

Educação Profissional Paulista

Técnico em
**Ciência de
Dados**

Secretaria da
Educação



SÃO PAULO
GOVERNO DO ESTADO

Bibliotecas: Pandas, NumPy, SciPy, Matplotlib e Seaborn

Pandas: visualização gráfica

Aula 1

Código da aula: [DADOS]ANO1C2B4S28A1

Mapa da Unidade 5 Componente 3

Bibliotecas: Pandas,
NumPy, SciPy,
Matplotlib e Seaborn

semana
23

Pandas: acesso
e seleção

Matplotlib:
estrutura

semana
29

29

Você está aqui!

Pandas: visualização
gráfica

semana
28

28

Matplotlib:
gráficos básicos

semana
30

30

Bibliotecas: Pandas,
NumPy, SciPy,
Matplotlib e Seaborn

Mapa da Unidade 5 Componente 3

Você está aqui!

Pandas: visualização
gráfica

Aula 1

Código da aula: [DADOS]ANO1C2B4S28A1

28





Objetivos da Aula

- Conhecer as formas de plotar um gráfico, usando a biblioteca Pandas do Python.



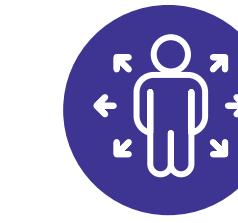
Recursos Didáticos

- Recurso audiovisual para exibição de vídeos e imagens.
- Acesso ao laboratório de informática e/ou à internet.
- Software Anaconda/Jupyter Notebook instalado ou similar.



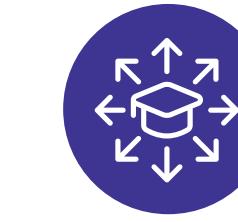
Duração da Aula

50 minutos.



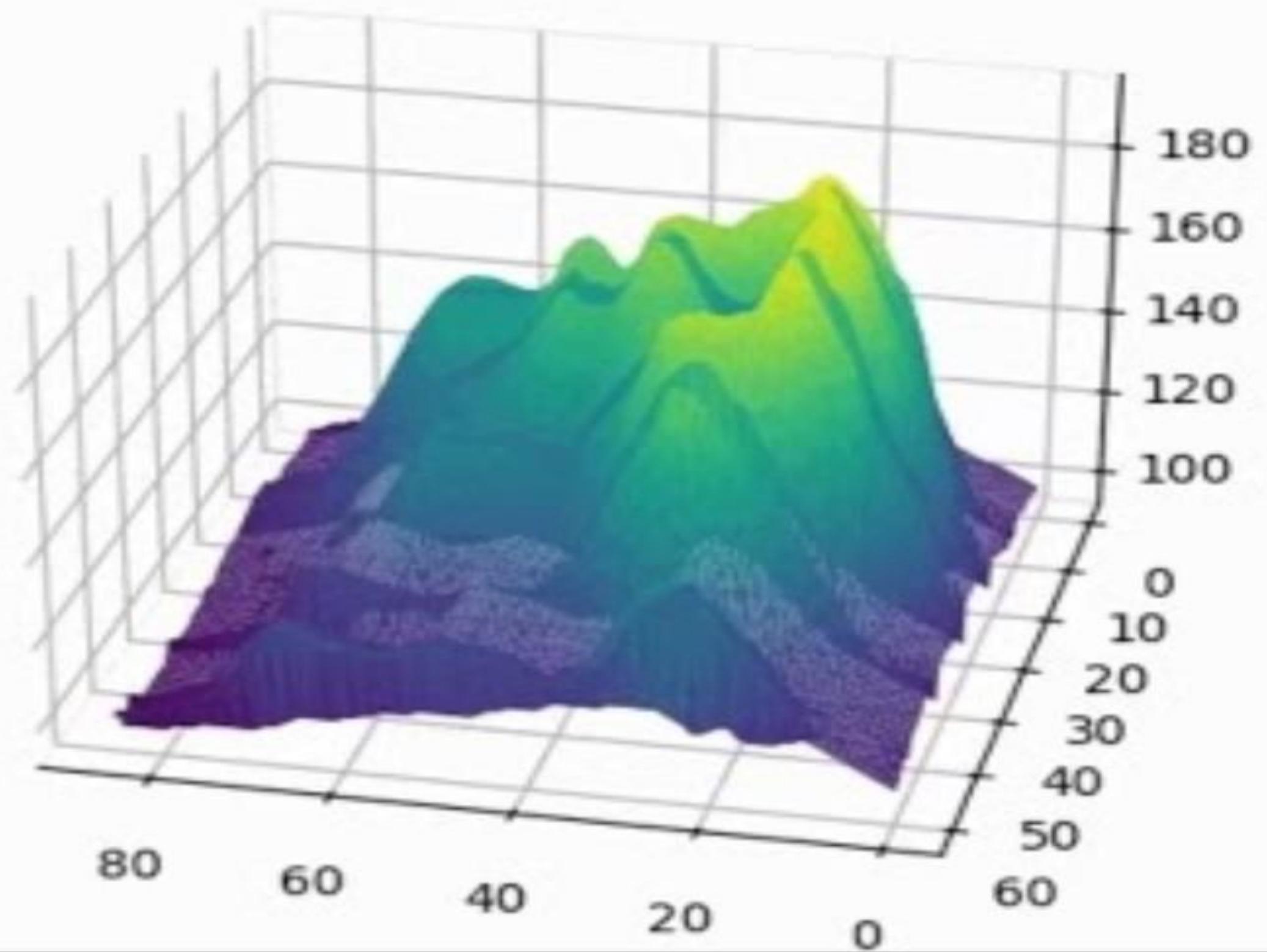
Competências Técnicas

- Ser proficiente em linguagens de programação para manipular e analisar grandes conjuntos de dados.
- Usar técnicas para explorar e analisar dados, aplicar modelos estatísticos, identificar padrões, realizar inferências e tomar decisões baseadas em evidências.



