



Engineering e Data Analytics com Python.

Uma exploração de como o Python serve como a ponte essencial entre a Engenharia de Dados (a construção de pipelines) e a Análise de Dados (a extração de valor).



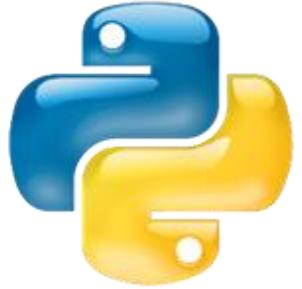
Prof. Dr. Nisston Moraes

Análise de Dados

A Arte e a Ciência de Conversar com IAs

1

Roteiro



Tópicos

- Módulo1 : Fundamentos de Análise de Dados
- Módulo2 : Introdução ao Python para Dados
- Módulo3 : Manipulação de dados com Pandas
- Módulo4 : Transformação de Dados com Python
- Módulo5 : Análise Estatística e Exploratória
- Módulo6 : Visualização de Dados
- Módulo7: Introdução à Modelagem Preditiva

Módulo 1: Fundamentos de Análise de Dados

- O que é Data Analytics
- Ciclo de vida da análise de dados
- Tipos de análise: descritiva, diagnóstica, preditiva e prescritiva
- Papel do Python na ciência de dados
- Conhecendo as ferramentas: Google Colab, Jupyter, VSCode, Anaconda

O que é Data Analytics

Conceito

- Data Analytics (ou Análise de Dados) é o **processo** de examinar, limpar, transformar e interpretar **dados** com o objetivo de descobrir informações úteis, apoiar decisões e identificar padrões, tendências ou relações entre variáveis.

Objetivos principais

- Descrever o que ocorreu (análise descritiva)
- Diagnosticar por que algo aconteceu (análise diagnóstica)
- Prever o que pode acontecer (análise preditiva)
- Prescrever ações com base nas previsões (análise prescritiva)

Envolve atividade como:

- Coleta de dados
- Limpeza e preparação dos dados
- Análise estatística e visualização
- Modelagem preditiva
- Comunicação dos resultados em relatórios e dashboards

Ferramentas utilizadas

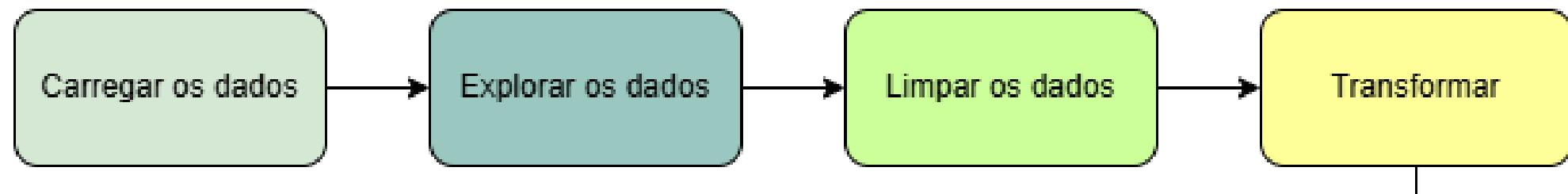
- Linguagens como Python e R
- Bibliotecas como Pandas, Numpy, Matplotlib, Scikit-learn
- Plataformas como Excel, Power BI, Google Data Studio

Ciclo de vida da análise de dados

Fluxo de trabalho

- **Carregar os dados:**
 - Importar de fontes como CSV, Excel, bancos de dados, APIs ou arquivos JSON.
- **Explorar os dados:**
 - Visualizar a estrutura, tipos de dados e valores nulos.
- **Limpar os dados:**
 - Remover duplicatas.
 - Preencher ou excluir valores ausentes.
 - Corrigir erros de formatação.
- **Transformar os dados:**
 - Criar novas colunas ou atributos derivados.
 - Alterar o formato dos dados (ex.: pivot ou melt).
- **Analigar os dados:**
 - Agrupar, filtrar e resumir para insights.
- **Salvar ou exportar:**
 - Exportar os dados tratados para arquivos ou bases.

Fluxo de manipulação dos dados



	<u>id</u>	<u>data_inversa</u>	<u>dia_semana</u>	<u>horario</u>	<u>Hora</u>	<u>Turno</u>	<u>uf</u>	<u>br</u>	<u>km</u>	<u>municipio</u>	...	<u>feridos_graves</u>	<u>ilesos</u>
0	331730	01/01/2021	sexta-feira	05:30:00	5	Manhã	SP	116.0	453.0	REGISTRO	...	0	0
1	331804	01/01/2021	sexta-feira	08:05:00	8	Manhã	AM	174.0	937.0	MANAUS	...	0	5
2	331815	01/01/2021	sexta-feira	10:10:00	10	Manhã	CE	222.0	128.5	ITAPAJE	...	0	0
3	331823	01/01/2021	sexta-feira	12:30:00	12	Tarde	RJ	493.0	18.0	MAGE	...	0	1
4	331843	01/01/2021	sexta-feira	14:40:00	14	Tarde	RJ	393.0	252.0	BARRA DO PIRAI	...	1	1

Etapas de um pipeline de dados

- Ferramentas para ETL (Extract, Transform, Load):
 - Apache Airflow.
 - Talend.
 - Apache NiFi.
 - Prefect.
- Soluções em nuvem:
 - AWS Glue.
 - Google Cloud Dataflow.
 - Azure Data Factory.
- Bibliotecas Python:
 - Pandas para manipulação de dados.
 - Luigi para criação de pipelines.
 - PySpark para processamento de grandes volumes.
 - Dask para computação paralela.

