
EGI notebooks

Release 1.0

Petr Pospíšil

Sep 21, 2021

Contents:

1	EGI Services	2
2	Přihlášení do EGI:	3
3	1) Vytvoření EGI účtu	3
4	2) Přihlášení do vo.notebooks.egi.eu VO	5
5	3) Přihlášení do EGI notebooks:	6
6	Co v takovém notebooku může být? A k čemu je dobrý?	9
7	Může v něm být nadpis - to už víme	9
8	Můžeme nahrát data a zpracovat je:	9
9	Udělejme to víc interaktivní za pomoci Panelu (ať se nemusíme pořád hrabat v kódu):	13
10	Dokonce se sem dá dát i gif	17
11	K čemu tedy ty notebooky jsou, nebo mohou být:	17
12	Co když někdo Jupyter nemá? Nebo nemá přístup do EGI nebo Metacentra?	17
13	EGI DataHub – OneData	18

1 EGI Services

<https://www.egi.eu/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

2 Přihlášení do EGI:

1. Vytvoření EGI účtu
 2. Přihlášení do vo.notebooks.egi.eu VO
 3. Přihlášení do EGI notebooks
-

3 1) Vytvoření EGI účtu

<https://aai.egi.eu/signup>



EGI User Community

Thank you for your interest in joining the EGI User Community. As part of this process, you will be assigned a personal EGI ID which is needed to access EGI tools and services.

SIGN UP



EGI User Community

EGI ID

Identifier

5fc5e8e871b10c76747835f56854b4f2dc578b5024364fd14

Name*

Your full name

Given Name*

Nicolas

Family Name*

Liampotis

Email

Your current email
address

Email*

nliam@admin.grnet.gr

Affiliation

Member

Organisation

Organisation*

grnet-hq.admin.grnet.gr

Agree to Terms and Conditions

You must agree to the following Terms and Conditions before continuing.

You must review the T&C before you can click I Agree, and you must agree before you can submit.

