

Conceitos
avançados de
Python focados
em análise
de dados + 4
indicações de
livros

Qual é a relevância do Python para a análise de dados?

Considerando que estamos falando sobre um trabalho extenso e que demanda o uso de ferramentas de automação para lidar com grandes volumes de dados, é importante que o processamento aconteça de forma **eficiente e ágil**. Além disso, também há a importância em ter informações limpas para o momento da análise.

É aí que entra o Python: a ferramenta prioriza a automação de tarefas, apresenta facilidade para lidar com recorrências e possui certo grau de simplicidade. O que isso significa? Produtividade!

Também é relevante destacar a **força da comunidade** para aqueles que utilizam a linguagem. O compartilhamento de recursos e bibliotecas disponíveis **ajudam o profissional a fazer a aplicação**, conduzir os processos analíticos e garantir que haja qualidade em cada um deles.

E você, como está o seu desenvolvimento profissional com relação ao Python? Para contribuir com os próximos passos na sua jornada profissional, preparamos um conteúdo com dicas para uma análise de dados ainda mais eficiente.

Boa leitura!

O que você irá encontrar neste e-book?

CLIQUE NO ASSUNTO
PARA SER DIRECIONADO
PARA À PÁGINA

1.

Quais são as bibliotecas
mais utilizadas para
análise de dados?

2.

9 conceitos para colocar
em prática

3.

8 dicas para visualizar
dados em um gráfico

4.

Quer se desenvolver em
Python? Saiba como
(+ dicas de livros)

5.

Encerramento

Quais são as bibliotecas mais utilizadas para análise de dados?



As **libraries** são o diferencial para que **Python** seja uma boa escolha para a análise de dados. Com elas, é possível obter **soluções completas** para que as tarefas sejam realizadas com sucesso. Saiba mais sobre as principais bibliotecas a seguir:

Quais são as bibliotecas
mais utilizadas para
análise de dados?

#1 PANDAS

[IR PARA O ÍNDICE](#)

A primeira ferramenta da lista se destaca por atuar com análise e manipulação de dados com facilidade, agilidade e flexibilidade.

Essas características tornam a análise produtiva e proporcionam um alto desempenho, ajudando a destacar o **Python** na sua função de atuar com o processamento dessas informações.

Diferenciais da pandas: remodelamento, expansão e filtragem de subconjuntos de dados.

[Saiba mais sobre a library clicando aqui.](#)

Quais são as bibliotecas mais utilizadas para análise de dados?

#2 NUMPY

Ideal para a computação numérica com matrizes multidimensionais, ela processa arranjos e matrizes grandes e multidimensionais. Também possui funções matemáticas para manipular esses arrays.

Quando o assunto é análise de dados, ela é utilizada como contêiner primário. Assim, possibilita-se o compartilhamento de dados entre algoritmos. **Por que os arranjos em NumPy se destacam?**

1. Método superior de armazenamento e manipulação de dados numéricos em comparação às estruturas nativas de Python;

2. Libraries de linguagens de níveis mais baixos podem ler e alterar os dados armazenados nesses arrays.

Você pode acessar a página da NumPy aqui.

Quais são as bibliotecas
mais utilizadas para
análise de dados?

[IR PARA O ÍNDICE](#)

Já essa biblioteca se destaca pela ampla possibilidade de produções de gráficos, como os tipos bidimensionais, de forma nativa. E não para por aí, pois há a possibilidade de utilizar extensões para maximizar suas possibilidades para:

1. Projeções
cartográficas;

2. Gráficos
tridimensionais;

3. Mais produtos
gráficos.

Aliás, você sabia que os gráficos gerados pela pandas possuem o Matplotlib como origem?

[Acesse o Matplotlib agora.](#)

#3 MATPLOTLIB

Além das
bibliotecas que
citamos acima,
ainda existem
as seguintes
alternativas:

Seaborn

Foco em visualização
de dados;