Competências Socioemocionais

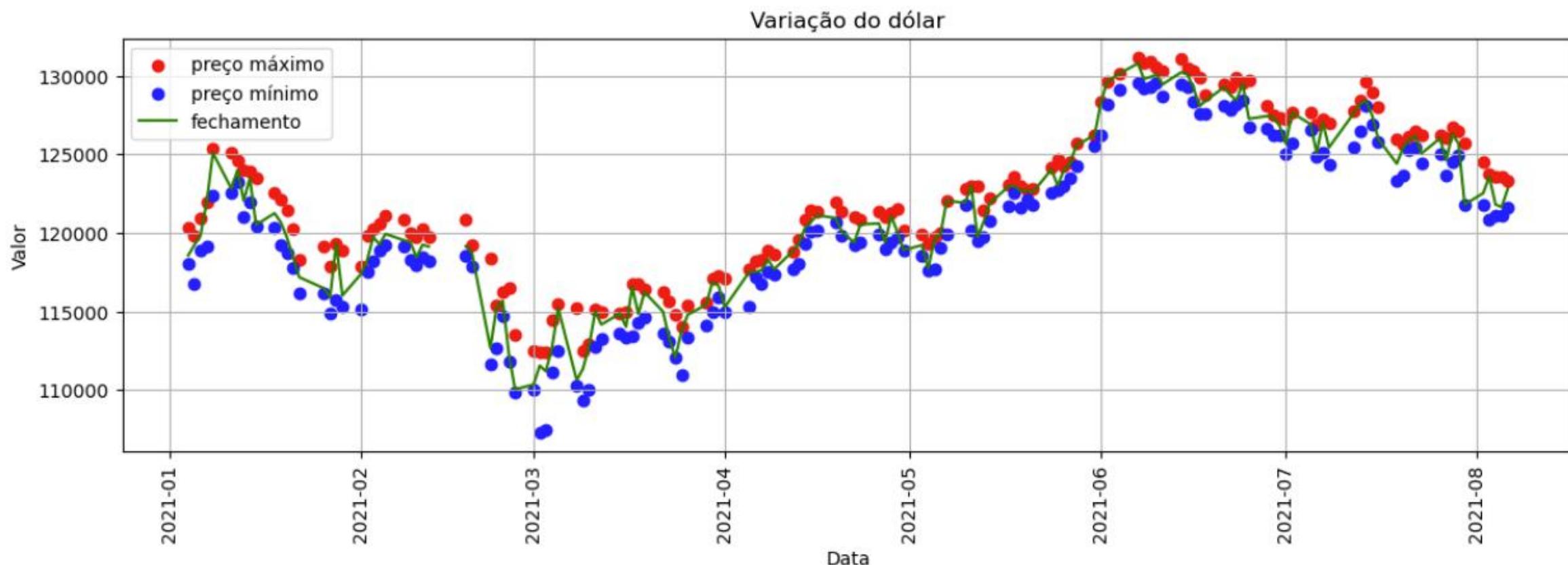
- Colaborar efetivamente com outros profissionais, como cientistas de dados e engenheiros de dados.
- Trabalhar em equipes multifuncionais, colaborando com colegas, gestores e clientes.



Primeiras ideias

-  Você visualiza dados na imagem mesmo estando em movimento?
-  Você acha que é possível fazer esse gráfico no Pandas?
-  Essa visualização parece ser difícil de ser feita?

Ponto de partida



Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

- ▶ Apenas ao olhar para o gráfico, você consegue identificar do que se trata?
- ▶ O gráfico é útil para obter informações?
- ▶ Como criar esse gráfico? Será que é possível criá-lo no Pandas?

Construindo
o conceito

Gráficos

Sabemos que a visualização gráfica é a melhor forma de entender alguns dados e tabelas.

Existem diversos tipos de gráficos para categorias diferentes.

Vamos, agora, descobrir como criar essas visualizações a partir da biblioteca Pandas?



Dica:

Na próxima semana, vamos ajustar o gráfico com tamanhos, cores, tipos, legendas e demais características.

Arquivos e bibliotecas

Carregue no notebook os seguintes arquivos csv:

Nome original	Renomear para
[DADOS]C3U5S28A1A2A3A4DADOSDE VENDAS.csv	df_vendas
[DADOS]C3U5S28A1A2A3A4MERCADO FRUTAS.csv	df_mercado
[DADOS]C3U5S28A1A2A3A4QUADRO MEDALHAS.csv	df_medalhas

E importe as seguintes bibliotecas:

- import pandas as pd
- import matplotlib.pyplot as plt

Histograma

Para criar um histograma diretamente do DataFrame, usando Pandas, você pode utilizar o método `.plot()` com o parâmetro `kind='hist'`.

```
1 import matplotlib.pyplot as plt
2 import pandas as pd
3
4 # Dados para o DataFrame
5 dados = {
6     'Valor': [10, 15, 13, 18, 16, 20, 15, 14, 17, 18, 22, 24, 25, 16, 15, 20, 18, 16, 20, 15, 20, 17, 18, 21, 24, 20]
7 }
8 df = pd.DataFrame(dados)
9
10 # Plotando o histograma diretamente do DataFrame
11 df.plot(kind='hist', bins=5, edgecolor='black')
12
13 # Exibindo o histograma
14 plt.show()
```

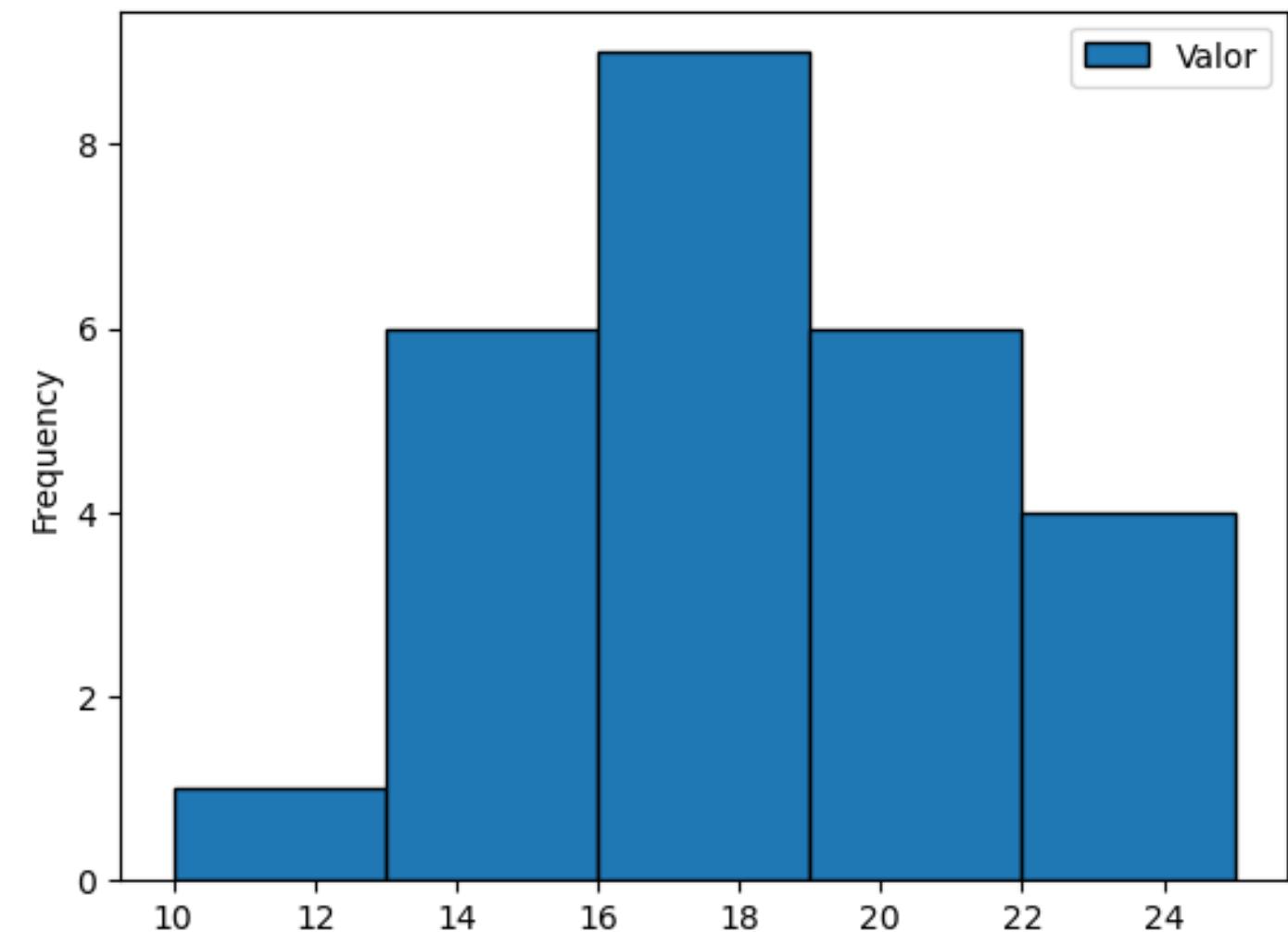
Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Histograma

O parâmetro **bins** define o número de bins ou intervalos no histograma. Neste exemplo, estamos usando 5 bins.

