



PROGRAMAÇÃO 101

dia 5

Imagens e Arquivos

Leitura de Arquivos

- Função `open()`:
 - Abre um arquivo para leitura.
- Método `read()`:
 - Lê o conteúdo do arquivo.
- Método `readlines()`:
 - Lê e retorna lista com as linhas do arquivo.

```
arquivo = open("exemplo.txt", "r")
conteudo = arquivo.read()
arquivo.close()
```

```
with open('meu_arquivo.txt', 'r') as arquivo:
    linhas = arquivo.readlines()
```



Leitura de Arquivos

- Escrita de arquivos:

```
with open('meu_arquivo.txt', 'w') as arquivo:  
    arquivo.write("Novo conteúdo do arquivo.\n")
```

- Modos de abertura de arquivos:
 - 'r': Leitura (modo padrão)
 - 'w': Escrita (substitui o arquivo se ele já existir)
 - 'a': Anexar (adiciona conteúdo ao final do arquivo)
 - 'b': Modo binário (combina com 'r', 'w' ou 'a')
 - 'x': Criação (gera um erro se o arquivo já existir)



Leitura de Imagens

- Biblioteca PIL (Pillow):
 - Utilizada para trabalhar com imagens.
- Método open():
 - Abre a imagem.
- Método show():
 - Exibe a imagem.

```
from PIL import Image  
  
imagem = Image.open("exemplo.jpg")  
imagem.show()
```



dia 5

Biblioteca NumPy

NumPy

Cálculos numéricos e Científicos.

- Manipulação de vetores, matrizes.
- Operações estatísticas.
- Cálculos de Álgebra Linear.

```
import numpy as np
```



Operações estatísticas

- Cálculos de média e desvio padrão

```
arr = np.array([1, 2, 3, 4, 5])
np.mean(arr)
np.std(arr)
```

É possível calcular também a soma e produto dos elementos, obter a moda, mediana, variância...

dia 5

Biblioteca Pandas

Pandas

Manipulação de Dados

- Tarefas de Planilhas/Excel
- Tratamento de Dados

```
import pandas as pd
```



Tarefas de Planilhas/Excel

- Leitura e Escrita de Arquivos CSV e Excel

```
df = pd.read_csv('data.csv')
df = pd.read_excel('data.xlsx')
```

```
df.to_csv('data_output.csv')
df.to_excel('data_output.xlsx')
```



Tarefas de Planilhas/Excel

- Manipulação de Colunas

```
df['new_col'] = df['col1'] + df['col2'] # Adicionar nova coluna  
df.drop('col1', axis=1, inplace=True) # Remover coluna  
df.rename(columns={'col1': 'new_name'}, inplace=True) # Renomear coluna
```

- Manipulação de Linhas

```
# Adicionando uma nova linha  
new_row = pd.DataFrame({'col1': [70], 'col2': [80]})  
df = pd.concat([df, new_row], ignore_index=True)  
  
df.loc[1, 'col1'] = 25 # Modificar o valor da linha de índice 1  
df.drop(0, axis=0, inplace=True) # Exclui a linha com índice 0
```



Tarefas de Planilhas/Excel

- Tratamento de Valores Faltantes

```
df.isna() # Verificar NaNs  
df.fillna(0, inplace=True) # Substituir NaNs por 0  
df.dropna(inplace=True) # Remover linhas com NaN
```

- Ordenação dos Dados

```
df.sort_values(by='col1', ascending=False, inplace=True)
```

dia 5

Biblioteca Matplotlib

Matplotlib

Visualização de Dados

- Criação de diversos gráficos como Histogramas, Gráficos de Barras, Dispersão, Pizza...

```
import matplotlib.pyplot as plt
```

