



Finanziert von der
Europäischen Union
Funded by
the European Union



Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Erasmus+

„Strato-Ballon Measurement an Environmental Consciousness“

2024-1-DE02-KA210-VET-000243591

Material for students and teachers allowing for the preparation of measuring equipment. Developed and used as part of the project implementation

<https://euballon.zslp.edu.pl/>

Analysis of measurement results taken during a stratospheric balloon flight. Recorded data and analysis methods.

Developed own automatic data analysis tools.

Analysis of measurement results taken during a stratospheric balloon flight. Recorded data and analysis methods.

Developed own automatic data analysis tools.

Below we present our tools made in Python, which allow for quick analysis saved in CSV text files during the flight. For the analysis, Python version 3 with the **numpy** (<https://numpy.org/>) and **matplotlib** (<https://matplotlib.org/>) libraries installed is required.

Shared MicroPython (see other our manuals) software from 3 Raspberry Pi Pico2 recorders used during flight creates CSV files named:

UVvolt.CSV (measuring the voltage from the solar panel, measuring the level of UV radiation)

1875	297	2499.0
1876	332	2508.0
1877	292	2493.0
1878	329	2484.0
1879	292	2493.0

presssuretemp.CSV (measuring the air pressure, temperature, humidity)

1839	-44.77	165.34	7.75
1840	-44.68	166.83	7.66
1841	-44.92	168.25	7.77
1842	-44.21	169.53	7.74
1843	-45.01	171.16	7.76

CO2TVOCresult.CSV (measuring CO2 and volatile organic compounds TVOC)

2010	1822	42537
2011	1885	43408
2012	1844	43129
2013	1815	42467
2014	1861	42850

In all data files, the **tab character** was used to separate table fields. All of these files can be imported into Excel, but the proposed solution is much easier, faster and more efficient, also due to several thousand records.