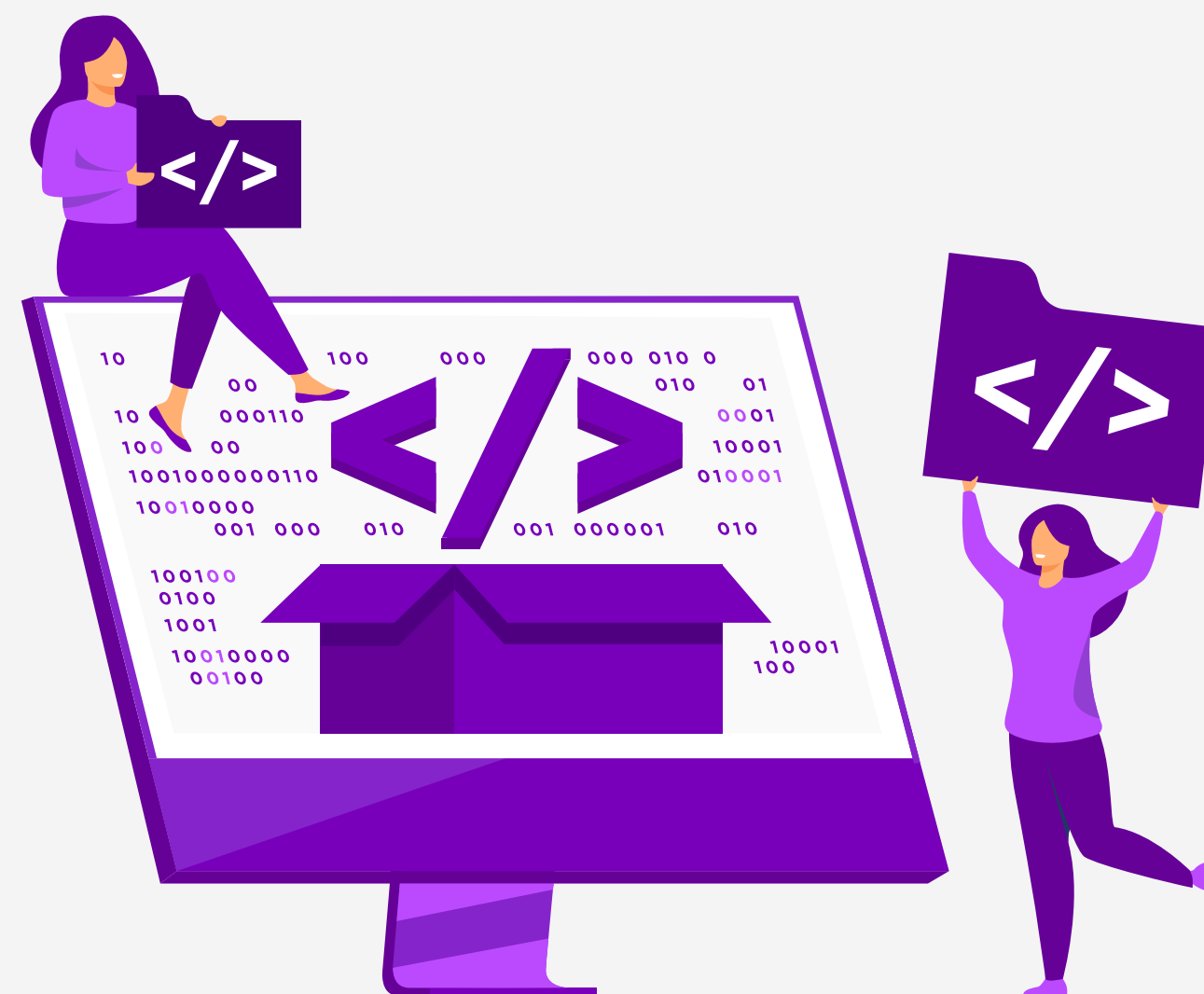
TensorFlow e Keras

Redes neurais, otimização
e modelos mais complexos
de machine learning.

Scikit-learn

Para a modelagem
estatística;

9 conceitos para colocar a mão na massa



As funções que vamos destacar a seguir podem ser utilizadas no **database Iris**. O dataset apresenta quatro variáveis de 50 amostras de três espécies: setosa, versicolor e virginica.

9 conceitos para colocar
a mão na massa

/pd.read_csv

1

A função é indicada para realizar a leitura do arquivo. O dataset é carregado para a memória, além de ser mantido na variável “planta”.

```
planta = pd.read_csv("iris.csv")
```

[IR PARA O ÍNDICE](#)

9 conceitos para colocar
a mão na massa

/import

Essa instrução diz respeito à biblioteca que será utilizada no código. Então, vamos supor que pandas seja a nossa primeira escolha. Veja:

2

“as” é utilizada para criar um apelido para a library. Assim, pode-se fazer um referenciamento depois. Lembrando que o mesmo raciocínio pode ser aplicado às demais bibliotecas.

```
import pandas as pd
```

Outro ponto é o estilo do gráfico, que pode ser utilizado da seguinte forma:

```
plt.style.use('ggplot')
```

[IR PARA O ÍNDICE](#)

9 conceitos para colocar
a mão na massa

/head()

3

Responsável por mostrar as
primeiras linhas da base de dados.
Caso o usuário não informe um
valor, a função exibirá até a 5ª linha.

```
In [3]: #Exibindo as 5 primeiras linhas do DataFrame
        planta.head()
```

```
Out[3]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

[IR PARA O ÍNDICE](#)

9 conceitos para colocar
a mão na massa

/describe()

4

A função agrega com informações sobre os dados que podem ser aproveitados para a geração de estatísticas, como desvio padrão e média.

```
In [4]: #resumo de informações das colunas
        planta.describe()
```

Out[4]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.829066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

[IR PARA O ÍNDICE](#)

9 conceitos para colocar
a mão na massa

/dtypes

5

Como o nome indica, a função está ligada ao tipo de dados, ou seja, a forma como Python interpreta os valores.

```
In [5]: #Tipo de dado em cada coluna
        planta.dtypes

Out[5]: sepal_length    float64
        sepal_width     float64
        petal_length     float64
        petal_width      float64
        species          object
        dtype: object
```

[IR PARA O ÍNDICE](#)

9 conceitos para colocar
a mão na massa

/shape

6

Apresenta dois valores: quantidade
de linhas e colunas da tabela.

