Biblioteca Pandas

Luís Fernandes Saucedo Souza
Leandro Andrade de Oliveira

VISÃO GERAL

O Pandas é um pacote de alto desempenho que fornece um conjunto abrangente de estruturas para trabalhar com dados. O Pandas se destaca no tratamento de dados estruturados, como conjuntos de dados contendo muitas variáveis, trabalhando com valores ausentes e mesclando vários conjuntos de dados. pandas é um componente essencial da ciência Python ao operar em dados. O Pandas também fornece métodos robustos e de alto desempenho para importar e exportar para uma ampla gama de formatos (SHEPPARD, 2020).

O nome *Pandas* é derivado do termo *panel data* (dados em painel), que é um termo que descreve dados compostos de múltiplas observações através do tempo para os mesmos indivíduos (tmfilho.github).

Tecnicamente, a Pandas Data Frame é uma estrutura de dados tabular bidimensional potencialmente heterogênea e de tamanho variável com eixos rotulados (linhas e colunas). Por sua vez, um quadro de dados é uma estrutura de dados bidimensional, o que significa que os dados são organizados em uma tabela em linhas e colunas. O Pandas Data Frame consiste em três componentes principais: dados, linhas e colunas (COUTINHO DE OLIVEIRA, 2021).

Podemos criar um Data Frame atribuindo o nome da coluna e uma lista de valores para essa coluna e o construtor criará um índice para cada um dos valores (Figura 1).

Exemplo:

```
import pandas as pd

df = pd.DataFrame({'Coluna 1': ["Elemento 0 coluna 1", "Elemento 1 coluna 1", "Elemento 2 coluna 1"],
                   'Coluna 2': ["Elemento 0 coluna 2", "Elemento 1 coluna 2", "Elemento 2 coluna 2"],
                   'Coluna 3': ["Elemento 0 coluna 3", "Elemento 1 coluna 3", "Elemento 2 coluna 3"]})

df
```

	Coluna 1	Coluna 2	Coluna 3
0	Elemento 0 coluna 1	Elemento 0 coluna 2	Elemento 0 coluna 3
1	Elemento 1 coluna 1	Elemento 1 coluna 2	Elemento 1 coluna 3
2	Elemento 2 coluna 1	Elemento 2 coluna 2	Elemento 2 coluna 3

Figura 1. Exemplo de Data Frame.
Fonte: Do Autor

Principais comandos

Criar:

Há várias maneiras de criar um Data Frame de Pandas. Na maioria dos casos, é usado o construtor Data Frame que fornecerá os dados, rótulos e outras informações. Os dados podem ser passados ao construtor como uma lista bidimensional, tupla[1] ou matriz NumPy. Também é possível passá-los como um dicionário, Serie Pandas[2], ou como um dos vários outros tipos de dados (STOJILJKOVIĆ, 2020).

Para este exemplo, suponha que você esteja usando um dicionário para passar os dados de alunos como nome, cidade, idade, nota e queira identificá-los pelo número de matrícula como na Figura 2:

[1] **tupla:** tipo de dado semelhante a uma lista, mas que não pode ser alterado.
[2] **Serie Pandas:** matriz unidimensional rótulos de linha e coluna.

```
data = {
     'Nome': ['Alberto', 'Bruno', 'Carlos', 'Davi', 'Estevão'],
     'Cidade': ['Porto Alegre', 'São Paulo', 'Belo Horizonte',
                'Rio de Janeiro', 'Salvador'],
     'Idade': [21, 28, 33, 24, 26],
     'Nota': [8.8, 7.9, 8.1, 8.0, 6.8]
 }

matriculas = [139227, 139228, 139229, 139230, 139231]

df = pd.DataFrame(data=data, index=matriculas)

df
```

	Nome	Cidade	Idade	Nota
139227	Alberto	Porto Alegre	21	8.8
139228	Bruno	São Paulo	28	7.9
139229	Carlos	Belo Horizonte	33	8.1
139230	Davi	Rio de Janeiro	24	8.0
139231	Estevão	Salvador	26	6.8

Figura 2. Rotulando um dicionário.
Fonte: Do Autor

Localizar:

Uma coluna pode ser acessada com o código: `dataframe['coluna']` ou `dataframe.coluna` e como output retorna um `pandas.Series` de uma coluna (Veja a Figura 3).

```
[6]  notas  = df["Nota"]
     notas

139227    8.8
139228    7.9
139229    8.1
139230    8.0
139231    6.8
Name: Nota, dtype: float64
```

