



*O Google Colab (Colaboratory) é uma ferramenta gratuita oferecida pelo Google que permite escrever e executar código Python diretamente no navegador, com foco especial em aplicações de ciência de dados e aprendizado de máquina. Ele oferece um ambiente baseado em Jupyter Notebook, onde você pode combinar código executável, visualizações e texto em Markdown.*

# O que o Google Colab oferece?

Ambiente de execução na nuvem (você não precisa instalar nada localmente).

Suporte a GPUs e TPUs gratuitas para projetos de machine learning.

Integração com o Google Drive para salvar e carregar notebooks.

Bibliotecas pré-instaladas como NumPy, pandas, matplotlib, scikit-learn, TensorFlow, entre outras.

Interface prática para prototipação rápida, especialmente útil em cursos e projetos colaborativos.

---

# O que é um notebook?

Um notebook é um documento interativo onde você pode escrever e executar código, colocar texto explicativo (como anotações, títulos, imagens), e até visualizar gráficos e resultados ali mesmo, tudo misturado. Ele é muito usado para programação, análise de dados, machine learning e ensino.

---

# Estrutura de um Notebook

## **Célula de Texto (Markdown):**

Usada para explicações, títulos, listas, links.

Pode usar formatação como negrito, itálico e código inline.

.....

# Estrutura de um Notebook

## **Célula de Código (Python):**

Executa instruções Python

Pode conter gráficos, cálculos, funções, etc.

O colab roda código em servidores do Google

Basta apertar Shift + Enter para executar

---

# Calculando no Colab

Operações básicas: +, -, \*, /, \*\*

Atribuição com =

Exibição com print()

---

# Tipos de Dados

int: números inteiros (ex: 10, -3)

float: números com ponto decimal (ex: 3.14)

str: textos (ex: "Olá", 'Python')

bool: booleanos (True ou False)

.....

# O que é variável?

Uma variável é como uma caixinha com um nome, onde você guarda alguma coisa para usar depois.





# Trabalhando com Listas

Sequência ordenada e mutável

Permite armazenar vários elementos



# Dicionários

Estrutura de dados com pares chave: valor

Muito usada para representar objetos do mundo real



# Tuplas

Sequência ordenada de elementos, como as listas

Imutável: não é possível alterar os elementos após criação

Usa parênteses: ()

---

# Conjuntos

Coleção não ordenada, sem elementos duplicados

Usa chaves: {} ou set()

Útil para remover duplicatas ou fazer operações de conjuntos

---

# Estatística Descritiva com statistics

Este módulo fornece funções para calcular estatísticas matemáticas de dados numéricos (valores reais).

<https://docs.python.org/3/library/statistics.html>

---

# Mão na massa

[https://colab.research.google.com/drive/12IKDKcd2G0BD3P9IjiTj7JlaHP8B\\_9l?usp=sharing](https://colab.research.google.com/drive/12IKDKcd2G0BD3P9IjiTj7JlaHP8B_9l?usp=sharing)

# Atividades

# Por que usar gráficos para representar dados?

Facilitam a compreensão visual de grandes volumes de dados.

Ajudam a identificar padrões, tendências e outliers.

Tornam a comunicação mais clara com públicos diversos.

São ferramentas essenciais na tomada de decisão baseada em dados.

---



# Introdução ao matplotlib

É a principal biblioteca de visualização gráfica em Python.

Permite a criação de gráficos 2D de forma altamente customizável.

Trabalha bem com numpy, pandas e outras bibliotecas.

Instalação (Colab): já vem embutida!

---

# Biblioteca Matplotlib

O matplotlib é uma das bibliotecas mais populares para gerar gráficos em Python.

Com ele, você pode criar histogramas, gráficos de linha, barras, dispersão, pizza e muito mais.

[https://matplotlib.org/stable/users/explain/quick\\_start.html#quick-start](https://matplotlib.org/stable/users/explain/quick_start.html#quick-start)

---

# Introdução ao Pandas

Biblioteca Python para análise e manipulação de dados

Construída sobre NumPy, focada em estruturas de dados flexíveis e poderosas

Estruturas principais:

- Series: array unidimensional rotulado

- DataFrame: tabela bidimensional com linhas e colunas rotuladas

- Muito usada em Ciência de Dados, Machine Learning, estatística e finanças

# DataFrames

Estrutura de dados tabular: linhas (observações) e colunas (variáveis)

Suporta dados heterogêneos (numéricos, texto, booleanos)

Indexação flexível por rótulos ou posição

# Tipos e variáveis

Tipos comuns em pandas:

- `int64`: inteiros
- `float64`: números reais
- `object`: texto/string
- `bool`: booleano
- `datetime64`: datas

# Tipos e variáveis

Visualizar tipos de dados:

```
df.dtypes
```

Importância de entender os tipos para manipulação e análise

Conversão de tipos

```
df['coluna'] = df['coluna'].astype('float')
```

# Arredondamento de valores

```
df['coluna'] = df['coluna'].round(2)
```

# Manipulação de arquivos com pandas

Leitura de diferentes formatos:

- CSV: `pd.read_csv('arquivo.csv')`
- Excel: `pd.read_excel('arquivo.xlsx')`
- JSON: `pd.read_json('arquivo.json')`

Salvando arquivos:

- CSV: `df.to_csv('saida.csv', index=False)`
- Excel: `df.to_excel('saida.xlsx', index=False)`



# Sites Seguros para Obter Dados para Análise

## Kaggle

- Plataforma líder em competições de Data Science
- Milhares de datasets públicos, diversos temas (saúde, finanças, imagens, texto etc.)
- Comunidade ativa com notebooks, discussões e tutoriais
- URL: <https://www.kaggle.com/datasets>

