MVPA of fMRI in Python

ICON 2017 workshop, Lukas Snoek & Steven Miletic,
University of Amsterdam

Who are we?

- Lukas Snoek
  - **PhD-student** at Steven Scholte's lab (University of Amsterdam)
  - Research on **emotion perception** and **neuroimaging** methods (machine learning/decoding)
  - Enjoy working on **open-source software** projects (mostly related to neuroimaging)
  - **Teaching neuroimaging** (fMRI) courses at the Research Master Psychology

Who are we?

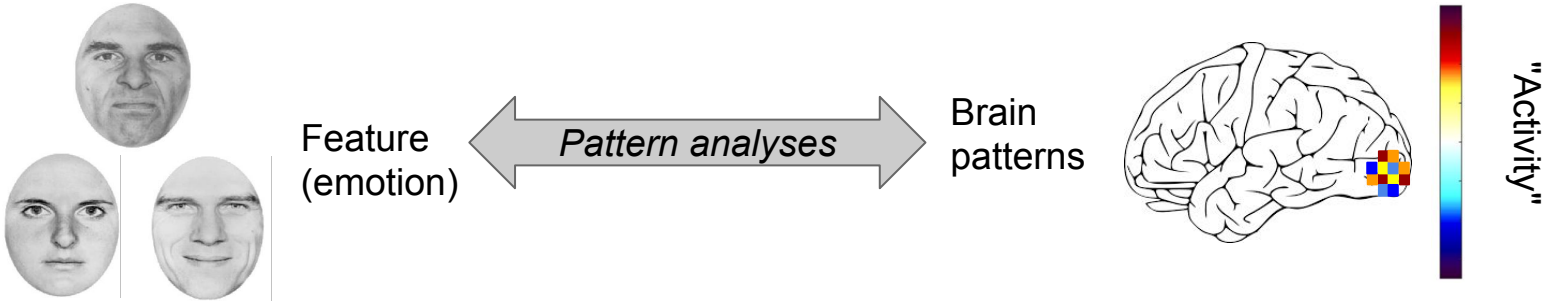
- Steven Miletic
  - Did a great internship on **multimodal decoding** (ensemble learning) at our lab!
  - **Soon-to-be-PhD-student** at Birte Forstmann's lab (University of Amsterdam)
  - Going to do research on **model-based cognitive neuroscience**
  - Currently working together on a project about **(dealing with) confounds in MVPA**

What's this workshop about?

- MVPA of fMRI data in Python

What's this workshop about?

- **MVPA** of **fMRI** data in **Python**
  - any analysis that relates **patterns** of voxels (or sensors in M/EEG) to features in the world (psychological processes, stimuli, traits, or behavior)



What's this workshop about?

- **MVPA** of **fMRI** data in **Python**
  - Specifically **machine learning** (no time for other pattern analyses like RSA ...)

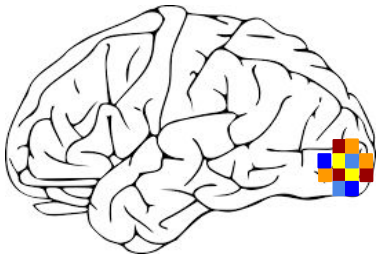
Pattern estimation

Cross-validation

y ← Machine learning algorithm X

Model evaluation

Feature selection



What's this workshop about?

- **MVPA** of **fMRI** data in **Python**
  - Specifically **machine learning** (no time for other pattern analyses like RSA ...)
  - Some parts specific to fMRI, but most material is directly applicable to **other modalities** (sMRI, EEG, MEG, etc.)

What's this workshop about?

- **MVPA** of **fMRI** data in **Python**
  - Specifically **machine learning** (no time for other pattern analyses like RSA ...)
  - Some parts specific to fMRI, but most material is directly applicable to **other modalities** (sMRI, EEG, MEG, etc.)
  - Using Python because of the awesome **scikit-learn** package

Format of this workshop

- Based on the "Neuroimaging: Pattern Analysis" course
- The tutorial is implemented in a jupyter notebook

This is a Markdown cell!

