

# Python for Data Science

Prof. Dr. Wolfgang Ecker, Lorenzo Servadei,  
Elena Zennaro  
19<sup>th</sup> October 2018



# Content

- 1 Overview of Python libraries for Data Science
- 2 NumPy
- 3 SciPy
- 4 Pandas
- 5 SciKit-Learn
- 6 Matplotlib
- 7 Seaborn
- 8 Anaconda and Jupyter Notebook

# Python Libraries for Data Science

## Data Analysis

- › NumPy
- › SciPy
- › Pandas
- › SciKit-Learn

## Data Visualization

- › Matplotlib
- › Seaborn

These are the libraries we will use during our Course!

# NumPy

- › Fundamental package for scientific computing with Python
- › Objects for multidimensional arrays and matrices, as well as functions that allow to easily perform advanced mathematical and statistical operations on those objects
- › Provides vectorization of mathematical operations on arrays and matrices which significantly improves the performance

**Link:** <http://www.numpy.org/>

# SciPy

- › Collection of algorithms for linear algebra, differential equations, numerical integration, optimization, statistics and more
- › It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data
- › Routines for computing integrals numerically, solving differential equations, optimization and sparse matrices

**Link:** <https://www.scipy.org/scipylib/>

# Pandas

- › Adds data structures and tools designed to work with table-like data (similar to Series and Data Frames in R)
- › Provides tools for data manipulation: reshaping, merging, sorting, slicing, aggregation etc.
- › Allows handling missing data

	rank	discipline	phd	service	sex	salary
0	Prof	B	56	49	Male	186960
1	Prof	A	12	6	Male	93000
2	Prof	A	23	20	Male	110515
3	Prof	A	40	31	Male	131205
4	Prof	B	20	18	Male	104800

**Link:** <http://pandas.pydata.org/>

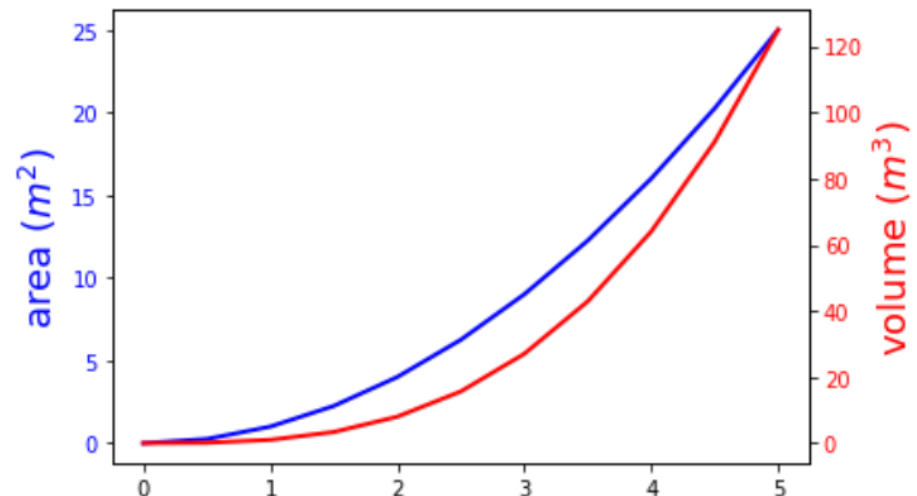
# SciKit-Learn

- › Provides machine learning algorithms: classification, regression, clustering, model validation etc.
- › Built on NumPy, SciPy and Matplotlib

**Link:** <http://scikit-learn.org/>

# Matplotlib

- › Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats
- › A set of functionalities similar to those of MATLAB
- › Line plots, scatter plots, barcharts, histograms, pie charts etc.
- › Relatively low-level; some effort needed to create advanced visualization

