DATA203 Foundational Python (Prof. Maull) / Spring 2025 / HW3

Points Possible	Due Date	Time Commitment (estimated)
25	Wednesday, April 30	*up to* 10 hours

- **GRADING:** Grading will be aligned with the completeness of the objectives.
- **INDEPENDENT WORK:** Copying, cheating, plagiarism and academic dishonesty *are not tolerated* by University or course policy. Please see the syllabus for the full departmental and University statement on the academic code of honor.

OBJECTIVES

- Practice using Pandas for data extraction.
- Practice using Pandas for data engineering.
- Practice using Pandas for data analysis.

WHAT TO TURN IN

You are being encouraged to turn the assignment in using the provided Jupyter Notebook. To do so, make a directory in your Lab environment called `homework/hw1`. Put all of your files in that directory. Then zip or tar that directory, rename it with your name as the first part of the filename (e.g. `maull_hw1_files.zip`, `maull_hw1_files.tar.gz`), then download it to your local machine, then upload the `.zip` to Canvas.

If you do not know how to do this, please ask, or visit one of the many tutorials out there on the basics of using zip in Linux.

If you choose not to use the provided notebook, you will still need to turn in a `.ipynb` Jupyter Notebook and corresponding files according to the instructions in this homework.

ASSIGNMENT TASKS

(30%) Practice using Pandas for data extraction.

In class we talked about the project to develop a platform to collect and store data for air quality measurements. I showed a system that took sensor data and displayed the data onto a dashboard (see https://cisl-chords.cloud.ucar.edu/dashboard). The data of most interest are the sensors labeled "Core Futures Labs / HU Spring 2025", which are sensors deployed by Dr. A's class.

Each system (we will call them "stations") is labeled "CFLXXX" and on each stations are several sensors in two categories "Air Quality" and "Atmospheric". In each of these categories we can have multiple sensors. In the case of these experiments being done in Dr. A's class, the diagram below shows the structure of a station:

```
Station
|--- Air Quality
|     |
|     |-- PM25
|
|--- Atmospheric
|     |---- Temperature
|     |
|     |-- Humidity
```

If you want to learn more about that entire platform, see the openIoTwx main page.

This assignment is 3 parts: data extraction, data filtration, data analysis.

To complete this assignment, you will need to use Pandas. Here are some key areas to bookmark:

- Pandas documentation
- Pandas in 10 minutes

IMPORTANT NOTE:

There are three stations we will be analyzing data and on 3 dates:

1. `CFL006` → dashboard link
2. `CFL007` → dashboard link
3. `CFL010` → dashboard link

You will not be analyzing all of these, but instead, you will analyze based on your last name and the table below:

Station ID	Last Name	Analysis Days	Data repo link
CFL006	A-H	4/16, 4/17, 4/18	spring25_data/src/branch/main/data/cfl006
CFL007	I-P	4/13, 4/14, 4/15	spring25_data/src/branch/main/data/cfl007
CFL010	Q-Z	4/14, 4/15, 4/16	spring25_data/src/branch/main/data/cfl010

Remember from lecture, you will need the **raw** link which would look something like this:

`https://.../spring25_data/raw/branch/main/data/cfl001/20250415_cfl001.csv`

§ Task: Retrieve data from a repository using Pandas.

In class lecture, we talked about using URLs to call the Dataframe function `pandas.read_csv()`.

You will use that function and place data into a DataFrame for each day in your station for you last name group listed above.

§ Task: Combine multiple files into one large Dataframe.

From the first task, use the Pandas [`pandas.concat()`](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.concat.html) to combine all files into a single Dataframe.

Make sure they are in the order of their date retrieval.

§ Task: Store the combined data in a single CSV file.

With a single Dataframe in hand, store that into a a file named `"cflXXX_combined_data.csv"`.

Use the `pandas.DataFrame.to_csv` function.

(40%) Practice using Pandas for data engineering.

The next part will be data retrieval and organizing. You might notice the data in the last column has a large number. It's a UNIX timestamp, representing seconds since a 1/1/1970. For an interesting rabbit hole on why that date was selected, see this lively discussion: [stackoverflow.com]: *Why is 1/1/1970 the "epoch time"?*

You may also notice that there is redundant and unnecessary data.

We'll use Pandas to clean all of this up. You'll see clues in the notebook on how to do it, but the template of what we're going to do is:

- convert the UNIX timestamp to a date/time
- remove unncessary columns
- set the index to the date/time
- filter the data to a single sensor measurement type (PM25 ENV)
- store the data in unfiltered and filtered form as CSV

To perform that task, you will need to study the example in the starter notebook:

- hw3_starter.ipynb

Now that we have an index for all values, filter data. Eliminate one column with identical values - `"sensor ID"`. It's the same value for every data point, so no need to keep it.

Use Pandas to drop this column. Then filter further. Only include PM25 ENV data points in your final file.

Store two files:

1. All data filtered with timestamp index
2. Filtered data with just`"PM25 ENV"`

§ Task: Convert UNIX time to a time-date string.

Study:

- `pandas.to_datetime()`, don't over think it!

§ Task: Eliminate unneeded columns.

Study:

- `pandas.DataFrame.drop()` → use the `axis=1` parameter
- the unneeded columns are the first and second

§ Task: Filter the remaining data to only include PM25 ENV data points.

- Study `pandas.Dataframe.query()` and see the examples in the notebook
- store the filtered Dataframe into a new one called `df_filtered`

§ Task: Store the data in a new CSV files.

- study the `pandas.DataFrame.to_csv()` function
- name the cleaned up complete file: `cflXXX_clean.csv`
- name the filtered file: `cflXXX_filtered.csv`

(30%) Practice using Pandas for data analysis.

The third part of the assignment will be taking the data that we have from the PM25 ENV data only, and we are going to answer three questions.

Recall from lecture, our discussion about descriptive statistics. Review:

- `pandas.Dataframe.describe()`

Of course we want to understand the exposure to PM2.5 and also understand the dynamics and variation within a day, and between days. These are the *basic* questions you must tune yourself to ask as a data scientist.

Remeber your greatest skill will be to *ask relevant and piercing questions about data* … your toolkit can be filled with technical skills **as** you hone your inquiry skills.

§ Task: Find the mean, median, minimum and maximum PM25 ENV values.

Study:

- `pandas.Dataframe.describe()`

§ Task: Identify when the maximum and minimum PM25 values occurred for each day separately.

Study:

- `pandas.Dataframe.groupby()`: you will need to group by day (see the example in the starter NB)
- notice how easy it is to use the datetime index with `groupby`

§ Task: Plot the PM25 data in a line plot.

Study:

- `pandas.Dataframe.plot.line()`

§ Task: Make a statement about your plot, noting the pattern you see. Keep your statement to less than 3 sentences.