O parâmetro **edgecolor** define a cor das bordas dos bins.



Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Boxplot

Para criar um boxplot diretamente do DataFrame, usando Pandas, você pode utilizar o método `.boxplot()`:

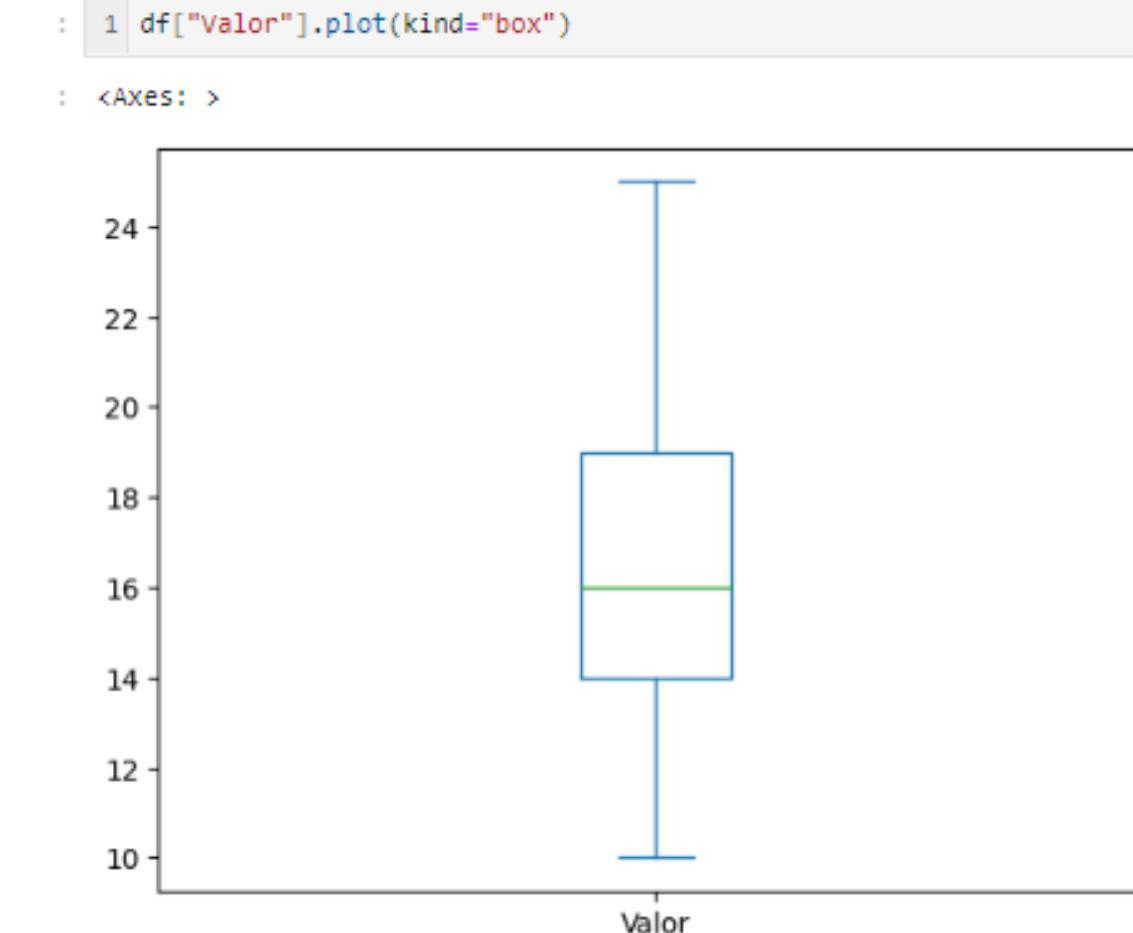
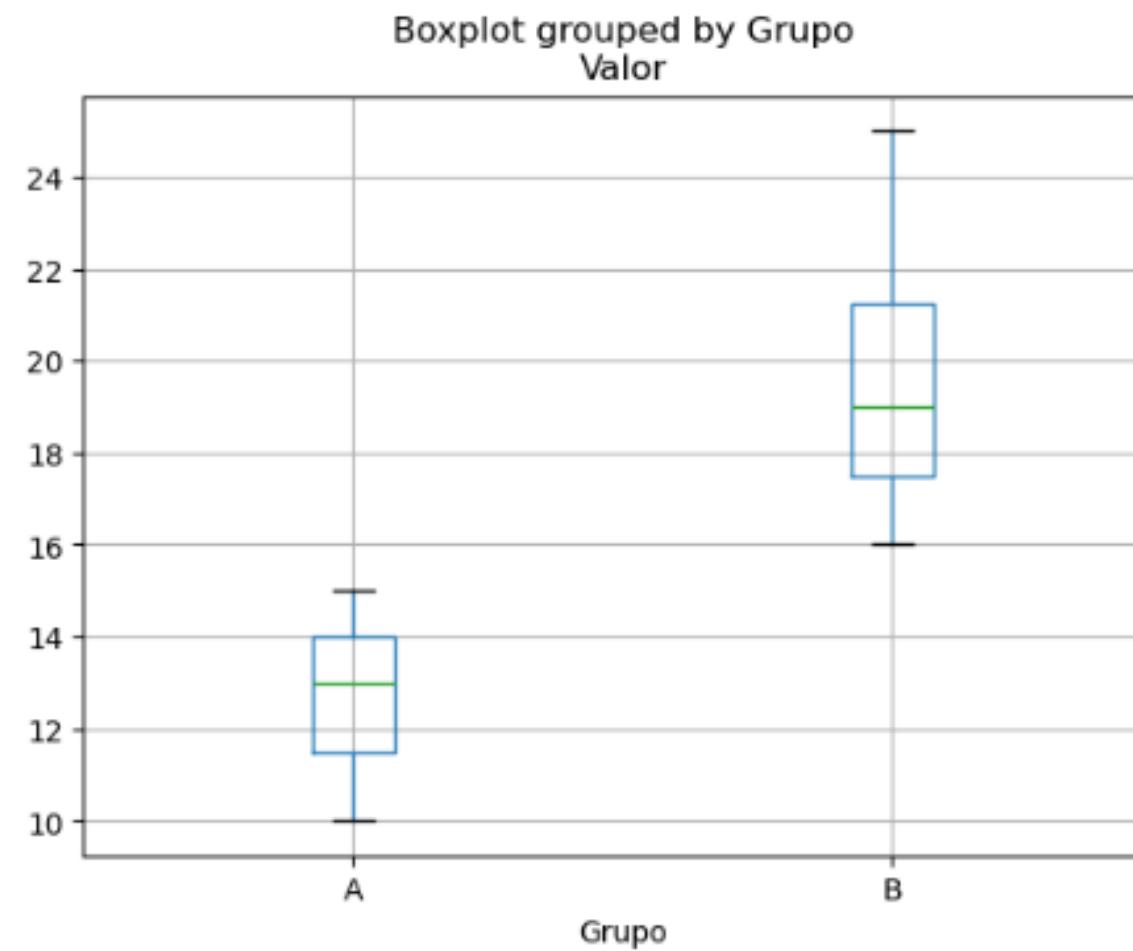
```
1 import pandas as pd
2
3 # Dados para o DataFrame
4 dados = {
5     'Grupo': ['A', 'A', 'A', 'B', 'B', 'B', 'B'],
6     'Valor': [10, 15, 13, 18, 16, 20, 25]
7 }
8 df = pd.DataFrame(dados)
9
10 # Plotando o boxplot diretamente do DataFrame
11 df.boxplot(by='Grupo', column='Valor')
12
13 # Exibindo o boxplot
14 plt.show()
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Boxplot

Também é possível utilizar o método `.plot()` com o parâmetro `kind='box'`:



Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Gráfico de pizza ou setores

Para criar um gráfico de pizza diretamente do DataFrame, usando Pandas, você pode utilizar o método `.plot()` com o parâmetro `kind='pie'`.