EGI AAI Terms of Use

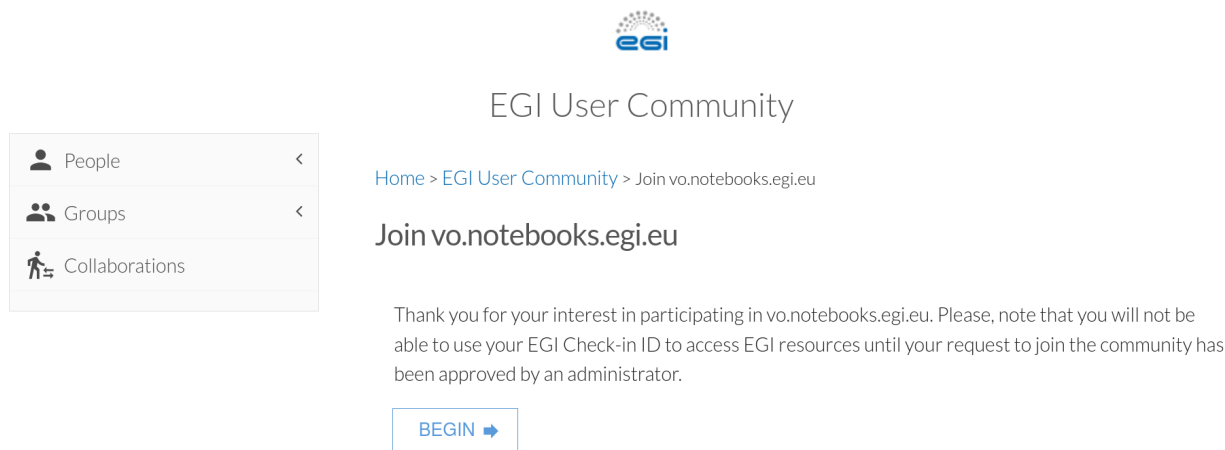
☒ Review Terms and Conditions

☒ I Agree

3.1 Check-in pošle email pro ověření požadavku a emailu (EGI Check-in Notifications)

Pokud nastane problém během celého procesu: support@egi.eu

4 2) Přihlášení do vo.notebooks.egi.eu VO





EGI User Community

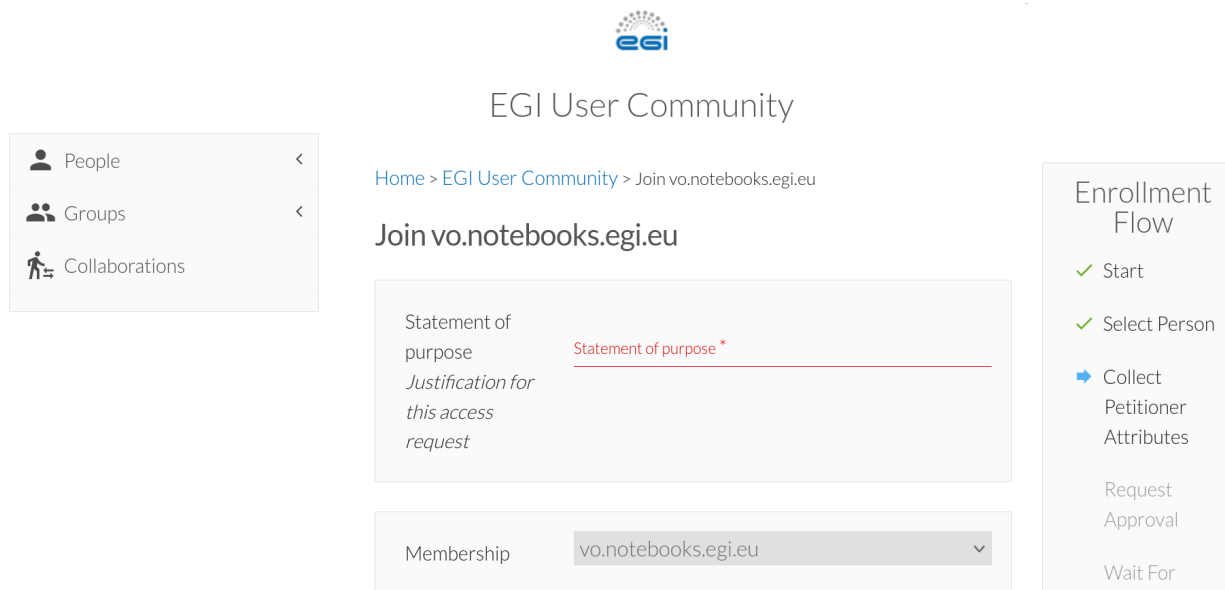
Home > EGI User Community > Join vo.notebooks.egi.eu


Join vo.notebooks.egi.eu

Thank you for your interest in participating in vo.notebooks.egi.eu. Please, note that you will not be able to use your EGI Check-in ID to access EGI resources until your request to join the community has been approved by an administrator.

[BEGIN ➔](#)

RT: 17237 [EGI-20210430-01] [TLP:AMBER] Security incident suspected at CESNET stále otevřené: update?





EGI User Community

Home > EGI User Community > Join vo.notebooks.egi.eu

Join vo.notebooks.egi.eu

Statement of purpose
Justification for this access request

Statement of purpose *

Membership vo.notebooks.egi.eu

Enrollment Flow

- ✓ Start
- ✓ Select Person
- ➔ Collect Petitioner Attributes
- Request Approval
- Wait For

5 3) Přihlášení do EGI notebooks:

<https://notebooks.egi.eu>



Notebooks

Notebooks is an environment based on [Jupyter](#) and the [EGI cloud service](#) that offers a browser-based, scalable tool for interactive data analysis. The Notebooks environment provides users with notebooks where they can combine text, mathematics, computations and rich media output.

Individual users can directly login by clicking the button below. The notebooks are limited to 1 CPU, 1GB RAM and 10GB of persistent storage per user.

Start your notebooks!

User communities/advanced users can have their customised EGI Notebooks service instance. EGI offers consultancy and support, as well as can operate the setup. Order a [community notebooks instance via the Marketplace](#).

The service is operated by and uses resources from [CESNET](#)





Check-in

Choose your academic/social account

29 Mayis University

A'Sharqiyah University

A*STAR - Agency for Science, Technology and Research

A. T. Still University

AAF Virtual Home

aai.lab.maeen.sa

AAU@EduHr Single Sign-On Service

or





Check-in

EGI AAI OpenID Connect Provider Proxy requires that the information below is transferred.

Entitlement regarding the service	
Mail	pospasilp@cesnet.cz
Display name	Petr Pospisil
Given name	Petr
Surname	Pospisil
Person's non-reassignable, persistent pseudonymous ID at	09189b6e155a6faaeae38c1dc007e6c24c0a310609e75d5757a4875694f275b4@egi.eu
<input type="checkbox"/> Remember	
Privacy policy for the service EGI AAI OpenID Connect	<input type="button" value="YES, CONTINUE"/> <input type="button" value="NO, CANCEL"/>

Celá přihlašovací procedura do EGI notebooků je zde: <https://docs.egi.eu/users/notebooks/>

5.1 Základní notebook:

Limit na uživatele: 2 CPU, 4GB RAM, 20GB storage

- Python
- Dirac/Python 2 (EGI Workload Manager)
- Julia
- R
- Octave

Home Token Services b10d032648d03b75a24a779533376e2376f9de61508522bce083ed850492b13a@egi.eu

Server Options

<input checked="" type="radio"/>	Default EGI environment - 4 GB RAM / 2 core The Default notebook environment includes Python, R, Julia and Octave kernels
<input type="radio"/>	MATLAB Environment - 4GB RAM / 4 cores The MATLAB environment (requires a valid license), includes Python and MATLAB kernels

5.2 MATLAB:

1. Individuální nebo akademické licence

2. Síťová licence

3. Zkušební licence (bez toolboxů)

V tuto chvíli musí mít každý svou MATLAB licenci – máme tam implementovat CESNET licenci?