## UCI Machine Learning Repository

- Um dos repositórios mais tradicionais e usados em pesquisas
- Datasets clássicos para aprendizado de máquina
- Arquivos simples, geralmente em CSV ou TXT
- URL: <https://archive.ics.uci.edu/ml/index.php>

# Sites Seguros para Obter Dados para Análise

## Google Dataset Search

- Ferramenta de busca específica para datasets na web
- Reúne dados de múltiplas fontes confiáveis
- Facilidade para encontrar dados por tema, formato e licença
- URL: <https://datasetsearch.research.google.com/>

## Data.gov

- Portal oficial de dados abertos do governo dos EUA
- Grande variedade de dados públicos (economia, saúde, meio ambiente, transporte)
- Dados atualizados e confiáveis
- URL: <https://www.data.gov/>

# Sites Seguros para Obter Dados para Análise

## Awesome Public Datasets (GitHub)

- Curadoria colaborativa de datasets públicos organizados por temas
- Link para repositórios e fontes variadas
- Atualizado frequentemente pela comunidade
- URL: <https://github.com/awesomedata/awesome-public-datasets>

## World Bank Open Data

- Dados globais sobre economia, desenvolvimento, indicadores sociais
- Atualização constante com fontes oficiais internacionais
- Formatos fáceis de usar para análise
- URL: <https://data.worldbank.org/>
-

# Biblioteca Matplotlib

O matplotlib é uma das bibliotecas mais populares para gerar gráficos em Python.

Com ele, você pode criar histogramas, gráficos de linha, barras, dispersão, pizza e muito mais.

[https://matplotlib.org/stable/users/explain/quick\\_start.html#quick-start](https://matplotlib.org/stable/users/explain/quick_start.html#quick-start)

---

# Biblioteca Seaborn

Seaborn é uma biblioteca de visualização de dados em Python, construída sobre o Matplotlib.

O objetivo dela é facilitar a criação de gráficos estatísticos bonitos e informativos.

Muito usada em análise exploratória de dados (EDA), quando precisamos entender padrões, distribuições e relações entre variáveis.

<https://seaborn.pydata.org/>

---

# O que é Correlação?

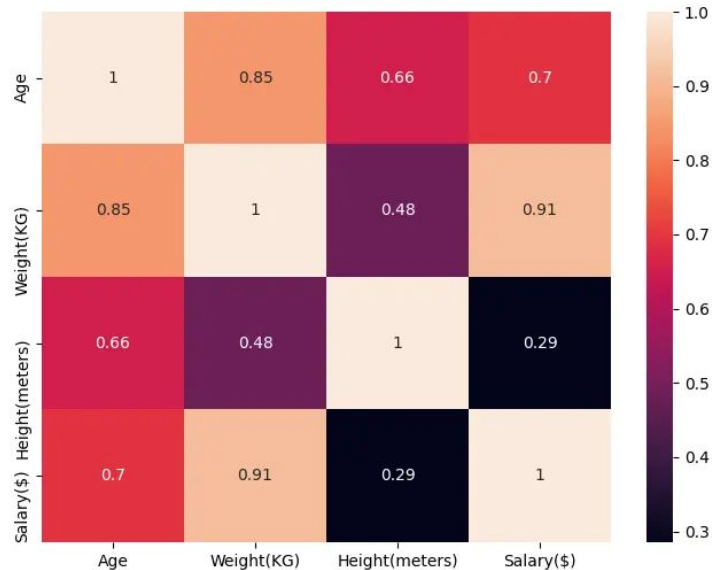
Correlação mede a força e direção da relação entre duas variáveis.

O coeficiente de correlação varia entre -1 e 1:

+1: relação positiva perfeita

-1: relação negativa perfeita

0: nenhuma correlação linear



# Exemplo simples:

```
# Importar bibliotecas
```

```
import seaborn as sns
```

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

```
import pandas as pd
```

```
# Carregar o dataset
```

```
tips = sns.load_dataset('tips')
```

```
# Calcular a correlação entre as variáveis  
numéricas
```

```
correlation = tips.corr()
```

```
print(correlation)
```

```
# Plotar o heatmap da correlação
```

```
plt.figure(figsize=(8,6))
```

```
sns.heatmap(correlation, annot=True,  
cmap='coolwarm', fmt=".2f")
```

```
plt.title("Mapa de Correlação - Dataset Tips")
```

```
plt.show()
```

---

# Biblioteca numpy

NumPy é o pacote fundamental para computação científica em Python. É uma biblioteca Python que fornece um objeto array multidimensional, vários objetos derivados (como arrays e matrizes mascarados) e uma variedade de rotinas para operações rápidas em arrays, incluindo operações matemáticas, lógicas, manipulação de formas, ordenação, seleção, E/S, transformadas discretas de Fourier, álgebra linear básica, operações estatísticas básicas, simulação aleatória e muito mais.

<https://numpy.org/doc/stable/>

---



# NumPy vs Listas do Python

Aspecto	Listas ( <b>list</b> )	NumPy ( <b>ndarray</b> )
Estrutura	Heterogênea (pode misturar int, str, float, etc.)	Homogênea (todos elementos do mesmo tipo, mais otimizado)
Desempenho	Mais lento em operações matemáticas	Muito mais rápido (implementado em C, usa vetorização)
Operações	Precisa de loops explícitos para somar, multiplicar etc.	Operações vetorizadas (soma, multiplicação, raiz, estatísticas, etc. já prontas)
Memória	Ocupa mais espaço, armazena metadados de cada objeto	Mais compacto e eficiente
Funcionalidade	Básico: append, remove, slice	Avançado: matrizes, álgebra linear, estatísticas, broadcasting