Exemplo prático

- **Carregar os dados**
- **Explorar os dados**
- **Limpar os dados**
- **Transformar os dados**
- **Analizar os dados**
- **Salvar ou exportar**

Exemplo Prático

- Web Scraping com Python
 - <https://github.com/nisston/coletadadoswebscraping>

Visualização dos Dados

Visualização dos Dados

- Durante a manipulação, é útil gerar visualizações

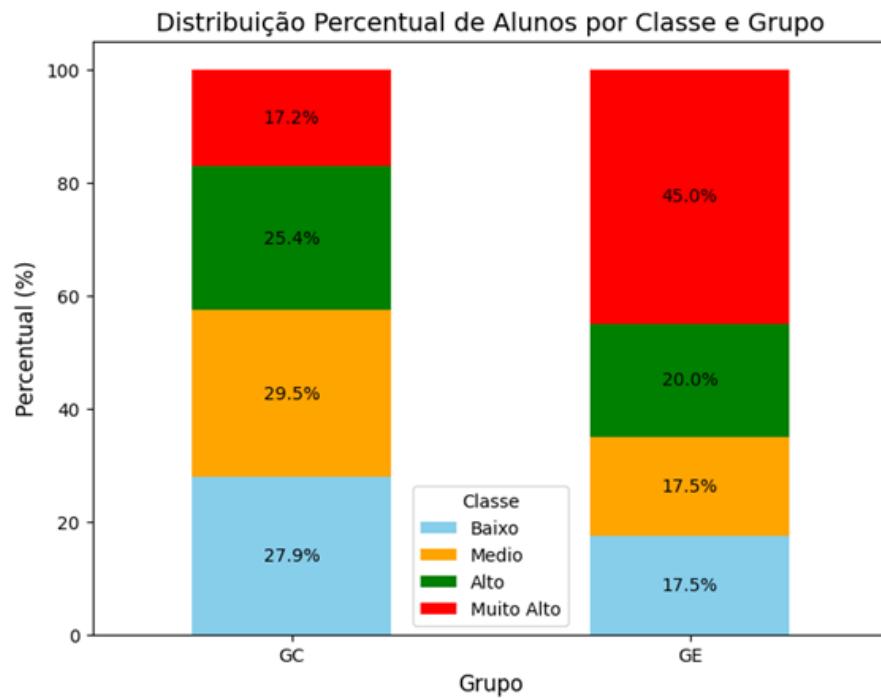
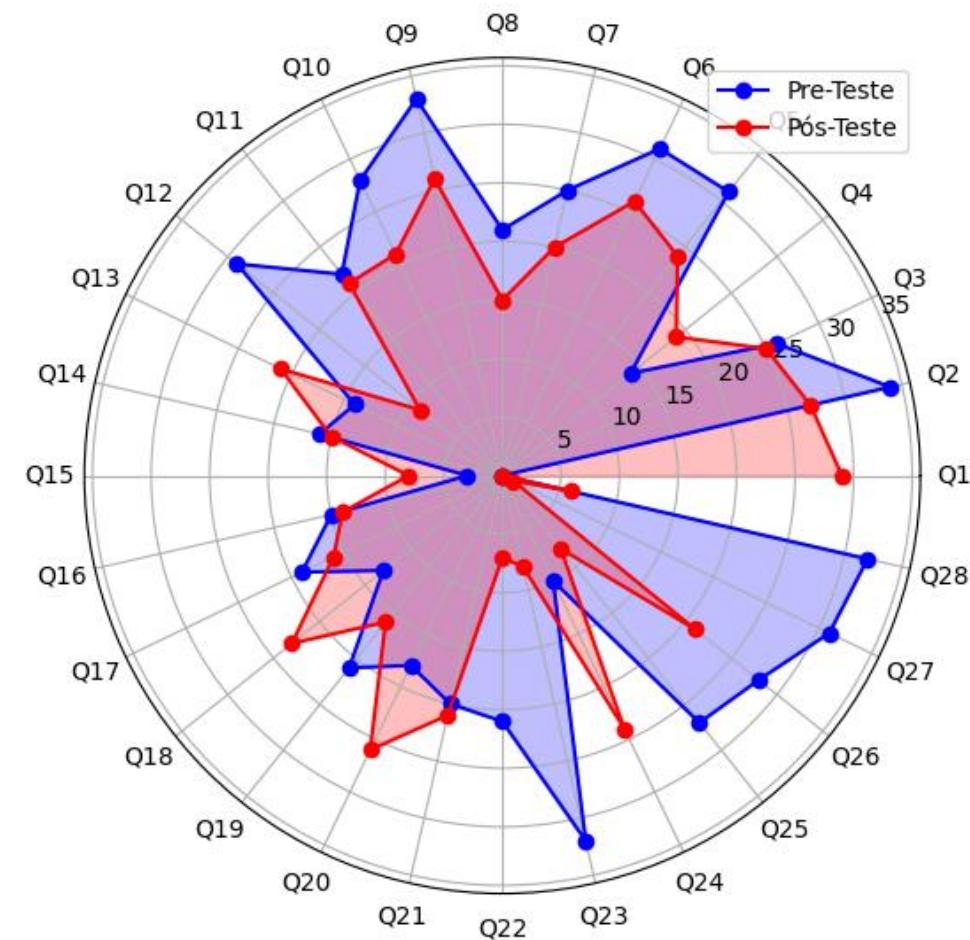