```
: 1 import pandas as pd
2
3 # Dados para o DataFrame
4 dados = {
5     'Categoria': ['A', 'B', 'C', 'D'],
6     'Valor': [30, 20, 25, 15]
7 }
8
9 # Criando o DataFrame
10 df = pd.DataFrame(dados)
11
12 # Plotando o gráfico de pizza diretamente do DataFrame
13 df.plot(kind='pie', y='Valor', labels=df['Categoria'], autopct='%1.1f%%', startangle=90)
14
15 # Exibindo o gráfico
16 plt.show()
17
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

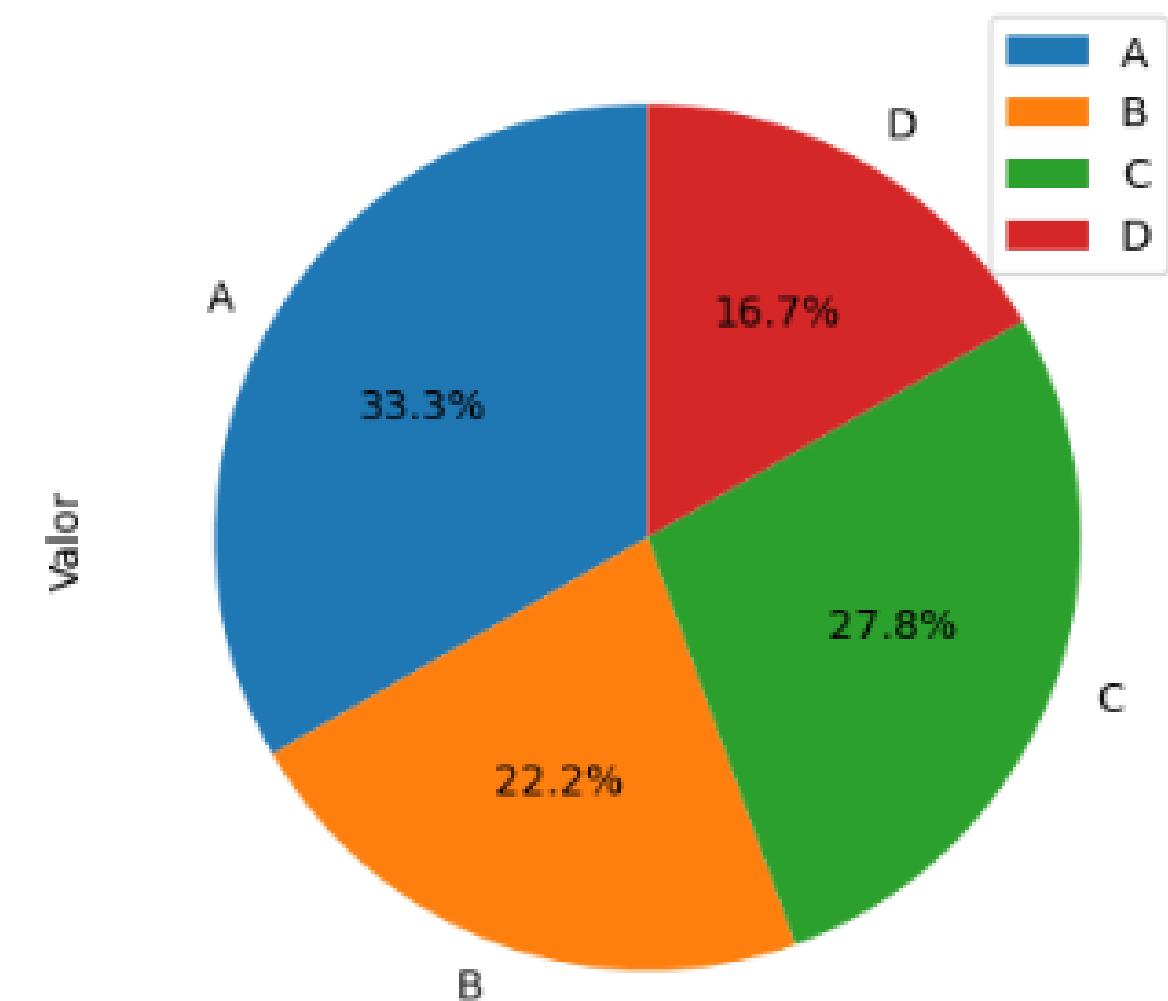
Este código vai criar e exibir um gráfico de pizza com os valores do DataFrame que criamos anteriormente. O parâmetro `y` especifica a coluna que contém os dados que serão usados para o gráfico de pizza.

Gráfico de pizza ou setores

Já o parâmetro **labels** permite especificar rótulos para as fatias da pizza.

O parâmetro **autopct** adiciona rótulos com a porcentagem de cada fatia.

O parâmetro **startangle** define o ângulo inicial da pizza.



Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Gráfico de pizza ou setores

Você pode criar um gráfico de pizza diretamente de uma série Pandas, usando o método `.plot()` com o parâmetro `kind='pie'`. Veja com `value_counts()`:

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 serie = pd.Series(['A', 'B', 'A', 'C', 'A', 'B', 'A', 'A'])
5
6 # Calculando a contagem de valores na série
7 contagem = serie.value_counts()
8
9 # Plotando o gráfico de pizza diretamente da contagem de valores
10 contagem.plot(kind='pie', autopct='%1.1f%%')
11
12 # Exibindo o gráfico
13 plt.show()
```

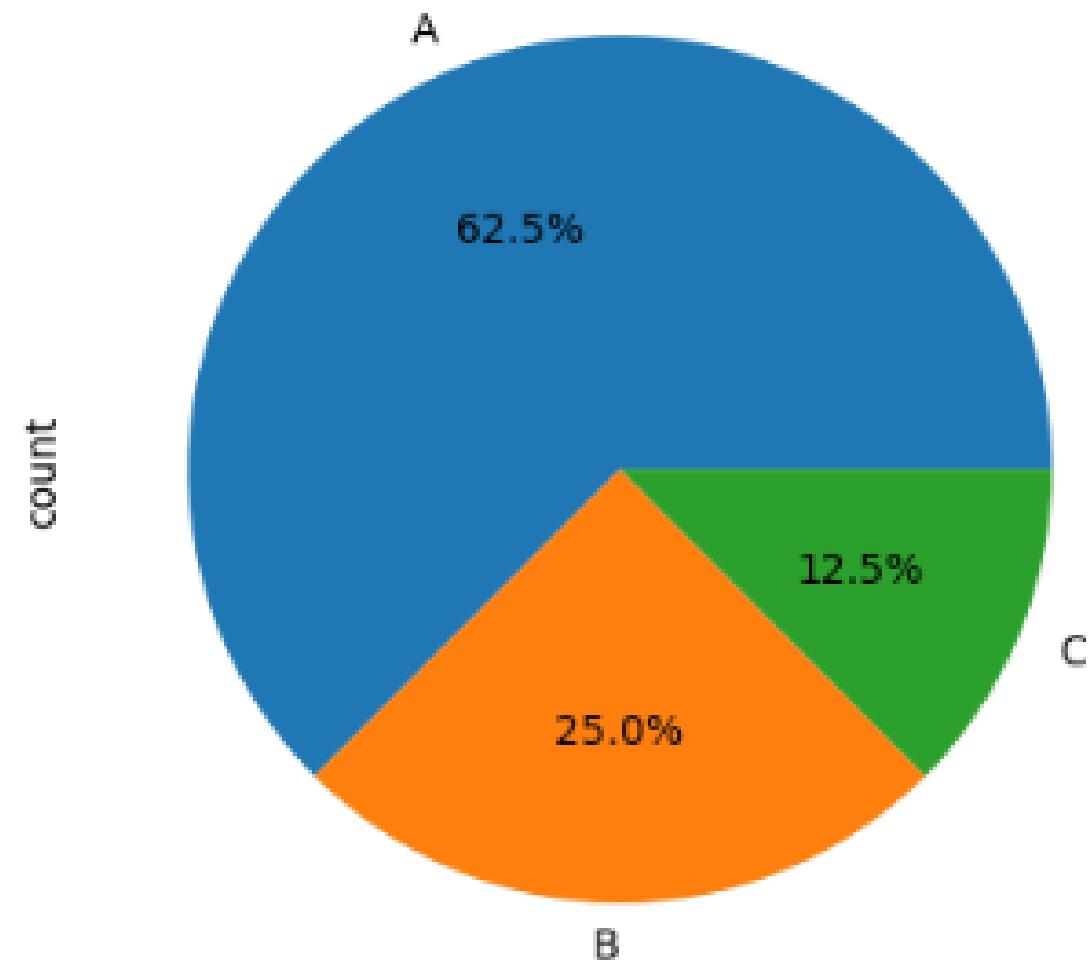
Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Neste exemplo, `serie.value_counts()` faz a contagem da ocorrência de cada valor na série.

Gráfico de pizza ou setores

Em seguida, o gráfico de pizza é plotado diretamente dessa contagem de valores usando `.plot(kind='pie')`.

O parâmetro **autopct** é usado para exibir as porcentagens das fatias.

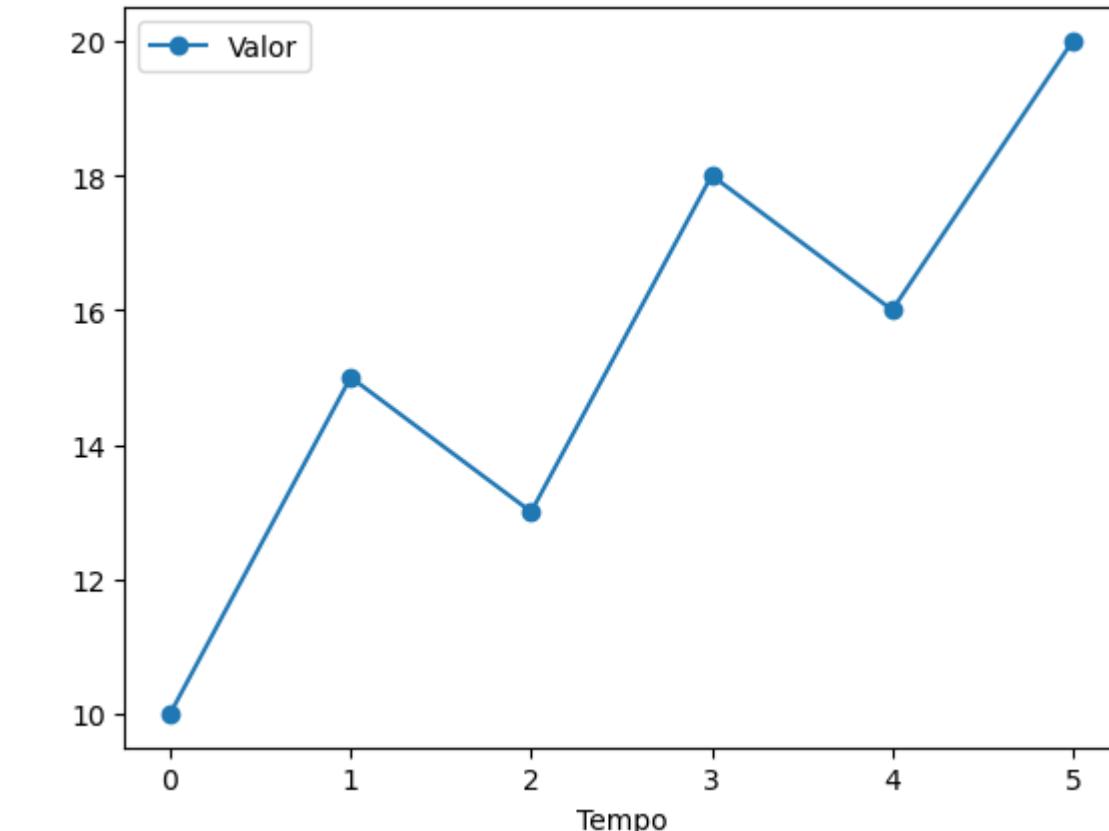


Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Gráfico de linha

Para criar um gráfico de linhas diretamente do DataFrame, usando Pandas, você pode usar o método `.plot()` do Pandas.

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 dados = {
5     'Tempo': [0, 1, 2, 3, 4, 5],
6     'Valor': [10, 15, 13, 18, 16, 20]
7 }
8
9 df = pd.DataFrame(dados)
10
11 # Plotando o gráfico de Linhas diretamente do DataFrame
12 df.plot(x='Tempo', y='Valor', kind='line', marker='o', linestyle='--')
13
14 # Exibindo o gráfico
15 plt.show()
```



Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Gráfico de barras

Para criar um gráfico de linhas diretamente do DataFrame, usando Pandas, você pode utilizar o método `.plot()` do Pandas.

```
1 import pandas as pd
2
3 # Dados para o DataFrame
4 dados = {
5     'Categoria': ['A', 'B', 'C', 'D'],
6     'Valor': [30, 20, 25, 15]
7 }
8
9 df = pd.DataFrame(dados)
10
11 # Plotando o gráfico de barras diretamente do DataFrame
12 df.plot(kind='bar', x='Categoria', y='Valor', color='skyblue')
13
14 # Exibindo o gráfico
15 plt.show()
```

