Demo

Release 1.0

Petr

Sep 21, 2021

CONTENTS:

CHAPTER

ONE

WELCOME TO THE JUPYTER DEMO

This is an example .ipynb notebook

[1]:
```
import sys
print(sys.version)
```

```
3.7.9 (default, Oct 19 2020, 15:13:17)
[GCC 7.5.0]
```

[2]:
```
from IPython.display import Image
```

[3]:
```
Image("http://sipi.usc.edu/database/preview/misc/4.2.03.png")
```

[3]:

1.1 Interactive widgets

Example that render an interactive widget

[4]:
```
import ipywidgets as ipw
```

[5]:
```
slider = ipw.IntSlider(min=0, value=10, max=20)
slider
```

```
IntSlider(value=10, max=20)
```

[6]:
```
slider
```

```
IntSlider(value=10, max=20)
```

[7]:
```
import panel as pn
pn.extension()
```

Data type cannot be displayed: application/javascript, application/vnd.holoviews_load.v0+json

Data type cannot be displayed: application/vnd.holoviews_load.v0+json, application/javascript

[8]:
```
def f(x):
    return x
```

[9]:
```
pn.interact(f, x=10)
```

[9]:
```
Column
    [0] Column
        [0] IntSlider(end=30, name='x', start=-10, value=10)
    [1] Row
        [0] Str(int, name='interactive00101')
```

CHAPTER

TWO

EGI SERVICES

Compute

Cloud Compute

Run virtual machines on demand with complete control over computing resources

Cloud Container Compute

Run Docker containers in a lightweight virtualised environment

High-Throughput Compute

Execute thousands of computational tasks to analyse large datasets

Workload Manager

Manage computing workloads in an efficient way

Storage and Data

Online Storage

Store, share and access your files and their metadata on a global scale

Data Transfer

Transfer large sets of data from one place to another

DataHub

Access key scientific datasets in a scalable way

Security

Check-in

Login with your own credentials

Applications

Applications on Demand

Share online applications for your data and compute-intensive research

Notebooks

Create interactive documents with live code, visualisations and text

Training

FitSM Training

Learn how to manage IT services with a pragmatic and lightweight standard

ISO 27001 Training

Learn how to manage and secure information assets

Training Infrastructure

Dedicated computing and storage for training and education

[10]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('data/datatest.txt')

Dx = data["date"]

data['date'] = data.date.astype('datetime64[ns]')
data = data.set_index('date')

data.tail()
```

```
Matplotlib is building the font cache; this may take a moment.
```

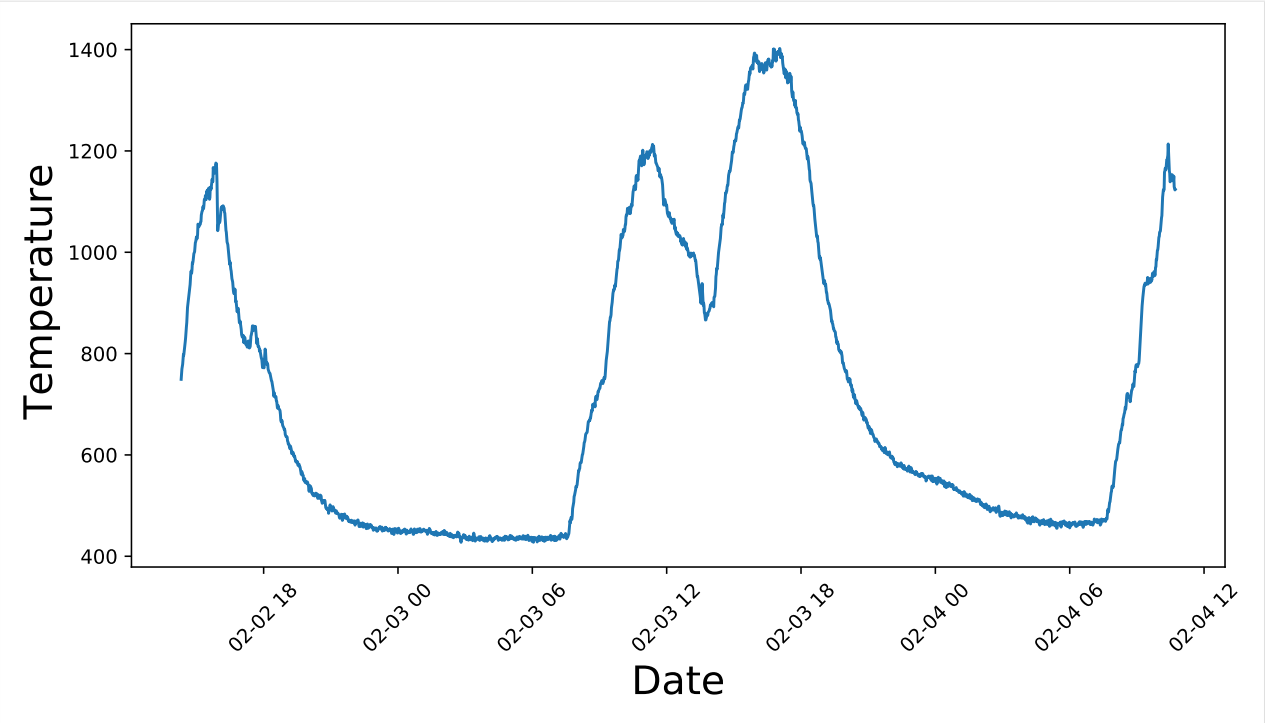
[10]:

date	Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
2015-02-04 10:38:59	24.290000	25.700000	808.0	1150.25	0.004829	1
2015-02-04 10:40:00	24.330000	25.736000	809.8	1129.20	0.004848	1
2015-02-04 10:40:59	24.330000	25.700000	817.0	1125.80	0.004841	1
2015-02-04 10:41:59	24.356667	25.700000	813.0	1123.00	0.004849	1
2015-02-04 10:43:00	24.408333	25.681667	798.0	1124.00	0.004860	1

[11]:
```
from matplotlib.figure import Figure
from matplotlib.backends.backend_agg import FigureCanvas
%matplotlib inline