Gráfico de Radar das Questões do Grupo Experimenta

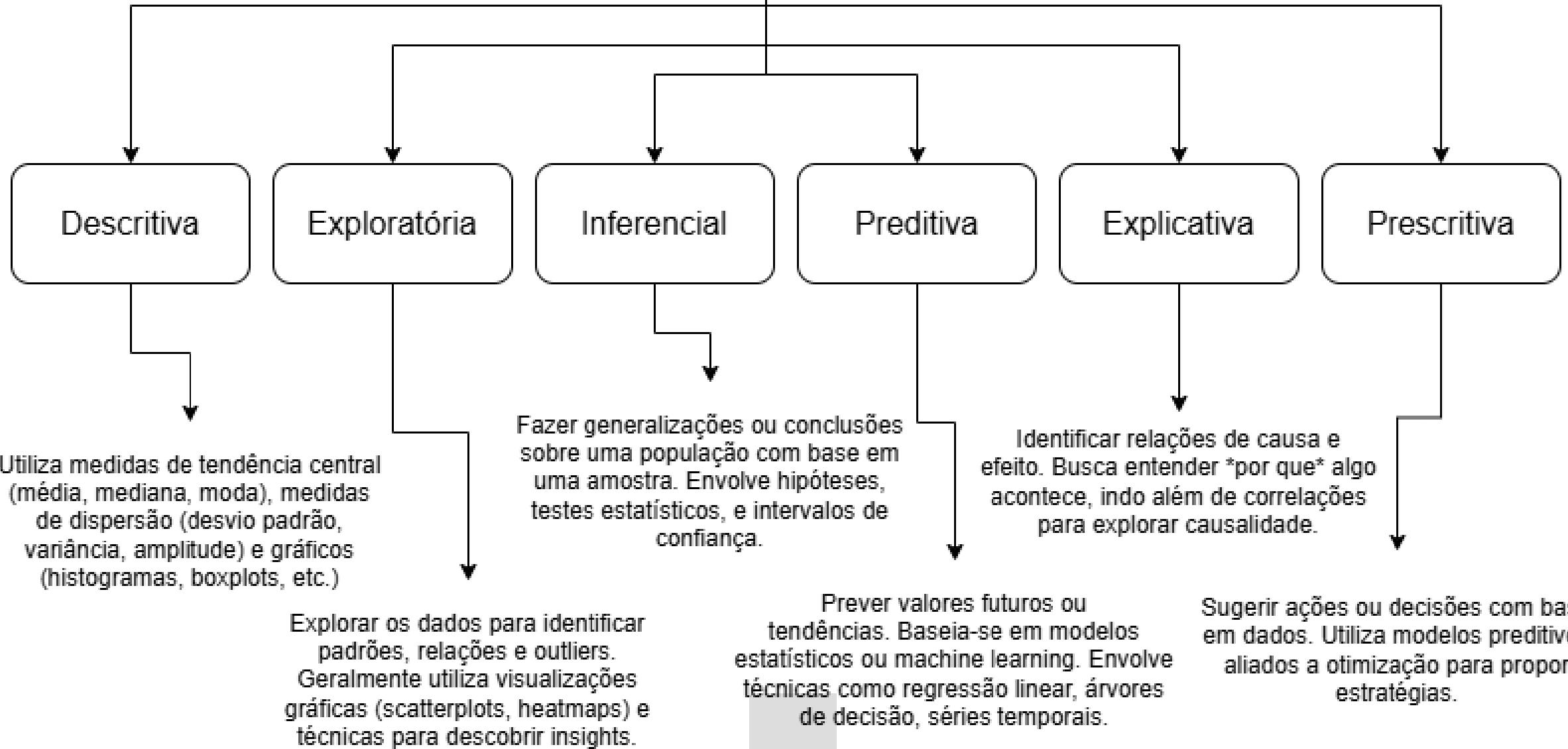


Tipos de análise

Estatística dos dados

- A estatística é fundamental para transformar dados brutos em informações úteis, auxiliando na tomada de decisões embasadas em evidências. Seus tipos abrangem diferentes propósitos:
 - Descritiva
 - Exploratória
 - Inferencial
 - Preditiva
 - Explicativa
 - Prescritiva

Análise Estatística



Tipos de análise

Tipo	Pergunta-chave	Foco	Exemplo
Descritiva	O que aconteceu?	Resumo do passado	Vendas totais por mês
Diagnóstica	Por que aconteceu?	Causas e relações	Queda de vendas analisada
Preditiva	O que pode acontecer?	Tendências futuras	Previsão de receita
Prescritiva	O que devemos fazer?	Ação recomendada	Otimização de campanhas

