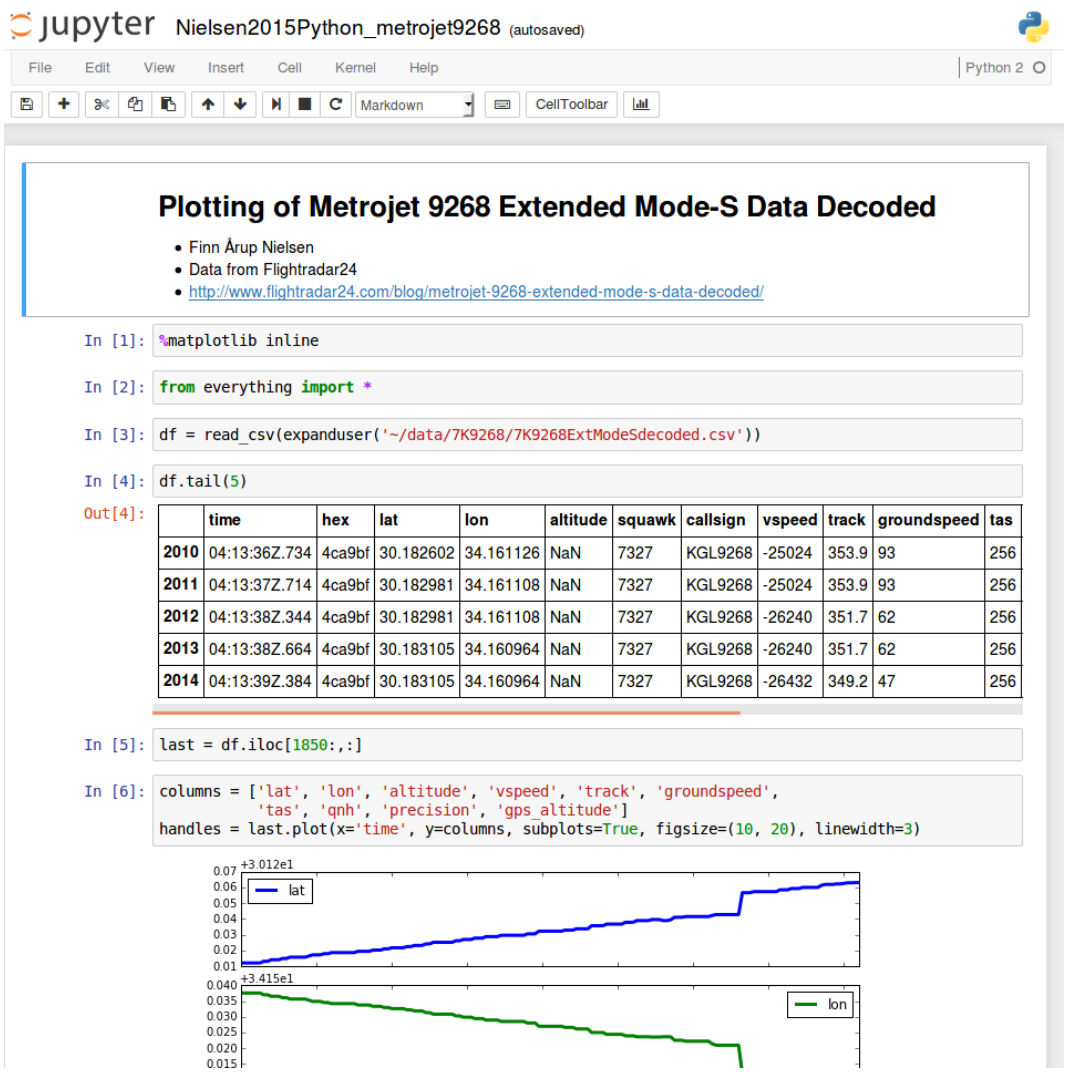
Jupyter (formerly IPython notebook)

Finn Årup Nielsen

DTU Compute
Technical University of Denmark

March 1, 2017

What is Jupyter?



	time	hex	lat	lon	altitude	squawk	callsign	vspeed	track	groundspeed	tas
2010	04:13:36Z.734	4ca9bf	30.182602	34.161126	NaN	7327	KGL9268	-25024	353.9	93	256
2011	04:13:37Z.714	4ca9bf	30.182981	34.161108	NaN	7327	KGL9268	-25024	353.9	93	256
2012	04:13:38Z.344	4ca9bf	30.182981	34.161108	NaN	7327	KGL9268	-26240	351.7	62	256
2013	04:13:38Z.664	4ca9bf	30.183105	34.160964	NaN	7327	KGL9268	-26240	351.7	62	256
2014	04:13:39Z.384	4ca9bf	30.183105	34.160964	NaN	7327	KGL9268	-26432	349.2	47	256