6 Co v takovém notebooku může být? A k čemu je dobrý?

7 Může v něm být nadpis - to už víme

Když je potřeba něco tučně, tak to jde udělat **tučně**

Nebo to může být *kurzívou*

Anebo to může být ***kurzívou a tučně***

8 Můžeme nahrát data a zpracovat je:

```
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.

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

(continues on next page)

(continued from previous page)

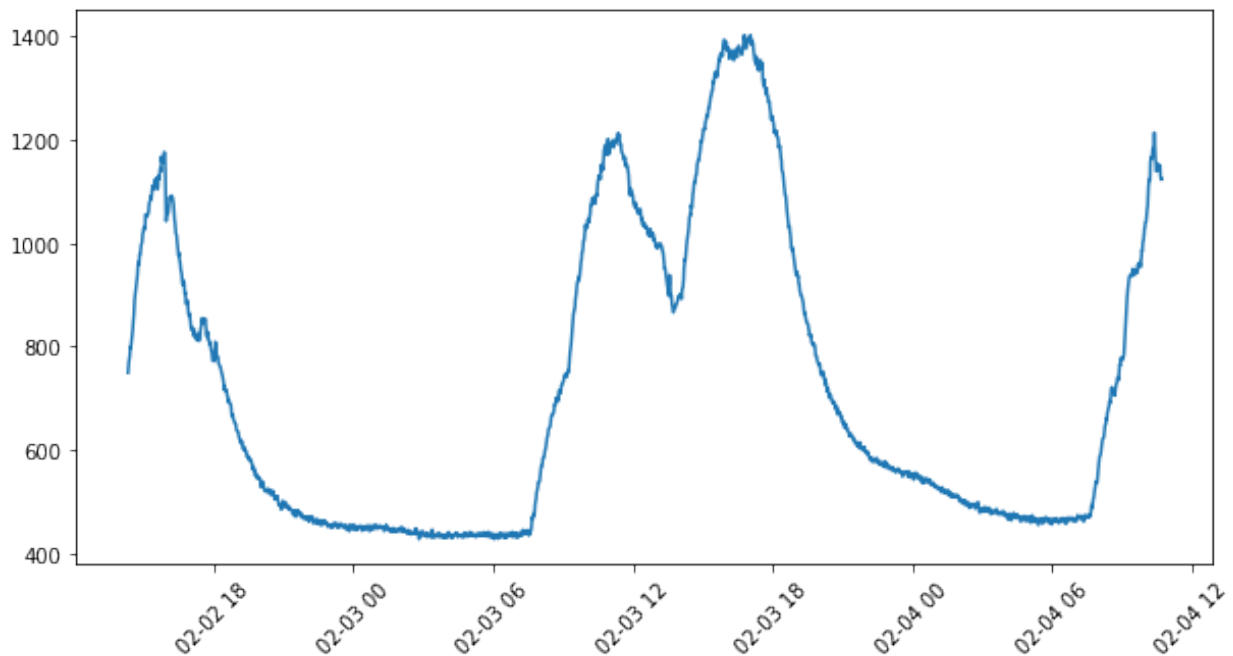
Occupancy	
date	
2015-02-04 10:38:59	1
2015-02-04 10:40:00	1
2015-02-04 10:40:59	1
2015-02-04 10:41:59	1
2015-02-04 10:43:00	1

8.1 Data obsahují měření v konferenční místnosti v čase

```
from matplotlib.figure import Figure
from matplotlib.backends.backend_agg import FigureCanvas
%matplotlib inline

#data['Occupancy'] = 100*data['Occupancy']
plt.figure(figsize=(10,5));
plt.plot(data.CO2);
plt.xticks(rotation=45);
#plt.ylabel('Temperature',fontsize = 20)