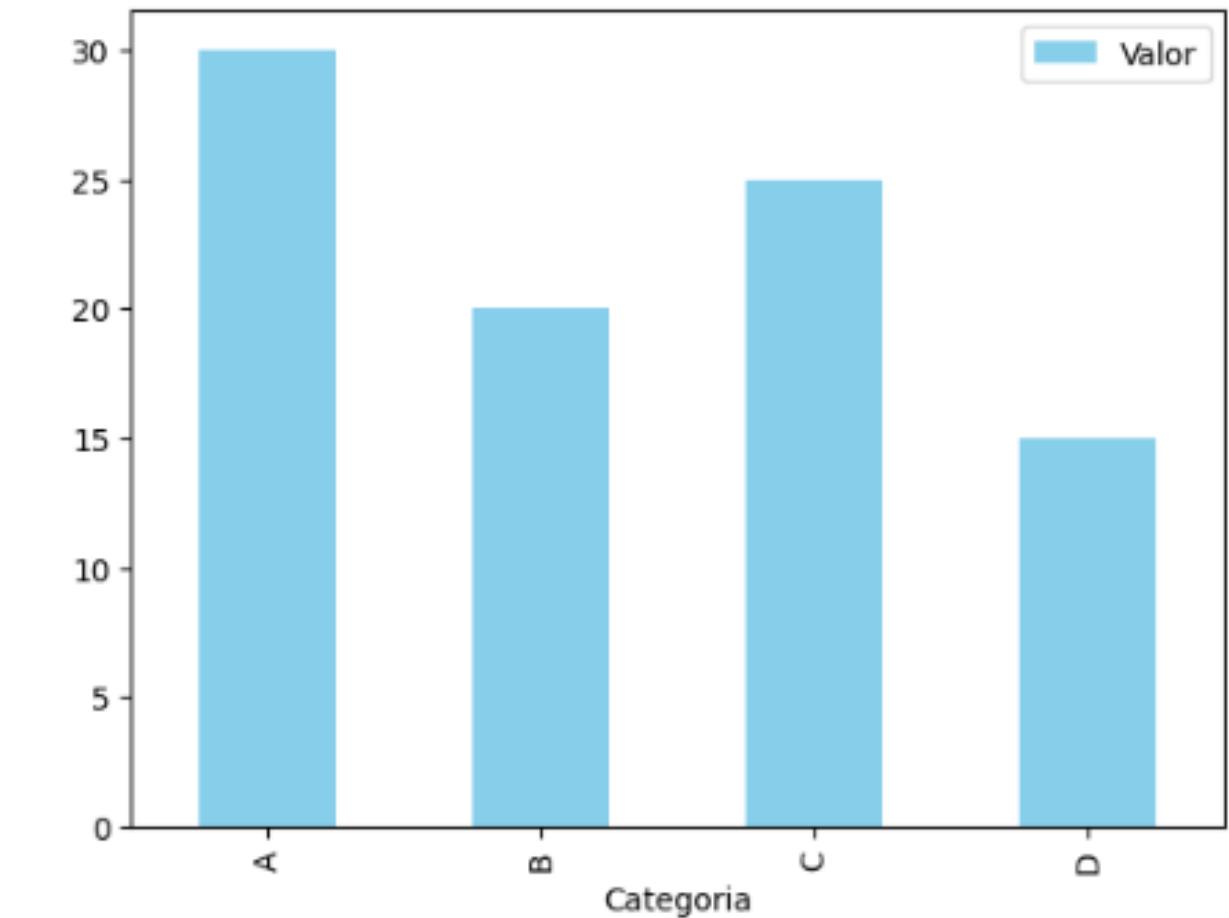
Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Gráfico de barras

Este código vai criar e exibir um gráfico de barras com os valores do DataFrame que criamos anteriormente.

O parâmetro `x` especifica a coluna que será usada para as categorias do eixo x, o parâmetro `y` especifica a coluna que será usada para os valores do eixo y.

O parâmetro `color` permite especificar a cor das barras.



Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Gráfico de barras

Para criar um gráfico de barras na horizontal no Pandas, você pode usar o parâmetro `kind='barh'` no método `.plot()`.

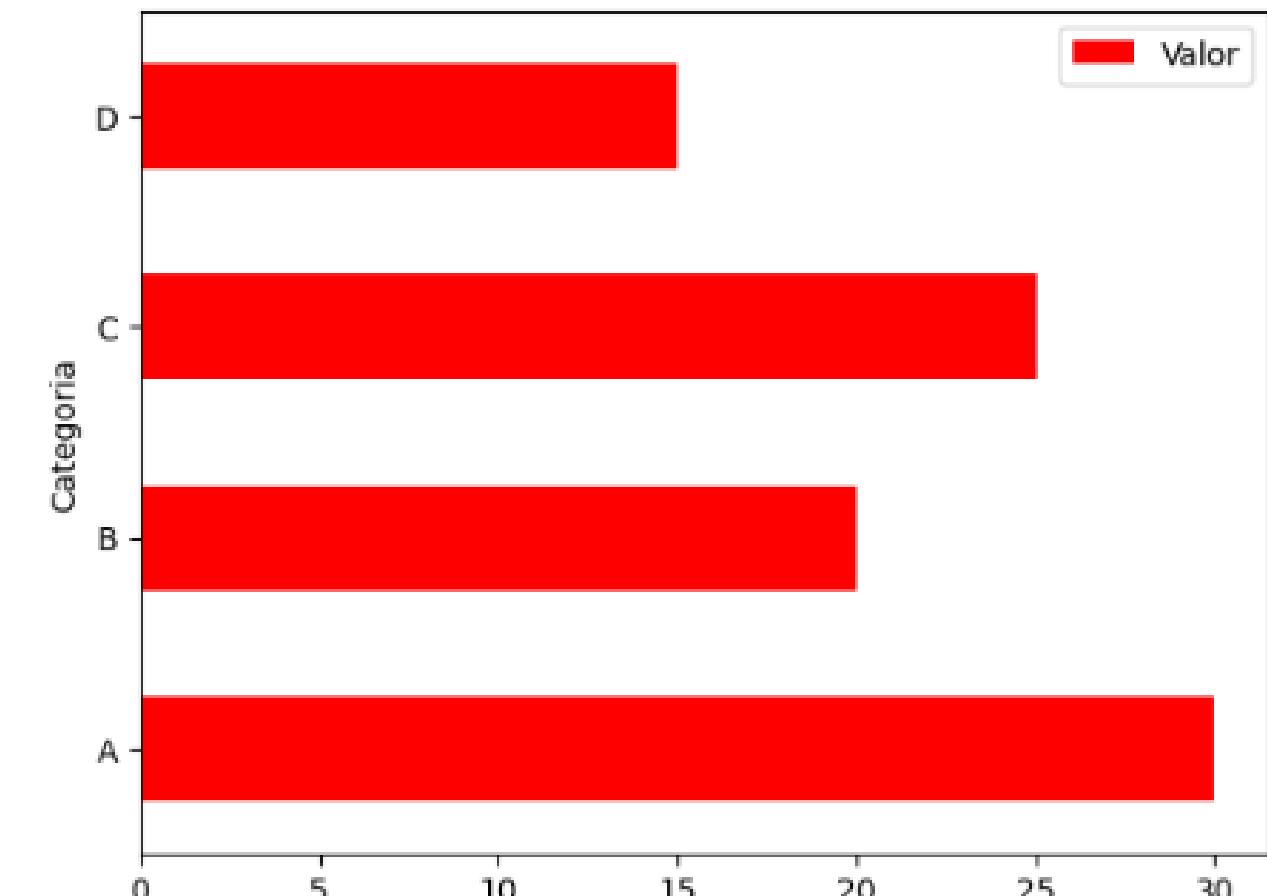
```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Dados para o DataFrame
5 dados = {
6     'Categoria': ['A', 'B', 'C', 'D'],
7     'Valor': [30, 20, 25, 15]
8 }
9
10 df = pd.DataFrame(dados)
11
12 # Plotando o gráfico de barras na horizontal diretamente do DataFrame
13 df.plot(kind='barh', x='Categoria', y='Valor', color='red')
14
15 # Exibindo o gráfico
16 plt.show()
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Gráfico de barras

Este código vai criar e exibir um gráfico de barras na horizontal com os valores do DataFrame que criamos anteriormente.

O parâmetro `x` especifica a coluna que será usada para as categorias do eixo `y`, o parâmetro `y` especifica a coluna que será usada para os valores do eixo `x`.



Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Gráfico de área

Já para criar um gráfico de área no Pandas, você pode usar o método `.plot()` com o parâmetro `kind='area'`.

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Dados para o DataFrame
5 dados = {
6     'Ano': [2010, 2011, 2012, 2013, 2014],
7     'Vendas': [100, 120, 90, 150, 110]
8 }
9
10 df = pd.DataFrame(dados)
11
12 # Plotando o gráfico de área diretamente do DataFrame
13 df.plot(x='Ano', y='Vendas', kind='area', color='skyblue')
14
15 # Exibindo o gráfico
16 plt.show()
17
```