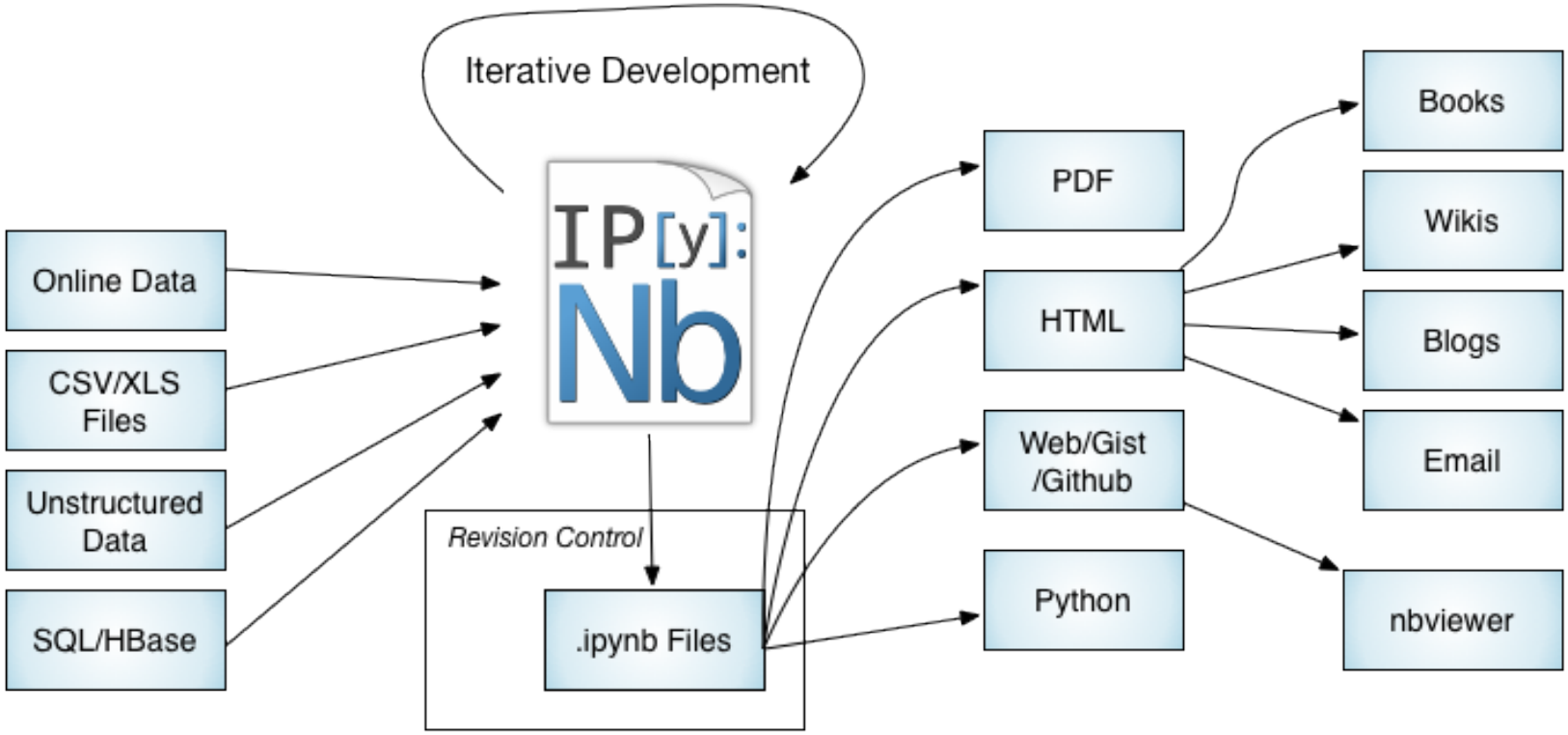
An interactive browser-based editing, computation and presentation environment

Runs as a Python program with interface through an ordinary webbrowser

Supports not only Python but several different computational languages.

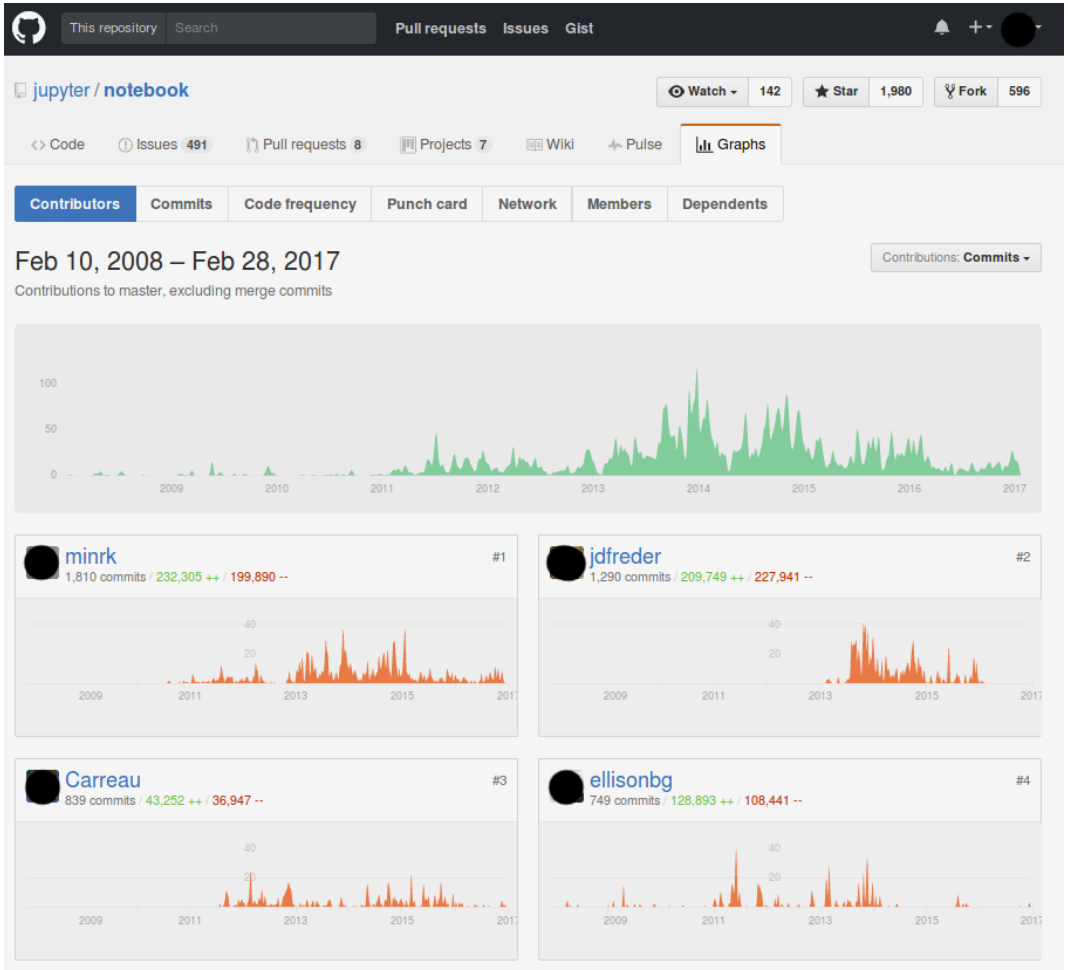
Like Mathematica and Maple but free/open software