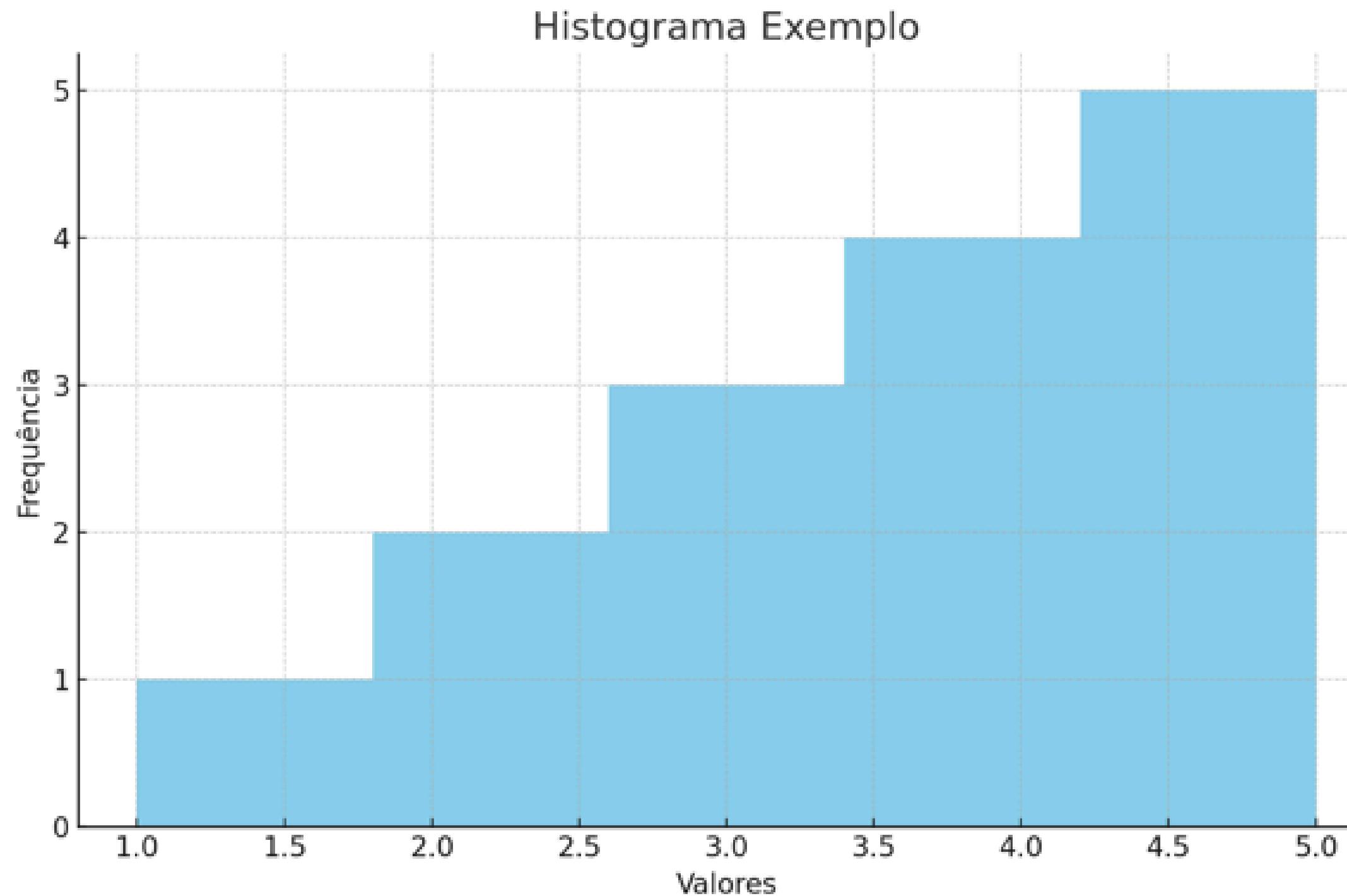


Exemplos de Gráficos

- Exemplo de Histograma

```
data_histogram = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5]
plt.figure(figsize=(10, 6))
plt.hist(data_histogram, bins=5, color='skyblue')
plt.title('Histograma Exemplo')
plt.xlabel('Valores')
plt.ylabel('Frequência')
plt.savefig('/mnt/data/histogram_example.png')
```