```
In [6]: #quantidade total de linhas e colunas  
        planta.shape  
Out[6]: (150, 5)
```

[IR PARA O ÍNDICE](#)

9 conceitos para colocar
a mão na massa

/columns

7

Aqui, a dica é aproveitar o atributo para renomear as colunas. Assim, a manipulação dos dados e o entendimento dessa base de informações tornam-se facilitados.

Utilize `head()` para verificar a nova versão dos nomes.

```
#Renomeando as colunas  
planta.columns = ['sepala_comprimento', 'sepala_largura', 'petala_comprimento', 'petala_largura', 'especie']
```

[IR PARA O ÍNDICE](#)

9 conceitos para colocar
a mão na massa

/isnull()

8

Com a função, as linhas que
apresentam valores nulos retornam
na coluna “sepala_comprimento”.

```
In [8]: #verificando se na coluna sepala_comprimento há algum valor nulo
        planta[planta['sepala_comprimento'].isnull()].head()

Out[8]:
```

sepala_comprimento	sepala_largura	petala_comprimento	petala_largura	espeidade
--------------------	----------------	--------------------	----------------	-----------

[IR PARA O ÍNDICE](#)

9 conceitos para colocar
a mão na massa

/value_counts

9

Quer saber as contagens de
valores totais?

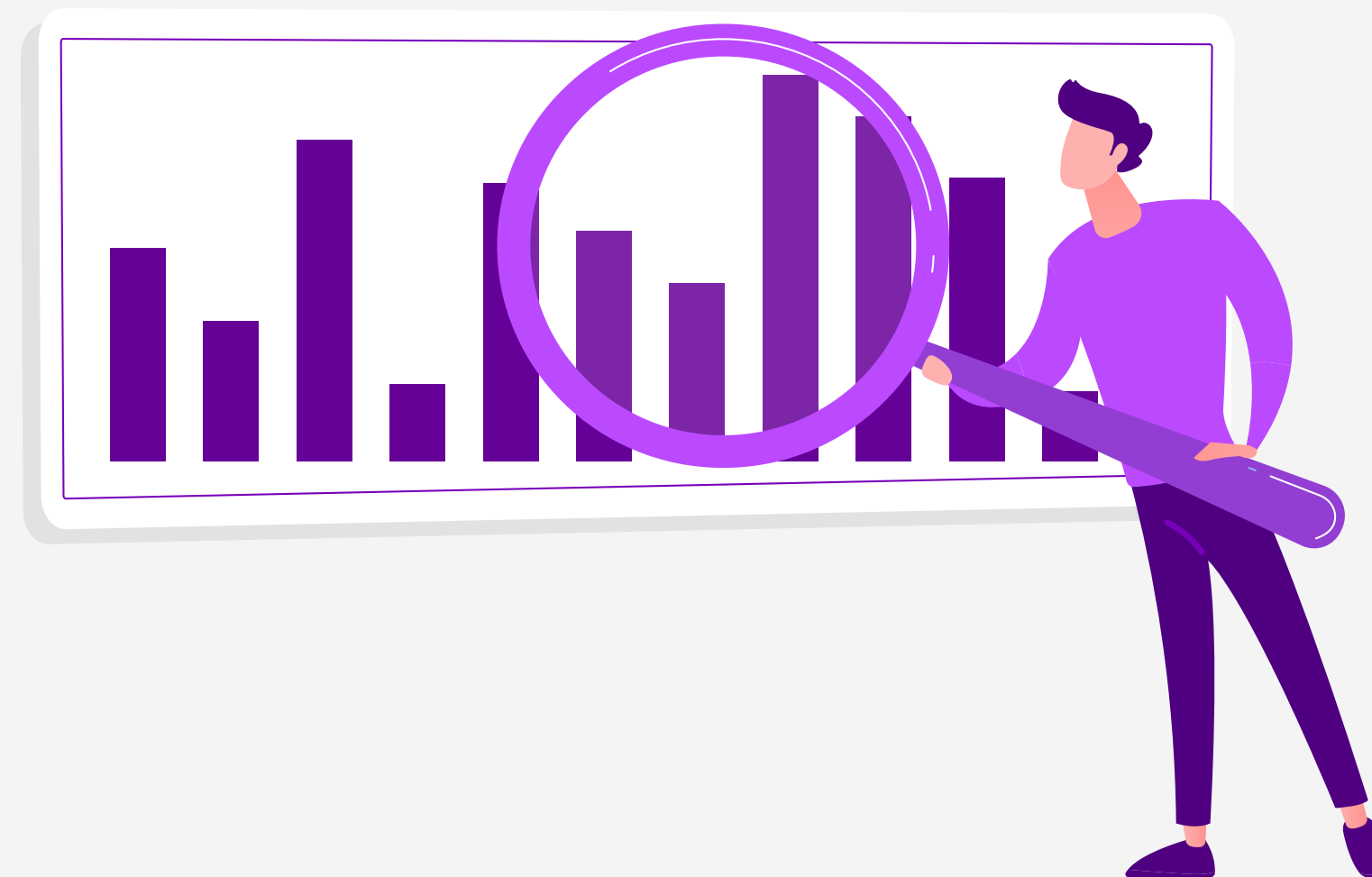
Você pode utilizar esta função.