The Jupyter workflow



By Joshua Barratt, IPynbWorkflows.png

Project Jupyter



Originally a spin-out of IPython, — an interactive shell (and from that formerly called IPython Notebook)

Homepage: https://jupyter.org/

Conference in 2017: jupytercon

Free/open development from https://github.com/jupyter

Lively development. Examples (that I do not know): nbgrader (grading), nbflow, cite2c (citations), Jupyter-drives (real-time collaboration) . . .

Installation

If you are on a Python-friendly platform:

```
pip install jupyter
```

i.e., Jupyter is available as a package from the Python package index, see https://pypi.python.org/pypi/jupyter/.

Otherwise, often recommended is to download and install Anaconda from the company *Continuum Analytics*.

After installation your should be ready to start the Jupyter from the command-line:

```
jupyter notebook
```

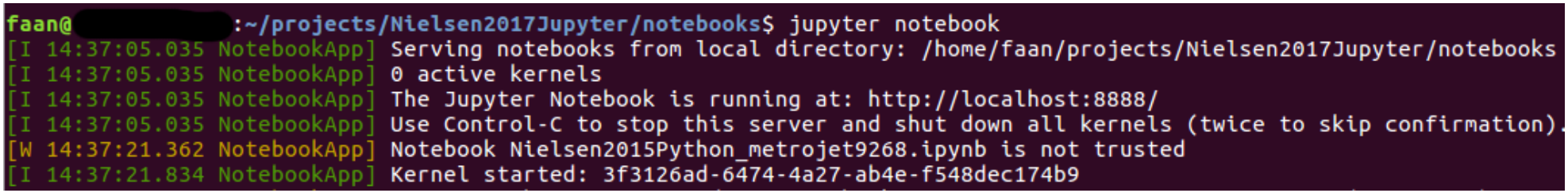
Starting Jupyter

Client-Server model: The client is the webbrowser that sends and receives from a (possibly remote) Python webserver to the kernel.

Start jupyter from the command line (in the working directory where you want to work from):

```
jupyter notebook
```

This will start the server and spawn a browser that is used for editing.



Do not close that server while you are running a notebook!

Text in Jupyter

Title in Markdown

In [1]:
```
# Here is some Python code which generates a result
1 + 1
```

Out[1]: 2

A subheading in markdown

And a bit of explanation together with some latex code

$$\frac{1}{\lambda} \approx 4$$

In [2]:
```
# In the above example the latex was written as $$\frac{1}{\lambda} \approx 4$$
```

Itemize in markdown

- You
- can
- do
- itemization

Tables

Column 1	Column 2	Third column
1	2	cell

Hyperlinks

Plain: http://dtu.dk or named links DTU's homepage.

Cell-oriented editing.

Cells with text type is formatted with *Markdown* markup language.

Markdown can do: headings, itemization, tables, hyperlinks, and a bit more, see John Gruber's Markdown site.

The markdown in Jupyter can handle LaTeX via "$$"

See also Simple notebook with various formatting

Code

Order of evaluation

```
In [3]: a = 1
        a
Out[3]: 1

In [4]: a += 1
        a
Out[4]: 2

In [5]: a -= 1
        a
Out[5]: 1
```

Default to Python.

Execute cells one at a time with shift+enter. This can be out of order.

Or run all cells in sequential order

Output is captured and printed under the cell.

Output is saved in the notebook file, the value of the variable is not save.

Kernels

You can choose different kernels that execute the code.

Not just for Python code: R, Julia, Matlab and many more, see list of Jupyter Kernels.

Kernels need to be installed separately.

An example with R code.

Data

In [1]:
```
n1 = 14 # All men
y1 = 9 # '5' men
n2 = 4 # All women
y2 = 2 # '5' women
```

Bayesian simulation

In [2]:
```
I = 100000 # simulations
hyperprior = 1
males = rbeta(I, y1+hyperprior, (n1-y1)+hyperprior)
females = rbeta(I, y2+hyperprior, (n2-y2)+hyperprior)
diff = males-females  # simulated diffs
```

Output

In [3]:
```
# OUTPUT
quantiles = quantile(diff,c(0.005,0.025,0.5,0.975,0.995))
print(quantiles,digits=2)

print("Probability higher % men than women '5':")
print(mean(males>females))
```

```
 0.5%  2.5%   50% 97.5% 99.5%
-0.41 -0.30  0.13  0.55  0.65
[1] "Probability higher % men than women '5':"
[1] 0.70275
```

Plot output with Python kernel

Plot output in the form of image can appear in different ways.