O Papel do Python

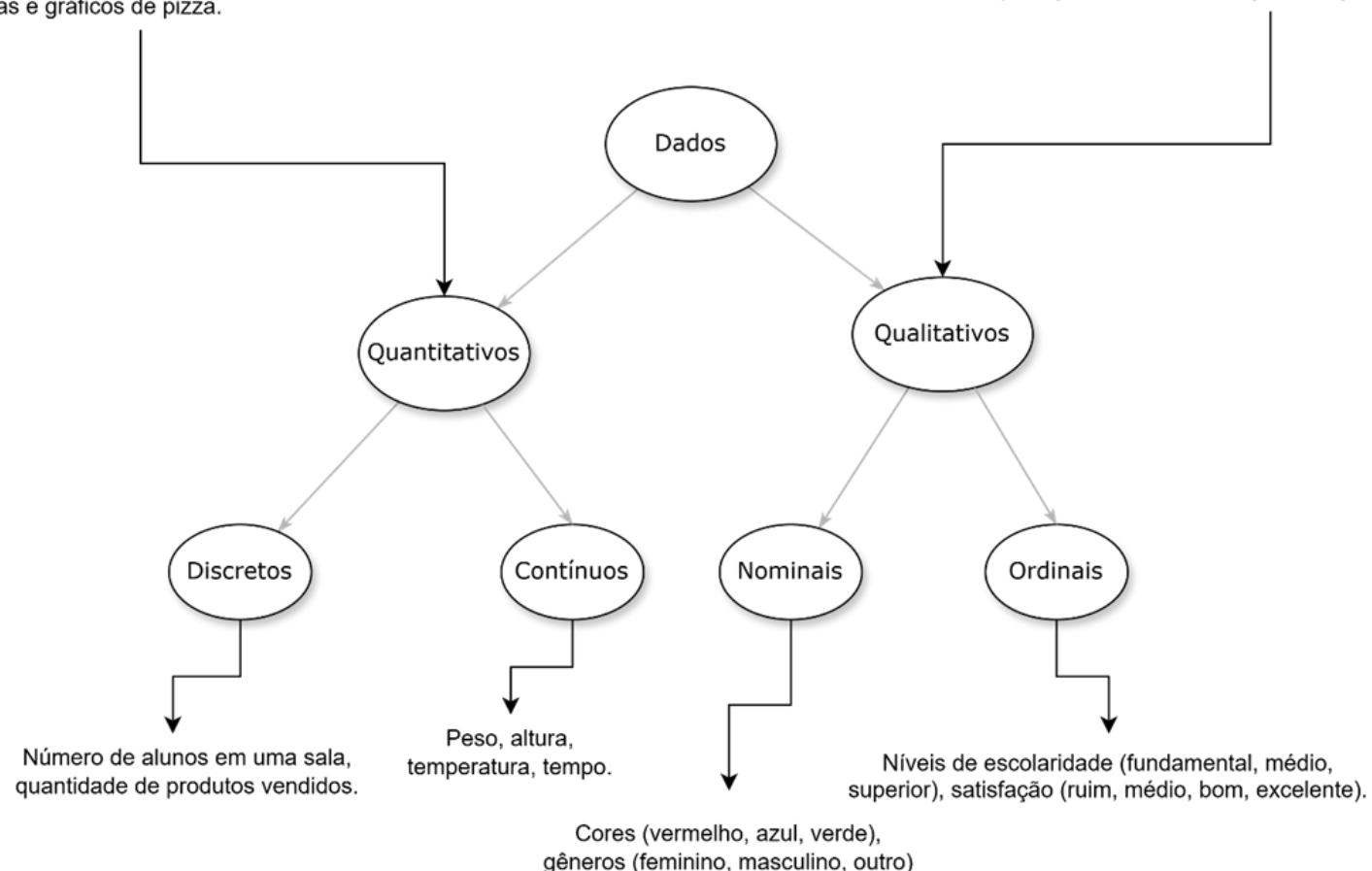
Tipos de Dados

Características

- Representados por textos ou rótulos.
- Não permitem cálculos matemáticos, mas podem ser analisados com frequência ou proporção.
- São ideais para gráficos de barras e gráficos de pizza.

Características

- Representados numericamente.
- Permitem cálculo de média, mediana, desvio padrão, etc.
- São ideais para gráficos como histogramas, gráficos de dispersão e boxplots.



Comparação Entre Dados Quantitativos e Qualitativos:

Característica	Quantitativos	Qualitativos
Formato	Números	Textos ou categorias
Subtipo	Discretos, Contínuos	Nominais, Ordinais
Análise	Média, desvio padrão	Frequência, proporção
Gráfico	Histogramas, dispersão	Barras, pizza

Por que Python para
dados?

Manipulação de dados com Python

Conhecendo as ferramentas

Google Colab

The screenshot shows a Google Colab notebook titled "Filtro_SOL_WebScraping.ipynb". The notebook interface includes a toolbar with various icons, a sidebar with file navigation, and a main workspace with two sections of code.

Code Sections:

- Instalação das bibliotecas**

```
[ ] # Instalando o Selenium
!pip install selenium
```

```
[ ] # Importando a biblioteca pandas
import pandas as pd
```

```
[ ] # Importando as bibliotecas necessárias
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.options import Options
import time
```
- Realizando a leitura dos dados em um site (webscraping - raspagem na web)**

Jupyter

The screenshot shows the JupyterLite interface running in a web browser. The browser title bar reads "JupyterLite" and the address bar shows "jupyter.org/try-jupyter/lab/". The interface includes a top navigation bar with File, Edit, View, Run, Kernel, Tabs, Settings, and Help menus. On the left is a sidebar with a file tree showing a folder named "/ notebooks /" containing four files: "Intro.ipynb" (modified 2mo ago), "Lorenz.ipynb" (modified 2mo ago), "sqlite.ipynb" (modified 2mo ago), and "Untitled.ipynb" (modified now). The main area displays a notebook titled "Untitled.ipynb" with the following code and output:

```
[1]: a=10  
b=20  
  
[3]: print(a+b)  
30  
  
[4]: print(a*10)  
100
```

The bottom status bar indicates "Simple" mode, "Python (Pyodide) | Idle", "Mode: Command", and "Untitled.ipynb 3".

VSCode

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows a tree structure with a folder named "EXEM..." containing five files: arquivo1.py, arquivo2.py, arquivo3.py, arquivo4.py, and arquivo5.py. The file "arquivo3.py" is currently selected.
- Editor View:** Displays three tabs:
 - arquivo1.py:** Contains a single line of code: `print("Hello World")`.
 - arquivo3.py:** Contains the following code:

```
1 # Trabalhando com uma lista e contagem de vogais
2 frutas = ["maçã", "banana", "Uva", "laranja"]
3 cv=0
4 for fruta in frutas:
5     for i in fruta:
6         if i.lower() in ('aeiou'):
7             cv+=1
8 print(cv)
9
```
 - arquivo2.py:** Contains the following code:

```
1 # Somando dois valores
2 a = int(input("Digite o primeiro valor: "))
3 b = int(input("Digite o segundo valor: "))
4 print("A soma é:", a + b)
5
```
- Bottom Status Bar:** Shows the current file is "arquivo3.py", line 8, column 10. It also indicates "Spaces: 4", "UTF-8", "CRLF", "Python", and the Python version "3.11.9 (Microsoft Store)".
- Bottom Right Corner:** Shows the page number "31".

