IL NUOVO CIMENTO **45 C** (2022) 229
DOI 10.1393/ncc/i2022-22229-x

Communications: SIF Congress 2021

The potential of using Jupyter Notebook in physics education: Experimentation for high school students

C. Sutrini(1), D. Passaro(2) and F. Pallotta(3)

(1) *Università di Pavia, Dipartimento di Fisica - Via Bassi 6, 27100 Pavia, Italy*
(2) *Sapienza Università di Roma, Dipartimento di Scienze Statistiche - Viale dell'Università 6, 00185 Roma, Italy*
(3) *Università dell'Insubria, Dipartimento di Scienze ed Alta tecnologia - Via Valleggio 11, 22100 Como, Italy*

received 31 January 2022

Summary. — The diffusion of Covid-19 has attracted the attention of the student population to the possibility of describing the spread of the epidemic by means of suitable mathematical models. Thanks to the data made available by the Civil Protection, an interdisciplinary learning activity was created and concretely proposed to students in two Italian secondary schools in Rome and Pavia between April and May 2020. We present the work about the possibility of introducing the use of Jupyter Notebook with Python more generally in physics teaching.

1. – Introduction

The COVID-19 global health crisis has unfortunately brought to light the need for creating educational contexts that foster scientific literacy competences [1]. The knowledge and skills needed to understand the nature of complex phenomena such as the development of a pandemic and its social and economic effects must therefore become part of the educational learning sequences that are carried out in schools [2].

The general aim of the project we are presenting is to support the scientific competences of secondary school students for the analysis of complex phenomena. The strategies are related to the acquisition of knowledge and skills in the field of modelling and the use of IT tools in an interdisciplinary context. The actions implemented concern the use of specific tools, such as the use of Python through Jupyter Notebook. This resulted in a self-consistent distance learning teaching sequence of 10 hours for secondary school students designed for the above purposes. This sequence, created to deal with the COVID-19 global crisis, provided students with analytical tools that could potentially be useful in addressing issues of interest to them with a scientific approach.

2. – Jupyter Notebook with Python

Jupyter [3] is a web-based interactive development environment for notebooks, code, and data. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. It was born in scientific fields to allow users to

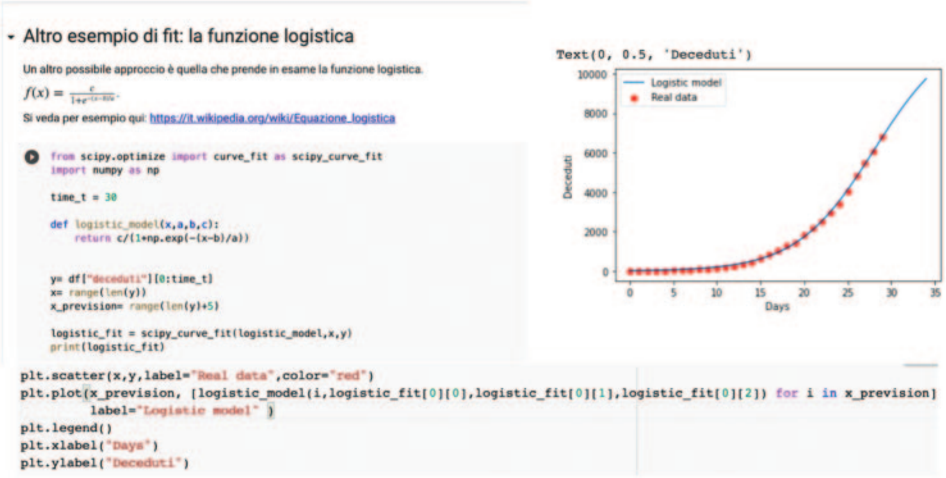
Table I. – *Structure of the sequence.*

Lesson	Contents	Learning goal
1. - 2. Introduction to Python with Jupyter Notebook (3 hours)	Values and operators; Variables; Functions; Strings and lists; "If" and "For"; Modules; Libraries	Learn the first useful elements of programming in Python by writing short codes
2. Introduction to Matplotlib (1 hour)	Graphics; Style, lines, colours; Title, axes, legend	Learn first element of Matplotlib
3. Matplotlib and CSV files (2 hours)	"Protezione Civile" data representation	Import CSV files on several topics and build meaningful graphs
4. Data Fit analysis (2 hours)	Linear data analysis and chi-square method in the context of Hooke's law	Learning about linear data analysis with Python and a Jupyter Notebook
5. Epidemiological models (2 hours)	Exponential, SI, SIS and SIR model	Understanding the SIR model by successive approximations

configure and share workflows (in particular in data science, scientific computing and machine learning) [4]. However, these characteristics are suitable for use also in the educational context. In fact, by using Jupyter Notebook it is possible to insert in the same place texts (including formulas written in LaTeX), live code and data.

With Jupyter Notebook, teachers and students can use a single environment instead of multiple software (word processing software for text edit, spreadsheet for storing and analyzing data, integrated development environment for code editing). A notebook can easily be shared with students and allows modifying the obtained results (*e.g.*, plots) by means of live code. For its simplicity, Python is currently considered an excellent entry level coding language [5]. It is also a natural choice considering that Jupyter Notebook is a spin-off project from the IPython project. Python offers high-level libraries for scientific computing (Numpy, Scipy), for graphing (Matplotlib) and data management (Pandas, which manages data by setting them up as dataframes). Both Jupyter and Python are free, open source and cross platforms and can be easily installed (in particular using Anaconda, a distribution of the Python and R programming languages for scientific computing aimed to simplify package management and deployment). Furthermore, if a teacher prefers to avoid installation, it is possible to use cloud solutions. Probably the most used is Google Colaboratory (also known as Colab), a free Jupyter Notebook, environment that runs in the cloud and stores its notebooks on Google Drive with many pre-installed data science libraries. Another well-known option is the Kaggle notebook, a product with similar functionalities to Colab. Like Colab, Kaggle provides free browser-based Jupyter Notebook.

In a notebook a user can essentially create two types of cells: cells with code and cells with text. Text cells are formatted using a simple markup language called Markdown that allows many possibilities including LaTeX. In a code cell, users can enter their own code and click "run cell" to execute it. The output, if there is one, will be shown directly below the cell. A tool like Google Colab has the advantage that it does not require to be



Altro esempio di fit: la funzione logistica

Un altro possibile approccio è quella che prende in esame la funzione logistica.

$f(x) = \frac{c}{1+e^{-(x-b)/a}}$.

Si veda per esempio qui: https://it.wikipedia.org/wiki/Equazione_logistica

```
from scipy.optimize import curve_fit as scipy_curve_fit
import numpy as np

time_t = 30

def logistic_model(x,a,b,c):
    return c/(1+np.exp(-(x-b)/a))


y= df["deceduti"][0:time_t]
x= range(len(y))
x_prevision= range(len(y)+5)

logistic_fit = scipy_curve_fit(logistic_model,x,y)
print(logistic_fit)
```

```
Text(0, 0.5, 'Deceduti')
```

```
plt.scatter(x,y,label="Real data",color="red")
plt.plot(x_prevision, [logistic_model(i,logistic_fit[0][0],logistic_fit[0][1],logistic_fit[0][2]) for i in x_prevision]
        label="Logistic model" )
plt.legend()
plt.xlabel("Days")
plt.ylabel("Deceduti")
```

Fig. 1. – Logistic model.

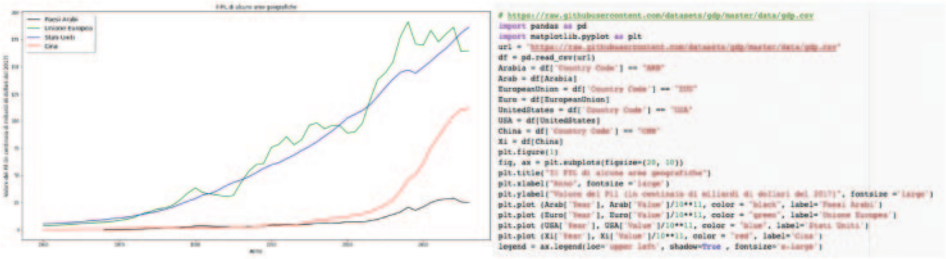
installed and can be used on all devices (including smartphones). It also allows creating a single environment shared (Drive folder) between students and teachers, so they can both edit the same notebook. For this characteristics we think that a joint use of Python and Jupyter offers great teaching potential, as shown in the next paragraph.