So-called magic commands can control the output. Default is in-line plots

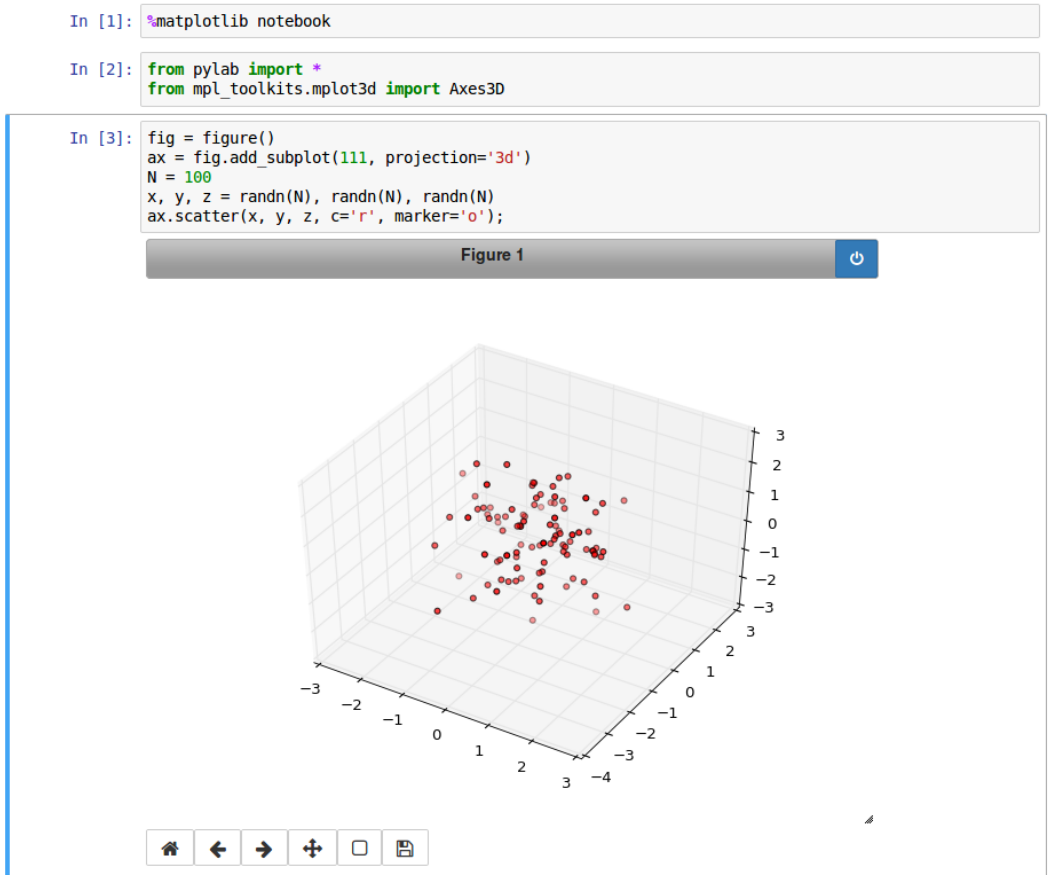
```
%matplotlib inline
```

The inline images are stored in the notebook in encoded binary format.

Examples of other display modes:

```
%matplotlib notebook
```

```
%matplotlib gtk
```



Control of IPython (Python kernel)

For the Python kernel there are a number of different magic commands beyond `matplotlib`

`!`	Execute command in the shell, e.g., `!ls`
`%%time`	Time the cell
`%cd`	Change working directory
`%magic`	Information about the magic system
`%%prun`	Profiling the code

...

File format: The Notebook document

Part of Jupyter Notebook is its *open data format*.

Plain modern *lingua franca* JSON (`.json`) with specific fields.

The file can easily be read by Javascript, Python, etc.

There are different versions, so old Jupyter versions might not be able to read new versions.

Read more here: https://nbformat.readthedocs.io.

File format: Example

```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 1,
   "metadata": {
    "collapsed": true
   },
   "outputs": [],
   "source": [
    "from __future__ import division\n",
    "\n",
    "from afinn import Afinn\n",
    "import numpy as np\n",
...
```

. . . and your can read it

Reading a public notebook with a small Python program:

```
import requests

url = ("https://raw.githubusercontent.com/fnielsen/afinn/"
       "master/notebooks/"
       "CrowdFlower%20Twitter%20sentiment%20analysis.ipynb")
notebook = requests.get(url).json()
print("␣".join(notebook['cells'][0]['source']))
```

Gives the output (the first cell of the notebook):

```
from __future__ import division

from afinn import Afinn
import numpy as np
import pandas as pd
```

Conversion with `nbconvert`

Convert to latex, html, pdf, . . . , see https://nbconvert.readthedocs.io

Extract the python code with "`--to python`"

You can write custom templates in the jinja language for output you specific way, see Customizing nbconvert.

Converting the notebook to LaTeX and further on to PDF:

```
jupyter nbconvert example.ipynb --to pdf
```

Nielsen2015Python_metrojet9268

February 28, 2017

1 Plotting of Metrojet 9268 Extended Mode-S Data Decoded

- Finn Årup Nielsen
- Data from Flightradar24
- http://www.flightradar24.com/blog/metrojet-9268-extended-mode-s-data-decoded/

```
In [1]: %matplotlib inline

In [2]: from everything import *

In [3]: df = read_csv(expanduser('~/data/7K9268/7K9268ExtModeSdecoded.csv'))

In [4]: df.tail(5)
```

Out[4]:

	time	hex	lat	lon	altitude	squawk	calls
2010	04:13:36Z.734	4ca9bf	30.182602	34.161126	NaN	7327	KGL9
2011	04:13:37Z.714	4ca9bf	30.182981	34.161108	NaN	7327	KGL9
2012	04:13:38Z.344	4ca9bf	30.182981	34.161108	NaN	7327	KGL9
2013	04:13:38Z.664	4ca9bf	30.183105	34.160964	NaN	7327	KGL9
2014	04:13:39Z.384	4ca9bf	30.183105	34.160964	NaN	7327	KGL9