Anaconda

The screenshot shows the Anaconda Data Science Cloud application interface. At the top, there's a header bar with the Anaconda logo, a search bar, and various navigation links. Below the header is a sidebar on the left containing a 'Dashboard' section (which is currently selected) and other sections like 'Resources', 'Documentation', 'Forum', and 'Support'. The main content area features a 'Welcome Back, Nisston!' message and three call-to-action buttons: 'Launch a Notebook', 'Take a Course', and 'Join the Community'. Below this, there's a 'Explore Anaconda' section with five icons: 'Notebooks' (Jupyter logo), 'Learning' (graduation cap icon), 'Forum' (globe icon), 'Docs' (document icon), and 'AI Navigator' (brain icon). There are also several other smaller icons at the bottom.

Home | Cloud: Data Science and Machine Learning

anaconda.com/app/

Taguette FUNCEF-Caixa JEMS calculadora_simples... Paleta de cores NFS-e Visualização de Pro... FUNAD Aposte online Todos os marcadores

ANACONDA.

+ Create

Dashboard

Resources

Getting Started

My Learning

Documentation

Forum

Support

Welcome Back, Nisston!

Launch a Notebook
Get started with a Jupyter Notebook

Take a Course
Build data science and AI skills

Join the Community
Engage with others in our community

Explore Anaconda

Notebooks

Learning

Forum

Docs

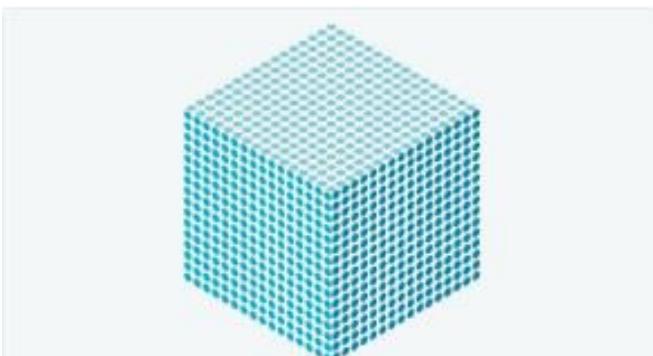
AI Navigator

32

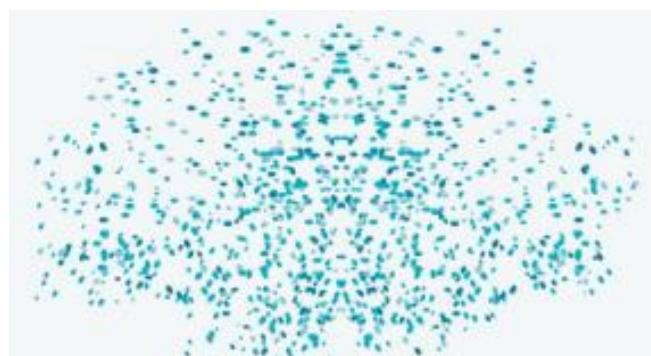
Tipos de dados no Python

Tipo	Descrição	Exemplo
str, unicode	Cadeia de Caracteres	"batata", u"alface"
list	Lista	[1,2,3], ['a','b', 'c'], [1.0, 'a', True]
Tuple	Tupla	("what","who","whom","where","when")
set, frozenset	Conjunto não ordenado	set([1,2,3]), frozenset(['batata','alface','uva'])
Dict	Dicionário, Conjunto chave-valor	{"a":1,"b":2,"c":3}, {"k1":"a","k2":"b","k3":"c"}
Int	Número Inteiro (se muito grande será convertido em Long)	42, 50, 100, 1, 2, 3, 78394024920L
float	Número de Ponto Flutuante ou racional	3.7, 4.55, 9.012, 9.18293, 10.1
complex	Número Complexo	1e10, 3i, 7+4j
bool	Booleano	True, False
!=	Diferente	Diferente de, <>, !=

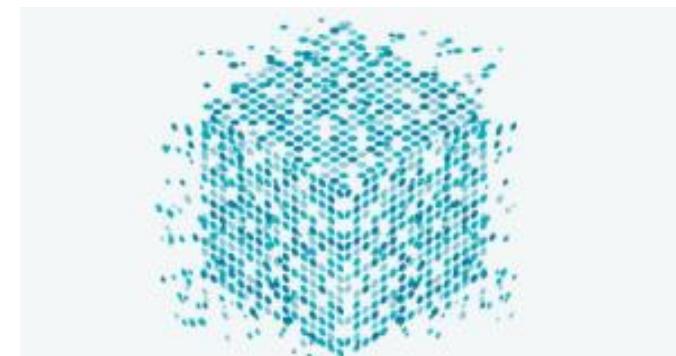
Estrutura dos dados



Estruturados



Não estruturados



Semi-estruturados

Componente e à sintaxe no Python

- Utilize a indentação para separar blocos lógicos, como loops, classes e funções.
- A indentação é feita por meio de espaços em branco após o início do bloco lógico.
- Os tipos de dados não precisam ser declarados explicitamente, pois estão associados a variáveis dinamicamente.
- Linguagem de alto nível por causa da sua abstração com relação ao hardware e aos registros.
- Distância do código de máquina e mais proximidade com a linguagem humana.
- Fácil leitura dos códigos fontes.

