



FACULDADE ESTÁCIO DE SÁ

CURSO: DESENVOLVIMENTO FULL STACK

4º SEMESTRE – MATRÍCULA 202302595341

Repositório GitHub - [alaimalmeida/tratandoImensidaoDosDados](https://github.com/alaimalmeida/tratandoImensidaoDosDados)

ALAIM ALMEIDA DE OLIVEIRA

Tratando a imensidão dos dados

Salvador – BA

2024

1. Introdução

A manipulação de dados é uma tarefa essencial em diversas áreas, como análise de dados, ciência de dados e gestão de projetos. O formato CSV (Comma-Separated Values) é amplamente utilizado para armazenar e transferir dados devido à sua simplicidade e compatibilidade com diversas ferramentas. Neste relatório, abordaremos como ler arquivos CSV utilizando a linguagem Python e a biblioteca Pandas.

2. Objetivo

Demonstrar o processo de leitura de arquivos CSV utilizando a biblioteca Pandas, configurando corretamente parâmetros como separador de colunas, engine e encoding.

Microatividade 1: Descrever como ler um arquivo CSV usando a biblioteca Pandas (Python)

1. Procedimento

- a. Instalar a biblioteca Pandas, caso ainda não esteja instalada:
“pip install pandas”
- b. Importar a biblioteca Pandas no código Python:
“import pandas as pd”
- c. Definir o nome do arquivo CSV que será lido:
“arquivo_csv = "dados.csv”
- d. Ler o arquivo CSV usando a função `read_csv`, especificando o separador de colunas (`sep`), a engine (`engine='python'`) e o encoding (`encoding='utf-8'`):
“tabela = pd.read_csv(arquivo_csv, sep=';', engine='python', encoding='utf-8')”
- e. Exibir as primeiras linhas do DataFrame para verificar a leitura dos dados:
“print(tabela)”

Ficando dessa forma do código completo:

```
projeto.py x
projeto.py > ...
1  import pandas as pd
2
3  # Definindo o nome do arquivo CSV
4  arquivo_csv = "dados.csv"
5
6  # Lendo o arquivo CSV
7  # Usando o separador ';', a engine 'python' e especificando o encoding (se necessário)
8  tabela = pd.read_csv(arquivo_csv, sep=';', engine='python', encoding='utf-8')
9
10 # Exibindo os dados lidos
11 print(tabela)
12
```

A parte do arquivo CSV ficou dessa forma:

```
projeto.py x  dados.csv
dados.csv > data
1  ID;Duration;Date;Pulse;Maxpulse;Calories
2  0;60;'2020/12/01';110;130;4091
3  1;60;'2020/12/02';117;145;4790
4  2;60;'2020/12/03';103;135;3400
5  3;45;'2020/12/04';109;175;2824
6  4;45;'2020/12/05';117;148;4060
7  5;60;'2020/12/06';102;127;3000
8  6;60;'2020/12/07';110;136;3740
9  7;450;'2020/12/08';104;134;2533
10 8;30;'2020/12/09';109;133;1951
11 9;60;'2020/12/10';98;124;2690
12 10;60;'2020/12/11';103;147;3293
13 11;60;'2020/12/12';100;120;2507
14 12;60;'2020/12/12';100;120;2507
15 13;60;'2020/12/13';106;128;3453
16 14;60;'2020/12/14';104;132;3793
17 15;60;'2020/12/15';98;123;2750
18 16;60;'2020/12/16';98;120;2152
19 17;60;'2020/12/17';100;120;3000
20 18;45;'2020/12/18';90;112;NaN
21 19;60;'2020/12/19';103;123;3230
22 20;45;'2020/12/20';97;125;2430 2
23 1;60;'2020/12/21';108;131;3642
24 22;45;NaN;100;119;2820
25 23;60;'2020/12/23';130;101;3000
26 24;45;'2020/12/24';105;132;2460
27 25;60;'2020/12/25';102;126;3345
28 26;60;20201226;100;120;2500
29 27;60;'2020/12/27';92;118;2410
30 28;60;'2020/12/28';103;132;NaN
31 29;60;'2020/12/29';100;132;2800
32 30;60;'2020/12/30';102;129;3803
33 31;60;'2020/12/31';92;115;2430
```

Ao executar o arquivo, fica da seguinte forma:

SAÍDA	CONSOLE DE DEPURACÃO		PORTAS	COMENTÁRIOS	TERMINAL	
ID	Duration	Date	Pulse	Maxpulse	Calories	
0	0	60	'2020/12/01'	110	130	4091
1	1	60	'2020/12/02'	117	145	4790
2	2	60	'2020/12/03'	103	135	3400
3	3	45	'2020/12/04'	109	175	2824
4	4	45	'2020/12/05'	117	148	4060
5	5	60	'2020/12/06'	102	127	3000
6	6	60	'2020/12/07'	110	136	3740
7	7	450	'2020/12/08'	104	134	2533
8	8	30	'2020/12/09'	109	133	1951
9	9	60	'2020/12/10'	98	124	2690
10	10	60	'2020/12/11'	103	147	3293
11	11	60	'2020/12/12'	100	120	2507
12	12	60	'2020/12/12'	100	120	2507
13	13	60	'2020/12/13'	106	128	3453
14	14	60	'2020/12/14'	104	132	3793
15	15	60	'2020/12/15'	98	123	2750
16	16	60	'2020/12/16'	98	120	2152
17	17	60	'2020/12/17'	100	120	3000
18	18	45	'2020/12/18'	90	112	NaN
19	19	60	'2020/12/19'	103	123	3230
20	20	45	'2020/12/20'	97	125	2430 2
21	1	60	'2020/12/21'	108	131	3642
22	22	45	NaN	100	119	2820
23	23	60	'2020/12/23'	130	101	3000
24	24	45	'2020/12/24'	105	132	2460
25	25	60	'2020/12/25'	102	126	3345
26	26	60	20201226	100	120	2500
27	27	60	'2020/12/27'	92	118	2410
28	28	60	'2020/12/28'	103	132	NaN
29	29	60	'2020/12/29'	100	132	2800
30	30	60	'2020/12/30'	102	129	3803
31	31	60	'2020/12/31'	92	115	2430

Microatividade 2: Descrever como criar um subconjunto de dados a partir de um conjunto existente usando a biblioteca Pandas (Python)

Procedimentos:

1. No mesmo arquivo/script utilizado na microatividade 1, crie uma nova variável;
2. Atribua, a essa nova variável, um subconjunto de dados contendo apenas parte das
3. colunas (recomenda-se a utilização de 3 colunas) disponíveis no conjunto de dados
4. original;
5. Salve as alterações realizadas;
6. Imprima/exiba em tela os dados da nova variável (que contém o subconjunto de
7. dados).

De acordo com o procedimento, fica dessa forma:

```
projeto.py > ...
1  import pandas as pd
2
3  # Definindo o nome do arquivo CSV
4  arquivo_csv = "dados.csv"
5
6  # Lendo o arquivo CSV
7  # Usando o separador ';', a engine 'python' e especificando o encoding (se necessário)
8  tabela = pd.read_csv(arquivo_csv, sep=';', engine='python', encoding='utf-8')
9
10 subconjunto = tabela[['ID', 'Date', 'Calories']]
11
12 # Exibindo os dados lidos
13 print(subconjunto)
14 |
```

Resultado ficando da seguinte forma:

# Definindo o nome do arquivo CSV			
SAÍDA	CONSOLE DE DEPURAÇÃO	PORTAS	COMENTÁRIOS
TERMINAL			
ID	Date	Calories	
0	0	'2020/12/01'	4091
1	1	'2020/12/02'	4790
2	2	'2020/12/03'	3400
3	3	'2020/12/04'	2824
4	4	'2020/12/05'	4060
5	5	'2020/12/06'	3000
6	6	'2020/12/07'	3740
7	7	'2020/12/08'	2533
8	8	'2020/12/09'	1951
9	9	'2020/12/10'	2690
10	10	'2020/12/11'	3293
11	11	'2020/12/12'	2507
12	12	'2020/12/12'	2507
13	13	'2020/12/13'	3453
14	14	'2020/12/14'	3793
15	15	'2020/12/15'	2750
16	16	'2020/12/16'	2152
17	17	'2020/12/17'	3000
18	18	'2020/12/18'	NaN
19	19	'2020/12/19'	3230
20	20	'2020/12/20'	2430 2
21	1	'2020/12/21'	3642
22	22	NaN	2820
23	23	'2020/12/23'	3000
24	24	'2020/12/24'	2460
25	25	'2020/12/25'	3345
26	26	20201226	2500
27	27	'2020/12/27'	2410
28	28	'2020/12/28'	NaN
29	29	'2020/12/29'	2800
30	30	'2020/12/30'	3803
31	31	'2020/12/31'	2430