```
In [10]: #contando o total de setosa
         planta['especie'].value_counts()

Out[10]: versicolor    50
         virginica     50
         setosa        50
         Name: especie, dtype: int64
```

[IR PARA O ÍNDICE](#)

8 dicas para visualizar dados em um gráfico



Para apoiar na tomada de decisão da empresa, a **precisão dos dados analisados é fundamental**. É neste momento que entra a importância em selecionar o gráfico corretamente. Confira, a seguir, algumas dicas para produzir diferentes formatos desse recurso.

8 dicas para visualizar dados em um gráfico

1

Uma linha de código, cinco argumentos

Kind

Tipo de gráfico.

Color

Cor do gráfico.

Figsiz

Tamanho do gráfico.

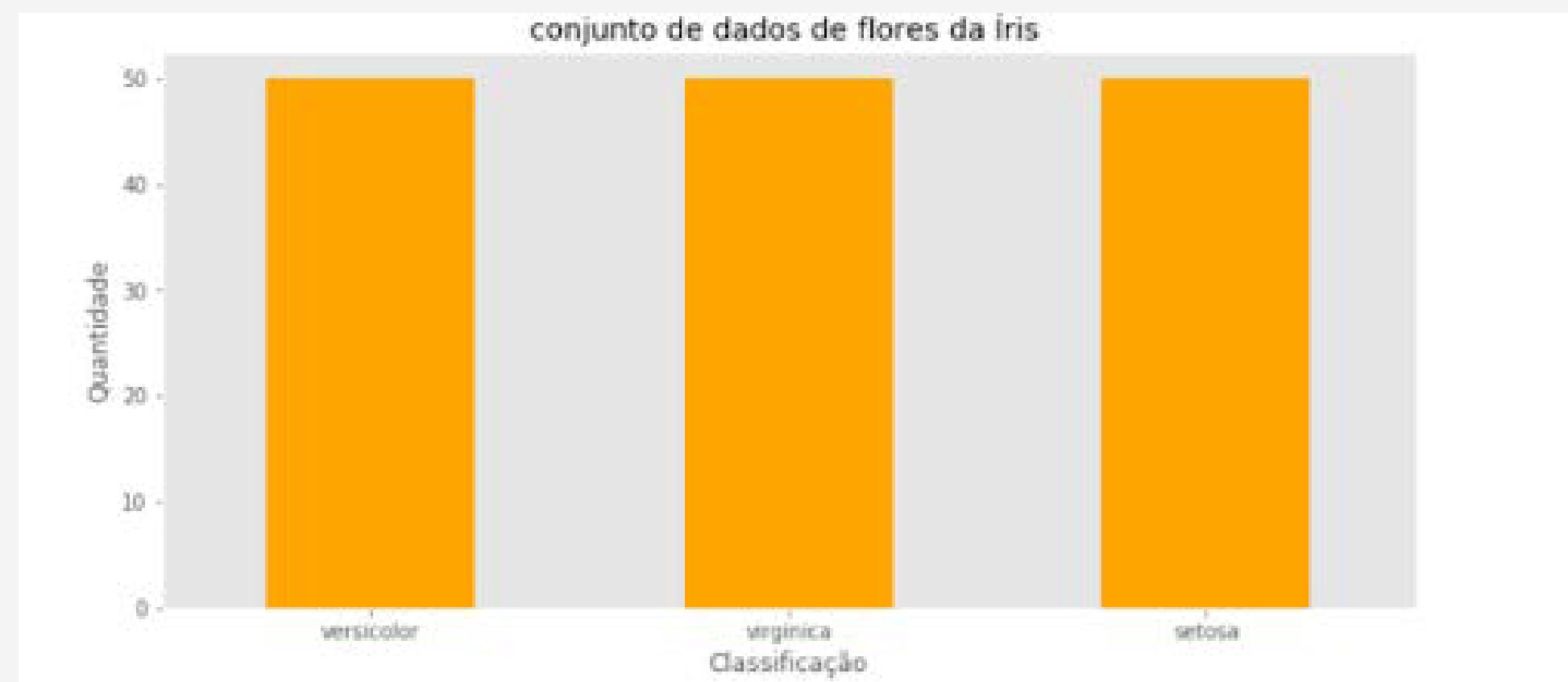
Grid

Para definir a linha de grade no gráfico.

Rot

Grau de rotação dos dados relacionados ao eixo X.

```
In [14]: #criando gráfico com apenas uma linha de código
planta['setosa'].value_counts().head(10).plot(kind='bar', figsize=(11,5), grid = False, rot=0, color='orange')
#desenhando o gráfico mais agradável
plt.title('conjunto de dados de flores da iris ')
plt.xlabel('Classificação') #nomeando o eixo x, onde fica o tipo de iris
plt.ylabel('Quantidade') #nomeando o eixo y, onde fica o total de classificação
plt.show() #exibindo o gráfico
```



[IR PARA O ÍNDICE](#)

8 dicas para visualizar
dados em um gráfico



Pandas profiling

A função amplia o DataFrame para uma ágil análise de dados. Uma linha de código gera volume de informações, além de um relatório HTML interativo.

```
df.describe()df.info()functionsdf.profile_report()
```

Veja, a seguir, quais são as estatísticas calculadas no pacote de criação do pandas profiling:

Como fazer a instalação?

