

Introdução a Análise de Dados com Python

Joyce Teixeira - jvt@cesar.school



Google Colab



Roteiro

- Google Colab
 - Introdução
 - Vantagens
 - Criando um notebook

Google Colab

- Plataforma de programação em ambiente Web
- Simples e com diversas funcionalidades para lidar de forma interativa com Python e outras linguagens
- Forma fácil e organizada para codificar ou demonstrar de forma plena algum código
- Interface do Jupyter Notebook e é amplamente usado na análise de dados, aprendizado de máquina e pesquisa.

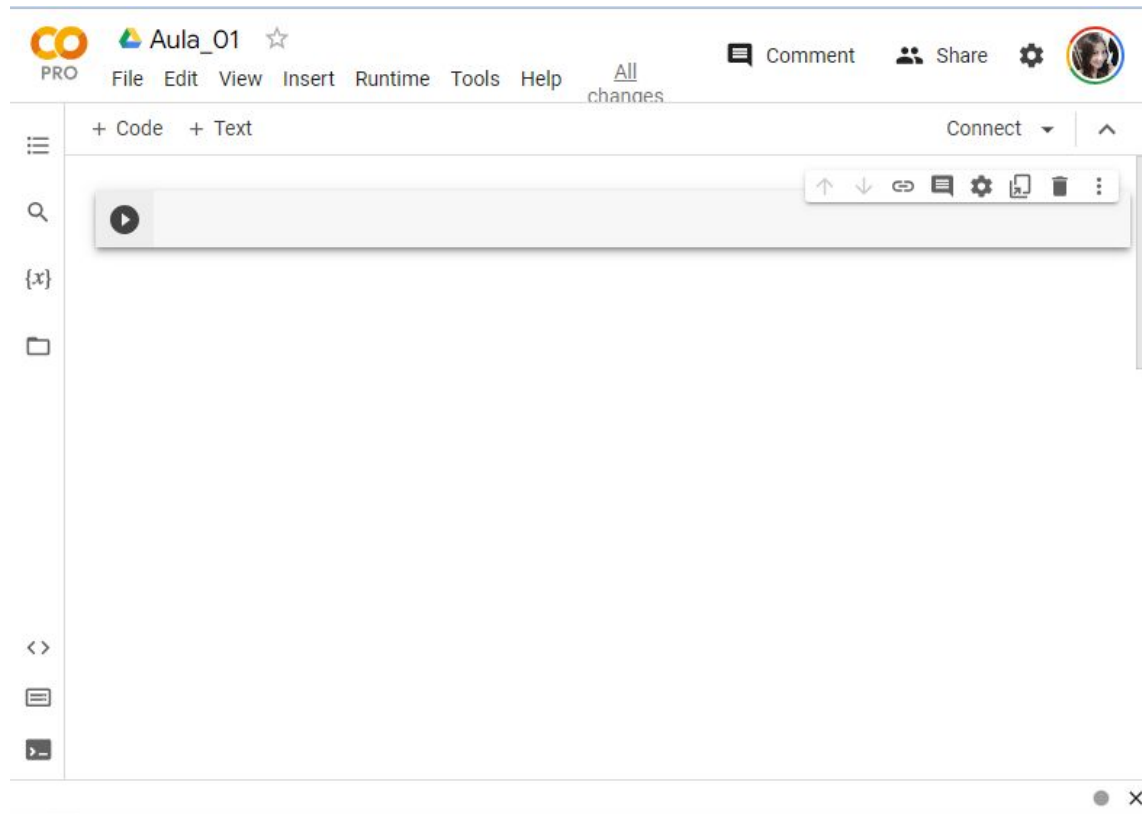
Google Colab

- Vantagens:
 - Acesso gratuito na nuvem: Não é necessário instalar software no computador.
 - Colaboração: Possibilidade de compartilhar e colaborar em tempo real com colegas.
 - GPU e TPU: Suporte a hardware acelerado para tarefas de aprendizado de máquina.
 - Integração com o Google Drive: Facilidade de armazenamento e compartilhamento de notebooks.

Google Colab

- Passo a passo para acessar:
 - Acesse: <https://colab.google/>
 - Crie um novo notebook
 - Por padrão, o Colab cria uma pasta Colab Notebooks onde armazena todos os notebooks criados

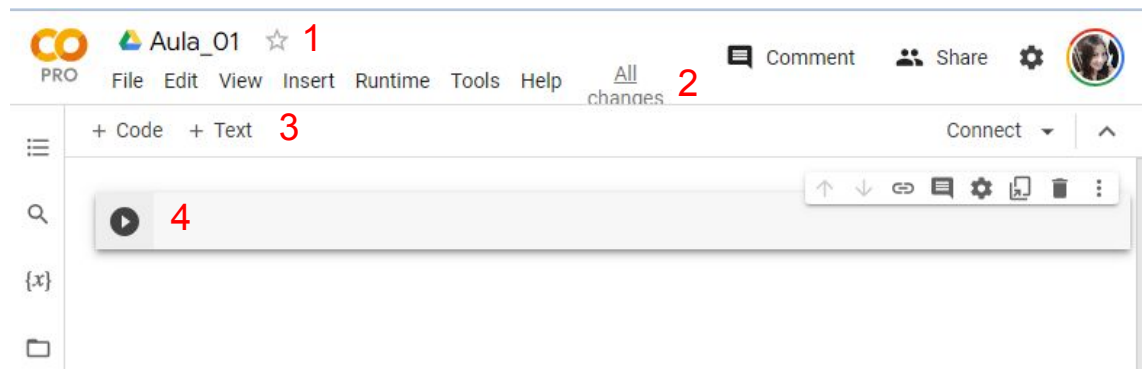
Google Colab



Entendendo o Google Colab

- Células e o ambiente de execução (Runtime) são fundamentais para entender o Colab e o que o torna mais do que apenas um processador de texto.
 - O Runtime é um “mecanismo computacional” que executa o código contido em um documento do notebook;
 - Uma célula é um recipiente para o texto a ser exibido no bloco de notas ou código a ser executado pelo ambiente de execução (Runtime) do notebook.