Microatividade 3: Descrever como configurar o número máximo de linhas a serem exibidas na visualização de um conjunto de dados usando a biblioteca Pandas (Python)

Ficando dessa forma o código completo:

```
projeto.py X dados.csv
projeto.py > ...
1 import pandas as pd
2
3 # Definindo o nome do arquivo CSV
4 arquivo_csv = "dados.csv"
5
6 # Lendo o arquivo CSV
7 tabela = pd.read_csv(arquivo_csv, sep=';', engine='python', encoding='utf-8')
8
9 pd.set_option('display.max_rows', 999)
10
11 # Exibindo os dados lidos
12 print(tabela.to_string())
```

Resultado ficando da seguinte forma:

	ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	60	'2020/12/01'	110	130	4091
1	1	60	'2020/12/02'	117	145	4790
2	2	60	'2020/12/03'	103	135	3400
3	3	45	'2020/12/04'	109	175	2824
4	4	45	'2020/12/05'	117	148	4060
5	5	60	'2020/12/06'	102	127	3000
6	6	60	'2020/12/07'	110	136	3740
7	7	450	'2020/12/08'	104	134	2533
8	8	30	'2020/12/09'	109	133	1951
9	9	60	'2020/12/10'	98	124	2690
10	10	60	'2020/12/11'	103	147	3293
11	11	60	'2020/12/12'	100	120	2507
12	12	60	'2020/12/12'	100	120	2507
13	13	60	'2020/12/13'	106	128	3453
14	14	60	'2020/12/14'	104	132	3793
15	15	60	'2020/12/15'	98	123	2750
16	16	60	'2020/12/16'	98	120	2152
17	17	60	'2020/12/17'	100	120	3000
18	18	45	'2020/12/18'	90	112	NaN
19	19	60	'2020/12/19'	103	123	3230
20	20	45	'2020/12/20'	97	125	2430 2
21	1	60	'2020/12/21'	108	131	3642
22	22	45	NaN	100	119	2820
23	23	60	'2020/12/23'	130	101	3000
24	24	45	'2020/12/24'	105	132	2460
25	25	60	'2020/12/25'	102	126	3345
26	26	60	20201226	100	120	2500
27	27	60	'2020/12/27'	92	118	2410
28	28	60	'2020/12/28'	103	132	NaN
29	29	60	'2020/12/29'	100	132	2800
30	30	60	'2020/12/30'	102	129	3803
31	31	60	'2020/12/31'	92	115	2430

Microatividade 4: Descrever como exibir as primeiras e últimas “N” linhas de um conjunto de dados usando a biblioteca Pandas (Python)

Procedimentos

1. Abra o arquivo/script utilizado nas microatividades anteriores;
2. Imprima na tela as apenas as primeiras 10 linhas do conjunto de dados original
3. (criado na microatividade 1);
4. Imprima na tela as apenas as últimas 10 linhas do conjunto de dados original
5. (criado na microatividade 1).

Ficando dessa forma o código completo:

```
projeto.py x dados.csv
projeto.py > ...
1 import pandas as pd
2
3 # Definindo o nome do arquivo CSV
4 arquivo_csv = "dados.csv"
5
6 # Lendo o arquivo CSV
7 tabela = pd.read_csv(arquivo_csv, sep=';', engine='python', encoding='utf-8')
8
9 print("Primeiras 10 linhas:")
10 print(tabela.head(10))
11
12 print("Últimas 10 linhas:")
13 print(tabela.tail(10))
14
15 # Exibindo os dados lidos
16 print(tabela.tail(10))
```

