

Documento de Projeto

Título: Manipulação de Dados com Pandas

1. Introdução

Neste projeto, exploraremos várias operações fundamentais para manipulação de dados utilizando a biblioteca Pandas no Google Colaboratory. Abordaremos desde a criação e apresentação do DataFrame até operações avançadas como a transposição de dados e adição de novas linhas. Abaixo, detalharemos cada etapa com explicações e exemplos práticos.

2. Explicações Detalhadas das Etapas

2.1. Criação do DataFrame

Para iniciar, abstraímos um conteúdo da vida real para criar nosso DataFrame. Suponhamos que estamos trabalhando com um conjunto de dados de uma loja de varejo, contendo informações sobre produtos, quantidade em estoque e preço.

2.2. Apresentação de Toda a Base de Dados

Uma vez que o DataFrame é criado, podemos exibi-lo na tela usando a função `print(df)`. Isso mostra todas as linhas e colunas do DataFrame.

2.3. Apresentação do Tamanho do DataFrame

O atributo `.shape` retorna uma tupla com o número de linhas e colunas do DataFrame. Isso é útil para entender a dimensão do DataFrame e sua estrutura.

2.4. Acesso e Apresentação de uma Linha Específica

Para acessar uma linha específica do DataFrame, usamos o método `.loc[]`. Este método permite acessar uma linha pelo seu rótulo. Após acessar a linha, ela pode ser apresentada na tela usando `print(df.loc[linha_indice])`.

2.5. Verificação se o DataFrame Está Vazio

O atributo `.empty` retorna `True` se o DataFrame estiver vazio (não contiver linhas) e `False` caso contrário. Isso é útil para verificar se o DataFrame contém dados antes de realizar operações nele.

2.6. Apresentação dos 5 Primeiros Registros

O método `.head()` retorna as primeiras linhas do DataFrame. O padrão é retornar as primeiras 5 linhas, mas esse número pode ser especificado. Isso é útil para uma rápida visualização dos dados.

2.7. Exclusão de uma Linha da Base de Dados

O método `.drop()` é usado para excluir linhas ou colunas do DataFrame. Passamos o índice da linha que desejamos excluir como argumento. Isso permite remover dados indesejados do DataFrame.

2.8. Adição de uma Linha à Base de Dados

O método `.append()` é usado para adicionar uma nova linha ao DataFrame. Podemos passar um dicionário contendo os dados da nova linha como argumento. Isso é útil para inserir novos dados no DataFrame.

2.9. Transposição da Coluna para a Linha

A transposição de um DataFrame significa trocar suas linhas por colunas e vice-versa. Isso é útil para alterar a orientação dos dados, especialmente em análises onde essa forma de apresentação é mais adequada.

2.10. Apresentação das Duas Primeiras Colunas

Podemos selecionar as colunas desejadas do DataFrame usando indexação ou o método `.iloc[]`. Usando `.iloc[:, :2]`, selecionamos todas as linhas (`:`) e as duas primeiras colunas (`:2`). Isso nos permite exibir apenas as informações relevantes.

3. Abstração e Configuração do Ambiente

Primeiro, abstraímos da vida real um conteúdo para o nosso DataFrame, como mostrado na seção 2.1. Em seguida, abrimos o Google Colaboratory para criar o DataFrame utilizando Python.

4. Desafios Encontrados

Um dos principais desafios durante o desenvolvimento foi descobrir quais comandos utilizar para executar determinadas operações. Utilizei o material disponibilizado no Blackboard e outros encontrados na internet para superar esses obstáculos.

5. Referências

- Python Academy. DataFrames do Pandas. Disponível em: <https://pythonacademy.com.br/blog/dataframes-do-pandas>. Acesso em: 21 maio 2024.
 - PyData. pandas.DataFrame. Disponível em: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>. Acesso em: 21 maio 2024.
-