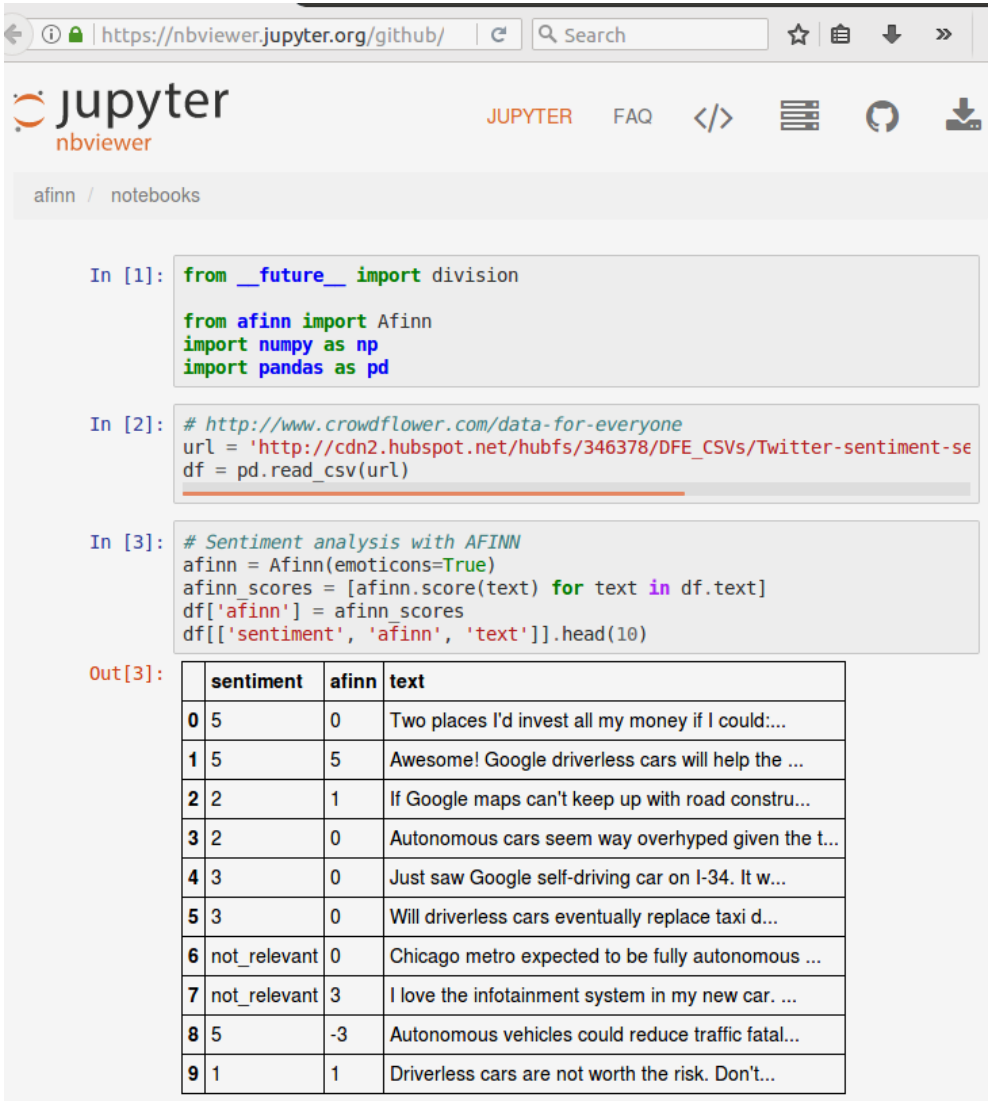
	vspeed	track	groundspeed	tas	qnh	mcp_alt	precision	gps_alti
2010	-25024	353.9	93	256	1012	32000	0	2
2011	-25024	353.9	93	256	1012	32000	0	2
2012	-26240	351.7	62	256	1012	32000	0	2
2013	-26240	351.7	62	256	1012	32000	0	2
2014	-26432	349.2	47	256	1012	32000	0	2

	oat
2010	-42
2011	-42
2012	-42
2013	-42
2014	-42

```
In [5]: last = df.iloc[1850:,:]
```

1

. . . and you can view it



	sentiment	afinn	text
0	5	0	Two places I'd invest all my money if I could:...
1	5	5	Awesome! Google driverless cars will help the ...
2	2	1	If Google maps can't keep up with road constru...
3	2	0	Autonomous cars seem way overhyped given the t...
4	3	0	Just saw Google self-driving car on I-34. It w...
5	3	0	Will driverless cars eventually replace taxi d...
6	not_relevant	0	Chicago metro expected to be fully autonomous ...
7	not_relevant	3	I love the infotainment system in my new car. ...
8	5	-3	Autonomous vehicles could reduce traffic fatal...
9	1	1	Driverless cars are not worth the risk. Don't...

You can "publish" notebooks

nbviewer is a webservice to view the Notebook files

Canonical viewer for public notebooks: https://nbviewer.jupyter.org/

For instance, a notebook on GitHub can be viewed with nbviewer with https://nbviewer . . . icons.ipynb

With some effort you can also setup nbviewer on your local computer/intranet.

Embedding content in a Python notebook

You can embed other content than text and code in a notebook.

