Reproducible Research with Jupyter

April 15, 2015

1 Working with the notebook

Lorem ipsum dolor sit amet, consectetur adipisici elit, sed eiusmod tempor incidunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodi consequat. Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

In [1]:
```
4 + 5
```

Out[1]: 9

In [2]:
```
x = 4
y = 3
```

In [3]:
```
print('The sum of {} and {} is {}.'.format(x, y, x + y))
```

```
The sum of 4 and 3 is 7.
```

In [4]:
```
def sum(x, y):
    return x + y
```

In [5]:
```
sum(3, 2)
```

Out[5]: 5

2 Example: Explorative analysis of grid frequency time series

In [6]:
```
PURPLE = (107.0/255, 36.0/255, 124.0/255)
FREQUENCY_COLOR = (16.0/255, 192.0/255, 225.0/255)

FREQUENCY_FILE = 'path-to-file/file'
```

In [7]:
```
import pandas as pd
%matplotlib inline
import seaborn as sns
```

In [8]:
```
freq = pd.read_hdf(FREQUENCY_FILE, '/frequency')
print('Index of time series: {} to {}.'.format(freq.index[0], freq.index[-1]))
print('Time resolution of time series: {}.'.format(freq.index.freq))
print('Size of time series: {:.2f} MB.'.format(freq.nbytes / 1024 / 1024))
```

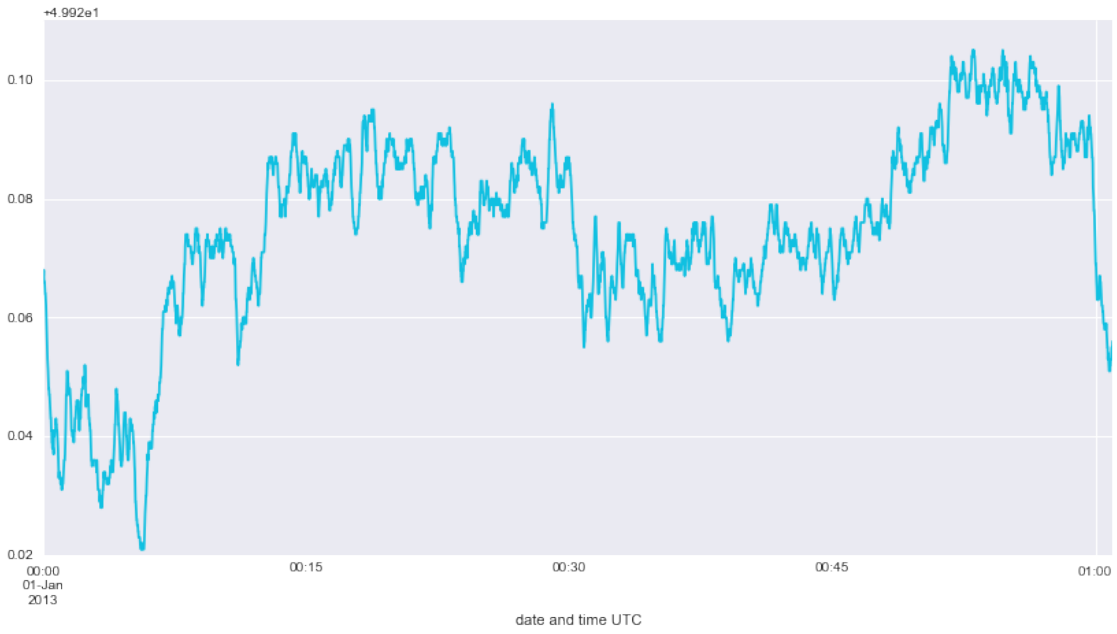
```
Index of time series: 2012-01-01 00:00:00+00:00 to 2013-12-31 23:59:59+00:00.
Time resolution of time series: <Second>.
Size of time series: 481.86 MB.
```

```
In [9]: freq.describe()
```

```
Out[9]: count    63158400.000000
        mean           49.999966
        std             0.021858
        min            49.849000
        25%            49.986000
        50%            50.000000
        75%            50.014000
        max            50.145000
        Name: grid frequency in Hz, dtype: float64
```

