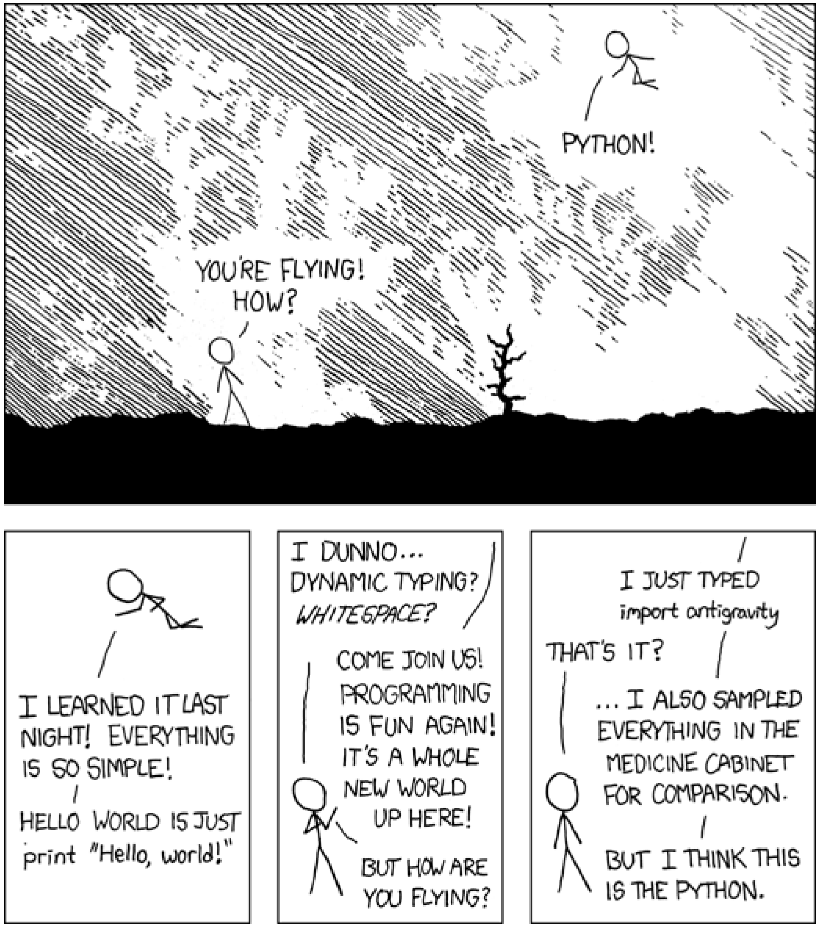
Introduction to Python & Jupyter Notebooks

Inf1cg Lab2

THE UNIVERSITY *of* EDINBURGH
informatics

Why Python?

- Great first programming language:
  - Easy to read and write.
  - Large community of users.
  - Loads of documentation, examples and support.
- Large number of implemented functionality (standard library), large collections of specialized add-on packages.
- Free, open source software.


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

http://xkcd.com/353/

Lab Structure

- Download the 'lab2' file to your home directory from the Learn page.
- Follow the instructions on setting up Python closely.
- Commands are written in `this font`
- The second part of the lab will be in notebook format.

Getting Python

- In this course we will use a virtual environment to run Python. More specifically, we will use the Conda http://conda.pydata.org/docs/ package management system.
- Using a virtual environment to run your project in offers many advantages:
  - Development is handled internally in your project without messing with your system-wide settings.
  - Platform independence.
  - Easy solution to access issues (no root access on DICE systems).
  - Dependency handling.
  - And many more...
- Here we will use the Python 3.5 version of Miniconda, a minimal distribution of Anaconda http://conda.pydata.org/miniconda.html
- We provide instructions to setup your Python environment on DICE, if you want to work on your own computers follow the instructions on installing Miniconda or Anaconda for Python 3.5 http://conda.pydata.org/miniconda.html https://www.continuum.io/downloads
- We encourage you to use the DICE setup!

