Jupyterhub-introduction-HPC-Cafe

October 13, 2020

1 A short JupyterHub introduction

2 Why use Jupyter Notebooks?

- Flexible
- Webbase -> easy access
- interactive code development
- good dokumentation

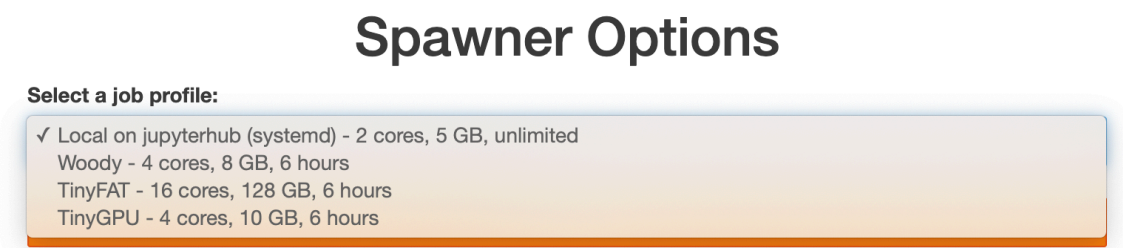
You can find the service at jupyterhub.rrze.uni-erlangen.de

For login use your HPC account information

Jupyter is a web-based interactive computational environment.

Not just for python!

There are ~40 kernels for various programming languages.



2.1 Notebooks

- Cell types
  - Marksdown (this)
  - Code (below)
- Execute / Render with Shift+Return

[1]:
```
# use any python code
print("Hallo HPC group")
```

```
Hallo HPC group
```

2.2 Code Cells

- (usually) running on IPython kernel
- Number in [brakets] gives execution order

[2]:
```
import math
foo, bar = 23, 42
math.pow(foo, bar)
```

[2]:
```
1.55800595299714e+57
```

[3]:
```
print(foo)
```

```
23
```

- silent execution with trailing semicolon (;)

[4]:
```
foo = -1
math.pow(foo, bar);
```

- Tab for completion
- Shift+Tab for help (twice for more help)

[ ]:
```
ma  #try here
```

2.3 IPython Magic

Prefixes: * !: execute shell commands with

[5]:
```
!cat /proc/cpuinfo | grep 'model name' | uniq
```

```
model name      : AMD Opteron(tm) Processor 6176 SE
```

also supports python variables in shell commands:

[6]:
```
path = "/etc/hostname"
!cat {path}
```

```
jupyterhub
```

- %: magic functions

[7]:
```
%lsmagic
```

[7]:
```
Available line magics:
%alias  %alias_magic  %autoawait  %autocall  %automagic  %autosave  %bookmark
%cat  %cd  %clear  %colors  %conda  %config  %connect_info  %cp  %debug  %dhist
%dirs  %doctest_mode  %ed  %edit  %env  %gui  %hist  %history  %killbgscripts
%ldir  %less  %lf  %lk  %ll  %load  %load_ext  %loadpy  %logoff  %logon
```

```
%logstart  %logstate  %logstop  %ls  %lsmagic  %lx  %macro  %magic  %man
%matplotlib  %mkdir  %more  %mv  %notebook  %page  %pastebin  %pdb  %pdef  %pdoc
%pfile  %pinfo  %pinfo2  %pip  %popd  %pprint  %precision  %prun  %psearch
%psource  %pushd  %pwd  %pycat  %pylab  %qtconsole  %quickref  %recall  %rehashx
%reload_ext  %rep  %rerun  %reset  %reset_selective  %rm  %rmdir  %run  %save
%sc  %set_env  %store  %sx  %system  %tb  %time  %timeit  %unalias  %unload_ext
%who  %who_ls  %whos  %xdel  %xmode

Available cell magics:
%%!  %%HTML  %%SVG  %%bash  %%capture  %%debug  %%file  %%html  %%javascript
%%js  %%latex  %%markdown  %%perl  %%prun  %%pypy  %%python  %%python2
%%python3  %%ruby  %%script  %%sh  %%svg  %%sx  %%system  %%time  %%timeit
%%writefile

Automagic is ON, % prefix IS NOT needed for line magics.
```

