

# The SciPy Stack

Data Analytics in Python

# Data Analytics/Scientific Computing

Gaining insight from data:

- ▶ Do instances fall into discernible groups?
  - ▶ Which characteristics differentiate groups?
- ▶ Do some characteristics of instances predict other characteristics?

Data are evidence. We seek predictive models and explanations.

# What is "data?"

First of all, data is the plural form of datum.

Data are measurements or assignments of values of attributes of instances of a class.

- ▶ Grades of students in a course. (Calculate grades for course.)
  - ▶ Grades of students in other courses. (Do grades from one course predict grades in another course?)
- ▶ DNA sequences. (Do parts of DNA predict diseases?)
- ▶ Pixel RGB intensities. (Do certain images contain faces? Which faces?)

Fundamental "linguistic" abstraction in data analytics/machine learning:  
**data are vectors of values.**

- ▶ Values can be numbers or categories.
- ▶ Multi-dimensional arrays can be "flattened" into 1-D vectors.

# The SciPy Stack

SciPy is a Python-based ecosystem of libraries and tools for scientific computing and data analytics (also the name of a particular python library).

- ▶ iPython
- ▶ Jupyter notebooks
- ▶ Numpy
- ▶ Pandas
- ▶ Matplotlib

iPython is the primary way of interacting with the SciPy stack – whether through the shell or a Jupyter notebook.

# iPython

Two modes:

- ▶ Interactive shell
  - ▶ Replacement for `python` REPL
- ▶ Jupyter notebook
  - ▶ Web-based documents mixing text, executable code, graphics

Before we proceed, make sure your computer is ready (OS shell):

```
$ conda update conda
$ conda update python ipython jupyter numpy pandas matplotlib
```

# A Taste of Data Analytics in iPython Shell

```
In [1]: cd analytics/  
/home/chris/vcs/github.com/datamastery/datamastery.github.io/code/analytics
```

```
In [3]: exam1grades = np.loadtxt('exam1grades.txt')
```

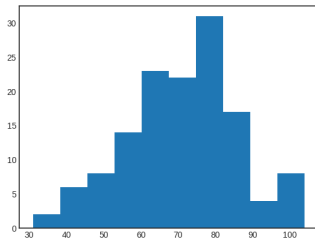
```
In [4]: import matplotlib.pyplot as plt
```

```
In [5]: %matplotlib qt5
```

```
In [6]: plt.hist(exam1grades)
```

```
Out[6]:
```

```
(array([ 2.,  6.,  8., 14., 23., 22., 31., 17.,  4.,  8.]),  
array([ 31. , 38.3, 45.6, 52.9, 60.2, 67.5, 74.8, 82.1,  
       89.4, 96.7, 104. ]),  
<a list of 10 Patch objects>)
```



# Jupyter Notebooks

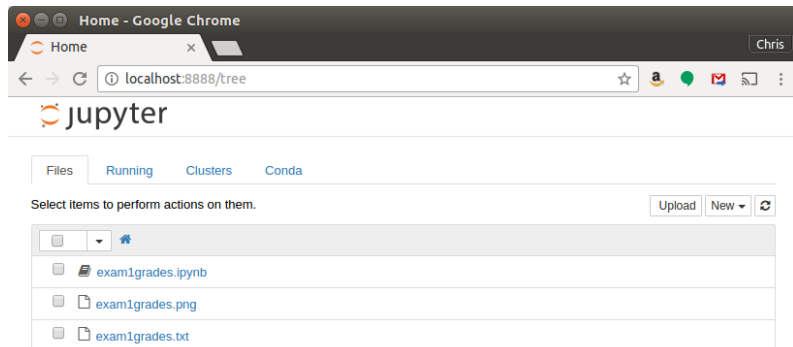
Go to the directory that holds your notebooks, or the class web site repo's code/analytics directory for this example and enter `jupyter notebook`.

```
[chris@bolshoi ~/vcs/github.com/datamastery/datamastery.github.io/code/analytics]
$ jupyter notebook
[I 15:06:15.705 NotebookApp] Serving notebooks from local directory:
    /home/chris/vcs/github.com/datamastery/datamastery.github.io/code/analytics
[I 15:06:15.705 NotebookApp] 0 active kernels
[I 15:06:15.705 NotebookApp] The Jupyter Notebook is running at:
    http://localhost:8888/
[I 15:06:15.705 NotebookApp] Use Control-C to stop this server and shut down all
    kernels (twice to skip confirmation).
Created new window in existing browser session.
```

Now a Jupyter Notebook server is running and you're ready to use iPython from the Jupyter Notebook web interface.

# Jupyter Web Interface

After running `jupyter notebook` from your OS command shell, open a browser and navigate to `localhost:8888`. You'll see a screen that looks like this:



Notice the listing of files in the directory in which you started the Jupyter notebook server.



# A Taste of Data Analytics in Jupyter Notebook

Select the `exam1grades.ipynb` file and you'll get this:

