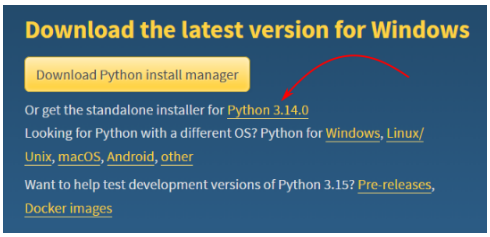
BSB - M1 Data Science : practicle

Setup your environment

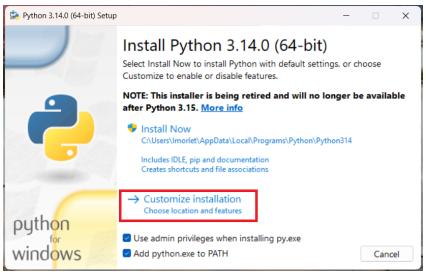
Before starting to use pandas, you will need to install Python on your computer, setup your Visual Studio Code to run it, and then download the required libraries.

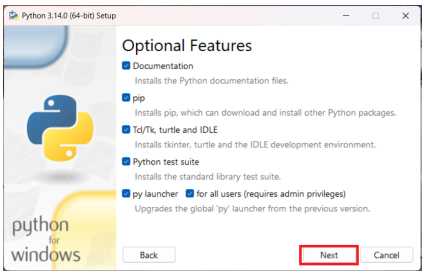
Install Python

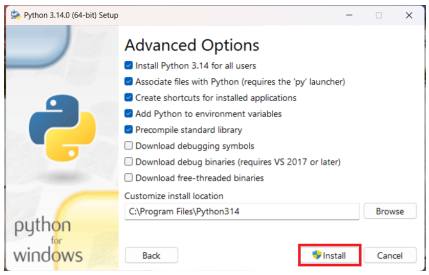
First of all, you will need to download the latest version of Python on your computer. Go to this URL : `https://www.python.org/downloads/` and click on the "standalone installer"



Double-click on the installer you have downloaded. On the following images, tick the checkboxes and then click on the button in the red rectangle





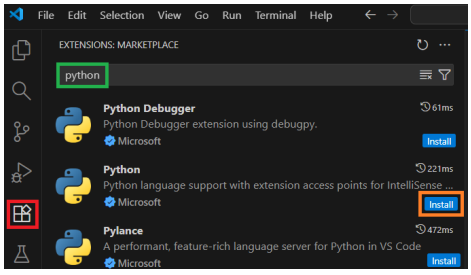


Install Visual Studio Code (VS Code)

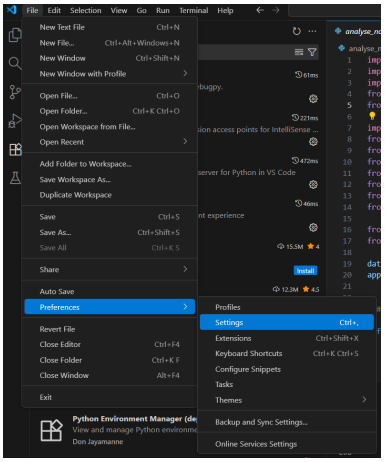
Once your Python interpreter is installed, you can go for the Visual Studio Code installer : `https://code.visualstudio.com/download` No specific instructions for it, just follow the recommended installation.

Link Visual Studio Code to Python

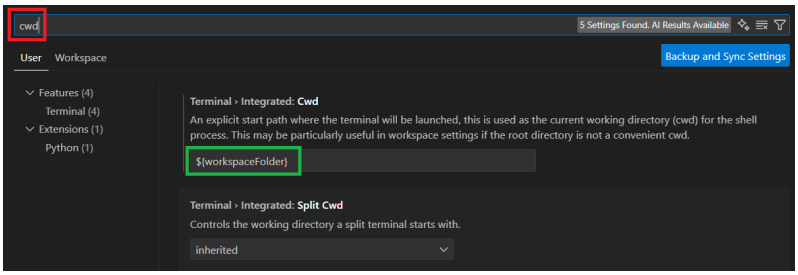
Run your VS Code. On the left panel, click on "Extension" (red rectangle in the following image). Then type "Python" in the search bar (green rectangle). To finish, click on "Install" (orange rectangle) for the Python extension proposed by Microsoft.



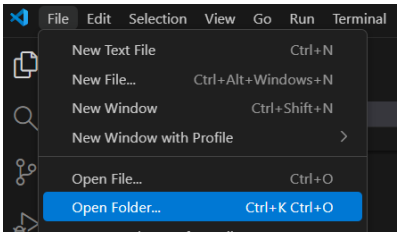
Open the ”Settings” by following the path presented in the following image



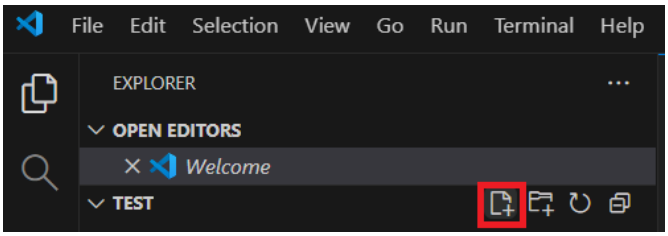
In the settings search bar (red rectangle in the following image), type ”cwd”. Then fill the ”Terminal integrated : Cwd” (green rectangle) with the value **${workspaceFolder}**



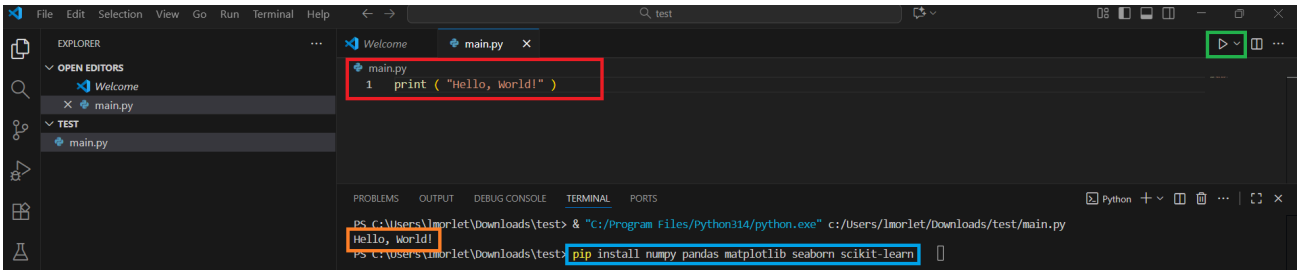
Open the folder where you want to store your script and data by following the path presented in the following image



In the file explorer, click on ”New file” (red rectangle in the following image) and name it ”main.py”



In the text editor (red rectangle in the following image), type **print (”Hello, World!”)**, then click on ”Run” (green rectangle). Your script will be executed and the result will be displayed inside the terminal (orange rectangle). In this terminal, type **pip install numpy pandas matplotlib seaborn scikit-learn** (cyan rectangle) to install all the required libraries.



Your VS Code is now ready!

Session nº 1 : First steps in Pandas

a) Start by downloading the CSV file ”burgundy_2023.csv” and place it in your workspace folder that you have opened in the previous part.

b) At the very beginning of your Python script, import the required libraries with :

```
import pandas as pd
import matplotlib.pyplot as plt
```

c) Then read the CSV and display its basic info

```
df = pd.read_csv('burgundy_2023.csv', sep=';')
print("Number of rows:", len(df))
print("Number of columns:", len(df.columns))
print("Column names:", df.columns)
```

d) How many rows and columns are in your CSV file ?

e) There are too many columns in this CSV, we need to check which ones are truly useful. Display the detailed info to find which columns are empty

```
print("Infos of dataframe")
df.info( verbose=True, show_counts=True)
```

f) Extract the useful columns

```
useful_cols = [ "NUM_POSTE", "NOM_USUEL", "LAT", "LON", "ALTI", "AAAAMMJJHH", "T" ]
df = df[useful_cols]
df.info(True, show_counts=True)
```

g) There are still missing values in your dataframe. To avoid this, you will delete every row that contains at least one missing value

```
df = df.dropna()
df = df.reset_index(drop=True)
df.info( verbose=True, show_counts=True )
```

h) Once your dataframe is ”clean”, let’s check what is inside of it. Display the cities where the data comes from. Why do we use the ”unique” function ?

```
print ( df["NOM_USUEL"].unique() )
```

i) By modifying the following code, create a sub-dataframe for each city.

```
dijon = df.loc[df["NOM_USUEL"] == "DIJON"]
dijon = dijon.reset_index(drop=True)
```

j) By modifying the following code, display the most common statistics about the temperature (”T”) of the studied cities.

```
print("Dijon temperature:")
print( dijon["T"].describe() )
```

k) What can you say about the statistics of the temperature of studied cities ?

l) Find a way to display what is inside the ”AAAAMMJJHH” columns. Can you identify it ?

m) You will need to convert it to a suitable format before using it. Use the following code to do it.

```
dijon["AAAAMMJJHH"] = pd.to_datetime(dijon["AAAAMMJJHH"], format="%Y%m%d%H")
```

n) Look at the format parameter, can you find what signify the value here ?

o) Plot the evolution of the temperature of Dijon during 2023

```
# Plot temperature over time for each city
plt.plot( dijon["AAAAMMJJHH"], dijon["T"], label="Dijon")
plt.xlabel("Date")
plt.ylabel("Temperature (°C)")
plt.title("Temperature over Time")
plt.legend()
plt.show()
```

p) Modify your code to display the curve of every city in the same figure