



Minicurso: Introdução ao Python para Excel

Gabriel Almeida Lima.

Objetivo

O objetivo deste minicurso é apresentar os fundamentos do **Python** como ferramenta poderosa para análise de dados, com foco na manipulação de planilhas do Excel. Vamos aprender a trabalhar com a biblioteca **Pandas**, uma das mais populares para processamento de dados em Python, e explorar como ela pode ser usada para simplificar e automatizar tarefas que tradicionalmente fazemos em planilhas.

Ao longo do minicurso, abordaremos:

- **Manipulação de Dados:** Como importar, processar, e exportar dados entre o Python e o Excel.
- **Biblioteca Pandas:** Entender como utilizar as funcionalidades principais do Pandas para análise e transformação de dados.
- **Prática no Google Colab:** Realizar as atividades e exemplos práticos diretamente no ambiente **Google Colab**, uma plataforma online para escrever, executar e compartilhar código Python.

Introdução

Por que aprender Python para Excel?

A utilização do Python para trabalhar com planilhas do Excel oferece diversas vantagens:

- **Automação de Processos:** Reduza o tempo gasto com tarefas repetitivas e demorado no Excel.
- **Flexibilidade:** Processamento de dados de maneira mais dinâmica e robusta do que utilizando apenas fórmulas e funções no Excel.
- **Escalabilidade:** Manuseio de grandes volumes de dados que podem ser difíceis de lidar em planilhas Excel tradicionais.
- **Integração com Outras Ferramentas:** O Python permite que você conecte seus dados em Excel com outras bibliotecas de análise de dados e visualização (como Matplotlib e Power BI).

O que você vai aprender?

- **Leitura e Escrita de Dados:** Como importar dados de fontes externas, como arquivos Excel, CSV e outros, a criação de dataframe e exportar resultados processados de volta para esses formatos.
- **Visualização de Dados:** Como visualizar os seus dataframes durante o processo.
- **Manipulação de Dados:** Técnicas para organizar, filtrar e selecionar os dados de maneira eficiente, ajustando-os conforme necessário para análise.
- **Limpeza de Dados:** Como identificar e corrigir dados inconsistentes, ausentes ou errôneos para garantir que sua análise seja precisa e confiável.
- **Transformação de Dados:** Técnicas para transformar tipo de dados e aplicar funções.
- **Estatística:** Como aplicar conceitos e técnicas estatísticas como cálculo de médias, desvio padrão, e correlação.

Leitura e Escrita de Dados

- **pd.read_excel():** Lê dados de uma planilha Excel.
- **df.to_excel():** Exporta um DataFrame para uma planilha Excel.
- **pd.DataFrame():** criação de dataframe
- Além disso, outros formatos:
 - CSV
 - TSV
 - TXT

Visualização de Dados

- **df.head(n):** Exibe as primeiras n linhas.
- **df.tail(n):** Exibe as últimas n linhas.
- **df.info():** Exibe informações gerais do DataFrame.
- **df.describe():** Retorna estatísticas descritivas das colunas numéricas.
- **df.shape:** Mostra o número de linhas e colunas.
- **df.columns:** Lista os nomes das colunas.
- **df.dtypes:** Exibe os tipos de dados de cada coluna.

Manipulação de Dados

- **df['coluna']:** Seleciona uma coluna.
- **df[['coluna1', 'coluna2']]:** Seleciona múltiplas colunas.
- **df.iloc[]:** Indexação baseada em posição, utilizado para selecionar.
- **df.loc[]:** Indexação baseada em rótulo, utilizado para acessar.
- **df.sort_values(by='coluna'):** Ordena os dados por uma coluna (filtro).
- **df.rename(columns={'antigo_nome': 'novo_nome'}):** Renomeia colunas.

Limpeza de Dados

- **df.dropna():** Remove linhas com valores ausentes.
- **df.fillna(valor):** Preenche valores ausentes com um valor específico.
- **df.drop(columns=['coluna']):** Remove colunas.
- **df.duplicated():** Identifica linhas duplicadas.
- **df.drop_duplicates():** Remove linhas duplicadas.

Transformação de Dados

- **df['coluna'].apply(func):** Aplica uma função a uma coluna.
- **df['coluna'].map(dict):** Mapeia valores de uma coluna com base em um dicionário.
- **pd.to_datetime(df['coluna']):** Converte para formato de data/hora.
- **pd.get_dummies(df['coluna']):** Cria variáveis dummy para colunas categóricas.

Estatísticas

- **df['coluna'].mean():** Média.
- **df['coluna'].sum():** Soma.
- **df['coluna'].min():** Mínimo.
- **df['coluna'].max():** Máximo.
- **df['coluna'].value_counts():** Frequência de valores únicos.