plt.figure(figsize=(10,5));
plt.plot(data.CO2);
plt.xticks(rotation=45);
plt.xlabel('Date', fontsize = 20)
plt.ylabel('Temperature',fontsize = 20)

plt.show()
```



2.1 Můžu použít nějakou statistiku - třeba rolling mean:

$$\tau = \frac{\varpi - \omega}{\sqrt{\frac{\sum \varpi - \omega}{N(N-1)}}}$$

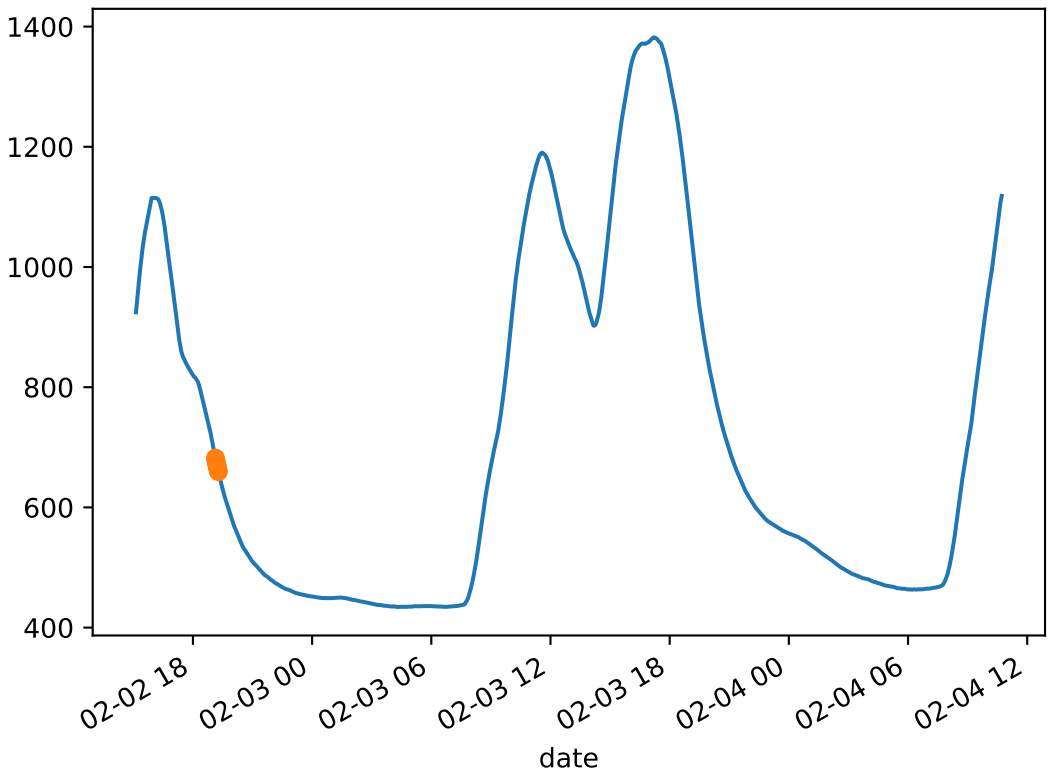
2.2 Tohle sice není rolling mean, ale aspoň vidíte, že se sem dá dát i LaTeX

[12]:
```
def mpl_plot(avg, highlight):
    fig = Figure()
    FigureCanvas(fig)
    ax = fig.add_subplot()
    avg.plot(ax=ax)
    if len(highlight): highlight.plot(style='o', ax=ax)
    return fig

def find_outliers(variable='Temperature', window=50, sigma=10, view_fn=mpl_plot):
    avg = data[variable].rolling(window=window).mean()
    residual = data[variable] - avg
    std = residual.rolling(window=window).std()
    outliers = (np.abs(residual) > std * sigma)
    return view_fn(avg, avg[outliers])
```

[13]:
```
find_outliers(variable='CO2', window=50, sigma=10)
```

[13]:



[14]:
```
import panel as pn
pn.extension()

pn.interact(find_outliers)
```

Data type cannot be displayed: application/javascript, application/vnd.holoviews_load.v0+json

Data type cannot be displayed: application/vnd.holoviews_load.v0+json, application/javascript

[14]:
```
Column
    [0] Column
        [0] IntSlider(end=150, name='window', start=-50, value=50)
        [1] IntSlider(end=30, name='sigma', start=-10, value=10)
    [1] Row
        [0] Matplotlib(Figure, name='interactive00112')
```

[ ]:

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search