

## Worksheet #5

---

*files*

---

1. Using the `many_values.txt` file :
  - 1.1. Calculate <sup>1</sup>the mean, median, and standard deviation of file values
  - 1.2. Calculate the mean and median of each row
    - 1.2.1. Write the result in a new file (each line shows the two values)
  - 1.3. Identify the line that originates the highest mean<sup>2</sup>
2. Using the file `many_values_commentaries.txt` (with comment lines – starting with `#` - that must be ignored)<sup>3</sup>:
  - 2.1. Calculate the mean, median, and standard deviation of file values
  - 2.2. Calculate the mean and median of each row
    - 2.2.1. Write the result in a new file (each line shows the two values)
  - 2.3. Identify the line that originates the highest mean
3. Considering the file `elements.txt`
  - 3.1. Structure the file data into a list of dictionaries and print the result

```
[{'name': 'Helio', 'atomicnr': 2, 'density': 0.1786},
{'name': 'Neon', 'atomicnr': 10, 'density': 0.9002},
{'name': 'Argon', 'atomicnr': 18, 'density': 1.7818},
{'name': 'Krypton', 'atomicnr': 36, 'density': 3,708},
{'name': 'Xenon', 'atomicnr': 54, 'density': 5,851},
{'name': 'Radonium', 'atomic nr': 86, 'density': 9.97}]
```
  - 3.2. Using the list created in 3.1:
    - 3.2.1. Print the name of all elements
    - 3.2.2. Calculate the average of the densities
    - 3.2.3. Identify the element with the highest atomic number
4. The Lusíadas <sup>4</sup>are composed of 10 songs, each with a variable number of octets (stanzas with 8 verses), which are preceded by a number. Using the `lusiadas.txt` file :

---

<sup>1</sup> Use the `statistics` module : <https://docs.python.org/3/library/statistics.html>

<sup>2</sup> You can use the `index` function to find out which index corresponds to a given value: [https://www.w3schools.com/python/ref\\_list\\_index.asp](https://www.w3schools.com/python/ref_list_index.asp). Other list functions in Python: [https://www.w3schools.com/python/python\\_ref\\_list.asp](https://www.w3schools.com/python/python_ref_list.asp)

<sup>3</sup> It is possible to use the `startswith` function to find out if a string starts with a certain sequence of characters: [https://www.w3schools.com/python/ref\\_string\\_startswith.asp](https://www.w3schools.com/python/ref_string_startswith.asp)

<sup>4</sup> Other curiosities about Os Lusíadas: <https://notapositiva.com/os-lusiadas-estrutura-interna-e-externa/>

- 4.1. Create stanza list <sup>5</sup>(size: 1102 stanzas)
- 4.2. Based on the created list, determine:
  - 4.2.1.How many verses does Os Lusíadas have? (R: 8816)
  - 4.2.2.How many words does Os Lusíadas have? (R: 55232)
  - 4.2.3.What is the average number of words per stanza? (R: 50.12)
  - 4.2.4.How many <sup>6</sup>times does the word “King” appear? (R: 149)
- 4.3. A class is analyzing Os Lusíadas in a group work. Each of the 5 groups in the class will have to analyze, at random, one of the stanzas. Develop the necessary code to make this task automatic: generate 5 files, each with a random stanza <sup>7</sup>to be later analyzed by the group.
  - 4.3.1.Automate the task further: allow the number of groups/students to be defined ( `input` ) by the user
5. A school needs a system to store its students and subjects, as well as the subjects students are enrolled in and grades obtained. It is intended to implement in Python and using object-oriented programming:
  - A **discipline** is characterized by a code (integer, sequential) and a name
  - A **student** is characterized by a student number (integer, sequential) and a name
  - An **enrollment** is characterized by a student, a course, and a list of grades obtained in the course. When creating the application, the list of notes is an empty list and the system must have a method that allows adding a note to an application
  - 5.1. Create the necessary classes for the implementation
  - 5.2. Create a main.py file, to test the code, with the following features:
    - 5.2.1.Create a dictionary with 2 subjects. Save course data in an XML file <sup>8</sup>. The dictionary must have the following format:
 

```
{"data": {"disciplines": [...disciplinas list...]}}
```
    - 5.2.2.Create a list of 10 students. Save student data in a JSON file <sup>9</sup>. The dictionary must have the following format:
 

```
{"students":liststudents}
```
    - 5.2.3.Create a list with 15 enrollments (it is not mandatory for all students to be enrolled in all disciplines).

<sup>5</sup> Hint: A stanza will be a list of lines. Read the entire file and create a list of lists with the stanzas. Use

- `isnumeric` to check if the string content is a numeric value  
[https://www.w3schools.com/python/ref\\_string\\_isnumeric.asp](https://www.w3schools.com/python/ref_string_isnumeric.asp)
- `enumerate` to help the verse reading cycle <https://realpython.com/python-enumerate/>

<sup>6</sup> Suggestion: use `count` <https://www.guru99.com/python-list-count.html>

<sup>7</sup> Generate random numbers in Python: [https://www.w3schools.com/python/ref\\_random\\_randint.asp](https://www.w3schools.com/python/ref_random_randint.asp)

<sup>8</sup> Use <https://jsonformatter.org/xml-parser> to check XML structure

<sup>9</sup> Use <https://jsonformatter.org/> to check JSON structure

5.2.4. For each entry, randomly create 3 values (between 0 and 20) for the grades and insert them into the entry.

5.2.5. Save the registration list in a CSV file with the following format:

```
nrStudent, studentName, subjectCod, subjectName, grade1, grade2, grade3
```

5.3. Read the CSV file where the registrations were saved and, based on the data obtained, calculate the final grade for each student in each subject, knowing that the evaluation moments are worth 20%, 30% and 50%, regardless of the subject

5.4. Create a folder `NotesDisciplinas` and save the results in CSV files inside that folder, one per discipline, in the format:

```
nrStudent, final grade
```

5.5. Read existing files in the folder<sup>10</sup> `gradesSubjects` and, for each subject, determine:

5.5.1. Number of positive and negative notes

5.5.2. Course average

5.5.3. Student with best and worst grade

<sup>10</sup> Consult [https://www.tutorialspoint.com/python/os\\_listdir.htm](https://www.tutorialspoint.com/python/os_listdir.htm) to learn how to list the files in a folder, so that they can be processed later