O resultado ficando da seguinte forma:

```
SAÍDA  CONSOLE DE DEPURAÇÃO  PORTAS  COMENTÁRIOS  TERMINAL
Primeiras 10 linhas:
   ID  Duration    Date  Pulse  Maxpulse  Calories
0  0         60 '2020/12/01'   110      130     4091
1  1         60 '2020/12/02'   117      145     4790
2  2         60 '2020/12/03'   103      135     3400
3  3         45 '2020/12/04'   109      175     2824
4  4         45 '2020/12/05'   117      148     4060
5  5         60 '2020/12/06'   102      127     3000
6  6         60 '2020/12/07'   110      136     3740
7  7         450 '2020/12/08'   104      134     2533
8  8          30 '2020/12/09'   109      133     1951
9  9         60 '2020/12/10'    98      124     2690
Últimas 10 linhas:
   ID  Duration    Date  Pulse  Maxpulse  Calories
0  0         60 '2020/12/01'   110      130     4091
1  1         60 '2020/12/02'   117      145     4790
2  2         60 '2020/12/03'   103      135     3400
3  3         45 '2020/12/04'   109      175     2824
4  4         45 '2020/12/05'   117      148     4060
5  5         60 '2020/12/06'   102      127     3000
6  6         60 '2020/12/07'   110      136     3740
7  7         450 '2020/12/08'   104      134     2533
8  8          30 '2020/12/09'   109      133     1951
9  9         60 '2020/12/10'    98      124     2690
```

Microatividade 5: Descrever como exibir informações gerais sobre as colunas, linhas e dados de um conjunto de dados usando a biblioteca Pandas (Python)

Procedimentos

1. Abra o arquivo/script utilizado nas microatividades anteriores;
2. Tendo como base o conjunto de dados original:
 1. Imprima as informações gerais sobre o conjunto – suas colunas, linhas e dados;
 2. Descubra a partir do comando acima:
 1. O total de linhas;
 2. O total de colunas;
 3. A quantidade de dados nulos, caso existam;
 4. O tipo de dado de cada coluna;
 5. A quantidade de memória utilizada pelo conjunto de dados.

Ficando dessa forma o código completo:

```
projeto.py X dados.csv
projeto.py > ...
1 import pandas as pd
2
3 # Definindo o nome do arquivo CSV
4 arquivo_csv = "dados.csv"
5
6 # Lendo o arquivo CSV
7 tabela = pd.read_csv(arquivo_csv, sep=';', engine='python', encoding='utf-8')
8
9 print("Informações gerais do conjunto:")
10 print(tabela.info())
11
12 totallinhas, totalColunas = tabela.shape
13 print(f"\nQuantidade de linhas: {totallinhas}")
14 print(f"Quantidade de colunas: {totalColunas}")
15
16 print("\nQuantidade de valores nulos por coluna:")
17 print(tabela.isnull().sum())
18
19 print("\nTipo de dado de cada coluna:")
20 print(tabela.dtypes)
21
22 print("\nMemória utilizada pelo conjunto de dados:")
23 print(tabela.memory_usage(deep=True))
```


Resultado das informações gerais do conjunto:

```
SAÍDA  CONSOLE DE DEPURACÃO  PORTAS  COMENTÁRIOS  TERMINAL

● Informações gerais do conjunto:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   ID           32 non-null    int64
1   Duration     32 non-null    int64
2   Date         31 non-null    object
3   Pulse        32 non-null    int64
4   Maxpulse     32 non-null    int64
5   Calories     30 non-null    object
```

Resultado de quantidade de linhas e colunas:

```
SAÍDA  CONSOLE DE DEPURACÃO  PORTAS  COMENTÁRIOS  TERMINAL

Quantidade de linhas: 32
Quantidade de colunas: 6
```

Resultado de valores nulos, caso existam:

```
SAÍDA  CONSOLE DE DEPURACÃO  PORTAS  COMENTÁRIOS  TERMINAL

Quantidade de valores nulos por coluna:
ID           0
Duration     0
Date         1
Pulse        0
Maxpulse     0
Calories     2
dtype: int64
```

Resultado de dado de cada coluna:

```
SAÍDA  CONSOLE DE DEPURACÃO  PORTAS  COMENTÁRIOS  TERMINAL

Tipo de dado de cada coluna:
ID           int64
Duration     int64
Date         object
Pulse        int64
Maxpulse     int64
Calories     object
dtype: object
```

Resultado da quantidade de memória utilizada pelo conjunto de dados:

SAÍDA	CONSOLE DE DEPURAÇÃO	PORTAS	COMENTÁRIOS	TERMINAL
Memória utilizada pelo conjunto de dados:				
Index	132			
ID	256			
Duration	256			
Date	1919			
Pulse	256			
Maxpulse	256			
Calories	1656			
dtype:	int64			