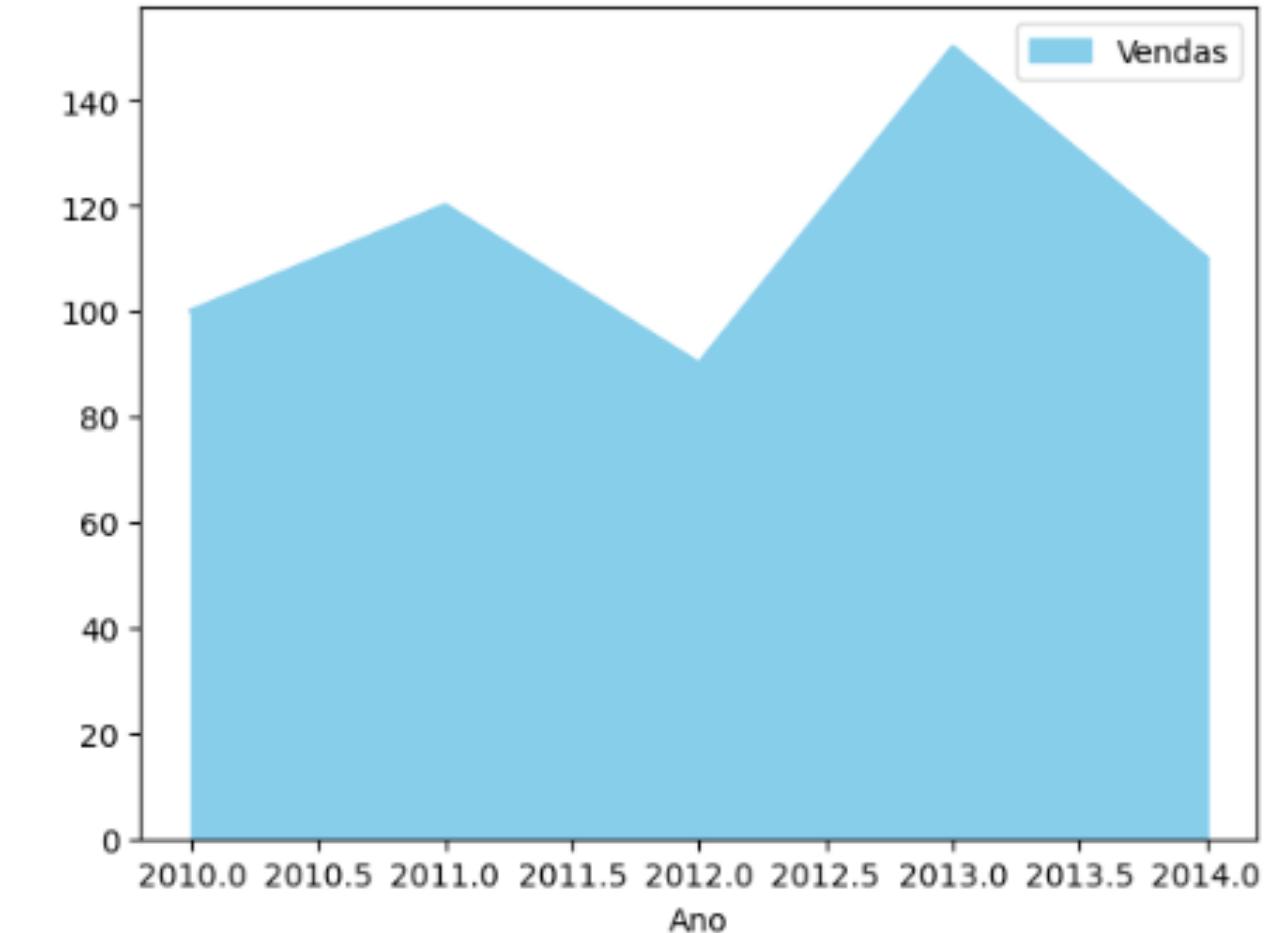
Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Gráfico de área

Este código vai criar e exibir um gráfico de área com os valores do DataFrame que criamos anteriormente.

O parâmetro `x` especifica a coluna que será usada para o eixo x, o parâmetro `y` especifica a coluna que será usada para o eixo y.



Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Gráfico de dispersão

E para criar um gráfico de dispersão no Pandas, você pode usar o método `.plot()` com o parâmetro `kind='scatter'`.