plt.show()
```



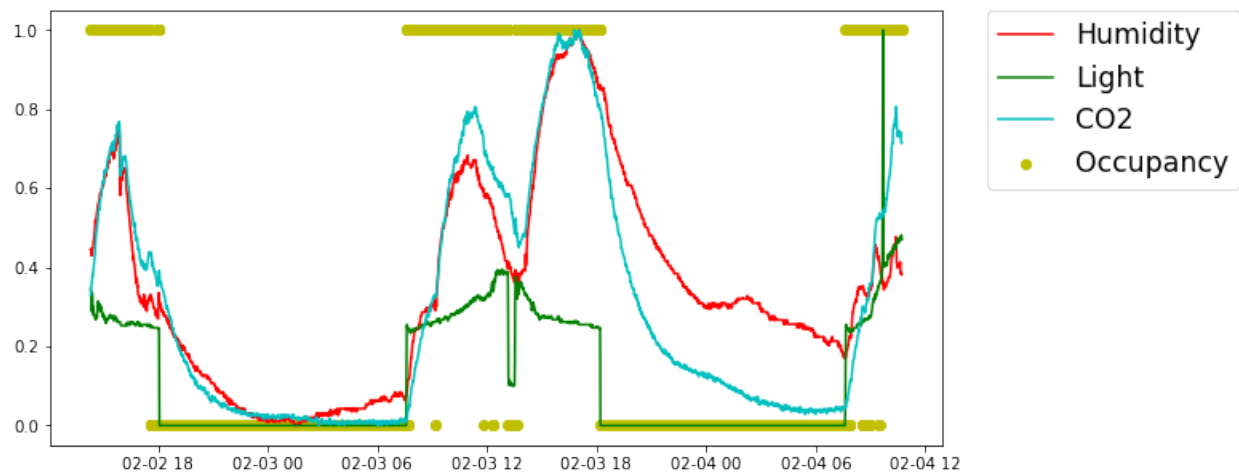
```
T = (data.Temperature-min(data.Temperature))
T = T/max(T)
H = (data.Humidity-min(data.Humidity))
H = H/max(H)
L = (data.Light-min(data.Light))
L = L/max(L)
```

(continues on next page)

(continued from previous page)

```
C = (data.CO2-min(data.CO2))  
C = C/max(C)
```

```
plt.figure(figsize=(10,5))  
#Temp = plt.plot(T,'b',label = 'Temperature')  
Humid = plt.plot(H,'r',label = 'Humidity')  
Light = plt.plot(L,'g',label = 'Light')  
CO = plt.plot(C,'c',label = 'CO2')  
Occup = plt.scatter(Dx,data.Occupancy,c = 'y',label = 'Occupancy')  
plt.legend(fontsize = 'xx-large',bbox_to_anchor=(1.05, 1), loc='upper left',  
borderaxespad=0.)  
plt.show()
```



8.2 Závěr nebo hypotéza:

1. datatest2 je divnej
 2. někdo se tam vloupal
 3. datatest je dobrej - sedí
- ...

Můžu použít nějakou statistiku - třeba rolling mean:

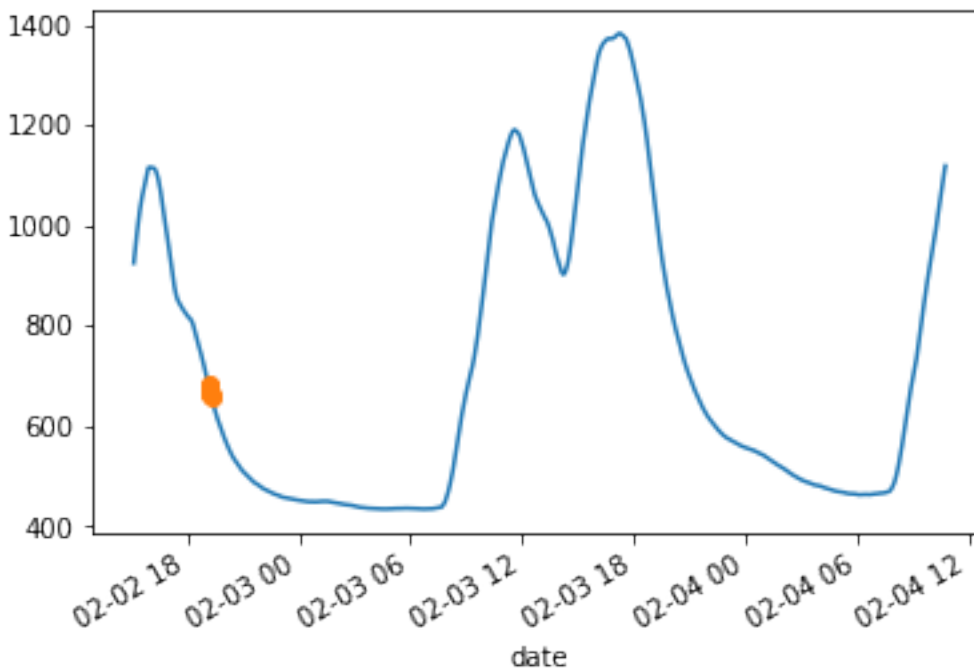
$$\tau = \frac{\bar{\omega} - \omega}{\sqrt{\frac{\sum \bar{\omega} - \omega}{N(N-1)}}}$$

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

```
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])
```

```
find_outliers(variable='CO2', window=50, sigma=10)
```



9 Udělejme to víc interaktivní za pomoci Panelu (ať se nemusíme pořád hrabat v kódu):

```
import panel as pn
pn.extension()

pn.interact(find_outliers)
```

```
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='interactive00101')
```