[8]:
```
%%latex
$\sum \frac{23}{42}$
```

$\sum \frac{23}{42}$

[9]:
```
import math
%timeit math.pow(3, 42)
```

```
398 ns ± 3.08 ns per loop (mean ± std. dev. of 7 runs, 1000000 loops each)
```

3 Magic function for package management

[10]:
```
%pip install --user --proxy http://proxy:80 minimal-lagrangians
```

```
Collecting minimal-lagrangians
  Using cached minimal_lagrangians-1.1.2-py3-none-any.whl (54 kB)
Installing collected packages: minimal-lagrangians
Successfully installed minimal-lagrangians-1.1.2
Note: you may need to restart the kernel to use updated packages.
```

[11]:
```
%pip uninstall -y minimal-lagrangians
```

```
Found existing installation: minimal-lagrangians 1.1.2
Uninstalling minimal-lagrangians-1.1.2:
  Successfully uninstalled minimal-lagrangians-1.1.2
Note: you may need to restart the kernel to use updated packages.
```

3.1 Quick Dip into NumPy, SymPy, Pandas and Matplotlib

Most important python packages for scientists: * NumPy: Scientific computing * SymPy: Symbolic mathematics * Matplotlib: Visualizations * Pandas: Data analysis

3.2 NumPy: Scientific Computing

[12]:
```
import numpy as np
```

[ ]:
```
np.lookfor('create array')
```

[ ]:
```
help(np.array)
```

[13]:
```
A = np.array([[1, 4, 3, 8], [3, 9, 5, 6], [8, 7, 1, 4], [5, 7, 1, 1]])
A
```

[13]:
```
array([[1, 4, 3, 8],
       [3, 9, 5, 6],
       [8, 7, 1, 4],
       [5, 7, 1, 1]])
```

[14]:
```
A.shape
```

[14]:
```
(4, 4)
```

[15]:
```
A*A
```

[15]:
```
array([[ 1, 16,  9, 64],
       [ 9, 81, 25, 36],
       [64, 49,  1, 16],
       [25, 49,  1,  1]])
```

Using NumPy is almost always faster:

[16]:
```
%%timeit
xvals = range(1000000)
[xval**2 for xval in xvals]
```

```
539 ms ± 2.23 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

[17]:
```
%%timeit
a = np.arange(1000000)
a**2
```

