

PÓS-GRADUAÇÃO

**Linguagens de
programação para
ciência de dados
(Python com Spark)**



PÓS-GRADUAÇÃO

Organização e visualização de dados


Bloco 1

Marcelo Tavares de Lima





► Objetivos

- Manipular planilha MS-Excel, usando a biblioteca Pandas.
 - Criar gráficos customizados, utilizando a biblioteca Matplotlib e Pandas.
 - Criar diferentes tipos de visualização (gráficos, tabelas, diagramas, histogramas etc.).
- 




► Organização e visualização de dados

- Visualização é uma técnica que consiste na criação de imagens diversas.
- Por meio de elementos visuais, a visualização de dados é uma maneira de analisar e entender as exceções, tendências, padrões ou anormalidade nos dados.



► Organização e visualização de dados

- A organização e visualização dos dados também é usada para o processo de tomada de decisão em empresas e, através de inspeção e análises apresentadas visualmente, é possível entender conceitos difíceis ou identificar novos padrões. (TOSI, 2009; YIM, 2018; HUNTER, 2019)
- 



► Introdução a biblioteca Matplotlib

- É a principal biblioteca de plotagem científica de dados em linguagem Python.
- Suporta visualização interativa e não interativa.
- Produz uma ampla variedade de visualizações.
- Desenvolvida em conjunto com a linguagem de programação MATLAB.



► Instalação e dependências da Matplotlib

- Na Leitura Fundamental há uma lista de bibliotecas que devem ser instaladas antes da instalação da Matplotlib.
- Existem diversas maneiras de instalação que dependem diretamente do sistema operacional da máquina onde será instalada.

► Gráficos customizados com Matplotlib

- Para criar um gráfico de linhas, é possível utilizar o seguinte código:

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
x = range(1, 7)
```

```
plt.plot(x, [xi * 2 for xi in x])
```

```
plt.plot(x, [xi * 3.0 for xi in x])
```

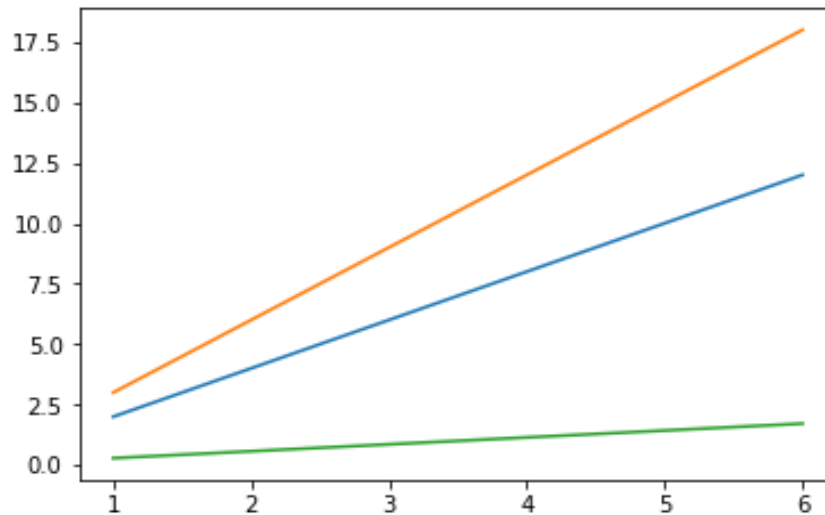
```
plt.plot(x, [xi / 3.5 for xi in x])
```

```
plt.show()
```


► Gráficos customizados com Matplotlib

- O resultado da execução do programa é:

Figura 1 – Gráfico de linhas, criado a partir de números gerados aleatoriamente



Fonte: Pereira (2019).



► Gráficos customizados com Matplotlib

- É possível aplicar uma série de customizações no gráfico apresentado, como grids, mudança no estilo das linhas etc.
- Mais detalhes da programação utilizada, assim como de resultados visuais, podem ser encontrados na Leitura Fundamental.

► Exportação de gráficos com Matplotlib

- É possível exportar para .pdf, .jpg ou .png.
- A seguinte codificação pode ser utilizada:

```
# -*- coding: utf-8 -*-
```

```
import matplotlib.pyplot as plt
```

```
plt.plot([1, 2, 3])
```

```
plt.savefig('grafico.pdf')
```


```
plt.savefig('grafico.jpg')
```

```
plt.savefig('grafico.png')
```

```
plt.show()
```



► Criação de gráficos a partir de informações

- Os utilizados para a geração de visualizações podem estar armazenados em diversos formatos: txt, csv, xls, xlsx etc.
 - É possível importar dados armazenados em qualquer um desses formatos para o ambiente Python.
 - As visualizações podem ser geradas com a biblioteca NumPy.
- 

► Criação de gráficos a partir de informações

- Exemplo de importação de dados em csv.
- Dados:

1,5

2,3

3,4

4,7

5,4

6,3

7,5

8,7

9,4

10,4

► Criação de gráficos a partir de informações

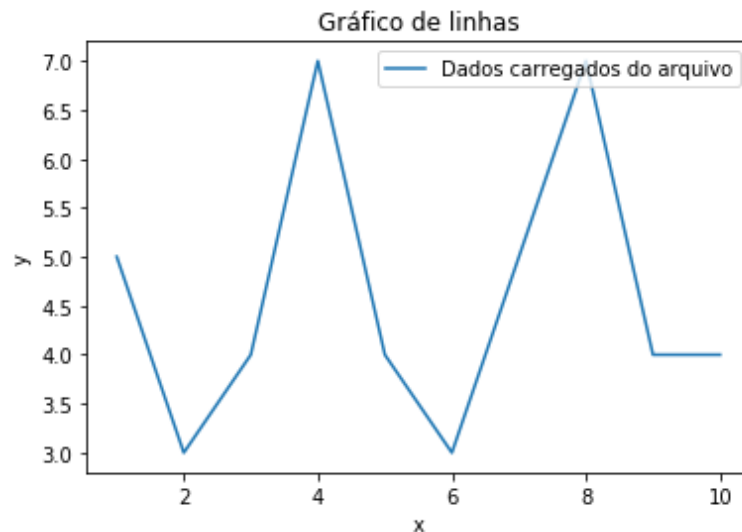
- Exemplo de importação de dados em csv.
- Código-fonte Python com Matplotlib:

```
# -*- coding: utf-8 -*-  
import matplotlib.pyplot as plt  
import csv  
  
x,y = [ ], [ ]  
  
with open('exemplo.txt','r') as csvfile:  
    plots = csv.reader(csvfile, delimiter=',')  
    for row in plots:  
        x.append(int(row[0]))  
        y.append(int(row[1]))  
  
plt.plot(x,y, label='Dados carregados do arquivo')  
plt.xlabel('x')  
plt.ylabel('y')  
  
plt.title(u'Gráfico de linhas')  
plt.legend()  
plt.show()
```

► Criação de gráficos a partir de informações

- Exemplo de importação de dados em csv.
- Output:

Figura 2 – Gráfico de linhas gerado a partir de dados externos



Fonte: Pereira (2019).



► Introdução a Pandas

- É um pacote Python que fornece estruturas de dados rápidas, flexíveis e expressivas, projetadas para facilitar o trabalho com dados relacionais.
- Dados manipulados no Pandas são, frequentemente, usados para trabalhar com análises estatísticas no SciPy, elaboração de gráficos com funções do Matplotlib e algoritmos de aprendizado de máquina no Scikit-learn, segundo McKinney (2019).



► Introdução a Pandas

- Contém um conjunto de ferramentas para manipulação de arquivos de diferentes formatos, como CSV, txt, JSON, xls, xlsx, bancos de dados SQL e formato HDF5.
- Possui duas estruturas principais: *Series* e *DataFrames*.



► Introdução a Pandas

- Estrutura *Series*: é um *array* de uma dimensão (1D). É como se fosse uma coluna de uma tabela.
- Estrutura *DataFrames*: é um encapsulamento da função *Series*, que se estende a duas dimensões (2D). Pode ser criada usando entradas, como listas, dicionários, series, *arrays* ou, outros *DataFrames*.

PÓS-GRADUAÇÃO

Organização e visualização de dados

Bloco 2

Marcelo Tavares de Lima





► Introdução a Pandas (continuação)

- Detalhes para sua instalação estão na Leitura Fundamental.
- Vamos criar gráficos mais customizados!

► Gráficos com o Pandas

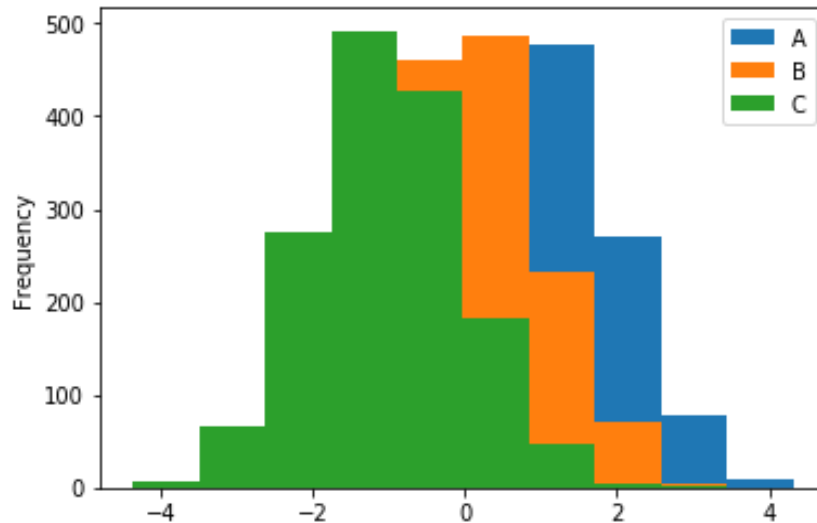
- Exemplo da elaboração de histograma:

```
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
  
df4 = pd.DataFrame({'A':  
np.random.randn(1000) + 1, 'B':  
np.random.randn(1000), 'c':  
np.random.randn(1000) - 1},  
columns=['A', 'B', 'C'])  
  
plt.figure();  
df4.plot.hist(stacked=True, bins=20)  
df4.plot.hist(alpha=0.5)
```

► Gráficos com o Pandas

- Exemplo da elaboração do histograma:

Figura 3 – Histograma para três conjuntos de dados gerados aleatoriamente

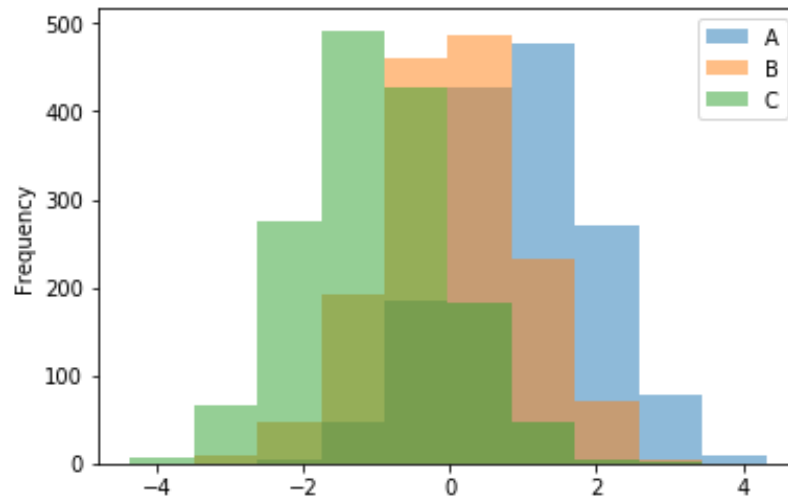


Fonte: Pereira (2019).

► Gráficos com o Pandas

- Exemplo da elaboração do histograma:

Figura 4 – Histograma para três conjuntos de dados gerados aleatoriamente



Fonte: Pereira (2019).

► Gráficos com o Pandas

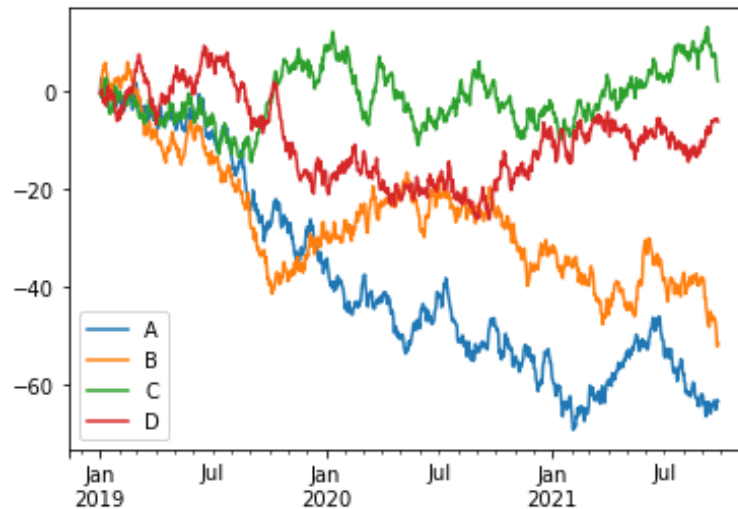
- Exemplo da elaboração de gráficos de linha:

```
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
  
ts = pd.Series(np.random.randn(1000),  
index=pd.date_range('01/01/2019',  
periods=1000))  
  
df =  
pd.DataFrame(np.random.randn(1000,  
4), index=ts.index, columns=list('ABCD'))  
df = df.cumsum()  
plt.figure(); Soma acumulativa, vale a  
explicação.
```


► Gráficos com o Pandas

- Exemplo da elaboração de gráficos de linha:

Figura 5 – Gráficos de linhas gerados a partir de quatro conjuntos de dados gerados aleatoriamente



Fonte: Pereira (2019).

► Manipulação de arquivos com o Pandas

Quadro 1 – Funções da biblioteca Pandas para manipulação de arquivos

| Formato | Descrição | Função para leitura | Função para escrita |
|---------|--------------------------------------|-----------------------------|---------------------------|
| Texto | CSV | read_csv | to_csv |
| Texto | JSON | read_json | to_json |
| Texto | HTML | read_html | to_html |
| Binário | MS Excel | read_excel | to_excel |
| Binário | Open Document Spreadsheet (ODS) | read_excel | - |
| Binário | HDF5 Format | read_hdf | to_hdf |
| Binário | Python Pickle Format | read_pickle | to_pickle |
| SQL | SQL | read_sql | to_sql |
| SQL | Google Big Query | read_gbq | to_gbq |

Fonte: McKinney (2019).

PÓS-GRADUAÇÃO

Teoria em prática


Bloco 3

Marcelo Tavares de Lima





► Planilhas MS Excel e Python

- As planilhas MS Excel representam uma ferramenta muito utilizada no controle e planejamento de diversos setores das empresas.
 - Entretanto, as planilhas, em muitos casos, não são suficientes para armazenar todas as informações conforme o tamanho da empresa.
 - Muitas vezes, é necessário buscar uma solução mais robusta e mais confiável.
- 



► Planilhas MS Excel e Python

- Diante desse cenário, imagine que você foi contratado pela empresa XPTO para fazer a integração dos dados, que, atualmente, estão salvos em planilhas MS Excel, e salvar essas informações em um sistema ERP (Enterprise Resource Planning).
- Utilizando Python e as bibliotecas Matplotlib e Pandas, como você faria a integração dos dados armazenados nas planilhas Excel para o sistema ERP?

Dica do professor

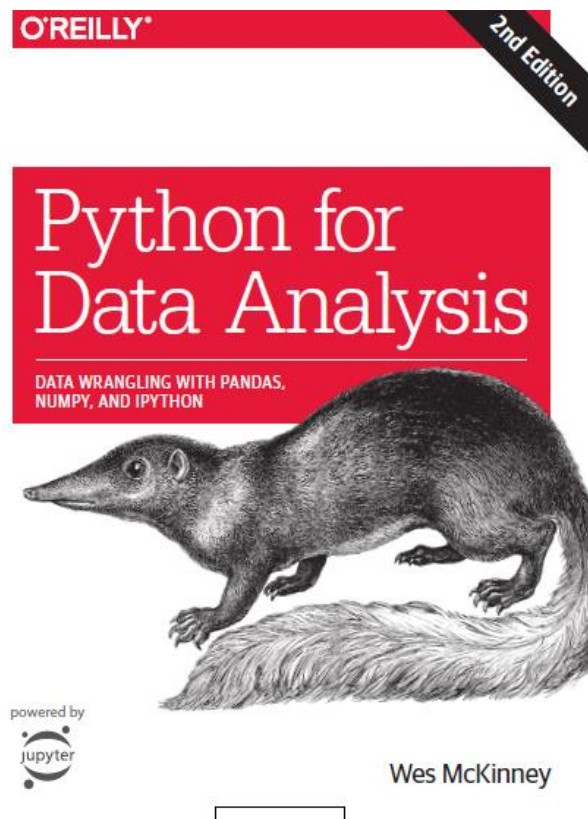
Bloco 4

Marcelo Tavares de Lima



► Dica de livro

Figura 6 - livro



Python for data analysis.

Autor: Wes McKinney (2018).

Editora: O'Reilly Media.

Fonte: <https://www.amazon.com/Python-Data-Analysis-Wrangling-IPython-ebook/dp/B075X4LT6K>. Acesso em: 21 jan. 2020



► Referências

HUNTER, J.; DALE, D.; FIRING, E. *et al.* **Matplotlib User's Guide**. Matplotlib Release 3.1.1, 2019.

MCKINNEY, W., **Pandas**: powerful Python data analysis toolkit. Release 0.25.3. Python for High Performance and Scientific Computing, 2019.

TOSI, S., **Matplotlib for Python developers**. Packt Publishing Ltd, 2009.

YIM, A.; CHUNG, C.; YU, A., **Matplotlib for Python Developers**: effective techniques for data visualization with Python, Packt Publishing Ltd, 2018.

