

Ficha de trabalho #5

Ficheiros

1. Utilizando o ficheiro `muitos_valores.txt`:
 - 1.1. Calcular¹ a média, a mediana e o desvio padrão dos valores do ficheiro
 - 1.2. Calcular a média e a mediana de cada linha
 - 1.2.1. Escrever o resultado num novo ficheiro (cada linha apresenta os dois valores)
 - 1.3. Identificar a linha que origina a média mais alta²

2. Utilizando o ficheiro `muitos_valores_comentarios.txt` (com linhas de comentário – começam com # - que devem ser ignoradas)³:
 - 2.1. Calcular a média, a mediana e o desvio padrão dos valores do ficheiro
 - 2.2. Calcular a média e a mediana de cada linha
 - 2.2.1. Escrever o resultado num novo ficheiro (cada linha apresenta os dois valores)
 - 2.3. Identificar a linha que origina a média mais alta

3. Considerando o ficheiro `elementos.txt`
 - 3.1. Estruturar os dados do ficheiro numa lista de dicionários e imprimir o resultado


```
[{'nome': 'Helio', 'nrAtomico': 2, 'densidade': 0.1786},
 {'nome': 'Neon', 'nrAtomico': 10, 'densidade': 0.9002},
 {'nome': 'Argon', 'nrAtomico': 18, 'densidade': 1.7818},
 {'nome': 'Cripton', 'nrAtomico': 36, 'densidade': 3.708},
 {'nome': 'Xenon', 'nrAtomico': 54, 'densidade': 5.851},
 {'nome': 'Radonio', 'nrAtomico': 86, 'densidade': 9.97}]
```
 - 3.2. Usando a lista criada em 3.1:
 - 3.2.1. Imprimir o nome de todos os elementos
 - 3.2.2. Calcular a média das densidades
 - 3.2.3. Identificar o elemento com o número atómico mais alto

¹ Usar o módulo `statistics`: <https://docs.python.org/3/library/statistics.html>

² É possível utilizar a função `index` para saber que índice corresponde a um determinado valor: https://www.w3schools.com/python/ref_list_index.asp. Outras funções de listas em Python: https://www.w3schools.com/python/python_ref_list.asp

³ É possível usar a função `startswith` para saber se uma string se inicia por uma determinada sequência de caracteres: https://www.w3schools.com/python/ref_string_startswith.asp

4. Os Lusíadas⁴ são compostos por 10 cantos, cada um com um número variável de octetos (estrofes com 8 versos), que são antecidas por um número. Utilizando o ficheiro `lusiadas.txt`:
 - 4.1. Criar uma lista de estrofes⁵ (tamanho: 1102 estrofes)
 - 4.2. Com base na lista criada, determinar:
 - 4.2.1. Quantos versos tem Os Lusíadas? (R: 8816)
 - 4.2.2. Quantas palavras tem Os Lusíadas? (R: 55232)
 - 4.2.3. Qual é o número médio de palavras por estrofe? (R: 50.12)
 - 4.2.4. Quantas⁶ vezes aparece a palavra “Rei”? (R: 149)
 - 4.3. Uma turma está a analisar Os Lusíadas num trabalho de grupo. Cada um dos 5 grupos da turma terá de analisar, aleatoriamente, uma das estrofes. Desenvolver o código necessário para tornar esta tarefa automática: gerar 5 ficheiros, cada um com uma estrofe aleatória⁷ para ser posteriormente analisada pelo grupo.
 - 4.3.1. Automatizar mais a tarefa: permitir que o número de grupos/alunos seja definido (`input`) pelo utilizador
5. Uma escola necessita de um sistema para guardar os seus alunos e disciplinas, bem como as disciplinas a que os alunos estão inscritos e as notas obtidas. Pretende-se implementar em Python e recorrendo a programação orientada a objetos:
 - Uma **disciplina** é caracterizada por um código (inteiro, sequencial) e um nome
 - Um **aluno** é caracterizado por um número de aluno (inteiro, sequencial) e um nome
 - Uma **inscrição** é caracterizada por um aluno, uma disciplina, e uma lista com as notas obtidas na disciplina. Ao criar a inscrição, a lista de notas é uma lista vazia e o sistema deve contar com um método que permite adicionar uma nota a uma inscrição
 - 5.1. Criar as classes necessárias para a implementação
 - 5.2. Criar um ficheiro `main.py`, para testar o código, com as seguintes funcionalidades:
 - 5.2.1. Criar um dicionário com 2 disciplinas. Guardar os dados das disciplinas num ficheiro XML⁸. O dicionário deve ter o seguinte formato:

```
{"dados": {"disciplinas": [...listaDisciplinas...]}}
```

⁴ Outras curiosidades sobre Os Lusíadas: <https://notapositiva.com/os-lusiadas-estrutura-interna-e-externa/>

⁵ Sugestão: uma estrofe será uma lista de versos. Ler o ficheiro completo e criar uma lista de listas com as estrofes. Utilizar

- `isnumeric` para verificar se o conteúdo da linha é um valor numérico https://www.w3schools.com/python/ref_string_isnumeric.asp
- `enumerate` para auxiliar o ciclo de leitura dos versos <https://realpython.com/python-enumerate/>

⁶ Sugestão: usar `count` <https://www.guru99.com/python-list-count.html>

⁷ Gerar números aleatórios em Python: https://www.w3schools.com/python/ref_random_randint.asp

⁸ Usar <https://jsonformatter.org/xml-parser> para verificar a estrutura do XML

5.2.2. Criar uma lista com 10 alunos. Guardar os dados dos alunos num ficheiro JSON⁹. O dicionário deve ter o seguinte formato:

```
{"alunos": listaAlunos}
```

5.2.3. Criar uma lista com 15 inscrições (não é obrigatório todos os alunos estarem inscritos em todas as disciplinas).

5.2.4. Para cada inscrição, criar aleatoriamente 3 valores (entre 0 e 20) para as notas e inseri-las na inscrição.

5.2.5. Guardar a lista de inscrições num ficheiro CSV com o seguinte formato:

```
nrAluno, nomeAluno, codDisciplina, nomeDisciplina, nota1, nota2, nota3
```

5.3. Ler o ficheiro CSV onde foram guardadas as inscrições e, com base nos dados obtidos, calcular a nota final para cada aluno em cada disciplina, sabendo que os momentos de avaliação valem 20%, 30% e 50%, independentemente da disciplina

5.4. Criar uma pasta `notasDisciplinas` e guardar dentro dessa pasta os resultados em ficheiros CSV, um por disciplina, no formato:

```
nrAluno, notaFinal
```

5.5. Ler os ficheiros existentes na pasta¹⁰ `notasDisciplinas` e, para cada disciplina determinar:

5.5.1. Quantidade de notas positivas e negativas

5.5.2. Média da disciplina

5.5.3. Aluno com melhor e pior nota

⁹ Usar <https://jsonformatter.org/> para verificar a estrutura do JSON

¹⁰ Consultar https://www.tutorialspoint.com/python/os_listdir.htm para aprender a listar os ficheiros existentes numa pasta, para depois poderem ser processados