```
6.03 ms ± 271 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

3.3 SymPy: Symbolic Mathematics

[18]:
```
import sympy as sp
```

[19]:
```
x,y = sp.var('x,y')
```

[20]:
```
x**y / 45
```

[20]: $\frac{x^{y}}{45}$

[21]:
```
expr = (x + y)**5
sp.expand(expr)
```

[21]: $x^{5} + 5x^{4}y + 10x^{3}y^{2} + 10x^{2}y^{3} + 5xy^{4} + y^{5}$

[22]:
```
sp.solve(expr)
```

[22]:
```
[{x: -y}]
```

[23]:
```
f = sp.lambdify((x,y), expr)
f(1, -1)
```

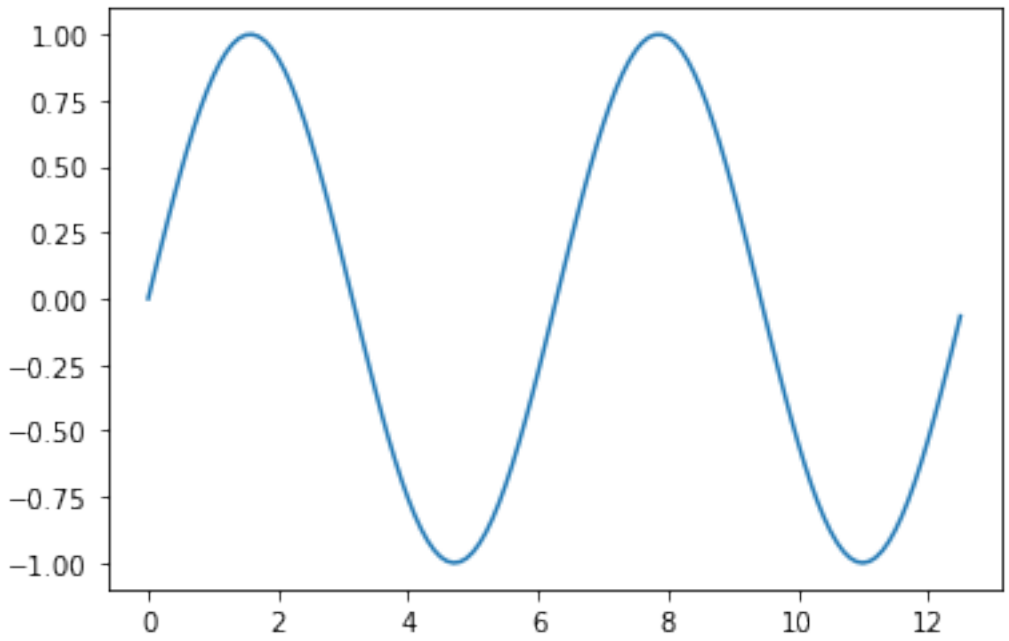
[23]:
```
0
```

3.4 Matplotlib: Visualizations

[24]:
```
from matplotlib import pyplot as plt
```

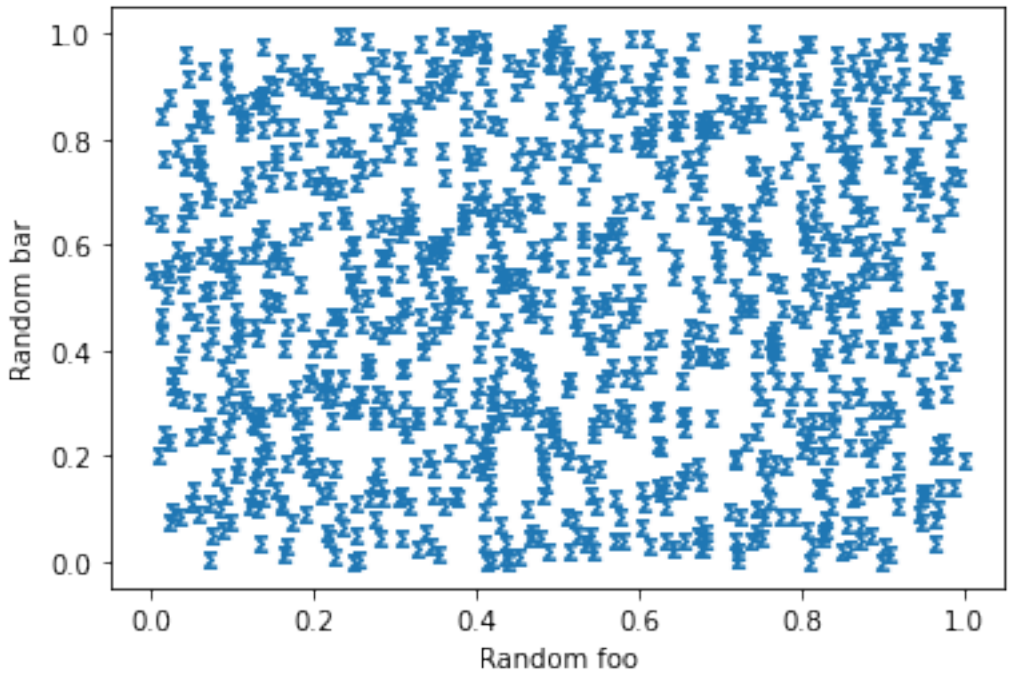
[25]:
```
from math import pi
x = np.arange(0, 4*pi, 0.1)
plt.plot(x, np.sin(x))
```

[25]:
```
[<matplotlib.lines.Line2D at 0x7ff284679490>]
```



[26]:
```
plt.scatter(np.random.rand(1000), np.random.rand(1000), marker='$\sum$')
plt.xlabel('Random foo')
plt.ylabel('Random bar')
```

[26]:
```
Text(0, 0.5, 'Random bar')
```



https://matplotlib.org/

Tipp: use stateful pyplot functions only for quick&dirty plotting

and use object-orietend interface pyplot.figure and pyplot.subplots for serious work