Criando um Notebook



1. Nome do notebook.
2. Barra de menus.
3. Barra de botões. Atalhos para algumas das rotinas mais utilizadas do Colab.
4. Célula. É aqui que a magia acontece.

Criando um Notebook

- Usando o Google Colab é possível adicionar células que podem ser independentes
- Diferente de um script Python padrão, podemos executar um bloco de códigos sem executar os blocos de código anteriores
- Textos explicativos podem ser adicionados juntamente com o código
- Um Markdown (célula de texto) é uma linguagem simples de marcação tipo HTML onde você pode incluir textos

Criando um Notebook

- Atalhos mais usados:
 - Para criar uma nova célula de código abaixo da célula atual, você pode pressionar Shift + Enter.
 - Para executar a célula atual, você pode pressionar Ctrl + Enter.
 - Para criar uma nova célula de código acima da célula atual, você pode pressionar Ctrl + M e depois A.
 - Para criar uma nova célula de código abaixo da célula atual, você pode pressionar Ctrl + M e depois B.
 - Para deletar uma célula, você pode pressionar Ctrl + M e depois D.

Numpy



Roteiro

- Numpy
 - Introdução
 - Criação de Arrays
 - Operações Matemáticas
 - Indexação e Fatiamento
 - Funções Estatísticas
 - Manipulação de Arrays
 - Números Aleatórios
 - Álgebra Linear
 - Funções de Comparação
 - Seleção e Filtragem
 - Exercícios

Introdução

- Numpy é um biblioteca de Álgebra Linear para Python
- É o bloco de construção de todas as outras bibliotecas de análise de dados
- A ideia fundamental do NumPy é o suporte à matrizes multidimensionais
- Pode ser considerado como a base para a computação numérica no Python e foi criado para permitir que o Python seja usado na resolução de problemas matemáticos e científicos

Introdução

- Seus principais métodos foram compilados em C, o que o torna bastante rápido:
 - Manipular arrays de Numpy é muito mais rápido que listas nativas de Python

```
import time

start_time = time.time()
a = range(10000000)
b = range(10000000)
c = []

for i in range(len(a)):
    c.append(a[i] + b[i])
elapsed_time = time.time() - start_time
print(elapsed_time)
```

4.71074366569519

```
import numpy as np
import time

start_time = time.time()
a = np.arange(10000000)
b = np.arange(10000000)
c = []

c = a + b

elapsed_time = time.time() - start_time
print(elapsed_time)
```

0.08148837089538574

Numpy - Funções e Métodos

- Criação de Arrays:
 - `np.array()`: Cria um array NumPy a partir de uma lista ou sequência.
 - `np.zeros()`: Cria um array de zeros com a forma especificada.
 - `np.ones()`: Cria um array de uns com a forma especificada.
 - `np.arange()`: Cria um array com valores igualmente espaçados dentro de um intervalo.
 - `np.linspace()`: Cria um array com valores igualmente espaçados entre dois limites.

Numpy - Funções e Métodos

- Operações Matemáticas:
 - Adição, subtração, multiplicação e divisão: Arrays NumPy permitem operações aritméticas elementares com facilidade.
 - `np.dot()`: Realiza multiplicação de matrizes.
 - `np.transpose()`: Retorna a matriz transposta.
- Indexação e Fatiamento:
 - Indexação por posição: Os elementos de um array podem ser acessados por sua posição.
 - Fatiamento: É possível selecionar partes específicas de um array.

Numpy - Funções e Métodos

- Funções Estatísticas:
 - `np.mean()`: Calcula a média dos elementos em um array.
 - `np.median()`: Calcula a mediana dos elementos em um array.
 - `np.std()`: Calcula o desvio padrão dos elementos em um array.
 - `np.sum()`: Soma os elementos de um array.
- Manipulação de Arrays:
 - `np.reshape()`: Redimensiona um array.
 - `np.concatenate()`: Concatena arrays.
 - `np.split()`: Divide um array em pedaços menores.

Numpy - Funções e Métodos

- Gerador de Números Aleatórios:
 - `np.random.rand()`: Gera números aleatórios em uma distribuição uniforme entre 0 e 1.
 - `np.random.randn()`: Gera números aleatórios em uma distribuição normal (gaussiana).
 - `np.random.randint()`: Gera números inteiros aleatórios dentro de um intervalo.

Numpy - Funções e Métodos

- Álgebra Linear:
 - `np.linalg.inv()`: Calcula a matriz inversa.
 - `np.linalg.det()`: Calcula o determinante de uma matriz.
- Funções de Comparação:
 - `np.where()`: Retorna elementos com base em uma condição.
- Seleção e Filtragem

Numpy - Revisão (Exercícios)

- Importe NumPy como np
- Crie uma array de 10 zeros
- Crie um array de inteiros de 10 até 50
- Crie um array dos números pares de 10 até 50
- Criei uma matriz 3x3 com valores variando de 0 até 8
- Crie uma matriz identidade 3x3
- Use NumPy para gerar números aleatórios entre 0 e 1
- Inserir 5 números inteiros em um array a partir do *input()*

Numpy

- Lista 01 disponível.