Getting Python (DICE)

- **Open a terminal** (Applications > Terminal).
- To download 64-bit Python 3.5 Miniconda we download the install script provided with:

```
wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

- wget is a small shell program that downloads files for you.
- After downloading, **execute the script** run:

```
bash Miniconda3-latest-Linux-x86_64.sh
```

Getting Python (DICE)

- **Go through the software licence agreement.**
- **Accept the default installation location** in the root of your home directory:

  ```
  ~/miniconda3/
  ```

- Do not append the Miniconda binaries to your PATH system environment (DICE differs from normal bash-startup). **Respond no**, we will set up the path in a moment. (If you setup on your own non-DICE machine accepting is fine).

Installing Miniconda

- Append Miniconda to your path by:

```
echo "export PATH=\""\$PATH":$HOME/miniconda3/bin\"" >> ~/.benv
```

- Update the Path variable with:

```
source ~/.benv
```

- For all future terminal sessions you should now be able to access Miniconda by typing `conda` in your terminal

Installing Miniconda

- Try `conda --help` to check that everything works fine. You should see the help page for Conda.
- If instead you get a *No command 'conda' found* error you didn't set up your PATH variable correctly (ask for help!).

Virtual Environment

- We now **create a virtual environment** named *cogsci* using Python 3.5 for all our projects in this course:

  ```
  conda create -n cogsci python=3.5
  ```

- Accept the setup with *y.*
- We still need to a**ctivate the environment** with:
- `source activate cogsci` (on Windows just `activate cogsci`)
- When a virtual environment is active your terminal prompt reflects the name of the virtual environment, on DICE:

  (cogsci) [machine-name]:~$

Virtual Environment

- **Every time** you want to use the virtual environment you will need to run `source activate cogsci`.
- When the virtual environment is used the commands will access the **Python interpreter defined inside the environment** and not other (external) interpreters.
- If you want to deactivate the environment run `source deactivate` (on Windows just use `deactivate`).

Installing Packages

- Finally, we install all dependencies for the forthcoming labs with:

```
conda install numpy scipy matplotlib jupyter
```

- This installs the **numeric calculation** package NUMPY http://www.numpy.org/, the **scientific python** package SciPy https://www.scipy.org, the **plotting library** MatPlotLib http://matplotlib.org/ and the Jupyter **notebook tools** http://jupyter.org/
- **Confirm the installation** with *y* and after the installation and you are ready to go!

The Python Interpreter

- We can run python code from the shell by using:

```
python pythonProgram.py
```

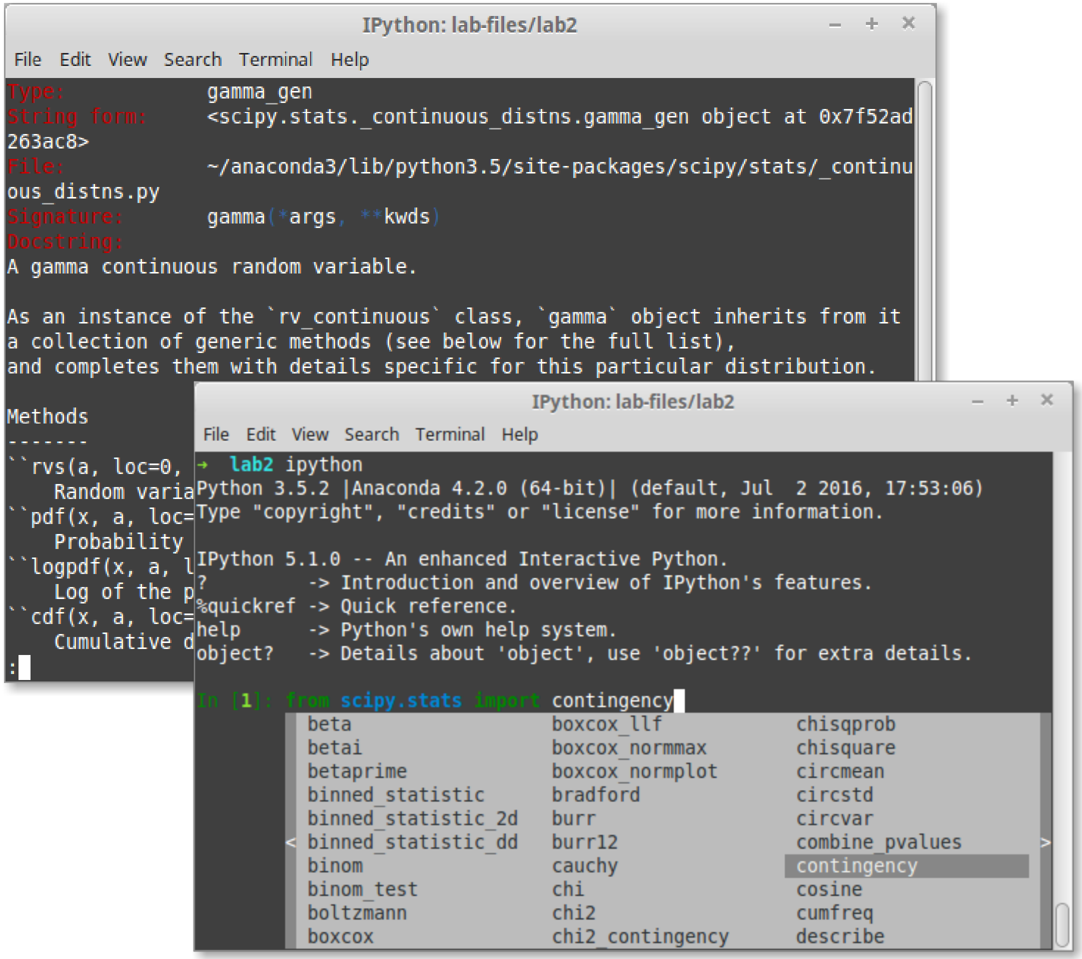
- The file format for Python programs is *.py*
- We can also use the interpreter interactively:
  - Start the interpreter by typing