3.5 Pandas: Data Analysis

[27]:
```
import pandas as pd
```

[28]:
```
# Some magic for HTTP GET to work
import os; os.environ['https_proxy'] = 'http://proxy:80'
```

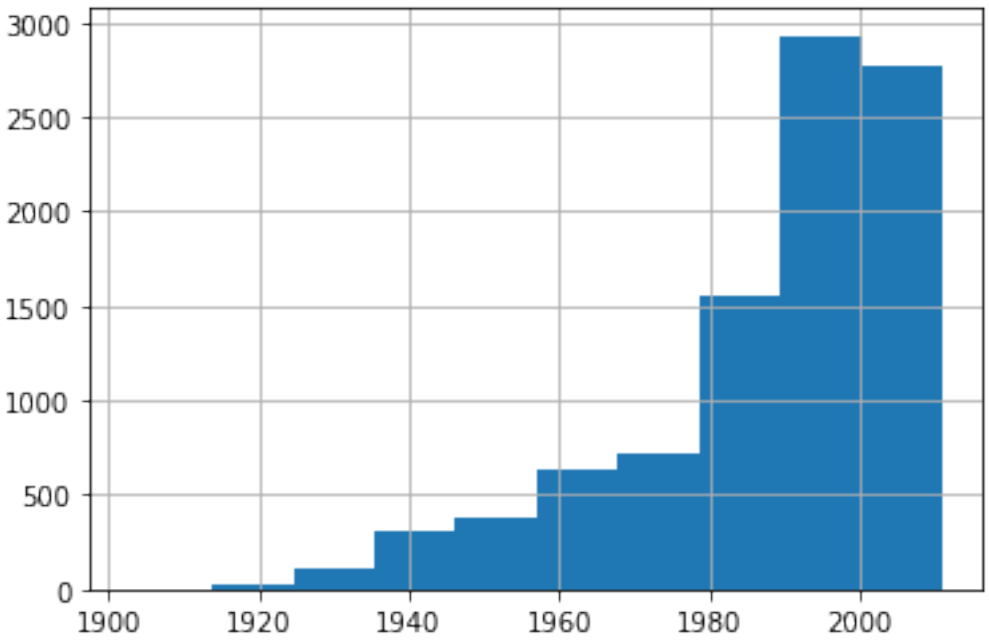
[29]:
```
df = pd.read_csv('https://raw.githubusercontent.com/cs109/2014_data/master/
↪movies.dat', sep='\t')
df.head()
```

[29]:
```
   id                         title  imdbID  \
0   1                     Toy story  114709
1   2                       Jumanji  113497
2   3                Grumpy Old Men  107050
3   4             Waiting to Exhale  114885
4   5  Father of the Bride Part II  113041

                                    spanishTitle  \
0                      Toy story (juguetes)
1                                   Jumanji
2                       Dos viejos gruñones
3                      Esperando un respiro
4  Vuelve el padre de la novia (Ahora también abu…

                                    imdbPictureURL  year  \
0  http://ia.media-imdb.com/images/M/MV5BMTMwNDU0…  1995
1  http://ia.media-imdb.com/images/M/MV5BMzM5NjE1…  1995
2  http://ia.media-imdb.com/images/M/MV5BMTI5MTgy…  1993
3  http://ia.media-imdb.com/images/M/MV5BMTczMTMy…  1995
4  http://ia.media-imdb.com/images/M/MV5BMTg1NDc2…  1995

                        rtID rtAllCriticsRating rtAllCriticsNumReviews  \
0                  toy_story                 9                      73
1              1068044-jumanji               5.6                      28
2               grumpy_old_men               5.9                      36
3            waiting_to_exhale               5.6                      25
4  father_of_the_bride_part_ii               5.3                      19
```

```
  rtAllCriticsNumFresh  … rtAllCriticsScore rtTopCriticsRating  \
0                   73  …               100                8.5
1                   13  …                46                5.8
2                   24  …                66                  7
3                   14  …                56                5.5
4                    9  …                47                5.4

  rtTopCriticsNumReviews rtTopCriticsNumFresh rtTopCriticsNumRotten  \
0                     17                   17                    0
1                      5                    2                    3
2                      6                    5                    1
3                     11                    5                    6
4                      5                    1                    4

  rtTopCriticsScore rtAudienceRating rtAudienceNumRatings rtAudienceScore  \
0               100              3.7               102338              81
1                40              3.2                44587              61
2                83              3.2                10489              66
3                45              3.3                 5666              79
4                20                3                13761              64

                                        rtPictureURL
0  http://content7.flixster.com/movie/10/93/63/10…
1  http://content8.flixster.com/movie/56/79/73/56…
2  http://content6.flixster.com/movie/25/60/25602…
3  http://content9.flixster.com/movie/10/94/17/10…
4  http://content8.flixster.com/movie/25/54/25542…

[5 rows x 21 columns]
```

