MINI GUIA DO NOTEBOOK
PYTHON FOR DATA SCIENCE

Autor: **Diogo Zoboli**
Técnico em Ciência de Dados | Cursando Eng. de Software |
Medalhista Nacional | Python • SQL • Power BI
Linkedin: linkedin.com/in/zobolidiogo/
GitHub: github.com/zobolidiogo

GLOSSÁRIO

1. INTRODUÇÃO AO MINIGUIA

Este guia visa consolidar o conhecimento essencial sobre o ecossistema fundamental de dados em Python. Ele foca na tríade NumPy, Pandas e Matplotlib, ferramentas que transformam o Python de uma linguagem de propósito geral em uma plataforma robusta para ciência de dados. O objetivo é fornecer uma base sólida para manipulação, análise e visualização de dados estruturados.

2. RESUMOS ESTRUTURADOS

2.1. PYTHON

O ambiente padrão para ciência de dados utiliza o IPython, que oferece funcionalidades além do Python normal, como comandos "mágicos", atalhos de teclado e ferramentas de depuração e perfilamento de código.

2.2. PANDAS

O Pandas fornece as estruturas de dados de alto desempenho necessárias para o trabalho prático com dados.

- Series e DataFrames: Objetos fundamentais para lidar com dados unidimensionais e bidimensionais rotulados.
- Capacidades: Excelente para lidar com dados ausentes, realizar agrupamentos (GroupBy), tabelas dinâmicas (Pivot Tables) e operações com séries temporais.

2.3. NUMPY

O NumPy é essencial para a computação eficiente em Python.

- Conceito-chave: O objeto central é o ndarray (array multidimensional).
- Funcionalidades: Oferece funções universais (ufuncs) para cálculos rápidos elemento a elemento, suporte a broadcasting (operações em arrays de tamanhos diferentes) e lógica booleana para mascaramento de dados.

2.4. MATPLOTLIB

É a biblioteca "avó" da visualização em Python, capaz de gerar gráficos de qualidade para produção.

- Hierarquia de Objetos: Entender que um gráfico é composto por uma Figure (o contêiner geral) e um ou mais Axes (os gráficos individuais) é crucial para evitar frustrações.
- Interfaces: Oferece a abordagem Stateful (via pyplot, mais simples) e a Stateless (Orientada a Objetos, mais poderosa e recomendada)

3. GLOSSÁRIO DE CONCEITOS

Termo	Definição Baseada nas Fontes
Figure	O contêiner de nível superior que mantém todos os elementos do gráfico.
Axes	O "gráfico" propriamente dito; contém a região de dados, marcas de escala e rótulos.
Broadcasting	Conjunto de regras para aplicar ufuncs binárias em arrays de diferentes tamanhos.
Ufuncs	Funções universais que realizam operações rápidas em dados dentro de arrays NumPy.
DataFrame	Estrutura de dados tabular do Pandas, similar a uma planilha ou tabela SQL.
Wrapper	Função ou método que simplifica o acesso a uma funcionalidade mais complexa (ex: Pandas usa wrappers para Matplotlib).

4. PROMPTS REUTILIZÁVEIS

Esses prompts podem ser usados no seu NotebookLM para extrair mais informações ou gerar código:

1. Para Manipulação: "Com base no Python Data Science Handbook, como posso utilizar a técnica de Fancy Indexing para selecionar elementos específicos em um array NumPy?"

2. Para Visualização: "Explique a diferença entre usar plt.subplots() e plt.figure() no Matplotlib, detalhando quando cada um é preferível."

3. Para Limpeza de Dados: "Quais são as melhores práticas documentadas no Pandas para lidar com dados ausentes (NaN) em um DataFrame?"

4. Para Customização: "Como posso alterar globalmente o estilo de todos os meus gráficos usando o objeto rcParams do Matplotlib?"

5. APLICAÇÕES PRÁTICAS

1. Análise Financeira: Utilizar Pandas para ler arquivos CSV de índices como o VIX e calcular médias móveis de 90 dias para identificar regimes de volatilidade.

2. Visualização Geográfica/Imobiliária: Combinar NumPy e Matplotlib para visualizar a densidade populacional e o valor de casas, utilizando grades complexas com subplot2grid.

3. Exploração Estatística: Criar histogramas e gráficos de dispersão lado a lado para entender a correlação entre variáveis usando o método plt.subplots(nrows, ncols).

6. PRÓXIMOS PASSOS

1. Aprofundamento em Visualização: Explorar o Seaborn, que é construído sobre o Matplotlib para gráficos estatísticos mais sofisticados.

2. Introdução ao Machine Learning: Estudar a biblioteca Scikit-Learn, focando em pré-processamento de dados e modelos de regressão linear ou classificação.

3. Análise de Dados em Larga Escala: Pesquisar ferramentas como o Datashader para lidar com datasets extremamente volumosos que o Matplotlib padrão pode ter dificuldade em renderizar.

7. REFERÊNCIAS

Hunter, J. D., & outros. Matplotlib Documentation.

VanderPlas, J. Python Data Science Handbook.

Solomon, B. Python Plotting With Matplotlib (Guide) – Real Python.

Comunidade Pandas. pandas documentation.

Comunidade NumPy. NumPy Manual.

8. SOBRE O AUTOR

O autor é Técnico em Ciência de Dados, com formação voltada à análise, organização e interpretação de dados. Possui interesse em transformar dados brutos em informações claras e úteis, apoiando a tomada de decisão de forma estruturada e objetiva. Seu perfil é prático, disciplinado e orientado à qualidade, prezando por soluções bem organizadas e funcionais.

Atualmente, cursa Engenharia de Software na modalidade EAD e realiza o curso CS50 de Harvard, buscando fortalecer sua base em programação, lógica computacional e desenvolvimento de sistemas. Sua formação técnica inclui conhecimentos em SQL, Power BI, Excel, Python, processos de ETL e modelagem de dados, além de estudos contínuos em IA Generativa e Machine Learning.

O autor mantém uma rotina constante de aprendizado e aprimoramento técnico, com foco no desenvolvimento profissional na área de dados. Acredita que a combinação entre uma base sólida, organização e aprendizado contínuo é essencial para a construção de soluções eficientes e escaláveis, aplicáveis a diferentes contextos e problemas reais.