Figura 3. Localizando a coluna.
Fonte: Do Autor

Leitura e exportação de dados:

Outros comandos:

- `df.index` (Retorna os valores dos índices das linhas)
- `df.columns` (Retorna os valores dos índices das colunas)
- `df.to_numpy()` (Transforma um Data Frame em uma array do NumPy)
- `df_.ndim` (Retorna a dimensão do Data Frame)
- `df_.shape` (Retorna o número de linhas e colunas do Data Frame)
- `df_.size` (Retorna o número de dados do Data Frame)
- `df.loc[i]` (Localiza a linha com índice 'i')
- `df.iloc[i]` (Localiza a linha na posição 'i')
- `df.loc[11:15, ['nome', 'cidade']]`(Como parâmetro pode filtrar a linha e coluna)
- `df.at[i, 'nota']` (Retorna o dado na linha 'i' da coluna)
- `df.insert(loc=x, column='nome coluna',value=[a,b, ...]))`(Insere uma coluna na localização 'x')
- `del df['coluna']` (Deleta a coluna)

Operações com colunas:

Pode-se fazer operações com as colunas e filtrá-las (veja Figura 4).

Exemplo:

```
df["trabalho1"] = [6.5, 7.1, 3.4, 5.8, 9.1] #criar novas colunas
df["trabalho2"] = [7.5, 3.1, 8.3, 3.6, 6.2]
df["trabalho3"] = [9.6, 9.5, 9.2, 9.2, 9.6]

df["Nota"] = 0.5*df["trabalho1"] + 0.3*df["trabalho2"] + 0.2*df["trabalho3"] #atribuindo peso e redefinindo coluna "Notas"

df["aprovado"] = df["Nota"] >= 7.0  #escolher os index que são maiores que 7

df["exame"] = (df["Nota"] >= 2.0) & (df["Nota"] < 7.0) #escolher os que estão entre 2 e 7

media = df["Nota"].mean() #fazer a média das notas

print(df[["aprovados", "exame"]])
print(f"Média das notas: {media}")
```

```
        aprovados  exame
139227       True  False
139228      False   True
139229      False   True
139230      False   True
139231       True  False
Média das notas: 6.796000000000001
```

Figura 4. Filtrando colunas.
Fonte: Do Autor

Importando e Exportando Dados

A importação e exportação de dados são tarefas essenciais no processo de análise e manipulação de dados. O Pandas oferece muitos recursos e funcionalidades para facilitar esse processo. (SHEPPARD 2020)

O Pandas permite trabalhar com vários formatos de dados, como Excel, CSV, WEB, JSON, APIS, Tabela de HTML, arquivos de texto entre outros.

Os comandos **read_hdf()**, **read_stata()**, e **read_csv()** são métodos do Pandas, que são chamados em um objeto Data Frame para realizar a importação de dados. Em Python, um método é uma função que pertence a um objeto específico. No caso do Pandas, esses métodos são chamados em um objeto Data Frame para realizar a leitura de dados de diferentes formatos.

Para importar dados no Pandas, podemos usar métodos como **'read_csv()'**, **'read_excel()'**, **'read_sql_query()'**, entre outros. Por exemplo, para importar dados de um arquivo CSV, basta usar o método **'read_csv()'** (Figura 5) e fornecer o caminho do arquivo como argumento. Isso criará um Data Frame, uma estrutura de dados do Pandas que permite manipular e analisar os dados de forma eficiente. Outros exemplos como

‘read_excel()’ (Figura 6) importam de uma tabela de Excel, assim como também podemos aplicar em tabelas SQL Query o método **‘read_sql_query()’** (Figura 7).

```
[4]  # Importar de um arquivo CSV:
     df_dsa = pd.read_csv('pnad_2015_clean.csv')
```

Figura 5. Importando de um arquivo CSV.
Fonte: Do Autor

Nas figuras 6 e 7 temos outros dois métodos de importação de dados que nos permite importar dados de uma tabela Excel e também tabelas SQL Query. Basta aplicarmos o método ‘**read_excel()’** para tabelas Excel e o método **‘read_sql_query()’** para tabelas Sql Query.

```
# Importando de um Excel
import pandas as pd

dados = pd.read_excel('caminho/do/arquivo.xlsx', sheet_name='nome_da_planilha')
```