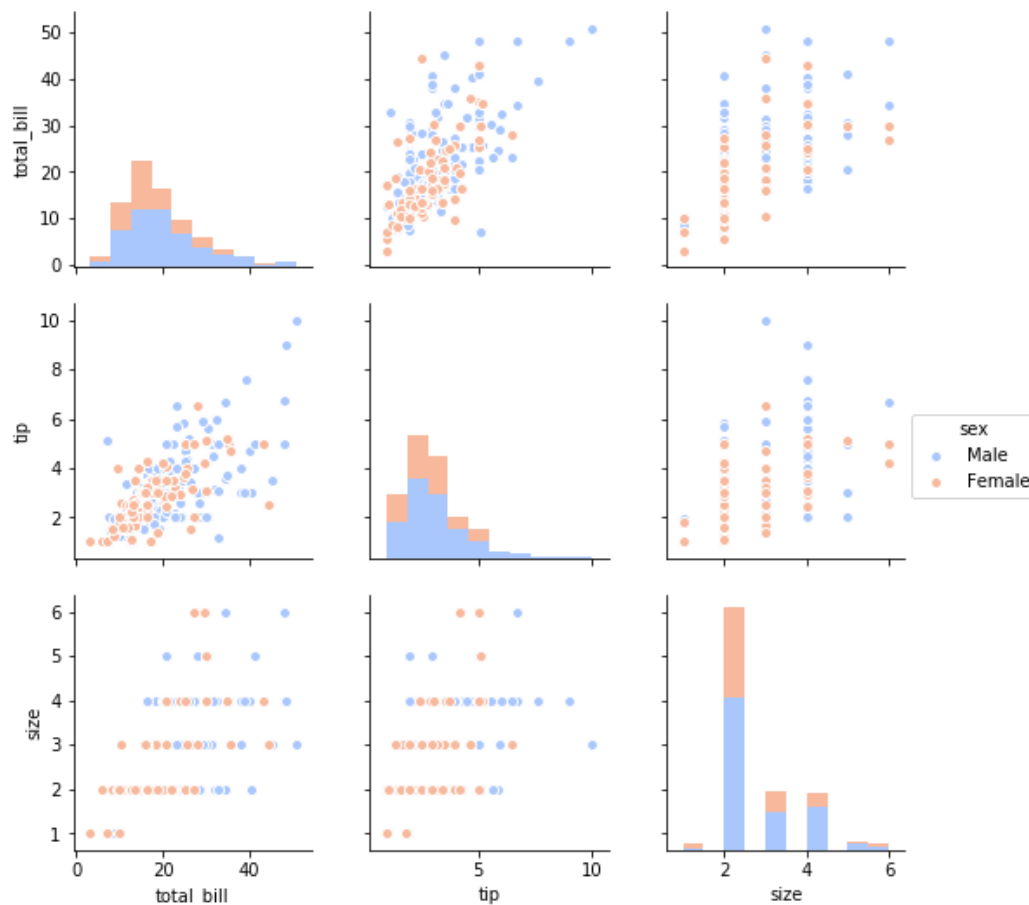


**Link:** <https://matplotlib.org/>



# Seaborn

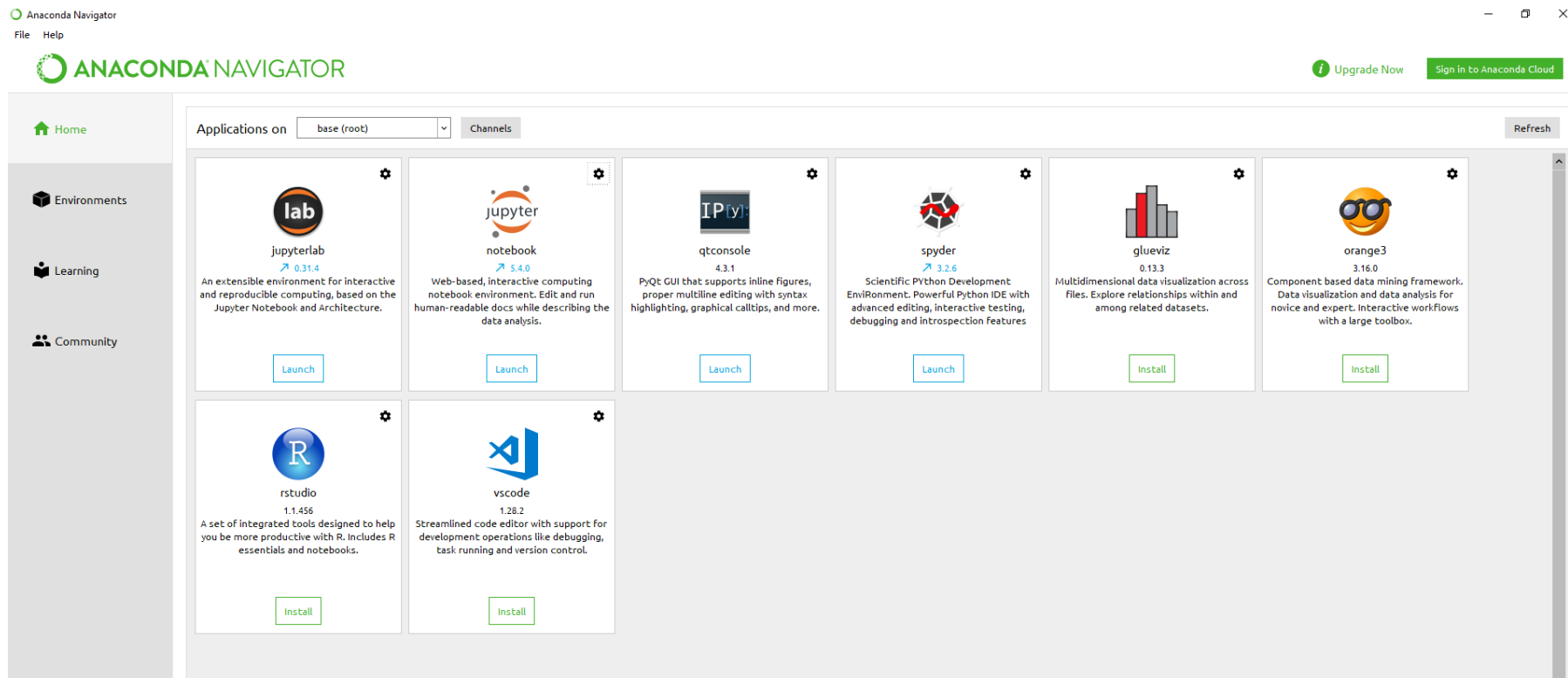
- › Based on Matplotlib
- › Provides high level interface for drawing attractive statistical graphics



**Link:** <https://seaborn.pydata.org/>

# Anaconda

## › Python Data Science Platform



To install Anaconda in Ubuntu: <https://www.digitalocean.com/community/tutorials/how-to-install-anaconda-on-ubuntu-18-04-quickstart>

# Jupyter Notebook

Jupyter Notebook is an open-source web application to create documents that contain:

- › Data cleaning and transformation
- › Numerical simulations
- › Statistical modeling
- › Data visualization
- › Machine Learning



**Link:** <https://jupyter.org/>

# How it looks like



Logout

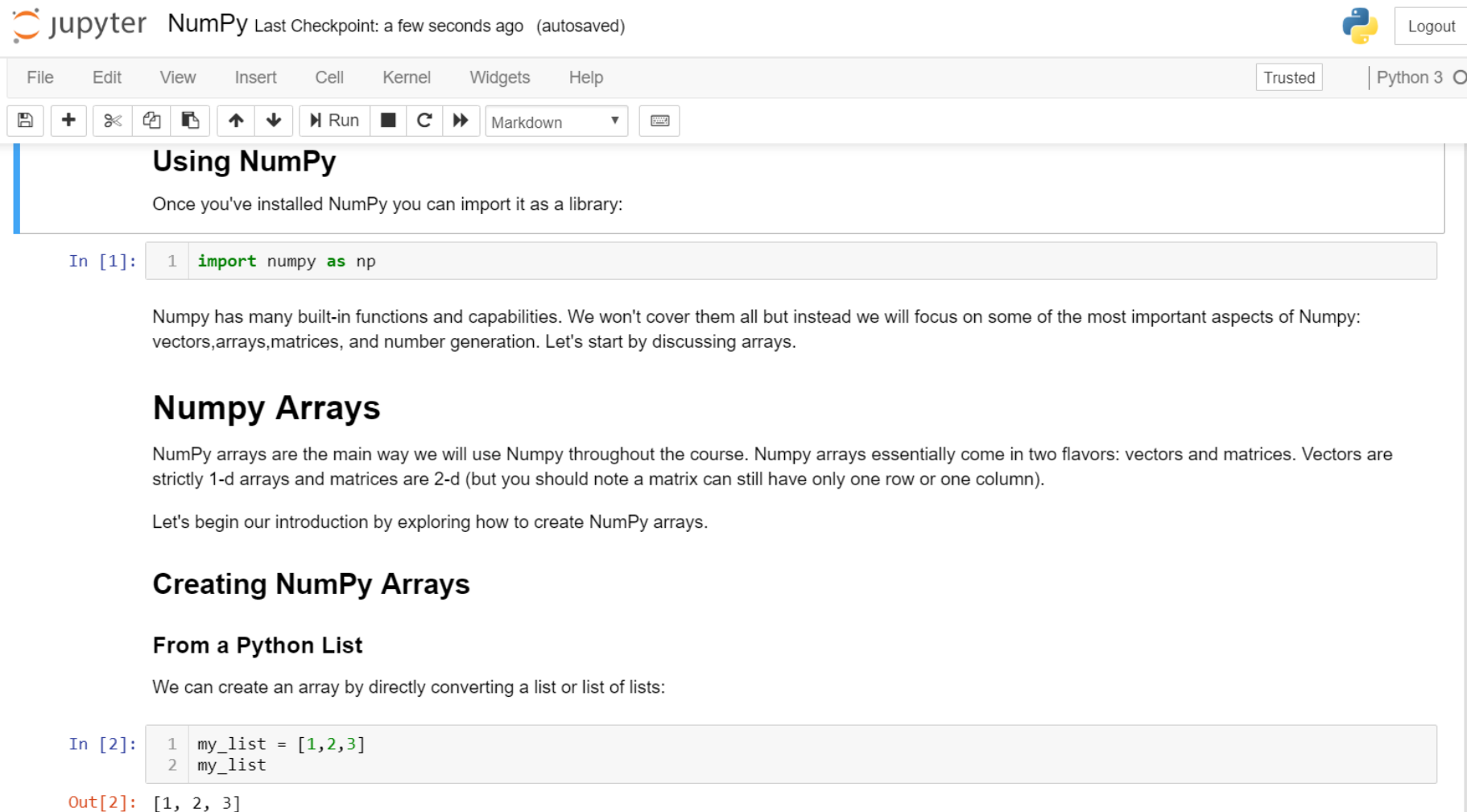
Files
Running
Clusters

Select items to perform actions on them.

Upload
New
Refresh

0 /		Name	Last Modified
<input type="checkbox"/>	Contacts		23 days ago
<input type="checkbox"/>	Desktop		an hour ago
<input type="checkbox"/>	Documents		30 minutes ago
<input type="checkbox"/>	Downloads		an hour ago
<input type="checkbox"/>	Favorites		23 days ago
<input type="checkbox"/>	jupyter_notebooks		3 months ago
<input type="checkbox"/>	Links		23 days ago
<input type="checkbox"/>	Music		23 days ago
<input type="checkbox"/>	No Backup		8 months ago
<input type="checkbox"/>	OneDrive		17 days ago
<input type="checkbox"/>	Outlook		4 months ago

# How it looks like



The screenshot shows a Jupyter Notebook interface. At the top, the Jupyter logo is followed by "NumPy" and "Last Checkpoint: a few seconds ago (autosaved)". On the right, there is a Python logo and a "Logout" button. Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu bar are "Trusted" and "Python 3" buttons. Below the menu bar is a toolbar with icons for saving, adding, undo, redo, running, and other notebook functions. The main content area has a title "Using NumPy" and a text block: "Once you've installed NumPy you can import it as a library:". Below this is a code cell with the input "In [1]: 1 import numpy as np". The output of this cell is a text block: "NumPy has many built-in functions and capabilities. We won't cover them