3. – Teaching experiment

3˙1. *Structure of the teaching sequence.* – The teaching sequence is divided into five parts (see table I): in the first two lessons, students learn the first elements of the Python language by using Jupyter Notebook. In the third lesson they develop some skills on the Mathplotlib library to import data to CSV files and represent them by means of graphs. In the fourth lesson the notions learned in the first three meetings are used to discuss data fits (in particular linear fits) through notebooks. In this way, students understand how this environment is able to contain the key steps of their traditional lab reports: importing images and videos, writing formulas, analysing data, writing text. In the last meeting they discuss together mathematical models for epidemiology and the codes to represent them (for example, see fig. 1). The lesson introduces the models in succession so that students can understand the limitations of each model and identify improvements.

3˙2. *Methods.* – The use of distance learning made it necessary to adopt an active approach. For this reason, the notebooks used [1] were organised so that the teacher (one of the authors) could explain theory using lines of text, could shows the codes, implement them and then give the students the opportunity to solve simple exercises and comment on them together. Jupyter Notebook had the merit of being a suitable environment for constructing these shared and interactive lessons.

[1] It is possibile find some of the notebooks used at `https://tinyurl.com/pdards25`.



```
# https://raw.githubusercontent.com/datasets/gdp/master/data/gdp.csv
import pandas as pd
import matplotlib.pyplot as plt
url = "https://raw.githubusercontent.com/datasets/gdp/master/data/gdp.csv"
df = pd.read_csv(url)
Arabia = df['Country Code'] == "ARB"
Arab = df[Arabia]
EuropeanUnion = df['Country Code'] == "EUU"
Euro = df[EuropeanUnion]
UnitedStates = df['Country Code'] == "USA"
USA = df[UnitedStates]
China = df['Country Code'] == "CHN"
Xi = df[China]
plt.figure(1)
fiq, ax = plt.subplots(figsize=(20, 10))
plt.title("Il PIL di alcune aree geografiche")
plt.xlabel("Anno", fontsize ='large')
plt.ylabel("Valore del Pil (in centinaia di miliardi di dollari del 2017)", fontsize ='large')
plt.plot (Arab['Year'], Arab['Value']/10**11, color = "black", label='Paesi Arabi')
plt.plot (Euro['Year'], Euro['Value']/10**11, color = "green", label='Unione Europea')
plt.plot (USA['Year'], USA['Value']/10**11, color = "blue", label='Stati Uniti')
plt.plot (Xi['Year'], Xi['Value']/10**11, color = "red", label='Cina')
legend = ax.legend(loc='upper left', shadow=True , fontsize='x-large')
```

Fig. 2. – GDP of some countries since 1960.

3˙3. *General aims*. – Starting from a rigorous discussion of technical and scientific aspects of current problems, such as the pandemic, the general objective was to provide students with tools suitable for making data analyses. Using data relevant to them, we wanted to emphasise the importance of the concept of a mathematical model and the process of building one. The exercises collected during the lessons show that some students produced relevant codes and graphs (fig. 2).

3˙4. *Context*. – The experimentation involved around 20 students of the fourth and fifth year of "Liceo Scientifico" and "Liceo Classico" Taramelli-Foscolo in Pavia (the complete sequence of 5 meetings) and a third Mathematics class at Liceo B. Russell in Rome (in which the introductory part on the Python language and the creation of graphs had already been done in the previous two years). The duration was about 10 hours for the complete sequence between April and May 2020.

4. – Conclusions

The work presented aims to create learning environments to support students in developing modelling skills through appropriate IT tools. The specific course related to the Covid-19 pandemic was an opportunity for a more general reflection on the scientific skills needed to analyse complex phenomena in an interdisciplinary context. The active participation of the students despite the difficulties due to distance learning and the comments made at the end of the course seem to promote the development of this approach.

REFERENCES

[1] Europe Council, *Council recommendations on key competences for lifelong learning*, available at https://ec.europa.eu/.
[2] UNESCO 2020, *Education in a post-COVID world: Nine ideas for public action*
[3] Jupyter Project (2019), Jupyer notebooks, Available at: `https://jupyter.org`.
[4] Rowe F., Maier G., Arribas-Bel D. and Rey S., *REGION*, **7** (2020) E1.
[5] Dierbach C., *J. Comput. Sci. Coll.*, **29** (2014) 153.