1)A. Measuring the level of UV radiation

Python 3 script (All blank lines and comment lines starting with the "#" character can be omitted)

```
#library accepts only real numbers as data source
# data must also be represented in numeric form

import numpy as np
import matplotlib.pyplot as plt

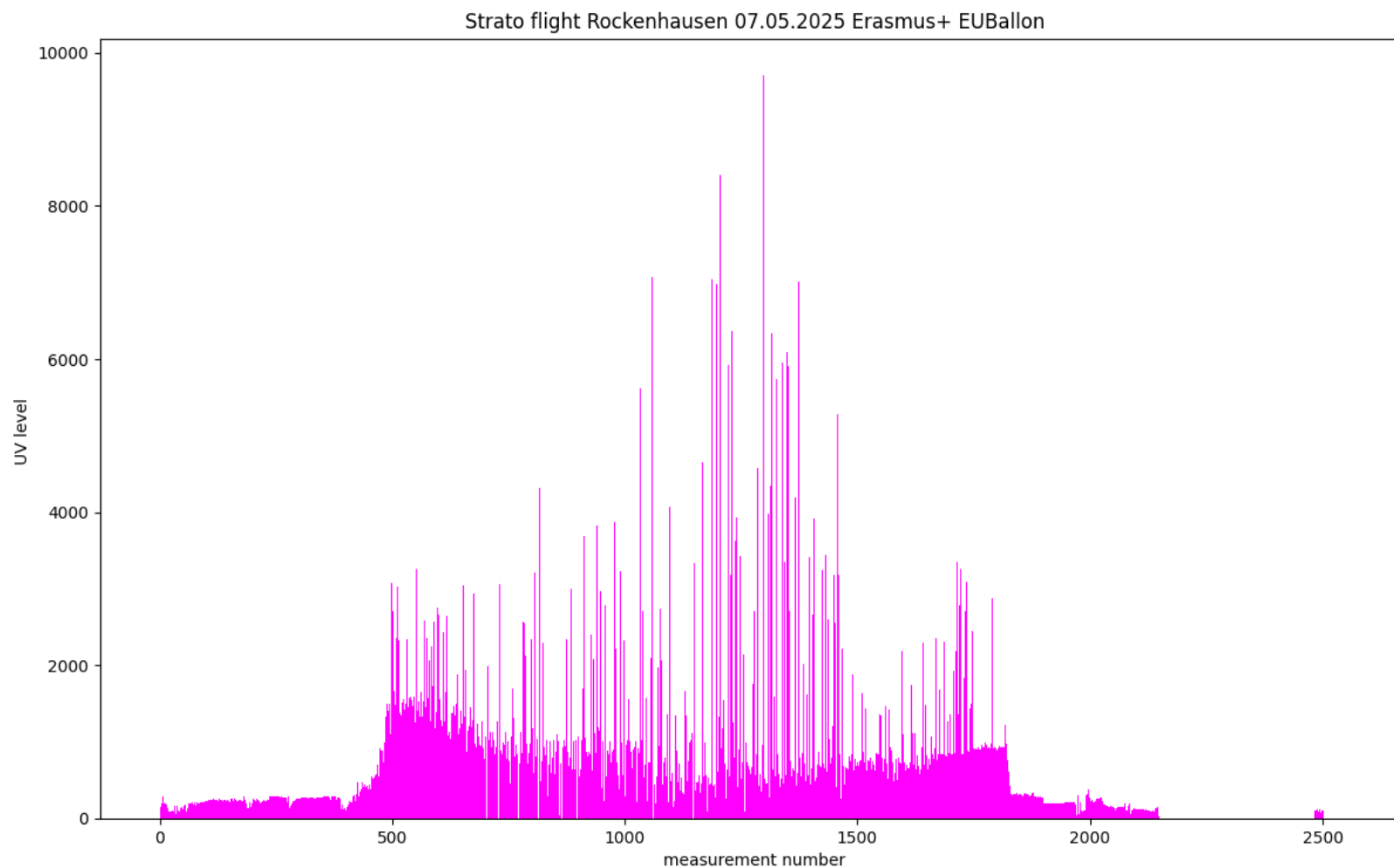
# filename with results: UVvolt.CSV
No,uvlevel=np.loadtxt('UVvolt.CSV',unpack=True,usecols=(0,1),delimiter ='\t')
print(No, uvlevel)
#Allocate space and determine the layout of the figure
fig, ax = plt.subplots()
ax.set_xlabel('measurement number')
ax.set_ylabel('UV level ')
plt.title('Strato flight Rockenhausen 07.05.2025 Erasmus+ EUBallon')
# plot bar graph
plot = plt.bar(No, uvlevel,width=1.0,color='magenta')
# Display chart
plt.show()
```

Intuitively, you can change the chart title, axis descriptions, and chart color.

We share this Python3 script with the name: **UVlevel_analysis.py**

We assume that the CSV data file and the above script are in the same folder

Chart UV radiation- png file generated by Python 3 UVlevel_analysis.py with matplotlib library based on data from CSV file: **UVvolt.CSV**



1) B. Measuring the voltage from the solar panel

Python 3 script (All blank lines and comment lines starting with the "#" character can be omitted)

