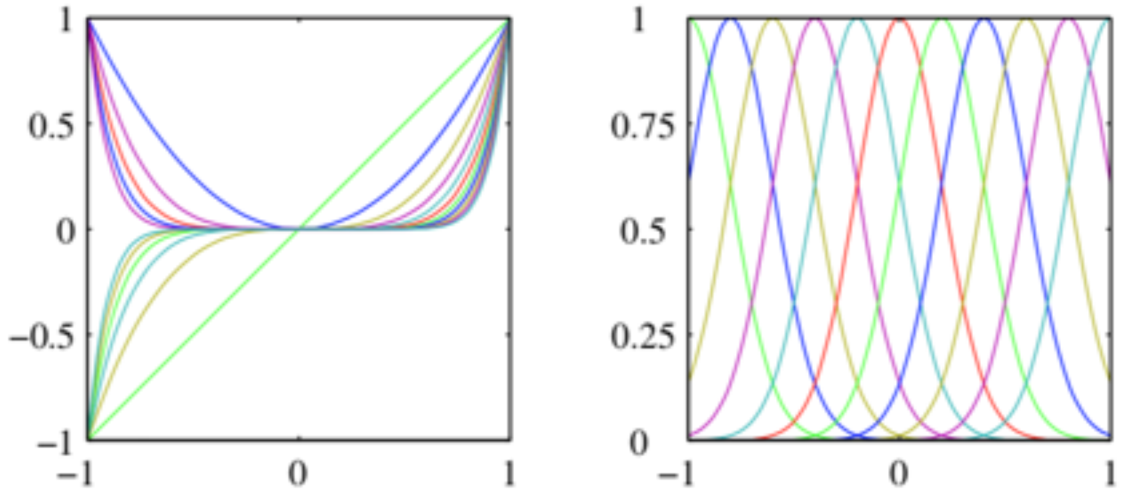
$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j x_j$$

Linear in x and w

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \boxed{\phi_j(\mathbf{x})}$$

Linear in w and phi(x)

Basis functions



$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x})$$

$$\phi_0(\mathbf{x}) = 1$$
$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \phi(\mathbf{x})$$

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j x_j$$

- **goal**: Predicting a continuous y as a function of the variables **x** and the parameters **w**

- **Model assumption**:
  - data generated by $t = y(\mathbf{x}, \mathbf{w}) + \epsilon$
  - Error ϵ ~ normal distribution with zero mean

- **Loss function**: sum-of-squares error between t and prediction y (least squares)