```
kw = dict(window=(1, 60), variable=sorted(list(data.columns)), sigma=(1, 20))
i = pn.interact(find_outliers, **kw)
#i.pprint()
```

```
text = "<br>\n# Room measurements\nSelect the variable, and the time window for smoothing  
↪"

p = pn.Column(pn.Column(text, i[0][0], i[0][1]), i[1][0])
p
```

```
Column
  [0] Column
      [0] Markdown(str)
      [1] Select(name='variable', options=['CO2', 'Humidity', ...], value='Temperature  
↪')
      [2] IntSlider(end=60, name='window', start=1, value=50)
  [1] Matplotlib(Figure, name='interactive00114')
```

```
i[0][2]
```

```
IntSlider(end=20, name='sigma', start=1, value=10)
```

```
import panel.widgets as pnw

variable = pnw.RadioButtonGroup(name='variable', value='Temperature',
                                options=list(data.columns))
window = pnw.IntSlider(name='window', value=10, start=1, end=60)

reactive_outliers = pn.bind(find_outliers, variable, window, 10)

widgets = pn.Column("<br>\n# Room measurements", variable, window)
occupancy = pn.Column(widgets, reactive_outliers)
occupancy
```

```

Column
  [0] Column
    [0] Markdown(str)
    [1] RadioButtonGroup(name='variable', options=['Temperature', ...], value=
↪ 'Temperature')
    [2] IntSlider(end=60, name='window', start=1, value=10)
  [1] ParamFunction(function)

```

```
import param
```

```

class RoomOccupancy(param.Parameterized):
    variable = param.Selector(objects=list(data.columns))
    window   = param.Integer(default=10, bounds=(1, 20))
    sigma    = param.Number(default=10, bounds=(0, 20))

    def view(self):
        return find_outliers(self.variable, self.window, self.sigma)

```

```

obj = RoomOccupancy()
obj

```

```
RoomOccupancy(name='RoomOccupancy00147', sigma=10, variable='Temperature', window=10)
```

```
pn.Column(obj.param, obj.view)
```

```

Column
  [0] Column(margin=5, name='RoomOccupancy', width=300)
    [0] StaticText(value='<b>RoomOccupancy</b>')
    [1] Select(name='Variable', options=OrderedDict([('Temperature...)], value=
↪ 'Temperature')
    [2] IntSlider(end=20, name='Window', start=1, value=10)
    [3] FloatSlider(end=20, name='Sigma', value=10)
  [1] ParamMethod(method)

```

```
import hvplot.pandas
```

```

def hvplot(avg, highlight):
    line = avg.hvplot(height=300, width=550)
    avg.hvplot(height=300) * highlight.hvplot.scatter(color='orange', padding=0.1)
    outliers = highlight.hvplot.scatter(color='orange', padding=0.1)
    return (line * outliers).opts(legend_position='top_right')

```

```

text2 = "## Room Occupancy\nSelect the variable and the smoothing values"
hvp = pn.interact(find_outliers, view_fn=hvplot, **kw)
pn.Column(pn.Row(pn.panel(text2, width=200), hvp[0]), hvp[1]).servable("Occupancy")

```

```

Column
  [0] Row
    [0] Markdown(str, width=200)
    [1] Column
      [0] Select(name='variable', options=['CO2', 'Humidity', ...], value=
↪ 'Temperature')

```

(continues on next page)

(continued from previous page)

```
[1] IntSlider(end=60, name='window', start=1, value=50)
[2] IntSlider(end=20, name='sigma', start=1, value=10)
[1] Row
[0] HoloViews(Overlay, name='interactive01621')
```

```
import holoviews as hv

tap = hv.streams.PointerX(x=data.index.min())

def hvplot2(avg, highlight):
    line = avg.hvplot(height=300, width=500)
    outliers = highlight.hvplot.scatter(color='orange', padding=0.1)
    tap.source = line
    return (line * outliers).opts(legend_position='top_right')

@pn.depends(tap.param.x)
def table(x):
    index = np.abs((data.index - x).astype(int)).argmin()
    return data.iloc[index]

app = pn.interact(find_outliers, view_fn=hvplot2, **kw)

pn.Column(
    pn.Column("## Room Occupancy\nHover over the plot for more information.", app[0]),
    pn.Column(app[1], table)
)
```

```
Column
[0] Column
[0] Markdown(str)
[1] Column
[0] Select(name='variable', options=['CO2', 'Humidity', ...], value=
→ 'Temperature')
[1] IntSlider(end=60, name='window', start=1, value=50)
[2] IntSlider(end=20, name='sigma', start=1, value=10)
[1] Column
[0] Row
[0] HoloViews(Overlay, name='interactive02379')
[1] ParamFunction(function)
```

