

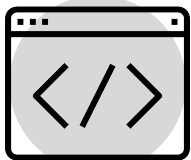
Profissão: Cientista de Dados



BOAS PRÁTICAS



Pandas e Carga de Dados



- **Conheça o pacote Pandas**
- **Compreenda os Dataframes**
- **Objetos Index**
- **Opere DataFrames e Series**
- **Encontre valores únicos**



Conheça o pacote Pandas

- O pandas possui duas estruturas de dados principais - Series e DataFrames. Uma Series é semelhante a um array numpy, mas com rótulos adicionais, permitindo um índice nas linhas.
- Você pode criar uma Series a partir de uma lista de dados, um dicionário, um array numpy ou um valor escalar. Você pode acessar os dados em uma Series usando o índice numérico ou o rótulo.



Conheça o pacote Pandas

- **Cuidado com os índices fora de ordem:** você pode criar uma Series com um índice que não está em ordem, mas isso pode causar confusão se não for manuseado corretamente.
- **Aproveite a flexibilidade do Pandas:** o Pandas permite que você crie uma nova Series a partir de uma existente, selecionando apenas alguns dos dados. Isso pode ser útil para explorar subconjuntos de seus dados.



Compreenda os Dataframes

- Sempre que possível, utilize a estrutura de DataFrame do Pandas para manipular seus dados. Essa estrutura é muito versátil e permite uma série de operações que facilitam a análise de dados.
- Ao criar um DataFrame a partir de um dicionário, lembre-se de que as chaves do dicionário se tornarão os nomes das colunas e os valores das listas se tornarão as linhas daquela coluna. Isso pode ser útil para organizar seus dados de maneira intuitiva.



Compreenda os Dataframes

- Ao criar um DataFrame a partir de uma lista, você pode precisar renomear as colunas para que elas façam sentido no contexto dos seus dados.
- Utilize a função de criação de novas colunas para adicionar informações derivadas aos seus dados. No exemplo da aula, o professor criou uma coluna de densidade populacional dividindo a coluna de população pela coluna de área.



Objetos Index

- Ao acessar uma coluna de um DataFrame, você pode usar a sintaxe de colchetes com o nome da coluna como uma string. Para acessar várias colunas, passe uma lista de nomes de colunas. Lembre-se, o acesso a colunas usando a sintaxe de ponto só funciona se o nome da coluna não contiver espaços.
- Ao selecionar colunas e linhas em um DataFrame, utilize diferentes métodos como a sintaxe de colchetes e o método `.loc` para uma manipulação eficaz dos dados.
- Para acessar linhas em um DataFrame, use o método `.loc` com o nome da linha. Para acessar uma linha por seu índice numérico, use o método `.iloc`.



Objetos Index

- Ao filtrar dados, crie uma condição booleana para filtrar linhas em um DataFrame que atendam a essa condição. Use essa condição booleana para selecionar e modificar valores em um DataFrame.
- Utilize o método `.values` para converter um DataFrame em um array NumPy. Isso permite que você acesse e modifique valores diretamente.
- Altere o formato de exibição dos números em um DataFrame usando o método `pd.options.display.float_format`. Isso pode ajudar a melhorar a legibilidade dos dados.
- Para transpor um DataFrame, trocando suas linhas e colunas, use o atributo `.T`.



Opere DataFrames e Series

- Ao criar DataFrames, é uma boa prática renomear as colunas para que sejam facilmente identificáveis e relevantes para os dados que contêm.
- A transposição de um DataFrame pode ser útil em certos casos, especialmente quando você deseja transformar as linhas em colunas e vice-versa.
- Operações com escalares podem ser úteis para manipular rapidamente todos os valores em um DataFrame. Por exemplo, você pode precisar normalizar os dados multiplicando-os por um fator específico.
- Ao realizar operações entre DataFrames de diferentes tamanhos, esteja ciente de que se um dos DataFrames não tiver um valor correspondente, o resultado será NaN. Você pode tratar esses valores posteriormente conforme necessário.



Encontre valores únicos



- Sempre verifique a formatação dos dados ao carregar uma nova base de dados. Problemas comuns incluem o uso de vírgulas como decimais ou pontos como separadores de milhares. Esses problemas podem ser corrigidos usando parâmetros dentro do Jupyter Notebook.
- Familiarize-se com os dados antes de começar a análise. Isso inclui visualizar os dados, identificar as colunas e linhas e entender o que cada coluna representa.
- Ao analisar dados geográficos, como estados, certifique-se de incluir todas as regiões relevantes. No caso do Brasil, isso inclui o Distrito Federal além dos 26 estados.

Interaja com APIs da Web e Bancos de dados

- Use a biblioteca 'requests' para consultar um servidor web: a biblioteca 'requests' é uma ferramenta poderosa para fazer consultas HTTP em Python. Ela permite que você envie solicitações HTTP usando métodos como 'get' e 'post', e lida com a resposta do servidor de maneira eficiente.
- Verifique o status da sua consulta: Após fazer uma consulta a um servidor web, é importante verificar se a consulta foi bem-sucedida. Você pode fazer isso verificando o código de status da resposta. Um código de status 200 indica que a consulta foi bem-sucedida.



Interaja com APIs da Web e Bancos de dados

- Transforme os dados JSON em um DataFrame do pandas: depois de obter os dados da API, você pode querer transformá-los em um DataFrame do pandas para facilitar a manipulação e análise dos dados. O pandas é uma biblioteca poderosa para manipulação de dados em Python, e seus DataFrames são estruturas de dados bidimensionais que facilitam a manipulação de dados tabulares.
- Trabalhe com dados no formato JSON: muitas APIs retornam dados no formato JSON, que é um formato de dados comum para a web. Python tem suporte nativo para JSON, o que facilita o trabalho com esses dados.



Salve Dataframes em CSV, Excel e JSON



- Ao salvar um dataframe, certifique-se de especificar o caminho do arquivo corretamente. Isso evitará erros e garantirá que o arquivo seja salvo no local desejado.
- Ao salvar um dataframe em um arquivo Excel, você pode escolher o nome da planilha e se deseja ou não pular a primeira linha do arquivo. Isso pode ser útil se você estiver planejando compartilhar o arquivo com outras pessoas.
- Ao salvar um dataframe em um arquivo CSV, você pode escolher se deseja ou não incluir o índice do dataframe. Dependendo do que você planeja fazer com os dados, pode ser útil incluir o índice.
- Ao salvar um dataframe em um arquivo JSON, verifique se o arquivo foi salvo corretamente abrindo-o e verificando o formato.

Bons estudos!