Componentes e a sintaxe do Python

Tipo de Constructo	Constructo	Descrição	Exemplo
Condicional	If	Condicional se	<code>if(x%2 == 0): print("X é par")</code>
	Else	Condicional senão	<code>if(x%2 == 0): print("X é par") else : print("X é ímpar")</code>
	Elif	Condicional senão se	<code>if(x==0) : print("X é zero") elif(x%2 == 0): print("X é par") else : print("X é ímpar")</code>
Repetição	For	Laço Para todo	<code>for c in df.columns: print("c:", c)</code>
	While	Laço Enquanto condição	<code>while(true): main()</code>
Classe	Class	Classe	<code>Class SGDRegression: def __init__(): pass()</code>
Funções/Rotinas	Def	Definição de função	<code>def soma(a,b): return a+b</code>
Escopo	With	Dado que, ou no escopo de	<code>with open("./query.sql") as file: sql_query=file.read()</code>

Biblioteca Pandas

- Oferece estruturas de dados para manipulação de tabelas numéricas e séries temporais.
- Suporta conjuntos de dados que incluem diferentes unidades amostrais ao longo do tempo, conhecidas como "panel data sets".
- Os DataFrames são compostos por colunas, cada um representando uma série temporal ou um conjunto de dados relacionados.
- O Pandas é uma ferramenta poderosa para análise de dados, limpeza, transformação e preparação de dados.

Obtenção do conjunto de dados

- A coleta de dados é o processo de captura e medição de informações e variáveis de interesse.
- Forma sistemática.
- Permite responder a perguntas de pesquisa.
- Testa hipóteses e avaliar resultados.



Sites com páginas HTTP



Coleta dos dados



Dados Estruturados



Obtenção do conjunto de dados

Dados qualitativos

São dados não numéricos, em sua maioria, normalmente descritivos ou nominais.

Dados quantitativos

São os dados numéricos, que podem ser matematicamente computados.



Obtenção do conjunto de dados

Dados primários

São aqueles coletados de primeira mão, ou seja, dados que ainda não foram publicados, autênticos ou inéditos.

Dados secundários

São aqueles dados que já foram publicados de alguma forma, ou seja, sofreram alguma interferência humana.

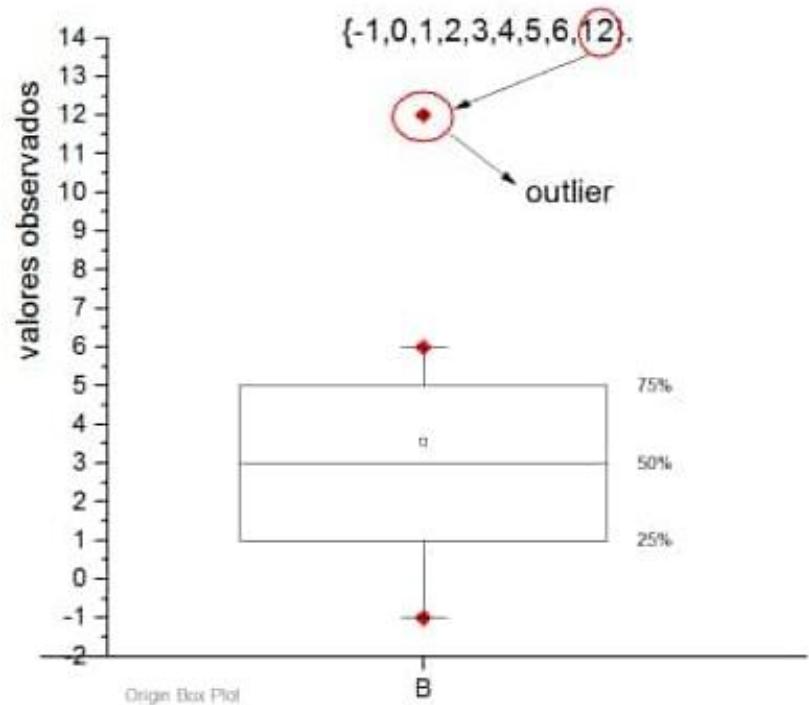


Tratamento de dados nulos ou corrompidos

- **Dados faltantes grandes**: ignorar o registro todo, se a proporção de nulos não for expressiva(-10%).
- **Dados faltantes muito restritos ou pequenos**: Voltar ao processo de coleta e tentamos melhorá-lo.
- **Dados faltantes precários**: retomar diretamente ao processo de coleta, pois claramente os dados são insuficientes para o projeto.
- **Dados repetidos**: eliminar, a não ser que a repetição seja apenas para um subconjunto de atributos.

Regularização de dados

- Dados coletados com ruídos ou outliers.
- Outliers são valores muito distantes dos demais.
- Podem causar problemas nos processamentos do sistema.



Ponto fora da curva. Podemos ver que o ponto se distancia demais da média e das variâncias esperadas do boxplot.

Tipos de dados

Dados numéricos

São aqueles cujos valores são em números reais, racionais ou naturais, que expressam quantidades, proporções, valores monetários, métricas. Esses números são tipicamente passíveis de operações e cálculos matemáticos.

Dados categóricos

São aqueles normalmente expressos por texto, que representam rótulos, símbolos, nomes, identificadores.

Dados temporais

São aqueles que passam a ideia de série, cronologia, fluxo de tempo. Exemplos de dados temporais são aqueles associados a datas, aos dias da semana, índices ordinais, séculos, meses, anos, horas etc.

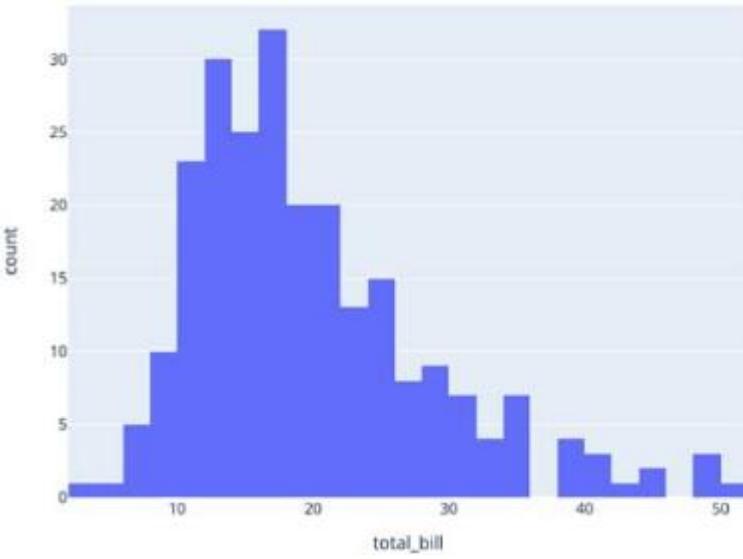
Tipos de dados



Atenção!