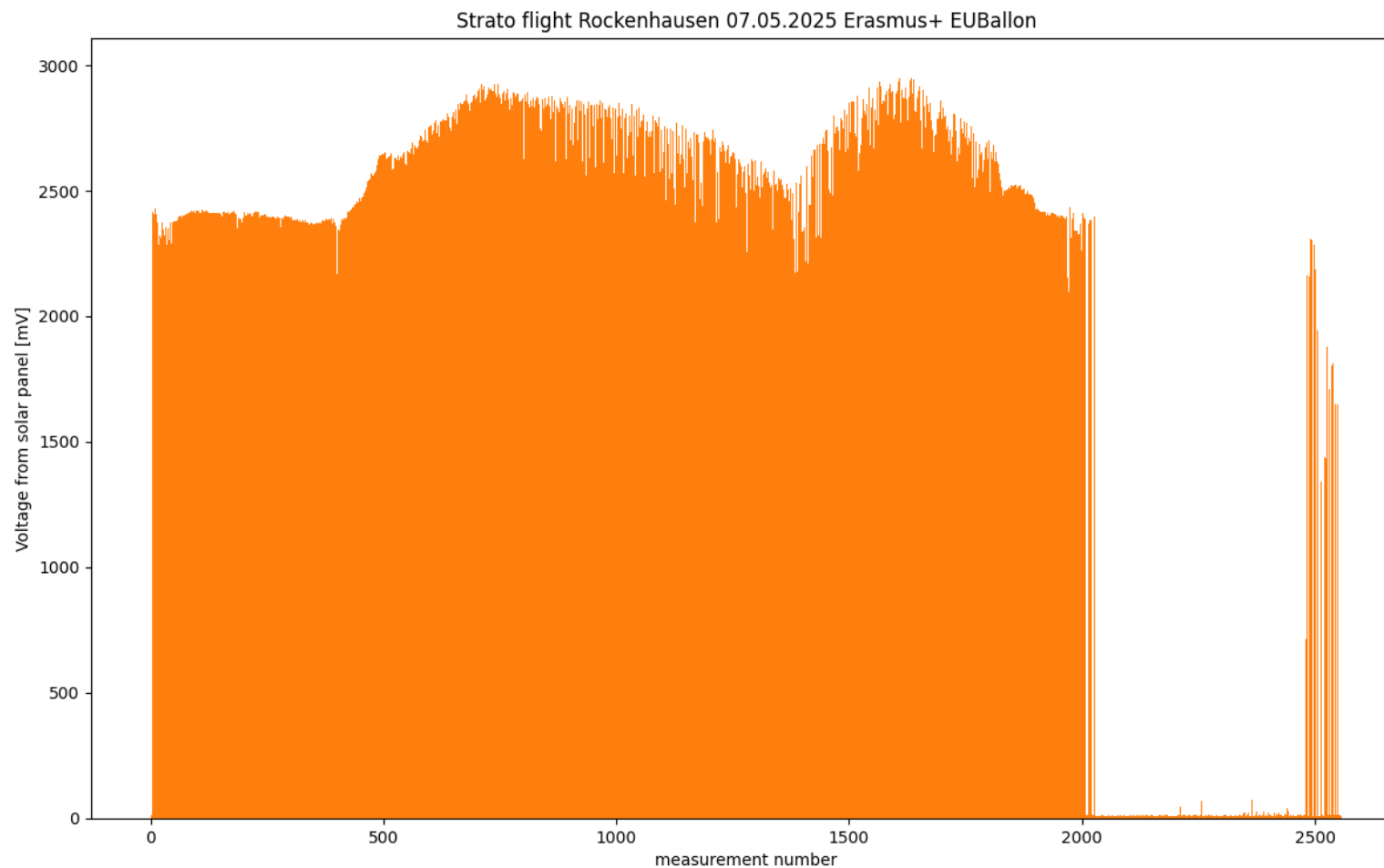
```
#library accepts only real numbers as data source
# data must also be represented in numeric form

import numpy as np
import matplotlib.pyplot as plt

# filename with results: UVvolt.CSV
No,voltage=np.loadtxt('UVvolt.CSV',unpack =True,usecols=(0,2),delimiter='\t')
print(No, voltage)
#Allocate space and determine the layout of the figure
fig, ax = plt.subplots()
ax.set_xlabel('measurement number')
ax.set_ylabel('Voltage from solar panel [mV]')
plt.title('Strato flight Rockenhausen 07.05.2025 Erasmus+ EUBallon')
# plot bar graph
plot = plt.bar(No, voltage,width=1.0,color="tab:orange")
# Display chart
plt.show()
```

Intuitively, You can change the chart title, axis descriptions, and chart color.
We share this Python3 script with the name: **solarpanelvoltage.py**.
We assume that the CSV data file and the above script are in the same folder.

Chart UV radiation- png file generated by Python 3 **solarpanelvoltage.py** with matplotlib library based on data from CSV file: **UVvolt.CSV** recorded during flight.