```
from IPython.display import Audio, YouTubeVideo
```

```
Audio('audio/simpleLoop.wav')
```

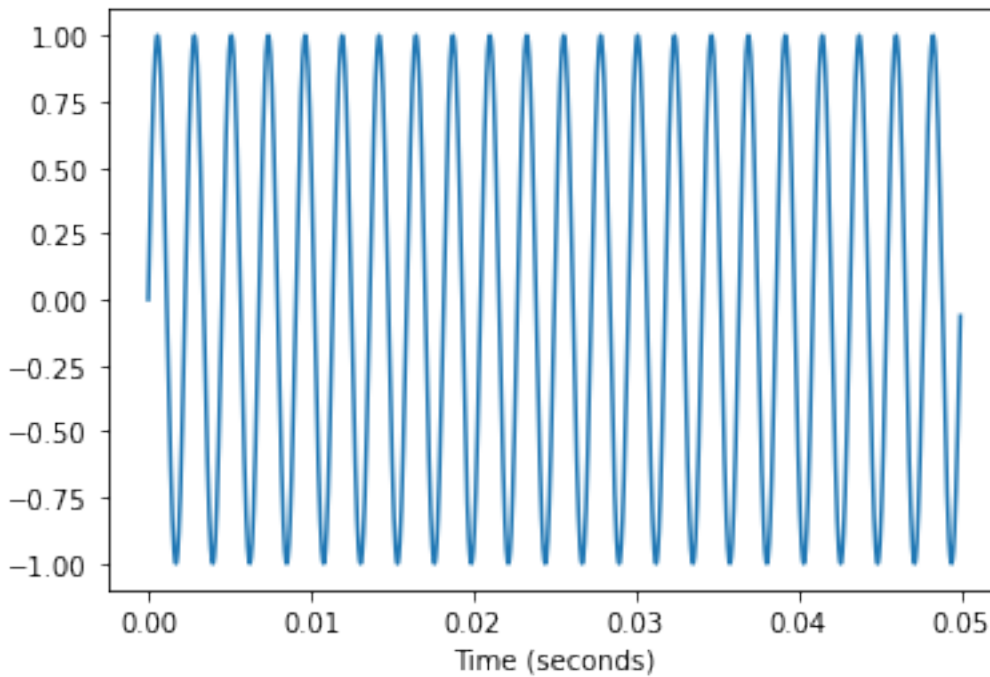
```
<IPython.lib.display.Audio object>
```

```
T = 0.05 # seconds
fs = 44100 # sampling frequency
t = np.linspace(0, T, int(T*fs), endpoint=False) # time variable
```

(continues on next page)

(continued from previous page)

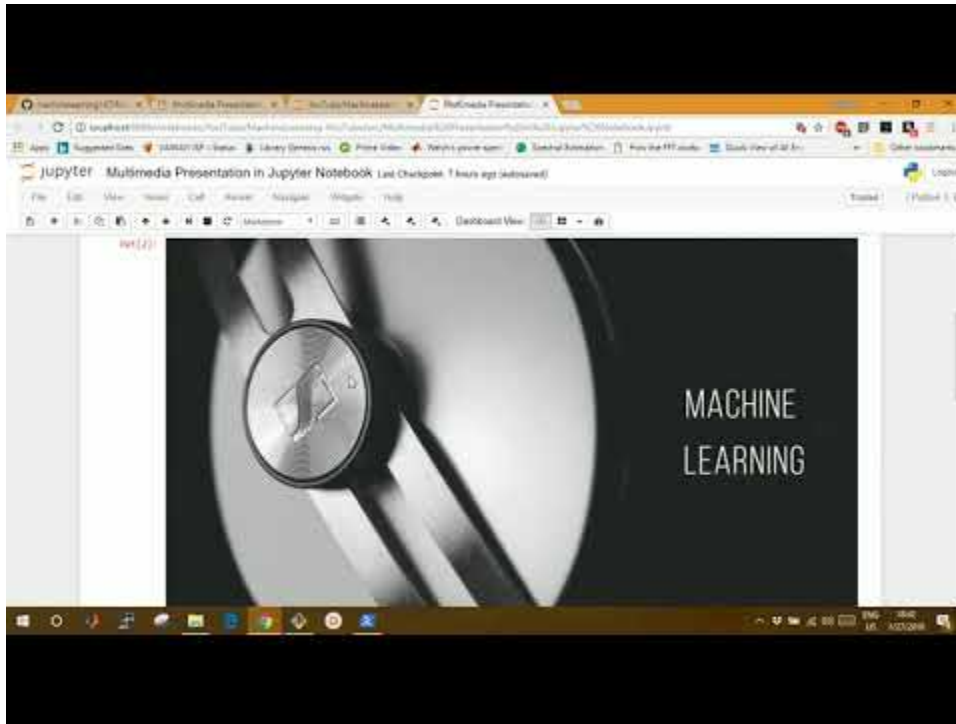
```
x = np.sin(2*np.pi*440*t)
plt.plot(t, x)
plt.xlabel('Time (seconds)')
plt.show()
```