```
pip install pandas-profiling  
ou  
conda install -c anaconda pandas-profiling
```

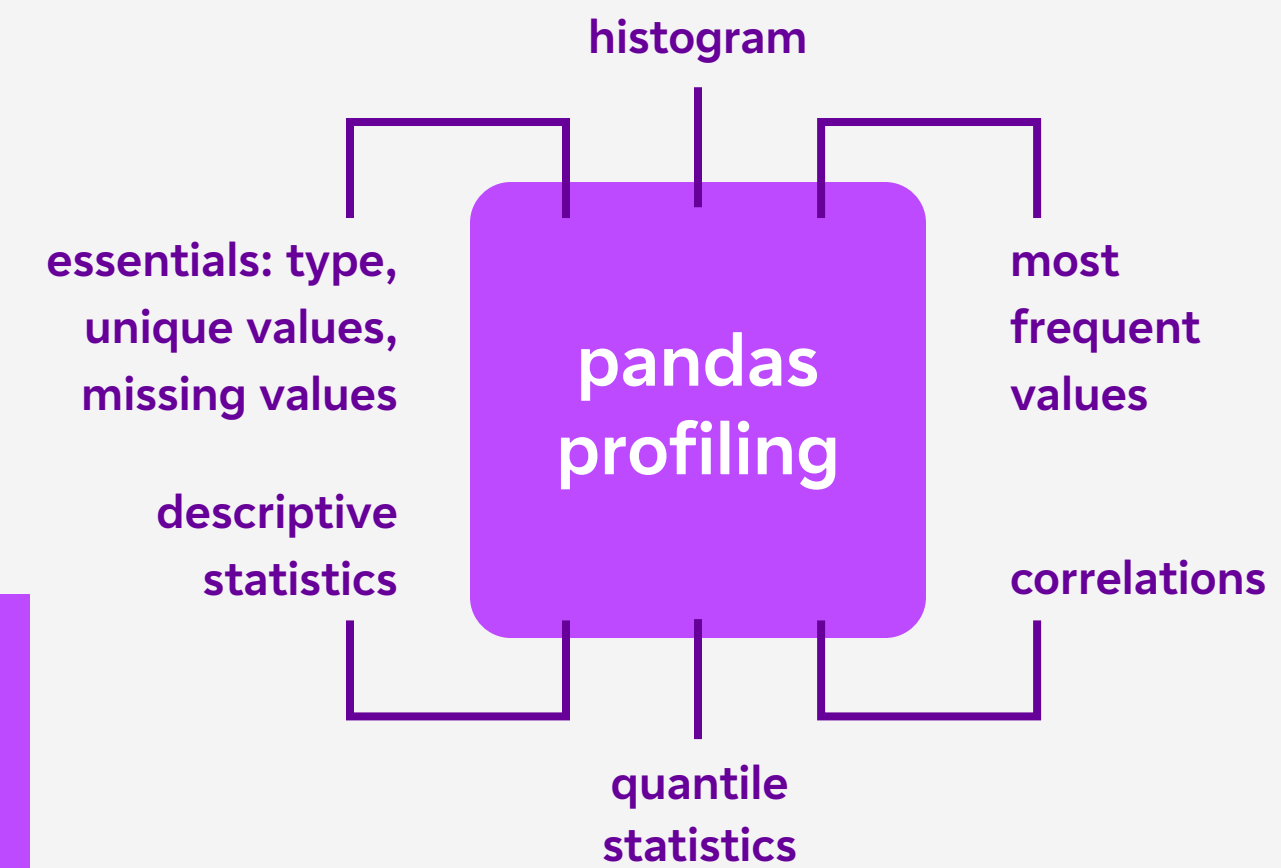
Como usar?

Utilize o conjunto de dados titânico para apresentar os recursos do criador de perfil Python.

Essa linha cumpre com o objetivo de exibir o relatório de criação de perfil de dados em um Jupyter Notebook de forma detalhada e com gráficos.

Para exportar o relatório para um arquivo HTML interativo, utilize o código:

```
profile = df.profile_report(title='Pandas Profiling Report')  
profile.to_file(outputfile="Titanic data profiling.html")
```



```
#importing os pacotes necessários  
import pandas as pd  
import pandas_profiling  
  