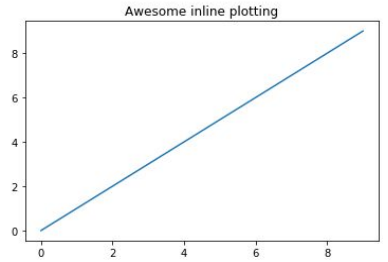
With some text.

In [1]:
```
print('But you can also write code')
x = 5
print('x squared = %i' % (x * x))
```

```
But you can also write code
x squared = 25
```

and plot stuff!

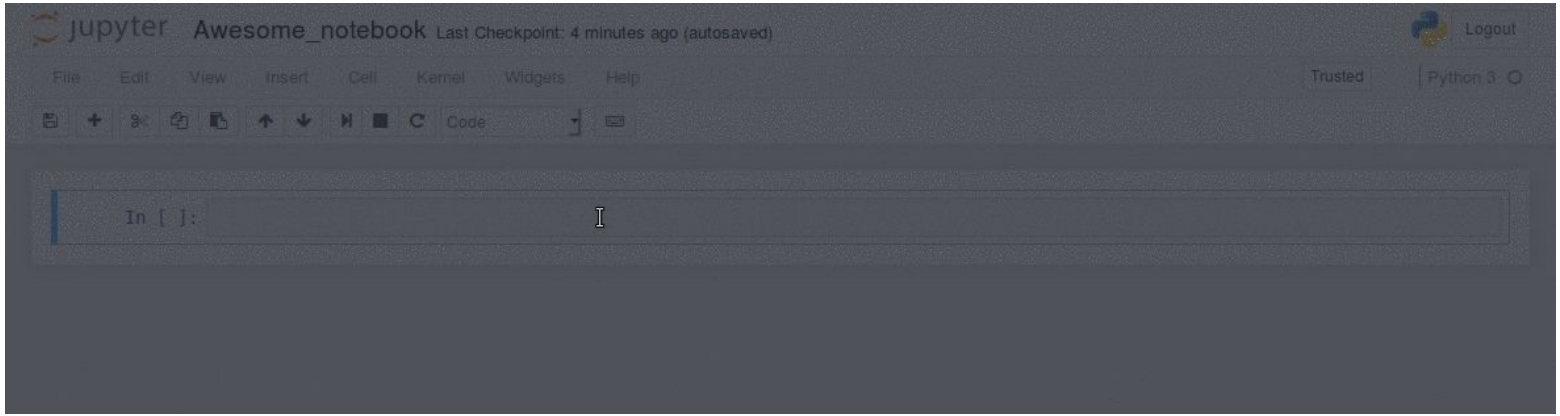
In [2]:
```
import matplotlib.pyplot as plt
%matplotlib inline
plt.title('Awesome inline plotting')
plt.plot(range(10))
plt.show()
```