2) A. Measuring the air pressure

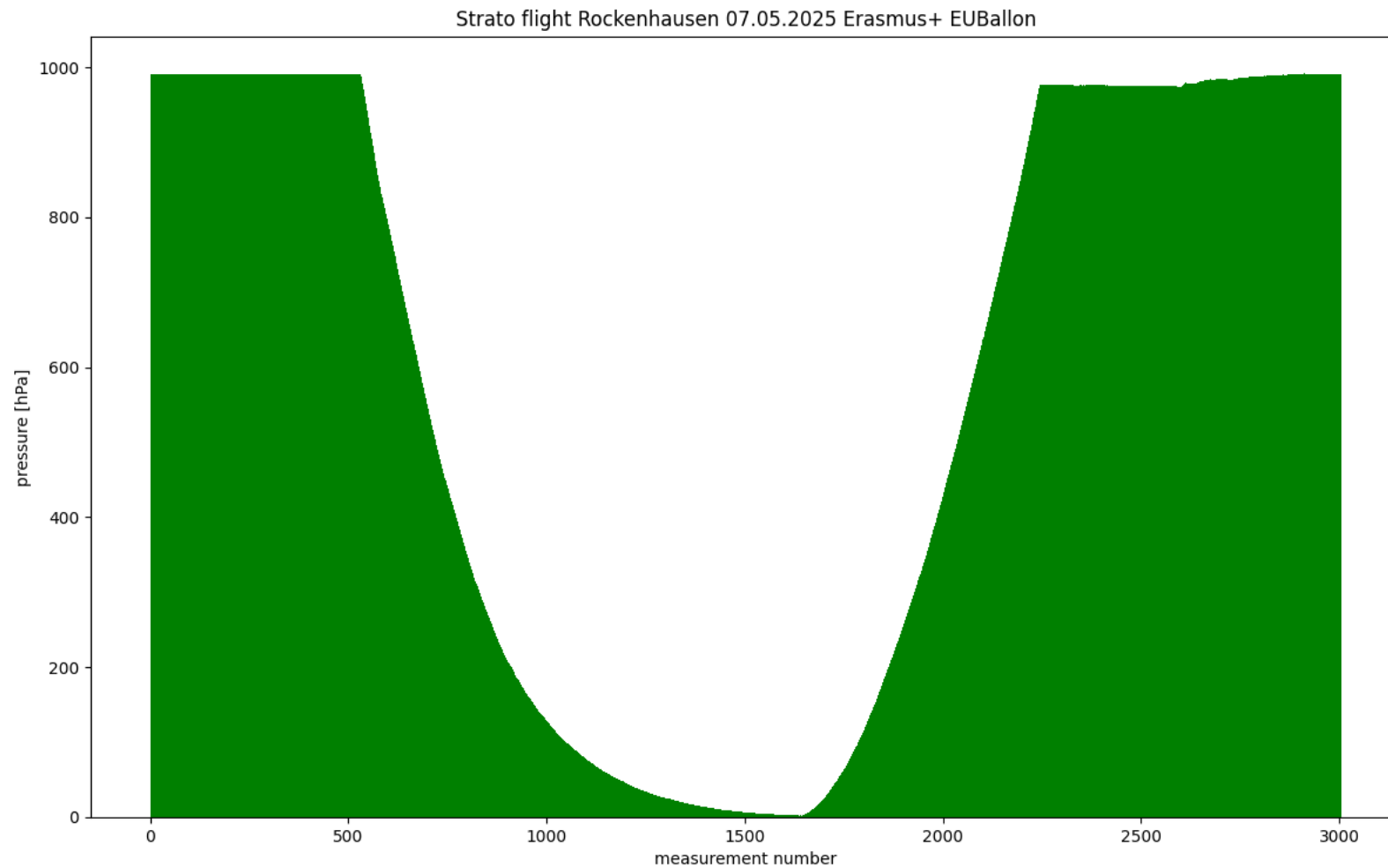
Python 3 script (All blank lines and comment lines starting with the "#" character can be omitted)

```
#library accepts only real numbers as data source
# data must also be represented in numeric form

import numpy as np
import matplotlib.pyplot as plt
# filename with results: presssuretemp.CSV
No,press=np.loadtxt('presssuretemp.CSV',unpack=True,usecols=(0,2),delimiter='\t')
print(No, press)
#Allocate space and determine the layout of the figure
fig, ax = plt.subplots()
ax.set_xlabel('measurement number')
ax.set_ylabel('pressure [hPa]')
plt.title('Strato flight Rockenhausen 07.05.2025 Erasmus+ EUBallon')
# plot bar graph
plot = plt.bar(No, press,width=1.0,color='green')
# Display chart
plt.show()
```

Intuitively, You can change the chart title, axis descriptions, and chart color.
We share this Python3 script with the name: **pressure_analysis.py**.
We assume that the CSV data file and the above script are in the same folder.