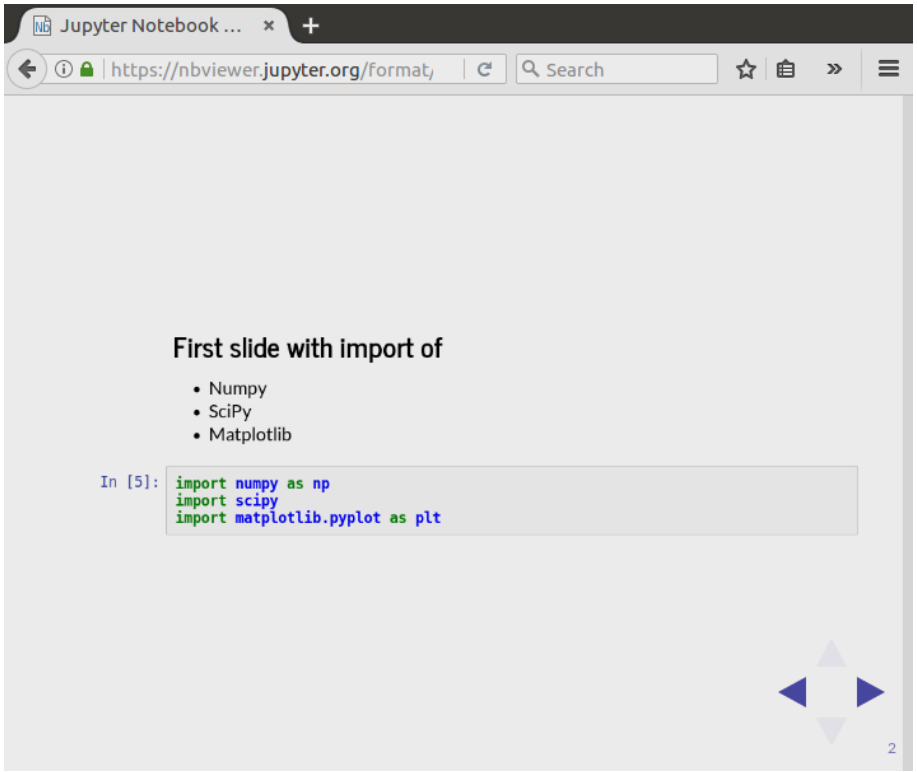
In a Jupyter notebook runing Python there is a range of embedding classes you can import from the `IPython.display` module, e.g., to include HTML, image files, YouTube videos and (content from) iframes.

```
from IPython.display import (HTML, Image,
                             YouTubeVideo, IFrame)
```

Note that viewers night not be able (or want to) to display the embedded content, see, e.g., Nielsen2017Jupyter_embeddings.ipynb on Github where GitHub does not display the YouTube video or the iframe.

Show with nbviewer!

Slideshows



Notebook can be styled to slideshows.

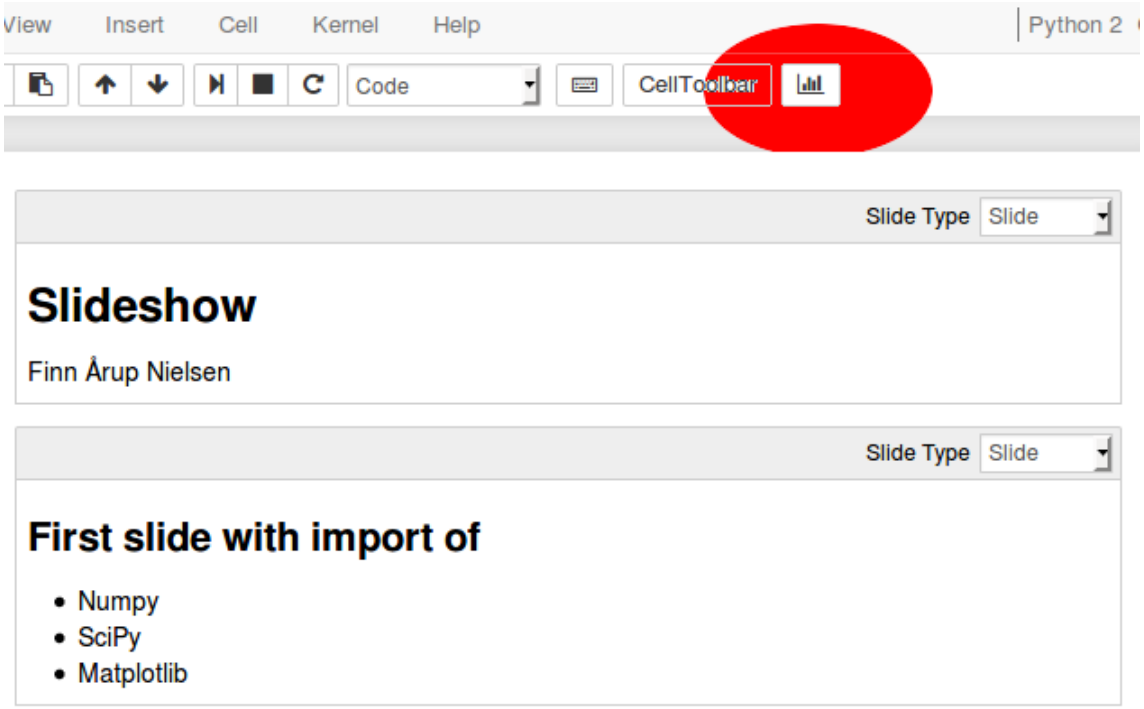
In an open notebook go to View – Cell Toolbar – Slideshow and mark cells with "slide".

View a public slide notebook with nbviewer: Example notebook as slide show

Or view the slide show, e.g., with

```
jupyter nbconvert example.ipynb --to slides --post serve
```

Live slideshows



The RISE notebook extension should be install for this functionality.

Install rise (`pip install rise`)

Install and enable it as a jupyter-nbextension

Start jupyter notebook as usual

. . . and you have an extra button "Enter/Exit Live Reveal Slideshow"

See more at https://github.com/damianavila/RISE.

"A bound notebook"?