Jupyter notebooks

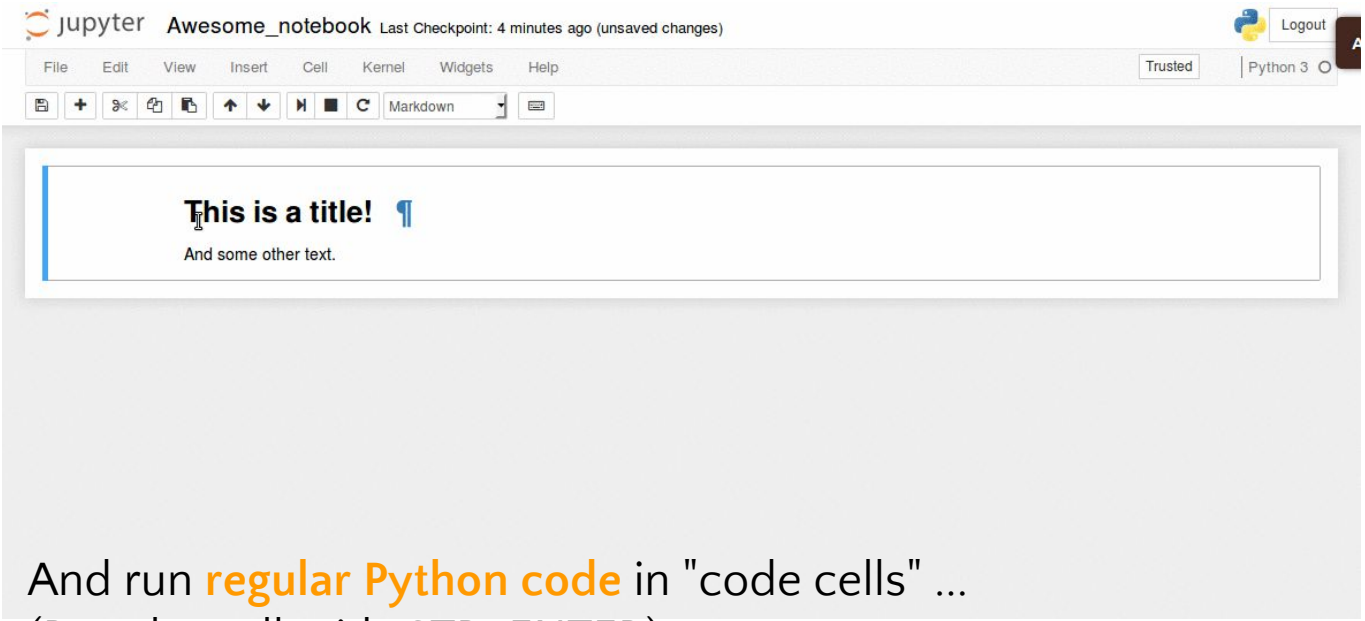
- Jupyter notebooks provide an **interactive Python environment** in which you can mix code, text, plots, and equations!

Jupyter notebooks



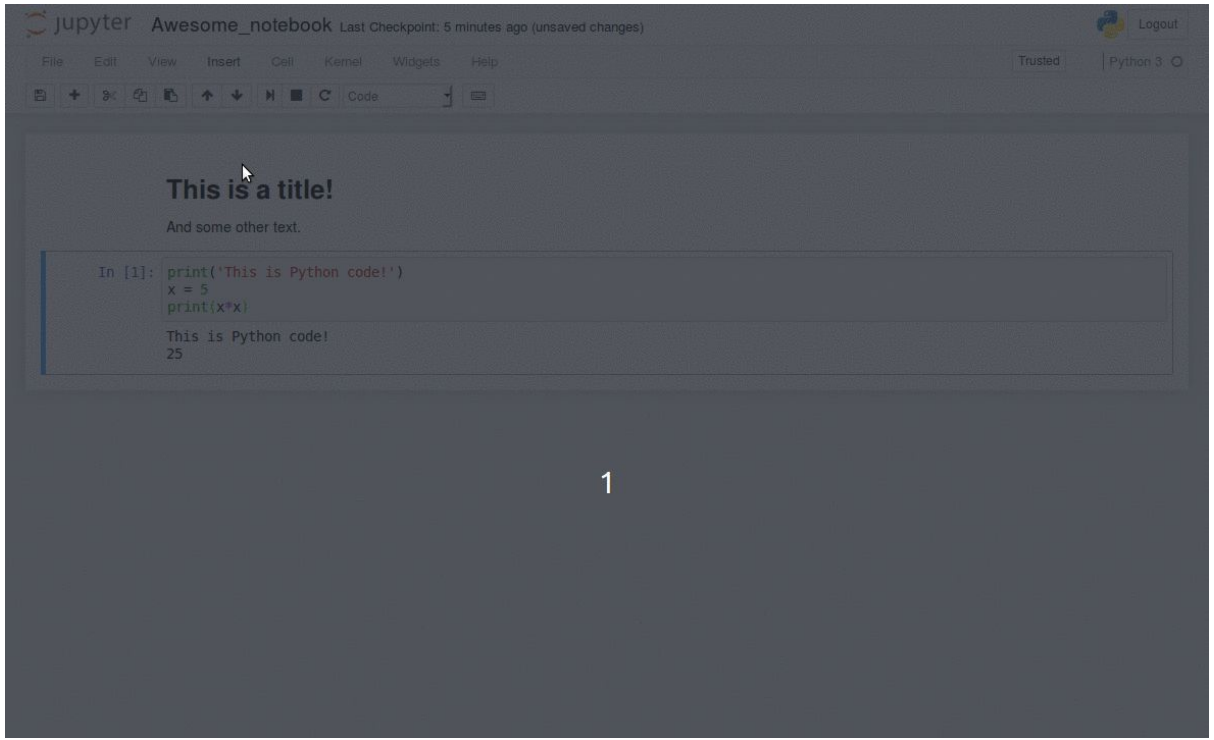
You can write **Markdown-formatted text** in "text cells" ...
(Render, or "run", the cell with CTR+ENTER)