Chart Air pressure- png file generated by Python 3 **pressure_analysis.py** script with matplotlib library based on data from CSV file: **presssuretemp.CSV** recorded during flight.



2) B. Measuring the air temperature

Python 3 script (All blank lines and comment lines starting with the "#" character can be omitted)

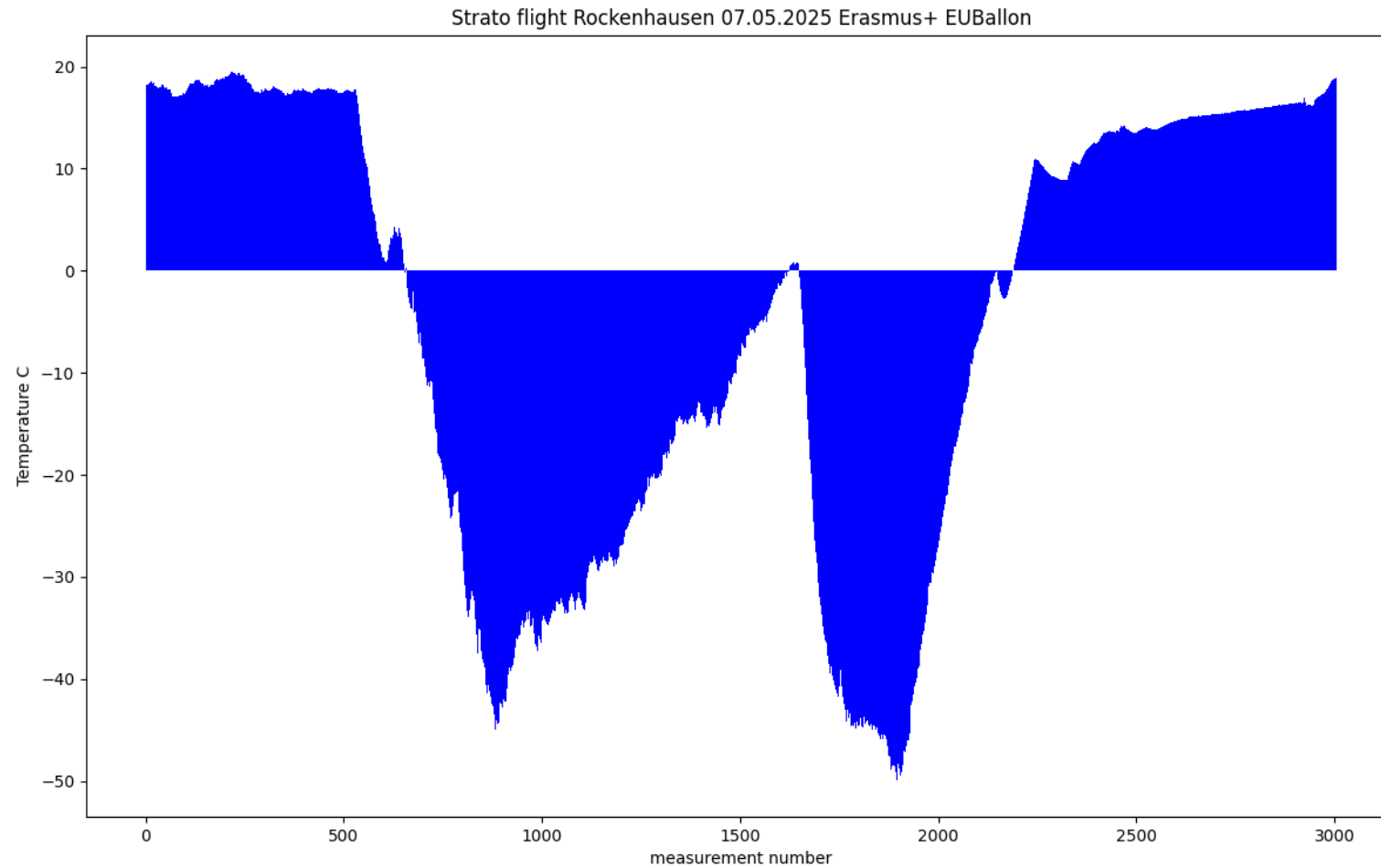
```
#library accepts only real numbers as data source
# data must also be represented in numeric form

import numpy as np
import matplotlib.pyplot as plt

# filename with results: presssuretemp.CSV
No,temp =np.loadtxt('presssuretemp.CSV', unpack =True,usecols = (0,1),delimiter ='\t')
print(No, temp)
#Allocate space and determine the layout of the figure
fig, ax = plt.subplots()
ax.set_xlabel('measurement number')
ax.set_ylabel('Temperature [C]')
plt.title('Strato flight Rockenhausen 07.05.2025 Erasmus+ EUBallon')
# plot bar graph
plot = plt.bar(No, temp,width=1.0,color='blue')
# Display chart
plt.show()
```