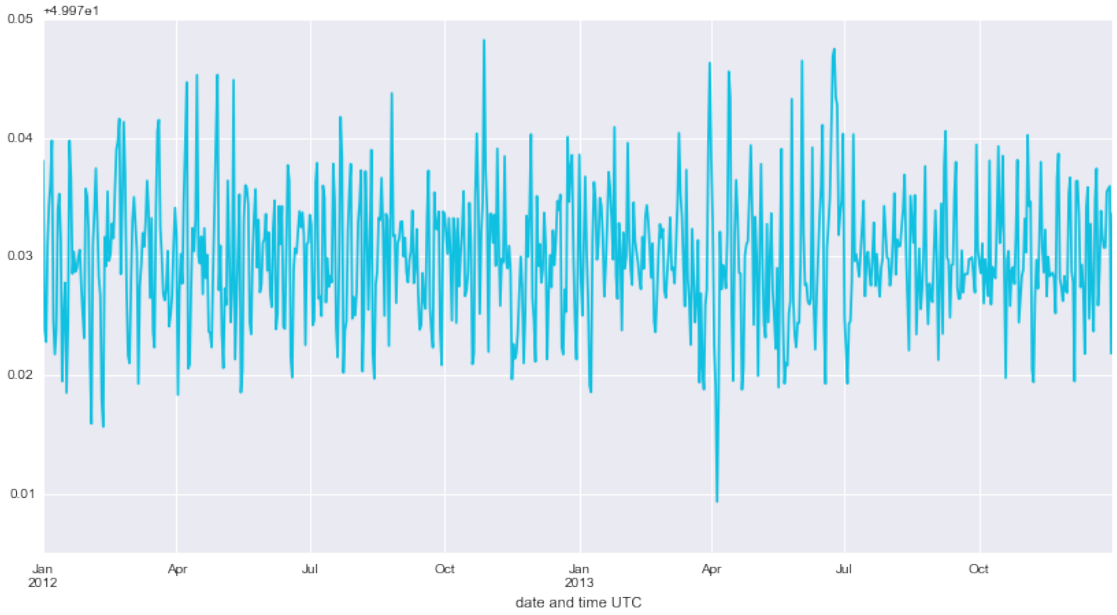
```
In [10]: freq['2013-01-01':'2013-01-01 01:00'].plot(figsize=(14,7), color=FREQUENCY_COLOR)
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x8e62c88>
```



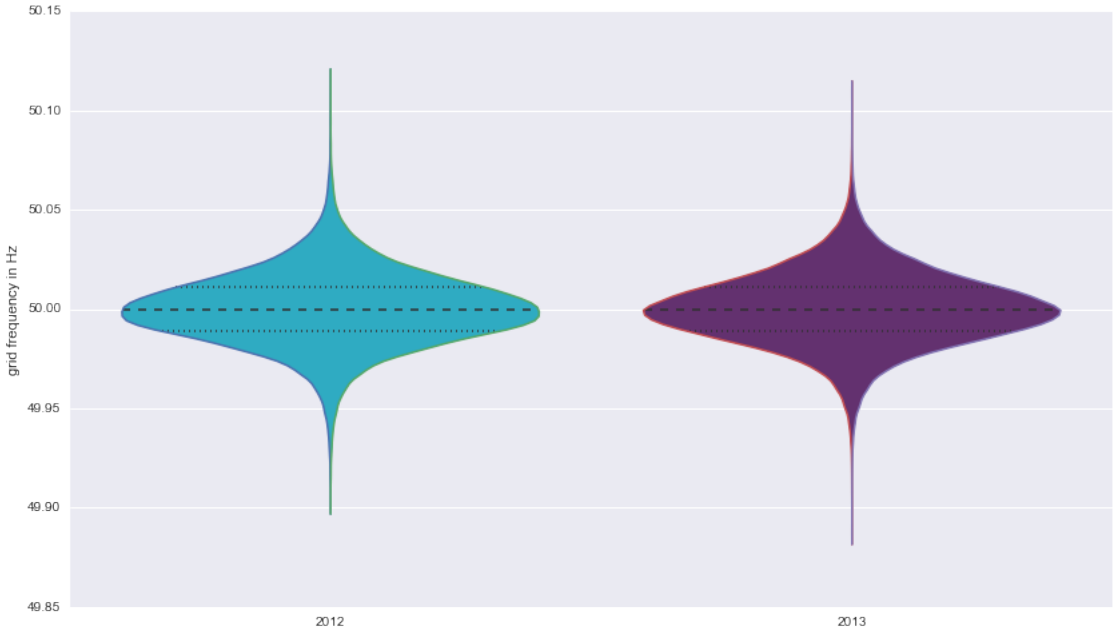

```
In [11]: freq.resample('D').plot(figsize=(14,7), color=FREQUENCY_COLOR)
```

```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x8eaf3c8>
```




```
In [12]: sns.set(rc={"figure.figsize": (14, 8)})
         sns.violinplot(freq.resample('10Min'), groupby=lambda x : x.year, color=[FREQUENCY_COLOR, PURPL
```

