

Lista de exercícios com foco na aplicação de NumPy.**Observações gerais (LEIA ANTES DE INICIAR):**

1. Utilize a biblioteca NumPy em **todos** os exercícios.
2. Quando não for especificado um método de criação de um array, escolha os valores como desejar ou, preferencialmente, use um método da biblioteca para criá-lo e testar seu programa.
3. Caso não seja mencionado no enunciado o tratamento de exceções, considere que os dados de entrada são coerentes com o solicitado. Por exemplo, se o programa pede para digitar um número, considere que o usuário irá seguir essa orientação (ele não vai digitar um texto, por exemplo). **No entanto**, vale ressaltar que essa consideração serve apenas para acelerar a execução do exercício. Na prática da vida real, você deve implementar todo tratamento de erros necessário para que seu código seja executado da forma correta, sem erros inesperados.
4. Aquilo que não estiver explicitamente especificado no enunciado é de escolha do programador. Ex: formatação da leitura e escrita na tela ou arquivo, tratamento de exceção, extensão de arquivos, etc.
5. A solução dos exercícios deve ser feita em um script a ser executado. Use o modo interativo, se desejar, somente para testes.
6. Dica para escrita e leitura de arquivos com NumPy:
<https://numpy.org/doc/stable/user/how-to-io.html>

Bons estudos!

- 1) Crie um array de 2 dimensões com valores em ponto flutuante escolhidos aleatoriamente dentro de um intervalo qualquer.

A partir desse array, escreva na tela:

- a) O valor máximo (e sua posição) e o valor médio, considerando o array completo;
- b) O valor mínimo na linha que apresenta o valor máximo; e
- c) O valor máximo e o valor médio em cada linha e coluna.

Salve o array em um arquivo texto, no formato tabular (linhas e colunas), com os valores separados por espaço.

- 2) Repita o exercício anterior para um array de números reais que será lido de um arquivo texto, com dimensões desconhecidas pelo programador. Considere que o arquivo não contém campos em branco e os dados estão separados por espaço. Sugestão: use o arquivo gerado no exercício anterior como entrada.

- 3) Considere dois conjuntos de dados com coordenadas x,y . O primeiro se refere a uma amostra qualquer. O segundo se refere a possíveis centróides. O objetivo é saber a qual centróide os pontos da amostra estão relacionados, isto é, são mais próximos, de acordo com a distância euclidiana. Faça um programa que cumpra esse objetivo.
- Bônus: use a biblioteca matplotlib para gerar gráficos com os pontos x,y e os centróides.
- 4) Considere o arquivo 'tabela_exemplo.txt' fornecido como anexo. O arquivo contém valores em ponto flutuante, no formato de 4 colunas separadas por um caractere de tabulação e um número arbitrário de linhas. Elabore um programa para executar as seguintes tarefas, considerando como entrada um arquivo no mesmo padrão do exemplo (mas não necessariamente igual):
- Escrever na tela, para cada coluna: média, mediana, desvio padrão, mínimo e máximo.
 - Salvar em um arquivo texto a soma das colunas para cada linha.
 - Emitir um alerta ao usuário no caso de haver linhas com valores repetidos, em qualquer coluna. Nesse caso, indique quais as linhas e escreva na tela os valores.
 - Escrever na tela o erro médio quadrático entre os dados da primeira coluna e os dados contidos em um segundo arquivo texto. Considere que esse segundo arquivo possui apenas uma coluna com valores em ponto flutuante com um número de linhas que pode ser diferente do primeiro. Obs: elabore um código à parte para gerar o segundo arquivo usando um dos métodos `np.random`.
- 5) Considere o arquivo 'notas.csv' fornecido como anexo. Crie um array estruturado com os dados de alunos cujos campos são: código do aluno, nome, código da turma, nome da turma e média geral. Escreva em um arquivo a relação de alunos em ordem de classificação, de acordo com a média geral (da maior para a menor). Caso haja empate, ordene pela turma e, em seguida, pela ordem alfabética dos nomes. Os dados devem ser escritos no mesmo formato do arquivo de entrada.