#Pandas-Profiling 2.0.0  
df = pd.read_csv('titanic/train.csv')  
df.profile_report()
```

[IR PARA O ÍNDICE](#)

8 dicas para visualizar
dados em um gráfico



Plotar com interatividade sem modificar o código de forma brusca? É possível

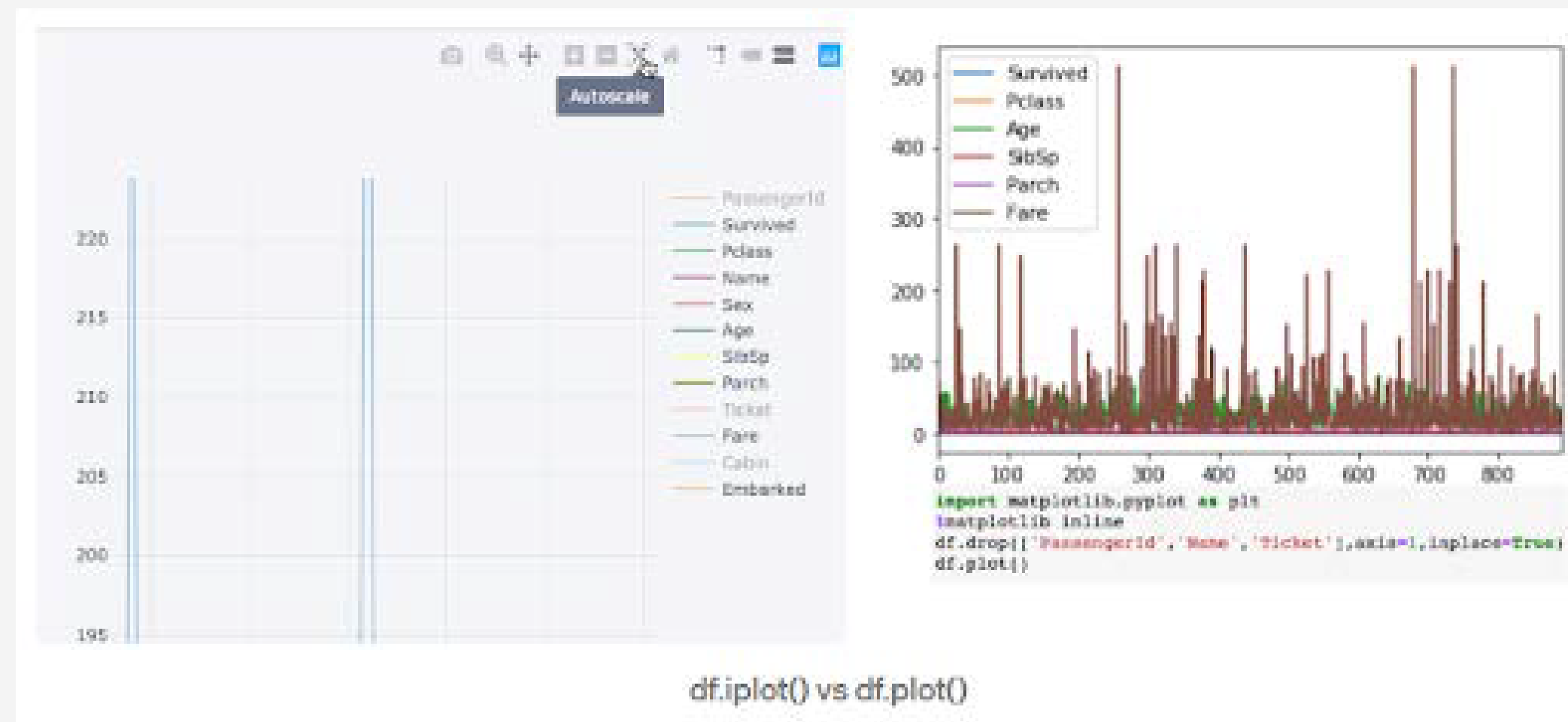
Com a library Cufflinks, você pode utilizar a plotagem
unida à flexibilidade dos pandas e gerar gráficos. Veja:

Como fazer a instalação?

```
pip install plotly # Plotly é um  
pré-requisito antes de  
instalar abotoaduras  
pip install abotoaduras
```

Como usar?

```
#importar pandas  
import pandas como pd  
#importar plotly e abotoaduras no modo offline  
import abotoaduras como cf  
  