Jupyter notebooks



And run regular Python code in "code cells" ...
(Run the cell with CTR+ENTER)

Jupyter notebooks



And create **inline plots**!

Jupyter notebook

1.4.2 tips & tricks to load and transform (nifti-)files

As a first thing, we need to find all the paths to the *t*-stat nifti-files. Python has a nifty (pun intended) tool called "`glob`" which can find files/directories on disk using wildcards. It is usually imported as follows:

In [ ]:
```
from glob import glob
```

`glob`, in Python, is a function that takes a path (as a string) with one or more wildcard characters (such as the `*`) and searches for files/directories on disk that match that. For example, let's try to find all the png-imagesin the "img" directory using glob (these are the images that I used inside this notebook).

In [ ]:
```
import os
# the images are in img/ on Linux/Mac systems, but in img\ on Windows (hence the "os.sep" thingie)
my_search_string = 'img' + os.sep + '*.png'
png_files = glob(my_search_string)
print(png_files)
```

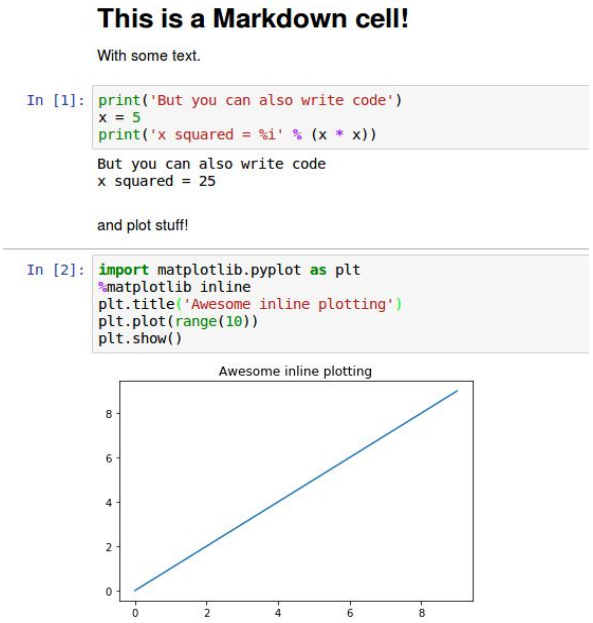
As you can see, it returns a list with all the files/directories that matched the search-string. Note that you can also search files outside of the current directory. To do so, we can simply specify the relative or absolute path to it.

> **ToDo**: Now you have the skills to actually "glob" all the *t*-stats from subject `pi0070` yourself! Use glob to find all the paths to the t-stats and store the results (a list with 40 strings) in a variable called `tstat_paths`. Note: the data directory is one directory above the current directory! Hint: watch out! There might be an `ftest.nii.gz` file in the stats-directory ...

In [ ]:
```
# Implement your ToDo here
```

Format of this workshop

- Based on the "Neuroimaging: Pattern Analysis" course
- The tutorial is implemented in a **jupyter notebook**
- The notebook walks you through all steps of MVPA
- With (optional) **programming exercises** and **review questions**
- Ask us (Lukas, Steven, Noor) questions!



Let's start!

`$ python download_data.py`

lukas-snoek.com/ICON2017

- Assuming you've downloaded the materials *and* data ...
- ... and you've set up your Python environment ...
- Start the notebook!

lukas-snoek.com/ICON2017/configure_python.html

  - `$ cd tutorial`
  - `$ jupyter notebook ICON2017_tutorial.ipynb`

- Need an account on our (pre-configured) server? Let us know!