Exemplos de Gráficos



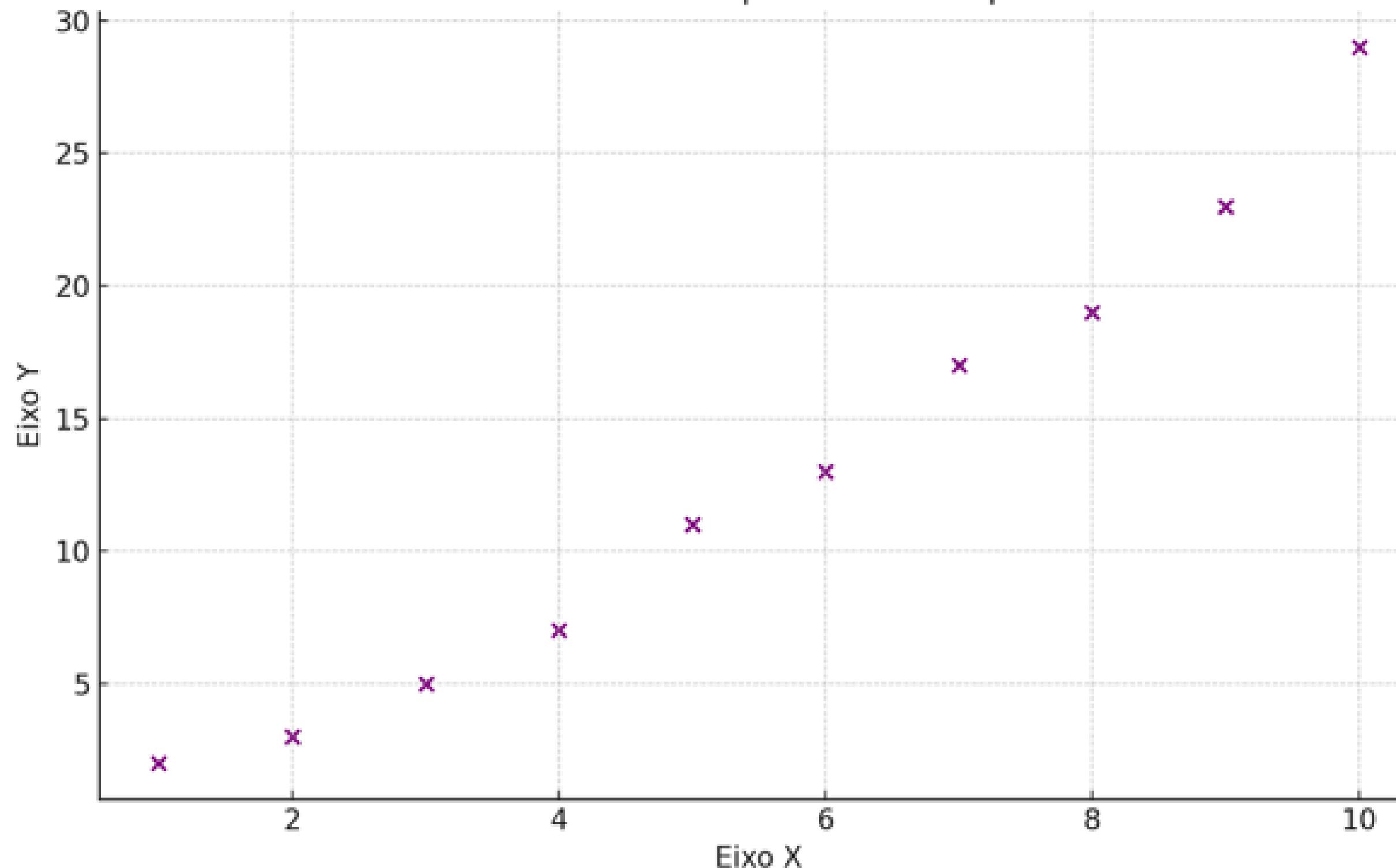
Exemplos de Gráficos

- Exemplo de Gráfico de Dispersão

```
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
plt.figure(figsize=(10, 6))
plt.scatter(x, y, color='purple')
plt.title('Gráfico de Dispersão Exemplo')
plt.xlabel('Eixo X')
plt.ylabel('Eixo Y')
plt.savefig('/mnt/data/scatter_example.png')
```

Exemplos de Gráficos

Gráfico de Dispersão Exemplo



Dia 5

Como continuar a
aprender?

Como continuar?

Para se familiarizar com Python, temos o Python Tutor que te auxilia a compreender como a linguagem funciona, te mostrando passo a passo a execução do seu código de maneira intuitiva.

<https://pythontutor.com/>



Como continuar?

Para exercitar lógica e programação, existem sites gratuitos que fornecem exercícios de maratonas



beecrowd



CODEFORCES
Sponsored by TON

Como continuar?

Existem muitos cursos gratuitos de Python disponíveis na Internet para aquilo que você deseja aprofundar. Você pode focar inicialmente em aprender a utilizar as bibliotecas e se acostumar com a linguagem antes de procurar cursos mais avançados de Ciências de Dados e Aprendizado de Máquina.

Recomendação: [Curso em vídeo](#)

https://www.youtube.com/watch?v=S9uPNppGsGo&list=PLHz_AreHm4dIKP6QQCekulPkyICiwmdl6&ab_channel=CursoemV%C3%ADdeo





PROGRAMAÇÃO 101