If you want to establish priority:

"The industrial researcher, whose work may lead to patents, has no choice accept to use a bound notebook for all laboratory notemaking." (Kanare, 1985)*

Research should be documented in a manner consistent with practices in the field of research in question, e.g. by keeping records, logbooks, journals or similar practices – if possible with dates and entries by the person(s) responsible for the conduct of the research. (Q28, 2014)

*Perhaps a dated claim? (Just get first to the patent office)

"A bound notebook"?

If you want to establish priority:

"The industrial researcher, whose work may lead to patents, has no choice accept to use a bound notebook for all laboratory notemaking." (Kanare, 1985)†

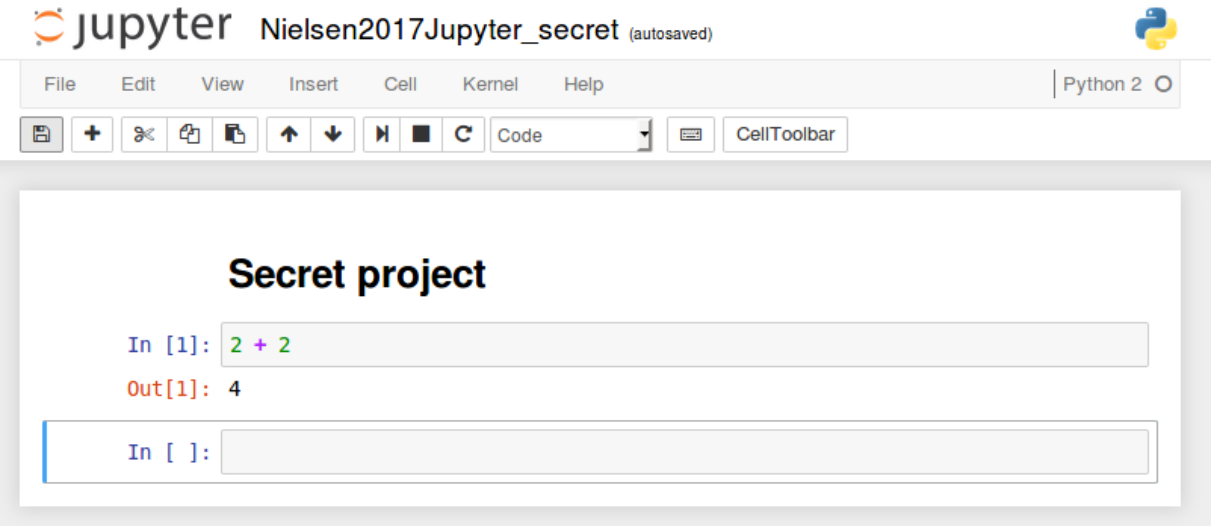
Research should be documented in a manner consistent with practices in the field of research in question, e.g. by keeping records, logbooks, journals or similar practices – if possible with dates and entries by the person(s) responsible for the conduct of the research. (Q28, 2014)

Jupyter Notebook has no sense of proven persistency: Old Jupyter Notebooks can be modified, so with Jupyter Notebook you cannot prove that the file was the original.

†Perhaps a dated claim? (Just get first to the patent office)

“A bound notebook”?

Consider a file:



Checksum the notebook file:

```
$ sha512sum Nielsen2017Jupyter_secret.ipynb
86d355942b95a36d723d94d50e78c3d2a3642c06dbff4c23704...
```

And publish the checksum. But the author and the date is still dangling.

Git — distributed revision control system

Jupyter Notebooks can be used with git (or other revision control system).

Just checkin the notebook file as a normal file (after creation of the repository):

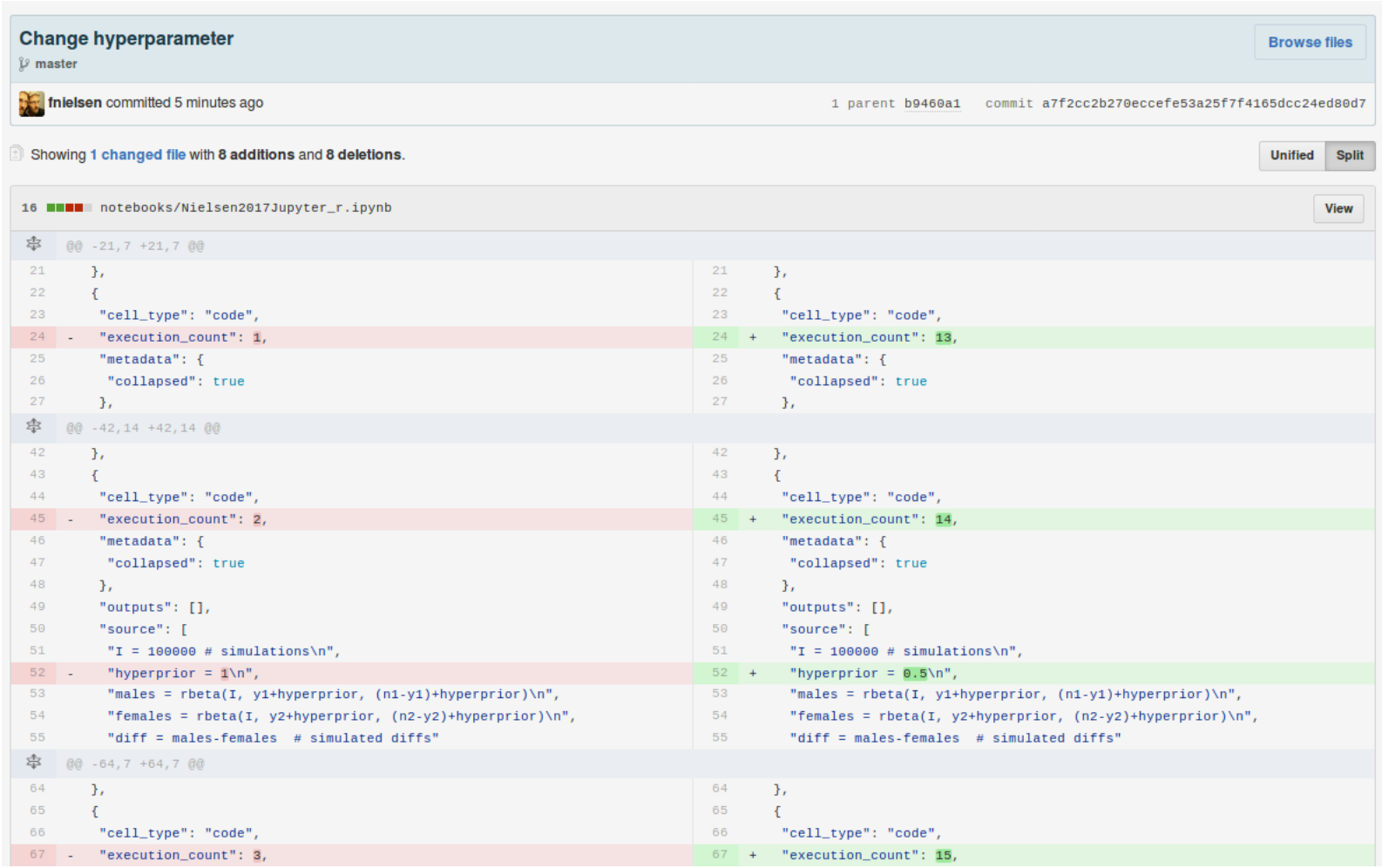
```
git add Nielsen2017Jupyter_simple.ipynb
git commit Nielsen2017Jupyter_simple.ipynb
git push
```