all but instead we will focus on some of the most important aspects of Numpy: vectors, arrays, matrices, and number generation. Let's start by discussing arrays." Below this is another section header "Numpy Arrays" followed by a text block: "NumPy arrays are the main way we will use Numpy throughout the course. Numpy arrays essentially come in two flavors: vectors and matrices. Vectors are strictly 1-d arrays and matrices are 2-d (but you should note a matrix can still have only one row or one column)." Below this is another text block: "Let's begin our introduction by exploring how to create NumPy arrays." Below this is a section header "Creating NumPy Arrays" followed by a sub-section header "From a Python List". Below this is a text block: "We can create an array by directly converting a list or list of lists:". Below this is a code cell with the input "In [2]: 1 my\_list = [1,2,3] 2 my\_list". The output of this cell is "Out[2]: [1, 2, 3]".

jupyter NumPy Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

## Using NumPy

Once you've installed NumPy you can import it as a library:

```
In [1]: 1 import numpy as np
```

NumPy has many built-in functions and capabilities. We won't cover them all but instead we will focus on some of the most important aspects of Numpy: vectors, arrays, matrices, and number generation. Let's start by discussing arrays.

## Numpy Arrays

NumPy arrays are the main way we will use Numpy throughout the course. Numpy arrays essentially come in two flavors: vectors and matrices. Vectors are strictly 1-d arrays and matrices are 2-d (but you should note a matrix can still have only one row or one column).

Let's begin our introduction by exploring how to create NumPy arrays.

## Creating NumPy Arrays

### From a Python List

We can create an array by directly converting a list or list of lists:

```
In [2]: 1 my_list = [1,2,3]
        2 my_list
```

```
Out[2]: [1, 2, 3]
```



Part of your life. Part of tomorrow.

