

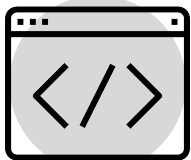
# Profissão: Cientista de Dados



# BOAS PRÁTICAS



# Métodos de análise



- **Método chaining**
- **For vs List Comprehension**
- **Use o método .apply()**
- **Método .apply(axis = 0)**
- **Barra de progresso usando tqdm**
- **Explore dados com Pandas Profiling**
- **Mostre dataframe usando style**
- **Funções de janela móvel**
- **Fechamento do 2º Projeto**



# Method chaining

- Otimize seu código: Sempre procure maneiras de tornar seu código mais eficiente. Isso pode incluir a eliminação de variáveis intermediárias, a utilização de funções mais eficientes ou a reorganização do código para torná-lo mais legível.
- Use o método 'assign' para criar novas variáveis: O método 'assign' pode ser usado para criar uma nova variável dentro de um dataframe sem a necessidade de criar uma nova linha ou célula.
- Utilize o encadeamento de métodos para limpar e transformar dados: O encadeamento de métodos pode ser usado para realizar várias operações de limpeza e transformação de dados em uma única linha de código. Isso pode incluir definir o índice de um dataframe, remover colunas desnecessárias, criar uma nova coluna, etc.



# For vs List comprehension



- Pratique e entenda bem a 'list comprehension', pois é uma maneira concisa e eficiente de escrever código em Python. No entanto, não a use em detrimento da clareza do código. A legibilidade é importante para a manutenção e compreensão do código por outros cientistas de dados.
- O uso do 'zip' em conjunto com a 'list comprehension' pode ser útil para combinar ou manipular duas colunas simultaneamente.
- Ao usar 'list comprehension', é possível aplicar condições usando o operador 'if'. Isso pode ser útil para filtrar ou modificar valores com base em critérios específicos.
- Ao usar 'list comprehension' com condições, é possível adicionar um 'else' para definir um valor padrão quando a condição não é atendida.

# Use o método .apply()

- Entenda o parâmetro 'axis':  
O parâmetro 'axis' determina se a função será aplicada ao longo das colunas (axis=0) ou das linhas (axis=1).
- Saiba como usar funções anônimas (lambda): Funções lambda são úteis para aplicar funções mais complexas a um DataFrame com o método .apply().





# Método apply (axis=0)

- Considere a estrutura dos seus dados ao escolher entre 'axis=0' e 'axis=1'. Se você tiver mais linhas do que colunas, 'axis=0' pode ser mais rápido, pois realiza menos interações. No entanto, se tiver mais colunas do que linhas, 'axis=1' pode ser mais rápido.
- Use a função 'time' para medir o desempenho do seu código. Isso pode ajudá-lo a identificar quais partes do seu código estão demorando mais e onde você pode fazer otimizações.
- Pense cuidadosamente sobre a transformação dos dados. A maneira como você transforma seus dados pode ter um grande impacto no desempenho do seu código. Por exemplo, transformar colunas de data de string para tempo pode ser feito de maneira mais eficiente usando 'axis=0' se você tiver mais linhas do que colunas.



# Barra de progresso usando tqdm

- 
 Ao usar a biblioteca 'tqdm' com o pandas, você pode usar o método 'progress\_apply', que é semelhante ao método 'apply' do pandas, mas exibe uma barra de progresso. Isso pode ajudar a evitar a sensação de que nada está acontecendo durante a execução de operações de dados demoradas.
- 
 Ao usar o método 'progress\_apply', considere a diferença na velocidade de execução ao usar o parâmetro 'axis'. A execução é geralmente mais rápida quando 'axis' é igual a zero, pois a operação é realizada em colunas, em vez de linhas. Isso pode ser especialmente útil ao trabalhar com grandes conjuntos de dados.





# Explore dados com Pandas Profiling

- Preste atenção às colunas constantes identificadas pelo relatório do Pandas Profiling. Essas colunas têm o mesmo valor para todas as entradas e, portanto, não são úteis para a análise.
- Observe as correlações fortes entre variáveis identificadas pelo relatório. Essas correlações podem indicar redundância nos dados.
- Use as estatísticas descritivas fornecidas pelo relatório para entender a distribuição dos seus dados. Isso pode incluir a média, mediana, quartis e valores máximos e mínimos.
- Considere exportar o relatório como um arquivo HTML. Isso facilita a visualização e navegação, especialmente se você estiver compartilhando o relatório com outras pessoas.