Standard Jupyter notebook has no time stamps and no author. Git has that for you.

The cloud-based git provider *GitHub* is able to format the notebook file for you.

Git diff

A Git diff on GitHub for an R notebook change.



Change hyperparameter

master

fnielsen committed 5 minutes ago — 1 parent b9460a1 commit a7f2cc2b270eccefe53a25f7f4165dcc24ed80d7

Browse files

Showing 1 changed file with 8 additions and 8 deletions. — Unified | Split

16 notebooks/Nielsen2017Jupyter_r.ipynb — View

```
@@ -21,7 +21,7 @@
   },
   {
    "cell_type": "code",
-   "execution_count": 1,
+   "execution_count": 13,
    "metadata": {
     "collapsed": true
    },
@@ -42,14 +42,14 @@
   },
   {
    "cell_type": "code",
-   "execution_count": 2,
+   "execution_count": 14,
    "metadata": {
     "collapsed": true
    },
    "outputs": [],
    "source": [
     "I = 100000 # simulations\n",
-    "hyperprior = 1\n",
+    "hyperprior = 0.5\n",
     "males = rbeta(I, y1+hyperprior, (n1-y1)+hyperprior)\n",
     "females = rbeta(I, y2+hyperprior, (n2-y2)+hyperprior)\n",
     "diff = males-females  # simulated diffs"
@@ -64,7 +64,7 @@
   },
   {
    "cell_type": "code",
-   "execution_count": 3,
+   "execution_count": 15,
```

Cloud-based?

Cloud-based notebooks have been shaky.

wakari.io ran notebooks for you. It was bought by Continuum Analytics and now amalcated into *Anaconda Enterprise Notebooks* ("Starting at $60.000" per year).

http://mybinder.org ? Mentioned in (Kluyver et al., 2016).

https://cloud.sagemath.com/. Starts at $79 per year (a small free notebook is available).

https://try.jupyter.org/ — only to try out.

Kaggle Kernels — for Kaggle datasets.

Why you shouldn't use Jupyter

Somewhat unusual editing behavior with the browser-based interface, — but you can probably get use to it (learn the keyboard shortcuts!).

Notebooks tend to be stand-alone, where code reuse is more difficult.

The notebook locks computation. You cannot to other computations while it is running. But you can still edit or start another notebook.

Less software developer-oriented: unit testing, lint.

Not really good for storing and editing data (input data, intermediate computation, or output) for you. This is not a spreadsheet.

Why you should use Jupyter

"Researchers today across all academic disciplines often need to write computer code in order to collect and process data, carry out statistical tests, run simulations or draw figures. The widely applicable libraries and tools for this are often developed as open source projects (such as NumPy, Julia, or FEniCS), but the specific code researchers write for a particular piece of work is often left unpublished, hindering reproducibility."

"Notebooks—documents integrating prose, code and results—offer a way to publish a computational method which can be readily read and replicated." (Kluyver et al., 2016)

Notebook examples

Sune Lehmann, Associate Professor, DTU Compute. Use Jupyter Notebook for assignment in his class, e.g., Social data analysis 2016 Assignment 1.

Peter Norvig, Google, http://www.norvig.com/ipython/. Didactic notebooks with explanations, Python code and latex equations, e.g., xkcd 1313: Regex Golf

Cameron Davidson-Pilon, Probabilistic Programming and Bayesian Methods for Hackers. This is an entire book written in Jupyter.

A few of my demonstrations, fnielsen/Nielsen2017Jupyter/ GitHub repo.

Thanks

References

(2014). *Danish Code of Conduct for Research Integrity*. Ministry of Higher Education and Science, Copenhagen, Denmark.

Kanare, H. M. (1985). *Writing the Laboratory Notebook*. American Chemical Society.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., and Team, J. D. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. pages 87–90. DOI: 10.3233/978-1-61499-649-1-87.