

ATMO-SEISM (A2-1)

Domain Experts:

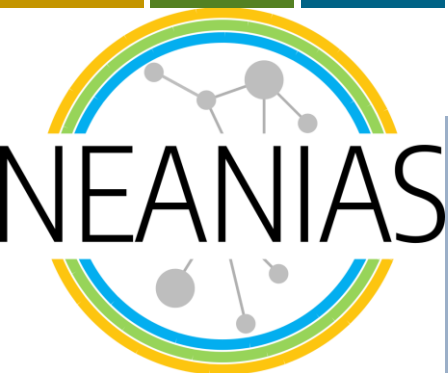
- Alessandro Tibaldi (UNIMIB)
- Fabio Luca Bonali (UNIMIB)
- Noemi Corti (UNIMIB)

Service development:

- Alekos Falagas (NKUA)
- Makis Ntouskos (NKUA)
- Konstantinos Karantzalos (ATHENA)



www.neanias.eu



Novel EDSC Services for Emerging
Atmosphere, Underwater & Space
Challenges

NEANIAS receives funding from
European Union under Horizon
2020 Research and Innovation
Programme under grant
agreement No. 863448



Goals of the service

- Correlate gas emissions with earthquakes and atmospheric conditions.

Rationale and scientific motivation

- Comprehension of the interplay between tectonic activity, volcanic eruptions and gas release through faults (e.g. radon, CO₂, SO₂ etc.).

Technology

- **Spatiotemporal correlation** of gas emissions and earthquakes
- Based on pipeline from Neri *et al.*, 2016
- It uses **statistical** and **machine learning** methods

Atmo-Seism Service: Jupyter Notebook

Neanias_Notebook - Jupyter Notebook

atmo-seism.neanias.eu/user/mdouskos/notebooks/Neanias_Notebook.ipynb

jupyter Neanias_Notebook Last Checkpoint: 10/20/2020 (autosaved)

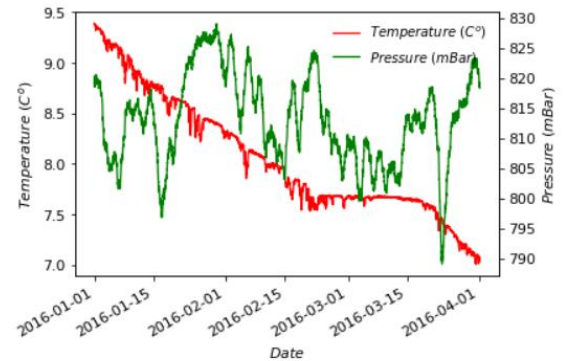
Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

2.2 Plotting radon measurements data

```
In [15]: # Convert to datetime
gas_data['Date / Hour'] = pd.to_datetime(gas_data['Date / Hour'], format = '%m/%d/%Y %H:%M')
# Plot Temperature and Pressure
df1 = gas_data[['Date / Hour', 'Temperature']]
df2 = gas_data[['Date / Hour', 'Pressure']]
# Plot Temperature and Pressure
Pyradon.plotlines(df1, df2, x_label = r'$Date$', y_label = [r'$Temperature\ (C^{\circ})$', r'$Pressure\ (mBar)$'], l1c = 'r',
del df1, df2
```



```
In [16]: # Plot Radon and Predicted Radon
df1 = gas_data[['Date / Hour', 'Radon']]
df2 = gas_data[['Date / Hour', 'Predicted Radon']]
Pyradon.plotlines(df1, df2, x_label = "Date", y_label = ["Radon", "Predicted Radon"], l1c = 'r', l1w = 1.3, l2c = 'g', l2w = 1.3,
del df1, df2
```

Plots of demo datasets