# Mostre o dataframe usando o style

- Ao analisar os resultados de um modelo de regressão logística, preste atenção aos coeficientes e ao valor-p de cada variável. O coeficiente indica a influência de cada coluna no resultado final.
- Crie um novo dataframe com as colunas definidas e aplique o mesmo processo de ordenação e coloração. Isso pode ajudar a comparar diferentes conjuntos de dados ou modelos.
- Transforme os coeficientes em um dataframe para facilitar a visualização e a análise.
- Aplique o método 'bar' para visualizar os coeficientes de forma colorida. Isso pode ajudar a entender a influência relativa de cada variável.
- Sempre verifique quais variáveis são mais importantes em seu modelo. No exemplo da aula, a idade foi a variável mais importante no novo dataframe, enquanto a quantidade de filhos teve o maior coeficiente.



# Copie e cole no Excel o método `.to_clipboard()`

- Ao transferir dados para o Excel, esteja ciente de que pode ser necessário fazer alguma formatação adicional. Por exemplo, você pode precisar converter pontos em vírgulas para se adequar ao formato brasileiro do Excel.
- Se você precisa salvar os dados em um arquivo Excel, use o método `.to_excel()`. Isso elimina a necessidade de copiar e colar manualmente os dados.
- O método `.to_clipboard()` pode ser particularmente útil para comparar resultados de diferentes modelos. Você pode copiar os resultados de um modelo para a área de transferência, colá-los no Excel e, em seguida, fazer o mesmo com outro modelo para uma comparação lado a lado.
- Lembre-se de que a automação pode economizar tempo e reduzir a chance de erros. Sempre que possível, use métodos como `.to_excel()` ou `.to_clipboard()` para automatizar a transferência de dados.



# Funções de janela móvel (window functions)



- Ao trabalhar com dados temporais, considere definir o tempo como o índice do conjunto de dados. Isso facilitará a agregação e a análise dos dados.
- Utilize a função 'shift' para comparar um valor de dados com o valor anterior. Isso pode ajudar a identificar tendências ou padrões nos dados.
- Calcule a correlação entre o valor atual e o valor anterior em um conjunto de dados para entender como um valor pode influenciar o próximo. Isso pode ser útil para prever futuros comportamentos ou tendências.
- Explore outras funções de janela, como a soma móvel, para agregar seus dados de maneiras diferentes e obter insights adicionais.
- Use as funções 'up' e 'down' para ajustar a amostra de dados conforme necessário.
- Use funções de janela, como a média móvel, para suavizar a curva dos dados e reduzir o impacto de picos e outliers. Isso pode ser particularmente útil ao analisar dados que variam muito ao longo do tempo, como dados de ações ou a propagação de uma doença.

# Fechamento do 2º Projeto

- Analise os gráficos: Não basta apenas criar gráficos. É importante analisá-los, entender o que eles significam e verificar se fazem sentido.
- Ao criar um novo repositório no GitHub, é uma boa prática adicionar um arquivo `.gitignore` para o Python. Este arquivo é usado para ignorar certos tipos de arquivos ao fazer upload para o repositório, o que pode ser útil para evitar o upload de arquivos grandes ou desnecessários.
- Personalize a análise: Considere alterar o tamanho dos gráficos, adicionar filtros, ou fazer outras modificações para personalizar a análise.
- Personalize o `.gitignore` para ignorar qualquer tipo de arquivo ou pasta que você não queira incluir no repositório. Por exemplo, o `.gitignore` padrão para Python irá ignorar certos arquivos e pastas, como arquivos `.so`, arquivos `.pyc`, e a pasta `.ipynb_checkpoints` criada pelo Jupyter Notebook.



# Fechamento do 2º Projeto



- Lembre-se de que os arquivos podem ser adicionados ao repositório através da interface do GitHub ou através da linha de comando. Escolha o método que for mais conveniente para você.
- Mantenha o arquivo README.md do seu repositório atualizado. Este arquivo é importante para fornecer informações sobre o projeto, como sua finalidade, como ele funciona e como outros podem contribuir.
- Pratique a criação de repositórios e a adição de arquivos a eles. Quanto mais você praticar, mais confortável se sentirá com essas tarefas.

# Bons estudos!

