

Lista de Exercícios: Introdução ao Matplotlib

Gabriel Rodrigues Caldas de Aquino

03/07/2025

1 Preço por kg de leite em pó

Descrição:

Você possui o arquivo de texto chamado `leite-preco-kg.txt`, contendo informações de diferentes marcas de leite em pó. Cada linha possui o nome do produto, o preço total do pacote em reais e o peso do pacote em kg, separados por vírgula.

```
Leite em pó A,38.90,2.0
Leite em pó B,22.50,1.0
Leite em pó C,18.00,0.8
Leite em pó D,45.20,2.5
Leite em pó E,12.90,0.4
Leite em pó Z,12.90,0.0
Leite em pó F,27.50,1.2
Leite em pó G,55.00,3.0
```

Tarefas:

- Ler o arquivo linha por linha, separando corretamente os campos.
- Calcular o preço por kg de cada produto (preço dividido pelo peso).
- Gerar um gráfico de barras com `matplotlib` mostrando:
 - eixo X: nome do produto
 - eixo Y: preço por kg calculado
- Tratar possíveis erros (por exemplo, divisão por zero) com `try-except`.

O gráfico deve conter título, legendas nos eixos e as barras identificadas.

2 Cálculo de vendas com produto matricial

Descrição:

Você é responsável pelo controle de vendas de três produtos (Lápis, Caderno e Borracha) em três lojas diferentes (Loja A, Loja B e Loja C). Foi montada uma planilha com a quantidade de produtos vendidos em cada loja, conforme abaixo (cada linha representa uma loja, e cada coluna representa um produto):

	Lápis	Caderno	Borracha
Loja A	100	50	200
Loja B	80	60	150
Loja C	120	40	100

Além disso, você possui os preços unitários de cada produto:

- Lápis: R\$ 2,00
- Caderno: R\$ 10,00
- Borracha: R\$ 1,50

Tarefa:

Utilizando o NumPy e a operação de produto matricial, calcule o valor total recebido em cada loja considerando todas as vendas.

3 Cálculo de vendas com preços diferenciados por loja

Descrição:

Você é responsável por três lojas (Loja A, Loja B e Loja C) que vendem três produtos (Lápis, Caderno e Borracha). Cada loja tem seu próprio preço para cada produto, e você já possui o levantamento das vendas conforme abaixo:

Quantidade de produtos vendidos:

	Lápis	Caderno	Borracha
Loja A	100	50	200
Loja B	80	60	150
Loja C	120	40	100

Preços dos produtos em cada loja:

	Lápis	Caderno	Borracha
Loja A	2.00	9.50	1.60
Loja B	2.20	10.00	1.40
Loja C	1.90	9.80	1.50

Tarefa:

Utilizando o NumPy, calcule o valor total recebido em cada loja, considerando a multiplicação elemento a elemento entre quantidade e preço, e depois somando os valores de cada loja.

4 Soma vetorial de forças

Descrição:

Um objeto está sendo puxado por duas forças representadas diretamente em coordenadas cartesianas:

- $\mathbf{F}_1 = (10, 5)$ N
- $\mathbf{F}_2 = (-4, 12)$ N

Tarefa:

Calcule a força resultante (soma vetorial) usando a biblioteca NumPy.

5 Plotando funções definidas pelo usuário

Descrição:

Em Python, crie uma função chamada $f(x)$ onde você mesmo(a) define a função matemática (por exemplo, $f(x) = x^2 + 3x + 1$) dentro dela.

Tarefas:

- Implemente a função $f(x)$ contendo a expressão matemática desejada dentro dela.
- Use o `matplotlib` para plotar o gráfico de $f(x)$ para valores de x no intervalo $[-10, 10]$, com pelo menos 100 pontos.
- Modifique a expressão dentro de $f(x)$ para testar diferentes funções e verifique os gráficos resultantes.
- Utilize o NumPy para gerar os valores de x .

6 Faça o tratamento dos possíveis erros que podem acontecer neste código mostrado em aula

```
def calcular_media(valores):  
    soma = sum(valores)  
    media = soma / len(valores)  
    return media  
  
def ler_arquivo(nome_arquivo):  
    with open(nome_arquivo, 'r') as file:  
        dados = file.read().split(',')  
    return [float(dado) for dado in dados]  
  
print("==== Calculadora de Média ===")
```

```

valores = input("Digite números separados por vírgula: ").split(',')
valores = [int(x) for x in valores]
print(f"Valores de entrada: {valores}")

media = calcular_media(valores)
print(f"Média do teclado: {media}")

dados_arquivo = ler_arquivo('dados.txt')
print(f"Média do arquivo: {calcular_media(dados_arquivo)}")

```

7 Manipulação de arquivos em Python

Descrição:

Analise o seguinte código em Python:

```

with open("dados.txt", "w+") as arquivo:
    arquivo.write("Primeira linha\n")
    arquivo.seek(0)
    print(arquivo.read())

```

Perguntas:

- O que será exibido no console após a execução do código?
- Qual seria o resultado se a linha `arquivo.seek(0)` fosse removida? Justifique sua resposta.

8 Uso do método seek() em arquivos

Descrição:

O método `seek()` reposiciona o cursor de leitura/escrita de um arquivo para um deslocamento específico, permitindo que operações subsequentes sejam feitas a partir de uma posição arbitrária dentro do arquivo.

Exemplo:

Considere que um arquivo chamado `letras.txt` contém o seguinte conteúdo:

ABCDEF

E que o seguinte código seja executado em Python:

```

arq = open("letras.txt", "a+")
arq.seek(3)
print(arq.read())
arq.close()

```

Pergunta:

- Qual será a saída exibida no console após a execução do código? Explique detalhadamente o motivo.

9 Verificação de triângulo equilátero

Descrição:

Considere o seguinte código em Python:

```
def verifica_equilatero(triangulo):
    if triangulo[0] == triangulo[1] == triangulo[2]:
        return True
    else:
        raise ValueError("Não é um triângulo equilátero!")
```

Perguntas:

- O que acontece se for passado os valores [4,5,4] como argumento?
- Escreva um código que chame essa função e trate a exceção de forma adequada, exibindo a mensagem de erro para o usuário.

10 Tratamento de exceções em Python

Descrição:

Considere o seguinte código Python:

```
try:
    arquivo = open("dados.txt", "r")
    dados = arquivo.read()
except FileNotFoundError:
    print("Arquivo não encontrado!")
finally:
    print("Fim da execução")
```

Suponha que o arquivo dados.txt não exista no diretório onde o código é executado.

Perguntas:

- Qual será a saída exibida na tela?
- Explique.