```
fs = 44100 # sampling frequency
T = 1.5    # seconds
t = np.linspace(0, T, int(T*fs), endpoint=False) # time variable
x = np.sin(2*np.pi*440*t)                       # pure sine wave at 440 Hz
Audio(x, rate=fs)
```

```
<IPython.lib.display.Audio object>
```

```
id = 'ER4WhBxM8DY'
YouTubeVideo(id=id, width = 700, height = 400)
```

10 Dokonce se sem dá dát i gif

...ale v pdf to nefunguje...

11 K čemu tedy ty notebooky jsou, nebo mohou být:

1. Interaktivní učební pomůcka nebo interaktivní tutorial
 2. Vytváření reportů nebo manuscriptů
 3. Workflow od dat až po interpretaci výsledků
-

12 Co když někdo Jupyter nemá? Nebo nemá přístup do EGI nebo Metacentra?

Nevadí, dáme to do [Binderu](#)



Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

New to Binder? Get started with a Zero-to-Binder tutorial in [Julia](#), [Python](#) or [R](#).

Build and launch a repository

GitHub repository name or URL

GitHub ▾

Git ref (branch, tag, or commit)


Path to a notebook file (optional)

File ▾

launch

Copy the URL below and share your Binder with others:

📄

Expand to see the text below, paste it into your README to show a binder badge:  [launch binder](#) ▶

Rychlý odkaz

13 EGI DataHub – OneData

<https://datahub.egi.eu/ozw/onezone/i#/login>

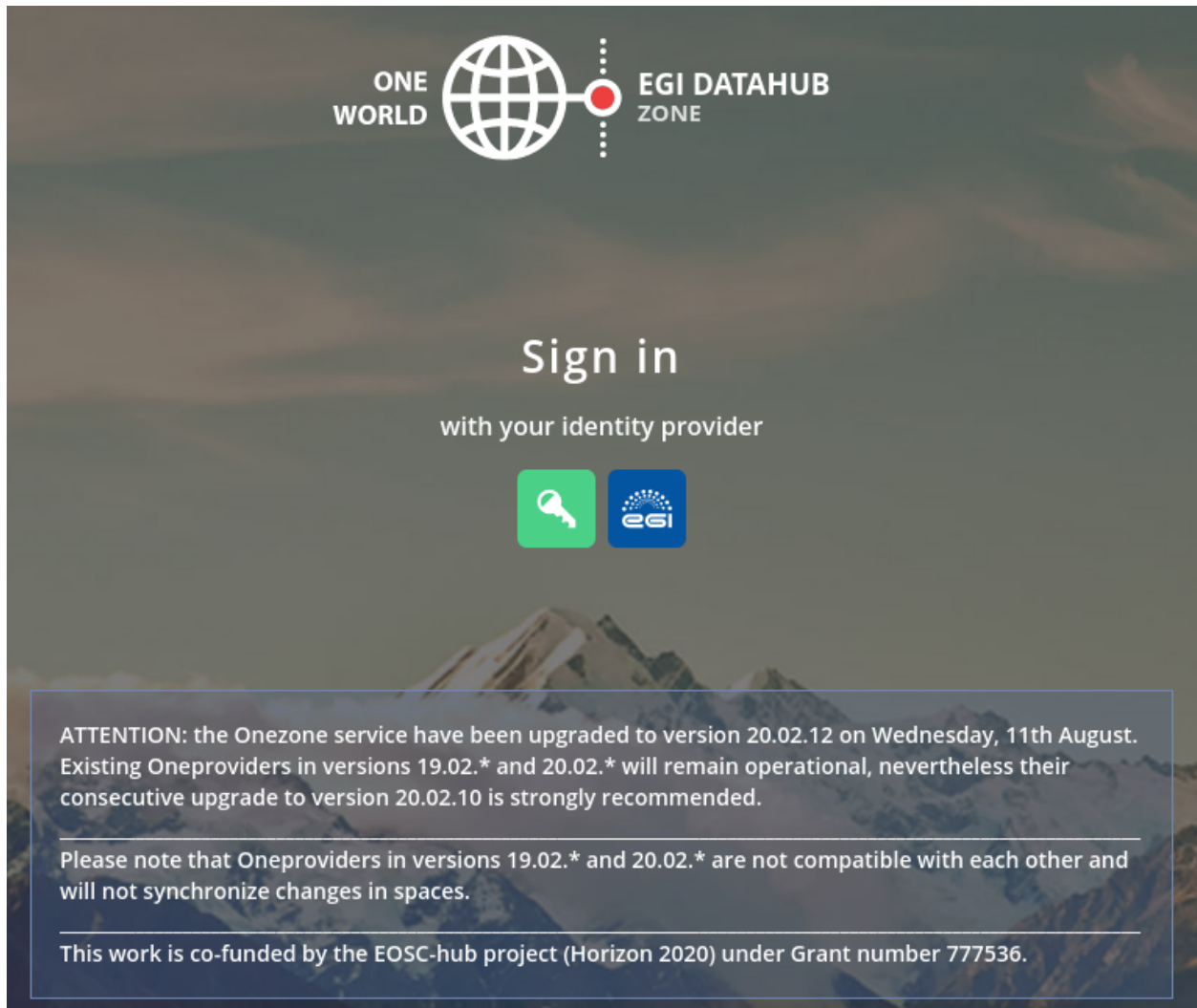
13.1 Běží zatím jen na training notebooks:

<https://training.notebooks.egi.eu>

13.2 Připojení DataHubu automaticky i pro:

<https://notebooks.egi.eu>

Jira ticket: <https://jira.egi.eu/browse/ACETA-52>