Figura 6. Importando de um Excel.
Fonte: Do ChatGPT

```python
# Importar de uma tabela SQL
import pandas as pd
import sqlite3

conexão = sqlite3.connect('banco_de_dados.db')
consulta = 'SELECT * FROM nome_da_tabela'
dados = pd.read_sql_query(consulta, conexão)
```

Figura 7. Importar de uma tabela SQL.
Fonte: Do ChatGPT

Comandos de importação:

- read_excel (Importa um arquivo Excel para um Data Frame)
- read_csv (Importa dados de um arquivo Comma-Separated Values)
- read_table (Importa dados tabulares de um arquivo em formato de texto delimitado)
- read_hdf (Importa dados de arquivo Hierarchical Data Format)
- read_stata (Importa dados de arquivo Stata)

Como Exportar um arquivo

A exportação de dados com o Pandas é igualmente simples e flexível. Para exportar um Data Frame para um arquivo CSV, por exemplo, podemos usar o método **'to_csv()'** (Figura 8) e fornecer o caminho do arquivo como argumento. É possível especificar outros parâmetros, como **'index'**, para controlar a inclusão do índice do Data Frame no arquivo CSV. (CHATGPT)

```python
[ ] # Exportando para um arquivo CSV
    dados.to_csv('caminho/do/arquivo.csv', index=False)
    # O parâmetro index=False evita a inclusão do índice do DataFrame no arquivo CSV
```

Figura 8. Exportando um arquivo CSV.
Fonte: Do ChatGPT

Da mesma forma, podemos exportar dados para um arquivo Excel usando o método **'to_excel()'** (Figura 9) . É possível definir o nome da planilha, bem como outros parâmetros, como **'index'**, para personalizar a exportação. Além disso, o Pandas permite exportar dados para bancos de dados SQL usando o método **'to_sql()'** (Figura 10). Podemos especificar o nome da tabela e a conexão com o banco de dados, além de opções adicionais, como **'if_exists'**, que define o comportamento em caso de tabela existente.

```
[ ]  # Exportando para um arquivo Excel
     dados.to_excel('caminho/do/arquivo.xlsx', sheet_name='nome_da_planilha', index=False)
```

Figura 9. Exportando para um arquivo Excel.
Fonte: Do ChatGPT

```
[ ]  # Exporta para uma tabela SQL
     import sqlite3

     conexão = sqlite3.connect('banco_de_dados.db')
     dados.to_sql('nome_da_tabela', conexão, if_exists='replace')
     # O parâmetro if_exists='replace' substitui a tabela existente, se houver
```

Figura 10. Exporta para uma tabela SQL.
Fonte: Do ChatGPT

Comandos de exportação:

- to_csv (Exporta dados para um arquivo Comma-Separated Values)
- to_excel (Exporta dados para um arquivo Excel)
- to_json (Exporta dados para um arquivo JavaScript Object Notation)
- to_sql (Exporta dados para uma tabela em um banco de dados SQL)
-

Gráficos

Com a biblioteca `matplotlib`, pode-se plotar gráficos, selecionando quais dados quer se comparar, conforme Figura 11.

```
[51] import matplotlib.pyplot as plt

     trabalhos = df[["trabalho1", "trabalho2", "trabalho3"]]

     ax = trabalhos.plot.hist(bins=10, alpha=0.5)
```

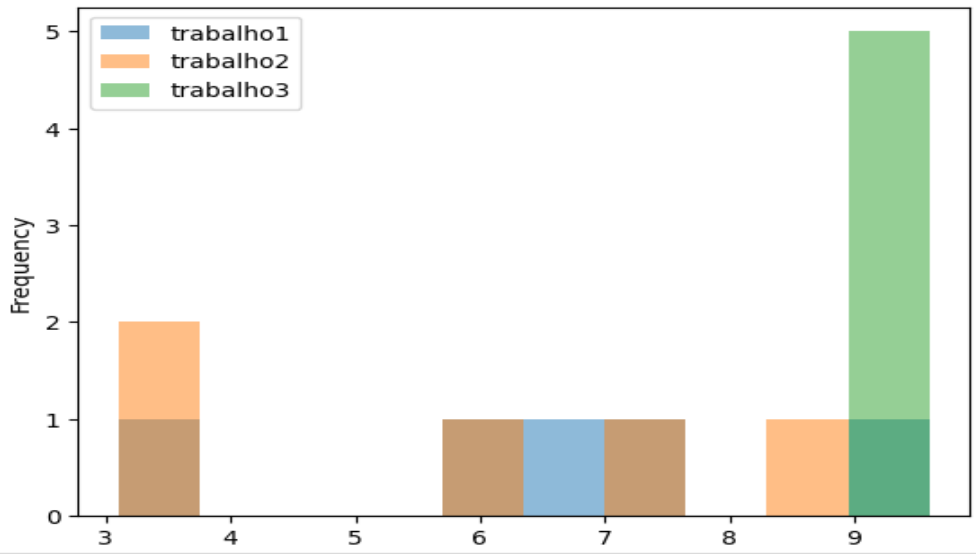

Figura 11. Plotando colunas.
Fonte: Do Autor

Importação e Exportação de dados com Pandas

Aplicações

Saúde :