Intuitively, You can change the chart title, axis descriptions, and chart color.
We share this Python3 script with the name: **temp_analysis.py**.
We assume that the CSV data file and the above script are in the same folder.

Chart Air temperature- png file generated by Python 3 script **temp_analysis.py** with matplotlib library based on data from CSV file: **presssuretemp.CSV** recorded during flight.



2) C. Measuring the air humidity

Python 3 script (All blank lines and comment lines starting with the "#" character can be omitted)

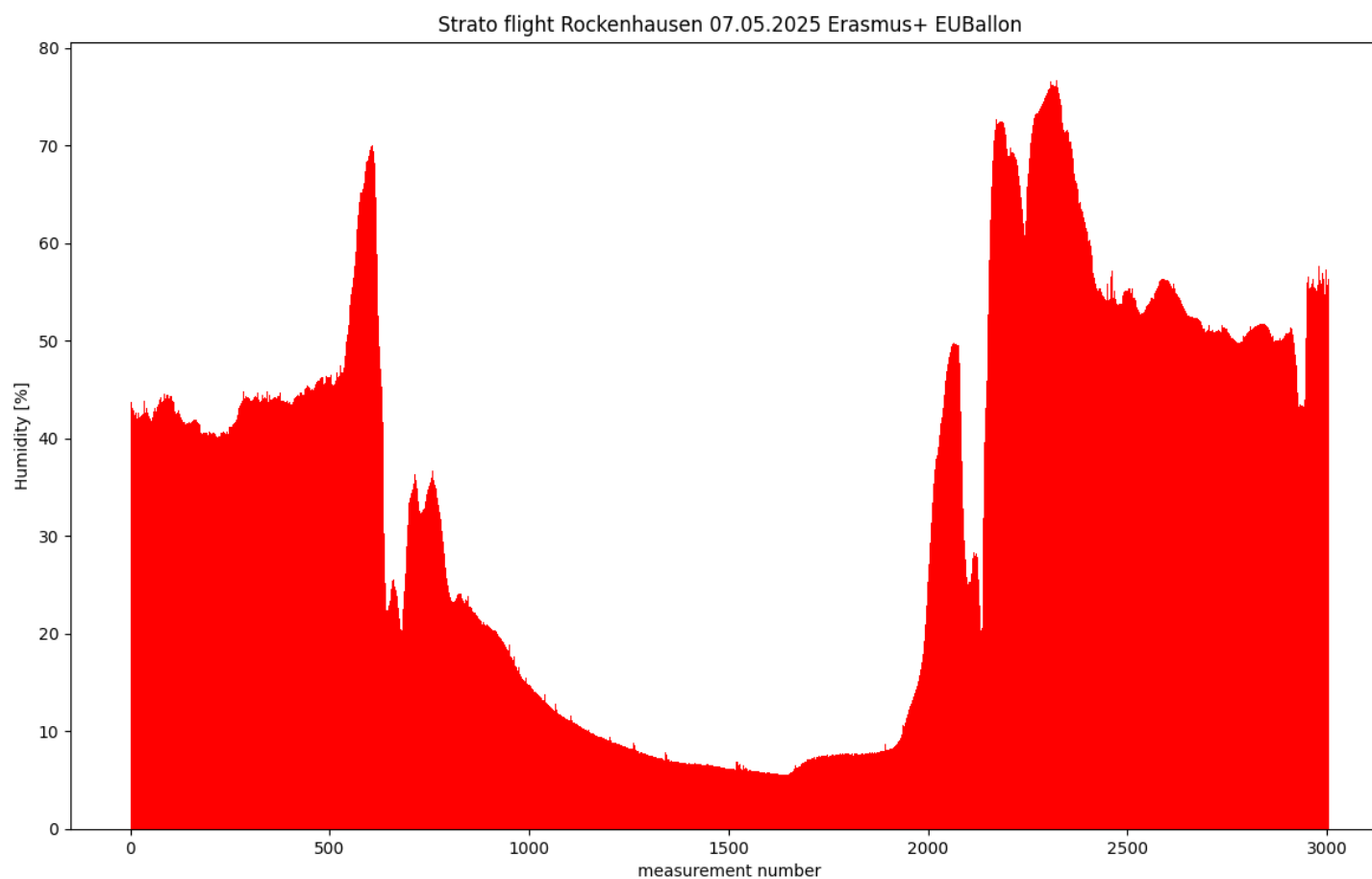
```
#library accepts only real numbers as data source
# data must also be represented in numeric form

import numpy as np
import matplotlib.pyplot as plt

# filename with results: presssuretemp.CSV
No,humid=np.loadtxt('presssuretemp.CSV',unpack=True, usecols=(0,3),delimiter ='\t')
print(No,humid)
#Allocate space and determine the layout of the figure
fig, ax = plt.subplots()
ax.set_xlabel('measurement number')
ax.set_ylabel('Humidity [%]')
plt.title('Strato flight Rockenhausen 07.05.2025 Erasmus+ EUBallon')
# plot bar graph
plot = plt.bar(No, humid,width=1.0,color='red')
# Display chart
plt.show()
```