```
plt.figure(figsize=(15,10))
plt.subplot(211)
plt.plot(data.Temperature)
plt.ylabel('degrees',fontsize = 20)

plt.title('Temperature',fontsize = 25)

plt.xticks(fontsize = 20)
plt.yticks(fontsize = 20)

plt.subplot(212)
plt.plot(data.Occupancy)
plt.ylabel('%',fontsize = 20)
```

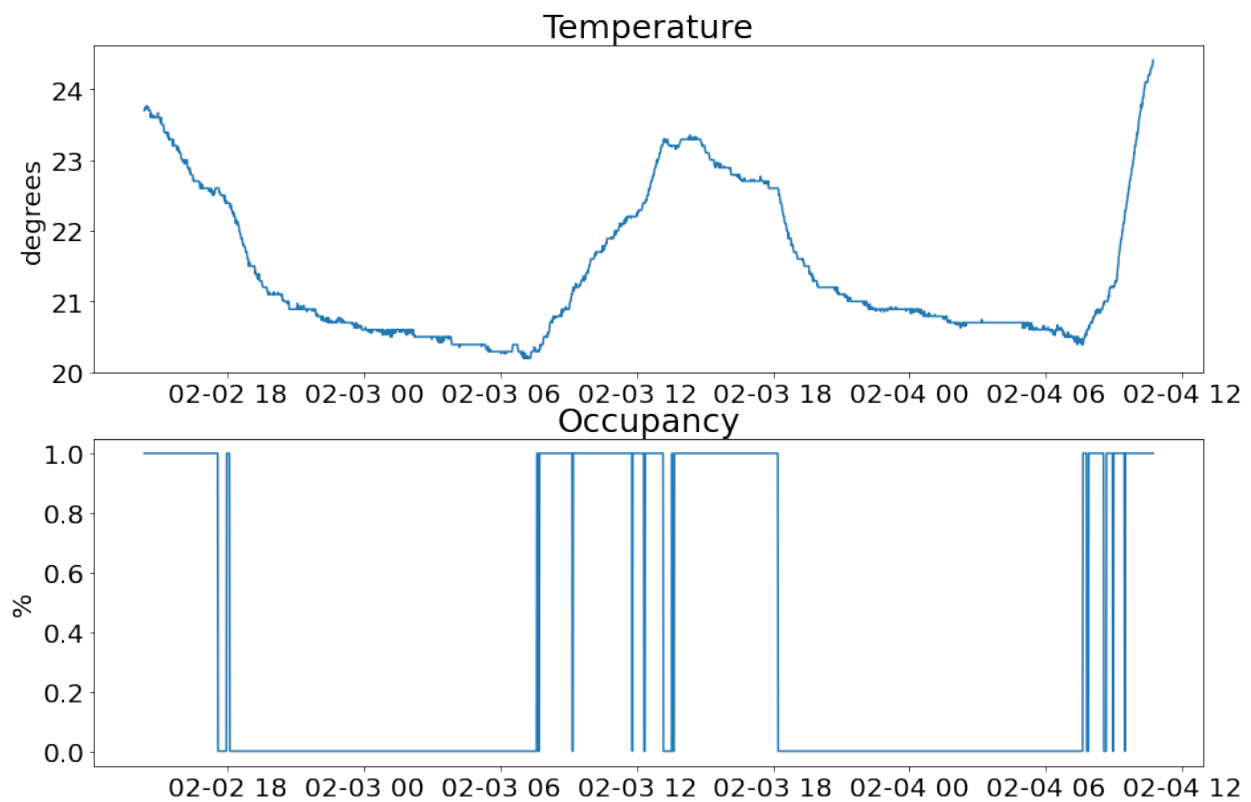
(continues on next page)

(continued from previous page)

```
plt.title('Occupancy',fontsize = 25)

plt.yticks(fontsize = 20)
plt.xticks(fontsize = 20)

plt.show()
```



```
x = np.linspace(-10,10, num = 100)
y = x*x
plt.plot(x,y)
plt.show()
```