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Dados para o DataFrame
5 dados = {
6     'X': [1, 2, 3, 4, 5],
7     'Y': [2, 3, 5, 7, 11]
8 }
9 df = pd.DataFrame(dados)
10
11 # Plotando o gráfico de dispersão diretamente do DataFrame
12 df.plot(x='X', y='Y', kind='scatter', color='blue')
13
14 # Exibindo o gráfico
15 plt.show()
```

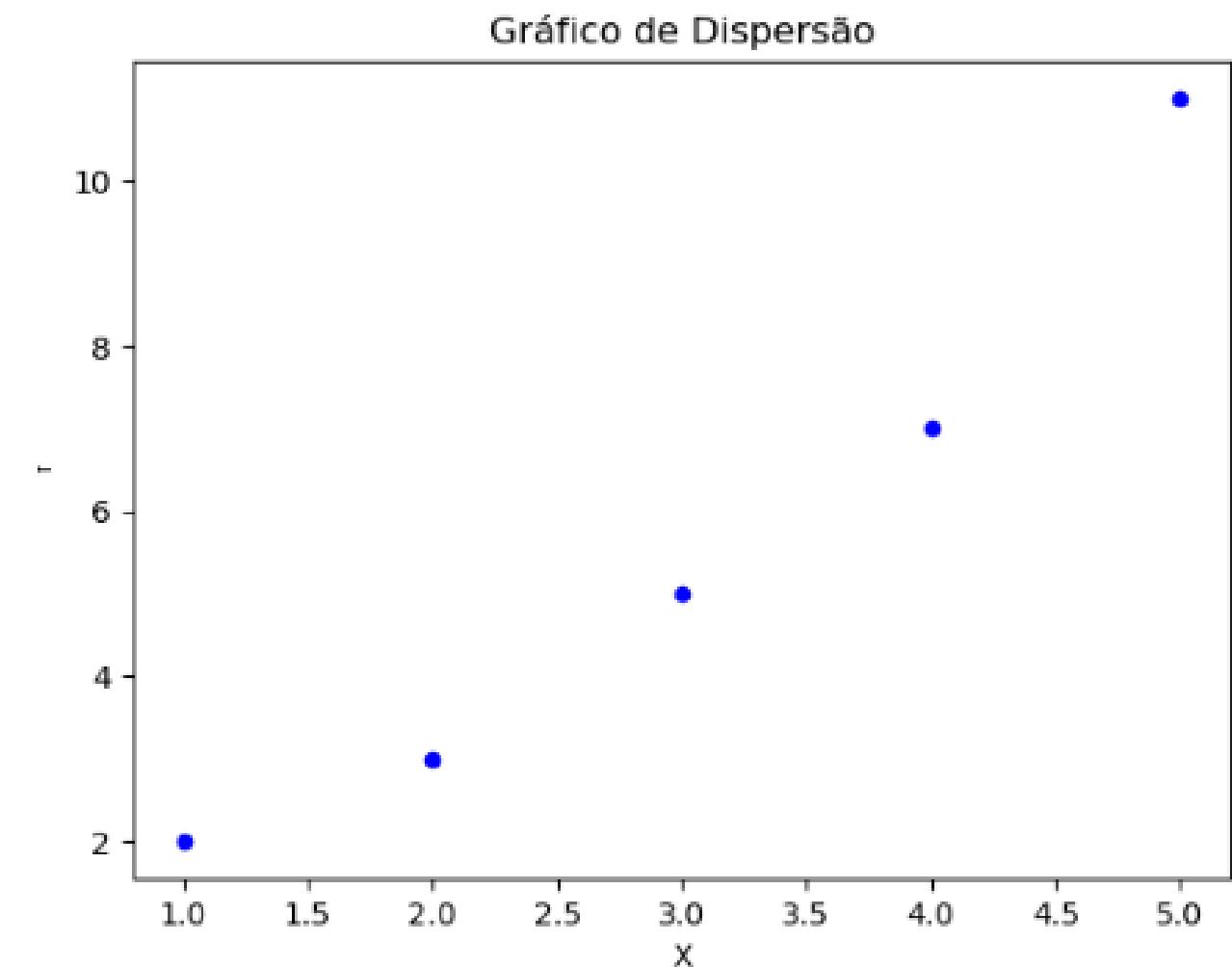
Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Construindo o conceito

Gráfico de dispersão

Este código vai criar e exibir um gráfico de área com os valores do DataFrame que criamos anteriormente.

O parâmetro x especifica a coluna que será usada para o eixo x, o parâmetro y especifica a coluna que será usada para o eixo y.



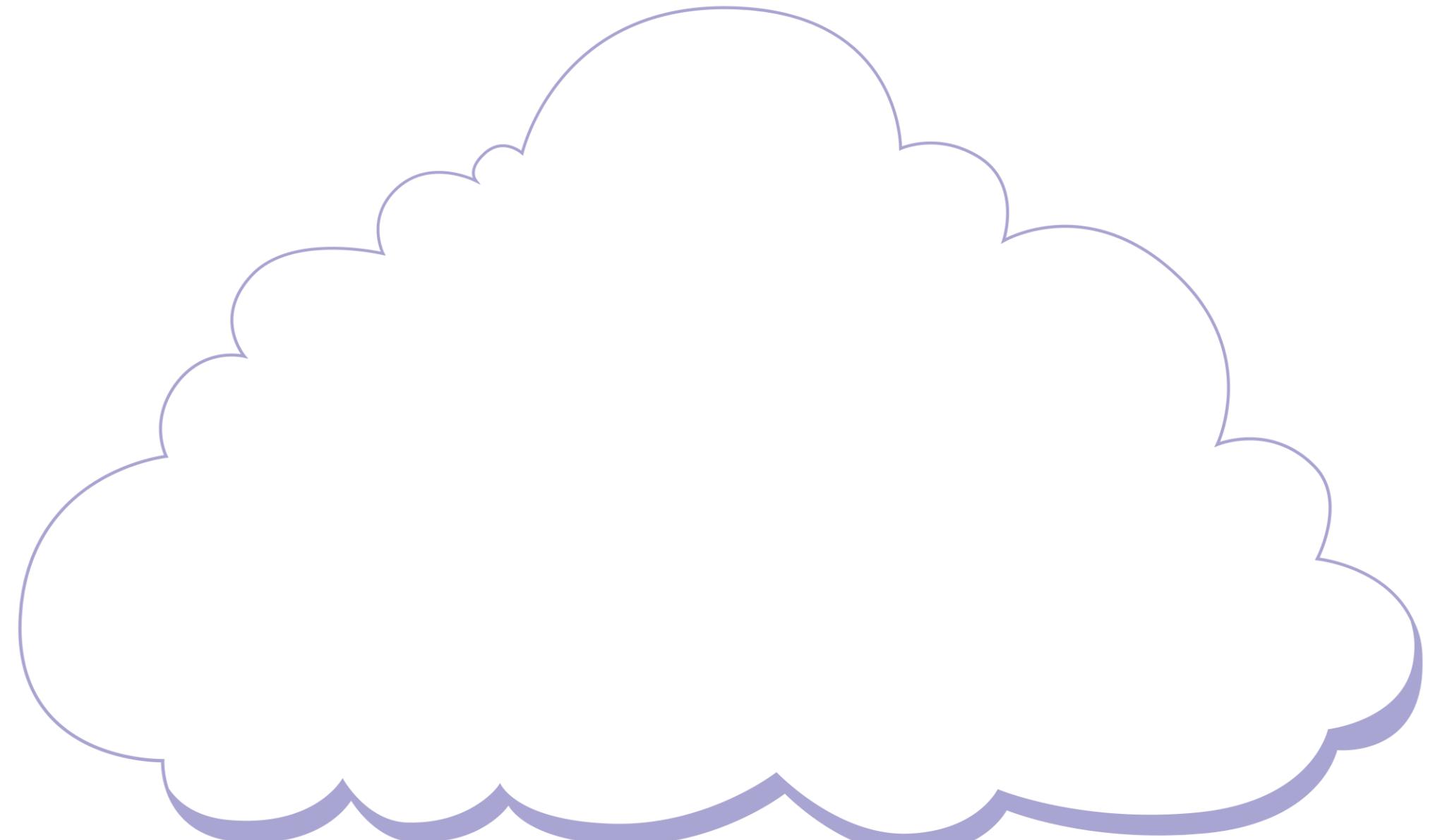
Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.



© Getty Images

O que nós
aprendemos
hoje?

Nuvem de palavras





© Getty Images

O que nós
aprendemos
hoje?

Então ficamos assim...

- 1 Entendemos que, para criar um gráfico no Pandas a partir de um DataFrame, usamos a **função plot()** e adicionamos o argumento “kind” para selecionar o tipo de gráfico que queremos;
- 2 Vimos diversos **tipos de gráficos**, como: histograma, boxplot, linhas, barras, pizza, área e dispersão;
- 3 Compreendemos a importância dos gráficos para a melhor **visualização** dos dados.

Saiba mais

Que tal aprender mais sobre visualização de gráficos de frequência?

Acesse:

MATHEUS, Y. Visualizando dados de frequência. Alura, 24 jul. 2019. Disponível em: <https://www.alura.com.br/artigos/visualizando-dados-de-frequencia>. Acesso em: 2 ago. 2024.

Referências da aula

MCKINNEY, W. *Python para análise de dados: tratamento de dados com Pandas, NumPy & Jupyter.* São Paulo: Novatec, 2023.

PANDAS. DataFrame, [s.d.]. Disponível em:

<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html>. Acesso em: 2 ago. 2024.

Identidade visual: imagens © Getty Images.

Educação Profissional Paulista

Técnico em
**Ciência de
Dados**

Secretaria da
Educação



SÃO PAULO
GOVERNO DO ESTADO