É usado na saúde para a análise de dados como os da pandemia do COVID-19, para estudar as relações de comorbidades como em (ROSA, 2022) ou para analisar a respostas dos governos, como em (IMTYAZ, 2020).

Biologia:

Análise de dados de florestas nativas, seu estado de preservação, seus recursos e sua composição como em (VAGIZOV, 2021).

Engenharia:

Usado para machine learning na engenharia agrícola para analisar os dados de produção, através de um método de tomada de decisão, a qualidade da soja produzida, por exemplo (OLIVEIRA, 2022).

Química:

Análise de dados de espectroscopia de concentração de partículas de ouro na água (HUGHES, 2015).

Economia:

Obter projeções e estimativas do mercado de ação através da análise de big data (ARAÚJO, 2016).

Referências

ARAÚJO, Alcides Carlos De ; MONTINI, Alessandra De Ávila . Técnicas de Big Data e Projeção de Risco de Mercado utilizando Dados em Alta Frequência. 3. ed. São Paulo: **FUTURE STUDIES RESEARCH JOURNAL**, 2016. 83 – 108 p. v. 8. ISBN: 2175-5825.

COUTINHO DE OLIVEIRA, Thiago. Quais são as vantagens e funcionalidades da biblioteca Pandas data frame?. **Voitto**, 2021. Disponível em: https://www.voitto.com.br/blog/artigo/dataframe. Acesso em: 18 maio 2023.

HUGHES, Adam ; LIU, Zhaowen ; REEVES, M. E. . Scikit-spectra: Explorative Spectroscopy in Python. 6. ed. Washington: **Journal of Open Research Software**, 2015. v. 3. DOI: http://dx.doi.org/10.5334/jors.bs

IMTYAZ, Ayman ; HALEEM, Abid ; JAVAID, Mohd . Analysing governmental response to the COVID-19 pandemic. 10. ed. New Delhi, India: **Journal of Oral Biology and Craniofacial Research**, 2020. 504–513 p. DOI: 10.1016/j.jobcr.2020.08.005.

OLIVEIRA, Daniela C. De ; BARBOSA, Uender C. ; BERGLAND, Alcídia C. R. O. Bergland ; RESENDE, Osvaldo; OLIVEIRA, Daniel E. C. De. G-SOJA - WEBSITE WITH PREDICTION ON SOYBEAN CLASSIFICATION USING MACHINE LEARNING. Goiás: **Journal of the Brazilian Association of Agricultural Engineering**, 2022. v. 42. ISBN 1809-4430.

ROSA, Ruy Roberto Porto Ascenso; LAVAREDA FILHO, Ronem Matos ; LINHARES, José Elislande Breno De Souza . Influência das comorbidades para a ocorrência de óbitos por COVID-19 em 2020: razão de chances no estado do Amazonas.48. ed. Amazonas: HU Revista, 2022. 1-8 p. DOI: 10.34019/1982-8047.2022.v48.37689.

SHEPPARD, Kevin . **Introduction to Python for Econometrics, Statistics and Data Analysis**. 4. ed. Oxford: University of Oxford, 2020. .

STOJILJKOVIĆ, Mirko. O DataFrame dos pandas: torne o trabalho com dados delicioso. **Real Python,** 2020. Disponível em: https://realpython.com/pandas-dataframe. Acesso em: 18 maio 2023.

User guide. **Pandas.** Disponível em: User Guide — pandas 2.0.1 documentation (pydata.org). Acesso em: 18 maio 2023.

VAGIZOV, M ; POTAPOV, A; KONZHGOLADZE, K; STEPANOV, S; MARTYN, I. Prepare and analyze taxation data using the Python Pandas library. 876. ed. St. Petersburg: **IOP Conf. Series: Earth and Environmental Science**, 2021. DOI: 10.1088/1755-1315/876/1/012078.