Intuitively, You can change the chart title, axis descriptions, and chart color.
We share this Python3 script with the name: **humid_analysis.py**.
We assume that the CSV data file and the above script are in the same folder.

Chart Air humidity- png file generated by Python 3 script **humid_analysis.py** with matplotlib library based on data from CSV file: **presssuretemp.CSV** recorded during flight.



3) A. Measuring CO2

Python 3 script (All blank lines and comment lines starting with the "#" character can be omitted)

```
#library accepts only real numbers as data source
# data must also be represented in numeric form

import numpy as np
import matplotlib.pyplot as plt

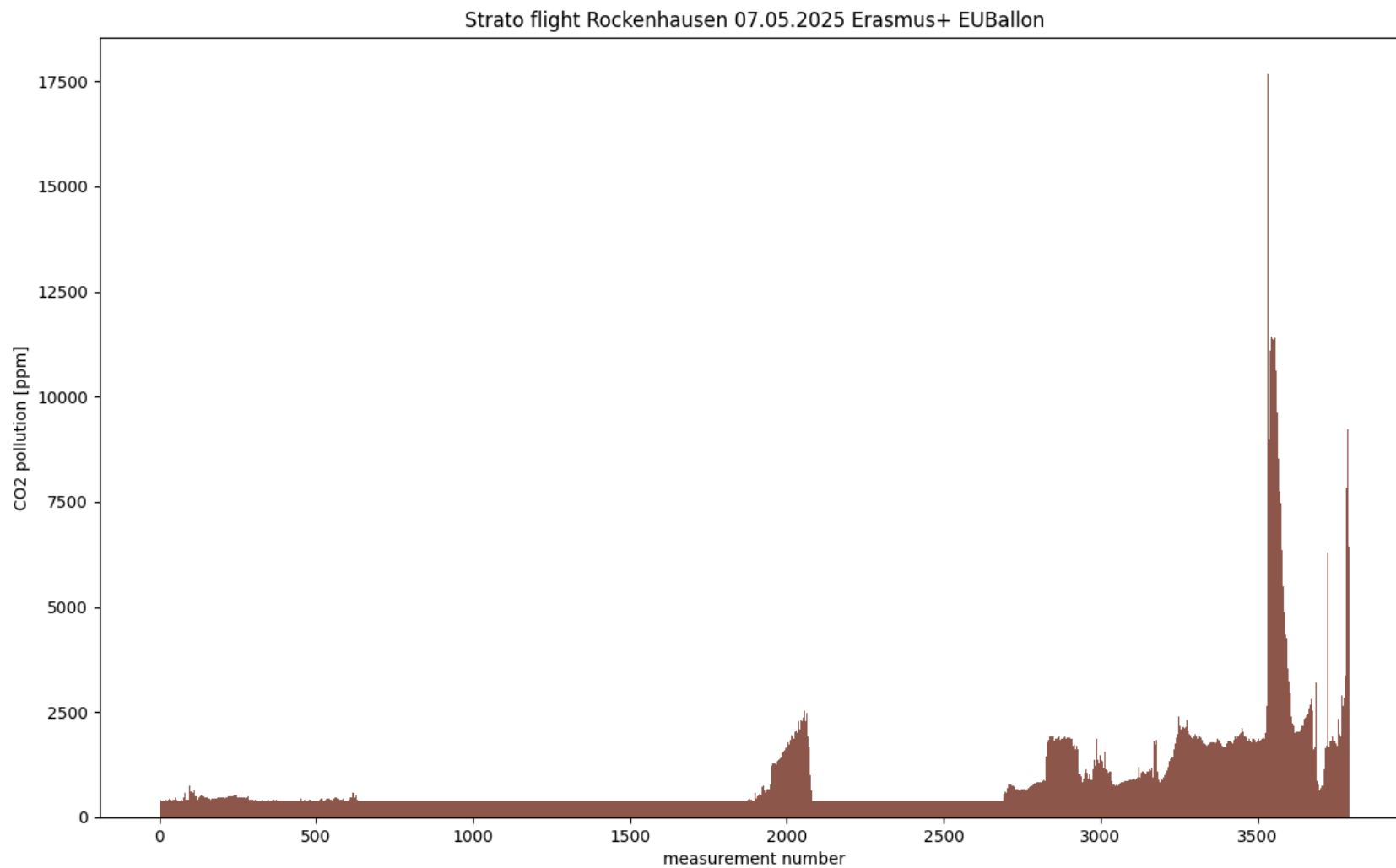
# filename with results: CO2TV0Cresult.CSV
No,co2 = np.loadtxt('CO2TV0Cresult.CSV',unpack =True,usecols=(0,1),delimiter ='\t')
print(No,co2)
#Allocate space and determine the layout of the figure
fig, ax = plt.subplots()
ax.set_xlabel('measurement number')
ax.set_ylabel('CO2 pollution [ppm]')
plt.title('Strato flight Rockenhausen 07.05.2025 Erasmus+ EUBallon')
# plot bar graph
plot = plt.bar(No, co2,width=1.0,color="tab:brown")
# Display chart
plt.show()
```

Intuitively, You can change the chart title, axis descriptions, and chart color.

We share this Python3 script with the name: **co2ppm.py**.