import plotly.offline  
cf.go_offline()  
cf.set_config_file(offline=False, world_readable=True)
```



[IR PARA O ÍNDICE](#)

8 dicas para visualizar dados em um gráfico

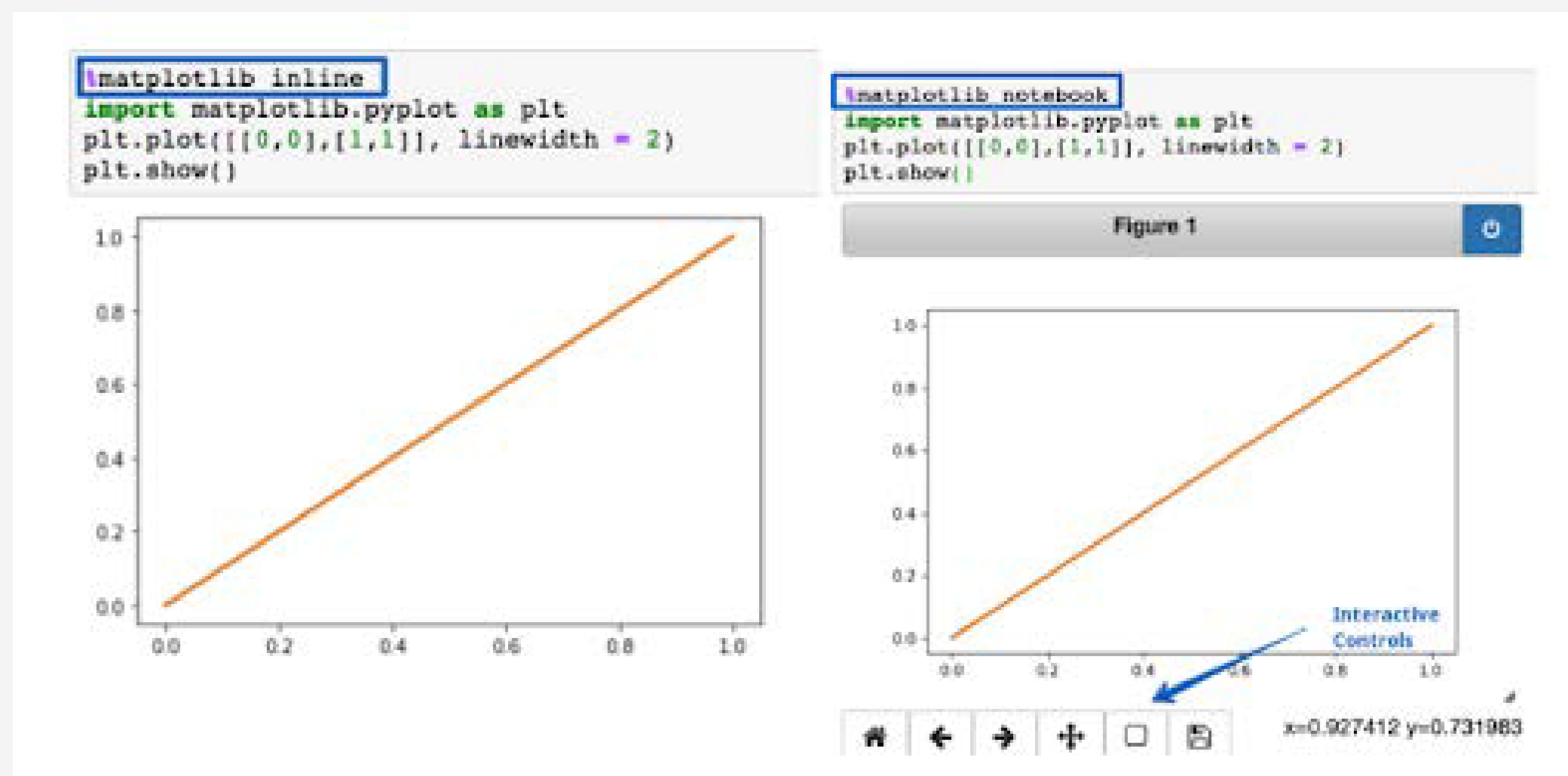
4

Comandos mágicos

Com a library Cufflinks, você pode utilizar a plotagem unida à flexibilidade dos pandas e gerar gráficos. Veja:

Eles ajudam a solucionar alguns dos principais desafios que aparecem na análise de `%ismagic`. Ao utilizar, você pode conferir todas as funções disponíveis.

Um exemplo é `%matplotlib inline` que ajuda a renderizar os gráficos de Matplotlib no Jupyter Notebook. Para que os gráficos tenham capacidade de zoom e redimensionamento, substitua “inline” por “notebook”, mas confira se a função foi chamada antes de importar a biblioteca.



8 dicas para visualizar
dados em um gráfico



Como melhorar a estética das estruturas de dados?

[IR PARA O ÍNDICE](#)

Utilize pprint para atingir esse objetivo! O módulo irá ajudar na impressão de dicionários ou dados JSON. Veja alguns exemplos a seguir:

```
# With Print

employee_records = {'Emp ID': '101', 'Emp Name': 'Tom',
                    'Project IDs': {'P1 ':1308, 'P2 ': 'A104', 'P4 ':2}}

print(employee_records)

{'Emp ID': '101', 'Emp Name': 'Tom', 'Project IDs': {'P1 ': 1308, 'P2 ': 'A104', 'P4 ': 2}}
```

```
# With Pretty Print

import pprint

employee_records = {'Emp ID': '101', 'Emp Name': 'Tom',
                    'Project IDs': {'P1 ':1308, 'P2 ': 'A104', 'P4 ':2}}

pprint.pprint(employee_records,width=-1)

{'Emp ID': '101',
 'Emp Name': 'Tom',
 'Project IDs': {'P1 ': 1308,
                  'P2 ': 'A104',
                  'P4 ': 2}}
```

8 dicas para visualizar
dados em um gráfico



Quer dar destaque
a algo importante?

Caixa azul para transmitir informações

```
<div class="alert alert-block alert-info">  
<b>Dica:</b> Use caixas azuis (alert-info) para dicas e notas. Se for uma nota, você não  
precisa incluir a palavra "Nota".  
</div>
```

Tip/Info: The Blue boxes are used for tips and notes.

Caixa amarela para emitir avisos

```
<div class="alert alert-block alert-warning">  
<b>Exemplo:</b> Caixas amarelas são geralmente usadas para incluir exemplos  
adicionais ou fórmulas matemáticas.  
</div>
```

Yellow Boxes are generally used to include additional examples or mathematical formulas.

Caixa verde para destacar um sucesso

```
<div class="alert alert-block alert-success">  
Use a caixa verde apenas quando necessário para exibir links para conteúdo relacionado.  
</div>
```

Use green box only when necessary like to display links to related content.

Caixa vermelha para indicar perigo