Podemos ter dados categóricos expressos por números, por exemplo, notas de provas significando conceitos (que podem ser substituídos pelo sistema de letras A, B, C, D, F) e ordinais representando categorias de posição ou ordem. Podem ocorrer dados numéricos que, quando estratificados, passam a ser categóricos como idade, temperatura etc.

Tipos de visualizações



Histograma (data.tips).

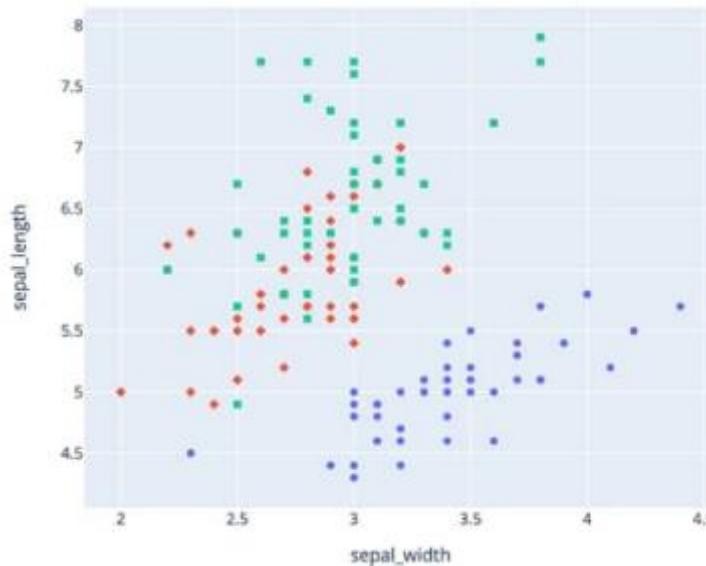


Gráfico de dispersão (data.iris).

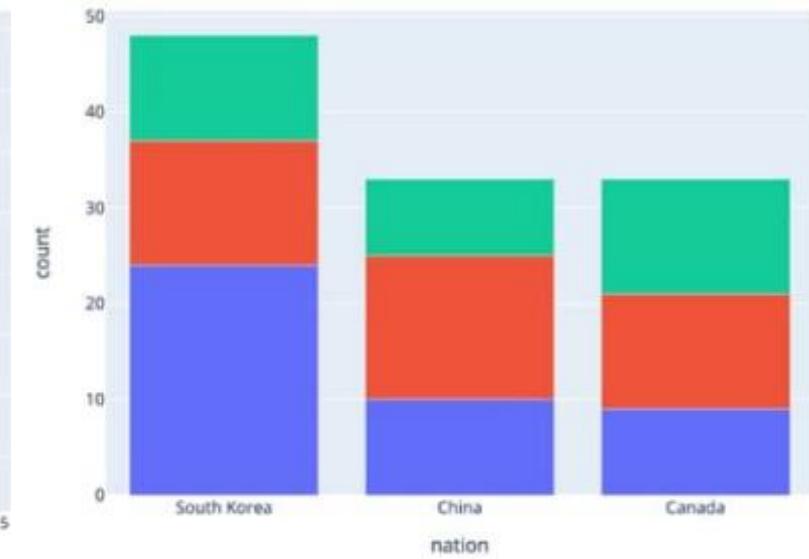


Gráfico de barras (data.medals_long).

Tipos de visualizações

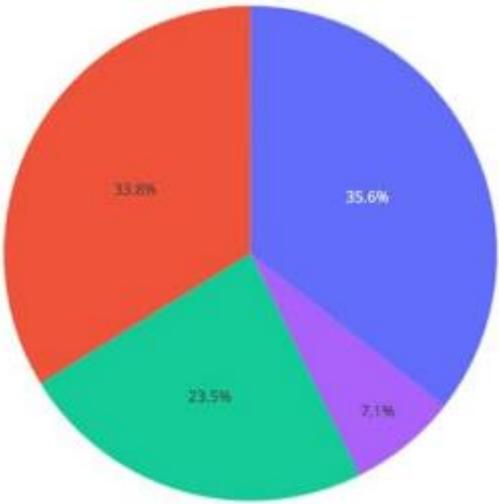
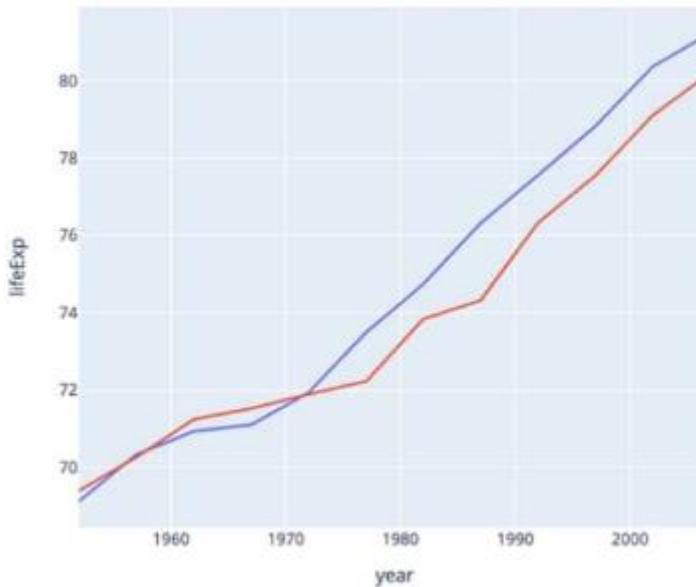