We assume that the CSV data file and the above script are in the same folder.

Chart CO2 pollution- png file generated by Python 3 script **co2ppm.py**. with matplotlib library based on data from CSV file: **CO2TVOCresult.CSV** recorded during flight.



3) B. Measuring volatile organic compounds TVOC

```
#library accepts only real numbers as data source
# data must also be represented in numeric form

import numpy as np
import matplotlib.pyplot as plt

# filename with results: C02TV0Cresult.CSV
No,tvoc=np.loadtxt('C02TV0Cresult.CSV',unpack=True,usecols=(0,2),delimiter='\t')
print(No, tvoc)
#Allocate space and determine the layout of the figure
fig, ax = plt.subplots()
ax.set_xlabel('measurement number')
ax.set_ylabel('organic pollution TVOC [ppb]')
plt.title('Strato flight Rockenhausen 07.05.2025 Erasmus+ EUBallon')
# plot bar graph
plot = plt.bar(No, tvoc,width=1.0,color="tab:olive")
# Display chart
plt.show()
```

Intuitively, You can change the chart title, axis descriptions, and chart color.
We share this Python3 script with the name: **tvocppb.py**
We assume that the CSV data file and the above script are in the same folder.

Chart CO2 pollution- png file generated by Python 3 script **tvocppb.py**. with matplotlib library based on data from CSV file: **CO2TVOCresult.CSV** recorded during flight.