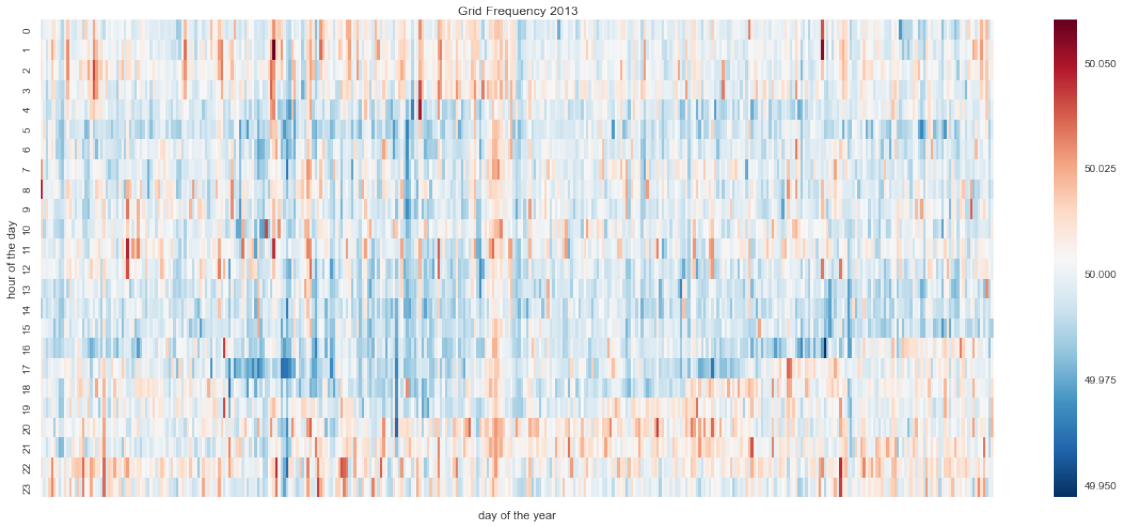
```
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x9458710>
```




```
In [13]: freq_per_hour = pd.DataFrame(freq['2013'].resample('H'))
         freq_per_hour['hour'] = [x.hour for x in freq_per_hour.index]
```

```
freq_per_hour['day'] = [x.dayofyear for x in freq_per_hour.index]
freq_per_hour = freq_per_hour.pivot("hour", "day", "grid frequency in Hz")
sns.set(rc={"figure.figsize": (20, 8)})
ax = sns.heatmap(freq_per_hour, xticklabels=False, linewidths=0, cmap="RdBu_r")
ax.xaxis.set_label_text('day of the year')
ax.yaxis.set_label_text('hour of the day')
ax.set_title('Grid Frequency 2013')
```

Out[13]: <matplotlib.text.Text at 0x984ab70>



3 Meta data

```
In [14]: %load_ext watermark
         %watermark
```

```
15/04/2015 13:53:10

CPython 3.4.2
IPython 3.1.0

compiler   : MSC v.1600 64 bit (AMD64)
system     : Windows
release    : 7
machine    : AMD64
processor  : Intel64 Family 6 Model 58 Stepping 9, GenuineIntel
CPU cores  : 4
interpreter: 64bit
```

4 The Notebook

- Use cases
  - scientist's notebook

  - interactive course material including assignments
  - writing papers
- Export to
  - latex
  - markdown
  - pdf
  - html
  - slideshow
- Supported languages:
  - Python
  - Julia
  - R
  - Octave
  - Matlab
  - Haskell
  - Perl
  - Ruby
  - Erlang
  - ...

5 The Project

Evolved from the IPython project

- 2001: `IPython 0.0.1` first release
- 2011: `IPython 0.12` first release including notebook
- 2013: `IPython 1.0` first official release
- 2015: `IPython 3.0` last monolithic release

Project Jupyter succeeds `IPython 3`'s language agnostic parts in different modules.

- `http://jupyter.org/`
- Try it online: try.jupyter.org
- Installation of the current version can be done easiest with Python Scientific Distribution Anaconda