```
python
```

  - Now you can type Python commands and execute them by hitting enter.
- Exercises:
  - Use the Python interpreter as a simple calculator.
- The standard interpreter is OK for quick calculation, but for more advanced use it has its limitations.

IPython

- Ipython is a much more user-friendly interactive shell and widely used in the Python community.
- It provides:
  - `Command history, which can be browsed with the up and down arrows on the keyboard.`
  - `Tab auto-completion.`
  - `In-line editing of code.`
  - `Documentation`
  - `Access to shell commands`
  - `And many more...`
- For more information, check: https://ipython.org/


IPython: lab-files/lab2
File Edit View Search Terminal Help
IPython: lab-files/lab2
File Edit View Search Terminal Help

```
Type:           gamma_gen
String form:    <scipy.stats._continuous_distns.gamma_gen object at 0x7f52ad
263ac8>
File:           ~/anaconda3/lib/python3.5/site-packages/scipy/stats/_continu
ous_distns.py
Signature:      gamma(*args, **kwds)
Docstring:
A gamma continuous random variable.

As an instance of the `rv_continuous` class, `gamma` object inherits from it
a collection of generic methods (see below for the full list),
and completes them with details specific for this particular distribution.

Methods
-------
``rvs(a, loc=0,
    Random varia
``pdf(x, a, loc=
    Probability
``logpdf(x, a, l
    Log of the p
``cdf(x, a, loc=
    Cumulative d
:
```

```
→  lab2 ipython
Python 3.5.2 |Anaconda 4.2.0 (64-bit)| (default, Jul  2 2016, 17:53:06)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: from scipy.stats import contingency
          beta                  boxcox_llf            chisqprob
          betai                 boxcox_normmax        chisquare
          betaprime             boxcox_normplot       circmean
          binned_statistic      bradford              circstd
          binned_statistic_2d   burr                  circvar
        < binned_statistic_dd   burr12                combine_pvalues    >
          binom                 cauchy                contingency
          binom_test            chi                   cosine
          boltzmann             chi2                  cumfreq
          boxcox                chi2_contingency      describe
```

Jupyter Notebooks

- In this course we will mainly use another way of writing Python code – Jupyter Notebooks.
- Jupyter notebooks are **HTML-based environments** and run in your browser.
- They are based on the IPython shell and provide all the functionality of the Ipython shell. In addition, they add great possibilities for mixing **text, Math, images, video and code**.
- They provide an environment to **document and illustrate your code** in a structured way.

Jupyter Notebooks

- We will continue the lab using the notebooks.
- Navigate into your *lab2* folder and run

```
jupyter notebook
```

- In the dashboard click on: `Introduction-to-Python`