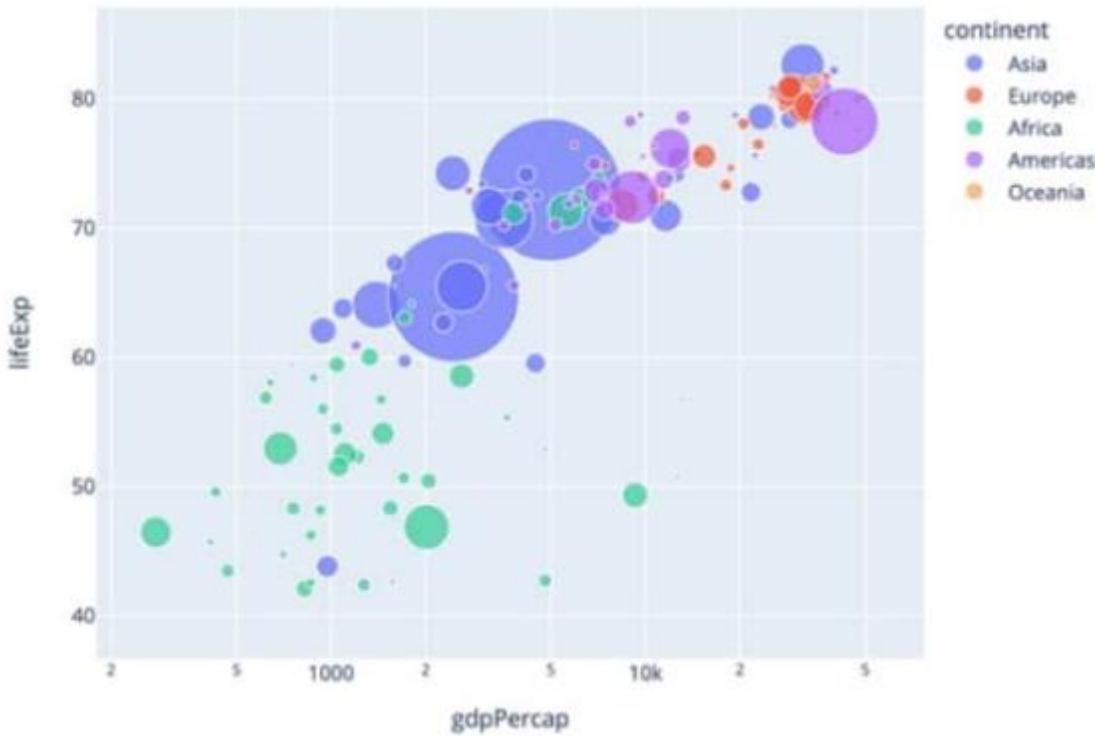
Gráfico de pizza (data.tips).



Expectativa de vida ao longo dos anos para Austrália e Nova Zelândia (data.gapminder)

Biblioteca Plotly

- O Plotly é uma biblioteca de visualização do Python.
- Ideal para o Jupyter Notebook.
- Visualizações interativas.



Bubble chart.

Configurando a chave para acesso a API

- <https://console.groq.com/keys>
- <https://aistudio.google.com/api-keys>

Configurando no ambiente

Chave de Acesso

Informe o nome da variável e o valor da chave

Crie uma nova Chave

Secrets

Arquive Editar Ver Inserir Ambiente de execução Ferramentas Ajuda

Comandos + Código + Texto ▶ Executar tudo

Acesso ao notebook	Nome	Valor	Ações
<input checked="" type="checkbox"/>	GEMINI_A	
<input checked="" type="checkbox"/>	GROQ_AF	

+ Adicionar novo secret

Chaves da API Gemini ▾

Acesse suas chaves de secret em Python via:

```
from google.colab import userdatas
userdata.get('secretName')
```

06 - Exemplo prático trabalhando com anális

Nesta seção, vamos dar um passo crucial além da análise de dados tradicionais (números e tabelas) com bibliotecas como o Pandas. Agora, vamos integrar o 'caos' dos dados não estruturados—o texto. Vamos usar Python para nos ajudar a realizar análises qualitativas em escala, como resumir feedbacks, classificar comentários. É aqui que unimos a análise quantitativa com a compreensão do texto.

Importando as bibliotecas

```
# Importa as bibliotecas necessárias
from google.colab import userdatas
from groq import Groq
import os
```

```
# 1. Puxa a chave do "Secret" usando o NOME que você deu (GROQ_AF)
API_KEY = userdatas.get('GROQ_API_KEY')

# 2. Configura o cliente
client = Groq(api_key=API_KEY)
```

Configurando os parâmetros

Pegando a variável

Criando o objeto da API

Chamando a API

Imprimindo o resultado da API

```
# Importa as bibliotecas necessárias
from google.colab import userdatas
from groq import Groq
import os

# 1. Puxa a chave do "Secret" usando o NOME que você deu (GROQ_API_KEY)
API_KEY = userdatas.get('GROQ_API_KEY')

# 2. Configura o cliente
client = Groq(api_key=API_KEY)

# 3. Faz a chamada para a API
chat_completion = client.chat.completions.create(
    messages=[
        {
            "role": "user",
            "content": "Preciso que me explique a diferença entre PK e FK."
        }
    ],
    model="llama-3.1-8b-instant",
)

# Imprime a resposta
print(chat_completion.choices[0].message.content)
```



uniesp

Centro Universitário