[30]:
```
df.year.hist()
```

[30]:
```
<AxesSubplot:>
```



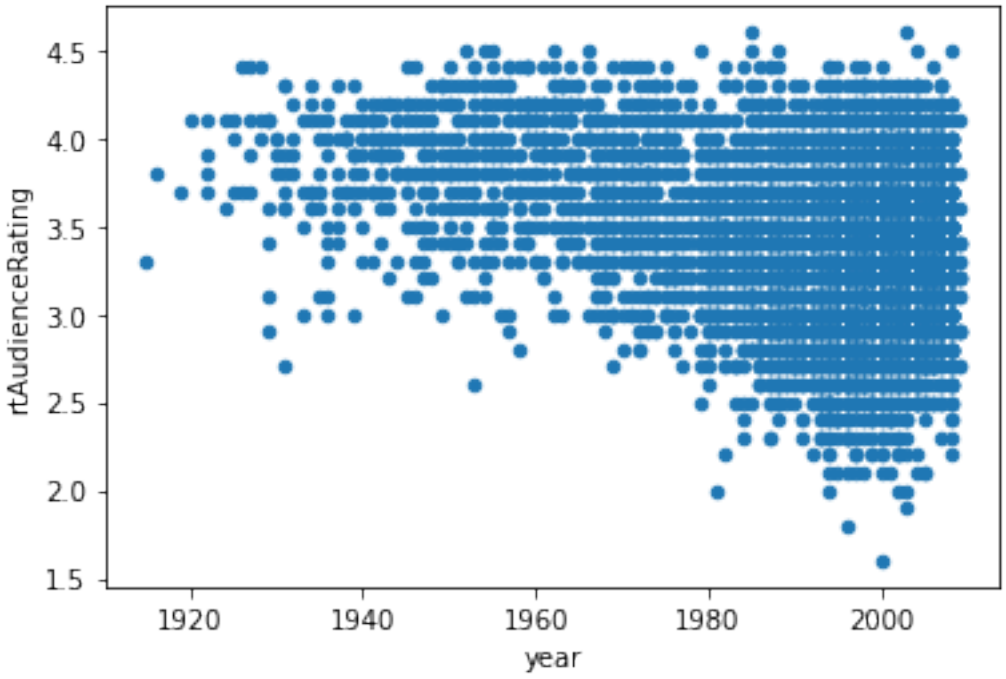
[31]:
```
df['rtAudienceNumRatings'] = pd.to_numeric(df['rtAudienceNumRatings'],␣
↪errors='coerce')
df['rtAudienceRating'] = pd.to_numeric(df['rtAudienceRating'], errors='coerce')
df_rated = df.query('rtAudienceNumRatings > 100')
```

[32]:
```
df_rated['rtAudienceRating'].describe()
```

[32]:
```
count    6594.000000
mean        3.389930
std         0.453881
min         1.600000
25%         3.100000
50%         3.400000
75%         3.700000
max         4.600000
Name: rtAudienceRating, dtype: float64
```

[33]:
```
df_rated.plot(x='year', y='rtAudienceRating', kind='scatter')
```

[33]:
```
<AxesSubplot:xlabel='year', ylabel='rtAudienceRating'>
```



4 Learn more about python

PythonDataScienceHandbook

WhirlwindTourOfPython

iPython