```
<div class="alert alert-block alert-danger">  
É bom evitar caixas vermelhas, mas pode ser usado para alertar os usuários para não  
excluir alguma parte importante do código etc.  
</div>
```

It is good to avoid red boxes but can be used to alert users to not delete some important part of code etc.

[IR PARA O ÍNDICE](#)

8 dicas para visualizar
dados em um gráfico



**Excluiu uma célula por
engano?
Saiba como restaurar**

Pressione **CTRL/CMD+Z** para recuperar o conteúdo de uma célula. Caso a restauração seja de uma célula inteira que foi apagada, utilize **ESC+Z** ou **CTRL/CMD+ZEDIT > Undo Delete Cells.**

8 dicas para visualizar
dados em um gráfico



Quer saber como usar
Python no Power BI?

Com o Python, você pode importar, transformar e visualizar dados na ferramenta. Veja algumas ações:

Criar o script de importação de dados (conjunto de dados Boston Housing, disponível no scikit-learn)

1. Utilizando para clustering, podemos usar a Análise de Componentes Principais para diminuir as dimensões e visualizar os dados em um espaço bidimensional;

2. Após, vamos aplicar o agrupamento k-means para identificar grupos homogêneos nos dados.

Obs.: crie scripts autônomos do código para testar e depurar possíveis problemas antes de utilizá-los no Power BI.

Como habilitar o Python no Power BI?

1. Configure o ambiente com as bibliotecas;

2. Para gerenciar o ambiente Python, é possível utilizar virtualenv e pipenv ou a distribuição conda.

[IR PARA O ÍNDICE](#)

Quer se desenvolver em Python? Saiba como (+dicas de livros)

O desenvolvimento profissional não pode parar, não é verdade? Por isso, aproveitamos para reforçar mais habilidades que são importantes para todo profissional que atua com Python:

Estatística/álgebra linear: são skills fundamentais para a realização do tratamento prévio da base de dados e a escolha de variáveis;

Comunidades: um dos grandes diferenciais da programação são esses espaços de troca de experiências com outros profissionais. Utilize sempre que necessário;

Para aprofundar o seu conhecimento, você também pode acessar os seguintes conteúdos:

Towards data science;
Data Hackers;
Hipster.tech (podcast);
Data science dojo;
Datacamp (podcast dataframed);
DeepLearning.AI (blog).

4 dicas de livros para você ir além no seu conhecimento de Python

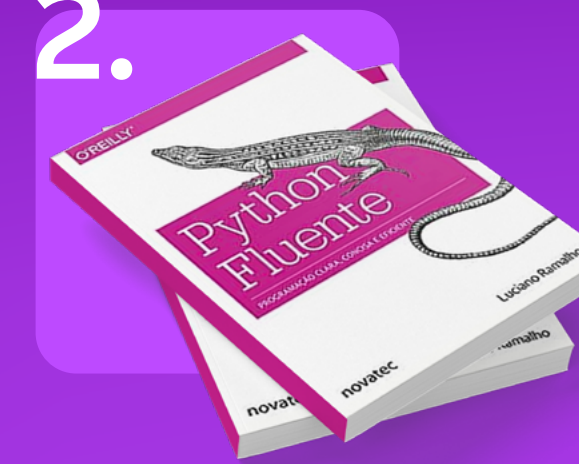
1.



Python para análise de dados:
Tratamento de dados com
Pandas, NumPy e IPython

Wes McKinney
Novatec Editora

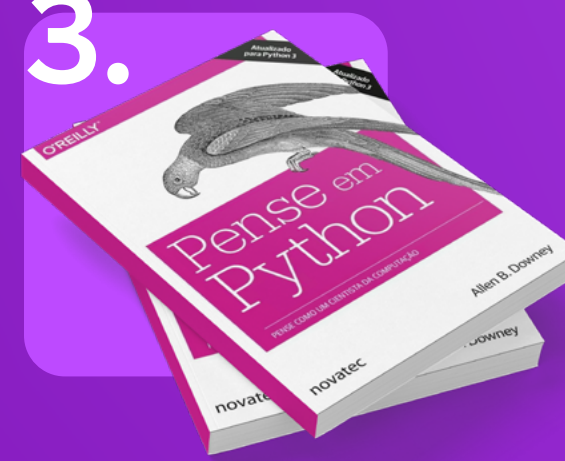
2.



Python Fluente:
Programação Clara,
Concisa e Eficaz

Luciano Ramalho
Novatec Editora

3.



Pense em Python:
Pense Como um Cientista
da Computação

Allen B. Downey
Novatec Editora

4.



Mãos à Obra: Aprendizado
de Máquina com Scikit
Learn & TensorFlow

Aurélien Géron
Alta Books

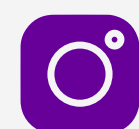
Esperamos que os conteúdos apresentados neste e-book possam contribuir com o desenvolvimento da sua carreira e no aprimoramento das habilidades técnicas que, assim como as soft skills, são importantes para quem deseja alcançar maior reconhecimento profissional ou até mesmo um cargo de gestão.

Por isso, desejamos que a sua jornada com Python seja de muito aprendizado e evolução constante! Quer inovar com a gente? Acesse o banco de talentos Vivo [clcando aqui](#) e venha digitalizar para aproximar.

[IR PARA O ÍNDICE](#)



E não deixe de nos acompanhar nas mídias sociais



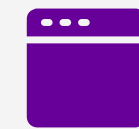
Instagram
#VemPraVivo



LinkedIn
Vivo (Telefônica Brasil)



Vídeo
Vem realizar com a gente



Blog
de Carreiras da Vivo

Até a próxima!