Missão Prática

Nível - 3

Tema:

Tratando a Imensidão dos Dados

Aluno: Yvo Murilo Santos D’Albuquerque

Microatividade - 1

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

- Procedimentos

1. Salve o conjunto de dados em formato CSV que utilizará num local acessível pela ferramenta de escrita de código que utilizará;

2. Crie um novo arquivo e:

a. Importe a biblioteca pandas;

b. Cria uma variável;

c. Leia o conteúdo do arquivo CSV, passando como parâmetros o separador de colunas, a engine – com o valor ‘python’ e o enconding relativo aos dados constantes no arquivo lido (esse último parâmetro pode ser opcional, dependendo do enconding existente);

d. Atribua os dados lidos do CSV à variável criada anteriormente; salve as
alterações;

e. Imprima/exiba em tela os dados da variável.

```
[ ] df = pd.read_csv('/content/drive/MyDrive/MP N3.csv', sep=';')
    display(df)
    display(df.head(4))
    df.tail(4)
```

10	10	60	'2020/12/11'	103	147	3293.0
11	11	60	'2020/12/12'	100	120	2507.0
12	12	60	'2020/12/12'	100	120	2507.0
13	13	60	'2020/12/13'	106	128	3453.0
14	14	60	'2020/12/14'	104	132	3793.0
15	15	60	'2020/12/15'	98	123	2750.0
16	16	60	'2020/12/16'	98	120	2152.0
17	17	60	'2020/12/17'	100	120	3000.0
18	18	45	'2020/12/18'	90	112	NaN
19	19	60	'2020/12/19'	103	123	3230.0
20	20	45	'2020/12/20'	97	125	2430.0
21	21	60	'2020/12/21'	108	131	3642.0
22	22	45	NaN	100	119	2820.0
23	23	60	'2020/12/23'	130	101	3000.0
24	24	45	'2020/12/24'	105	132	2460.0
25	25	60	'2020/12/25'	102	126	3345.0

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

colunas (recomenda-se a utilização de 3 colunas) disponíveis no conjunto de dados

original;

3.Salve as alterações realizadas;

4.Imprima/exiba em tela os dados da nova variável (que contém o subconjunto de

dados).

	ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	60	'2020/12/01'	110	130	4091.0
1	1	60	'2020/12/02'	117	145	4790.0
2	2	60	'2020/12/03'	103	135	3400.0
3	3	45	'2020/12/04'	109	175	2824.0

	ID	Duration	Date	Pulse	Maxpulse	Calories
28	28	60	'2020/12/28'	103	132	NaN
29	29	60	'2020/12/29'	100	132	2800.0
30	30	60	'2020/12/30'	102	129	3803.0
31	31	60	'2020/12/31'	92	115	2430.0

Microatividade – 3

- Procedimentos:

1. Abra o arquivo/script utilizado nas microatividades anteriores;
2. Usando as opções de configuração da biblioteca pandas, defina um novo valor para a propriedade “max_rows”, definindo o novo valor para 9999;
3. Salve as alterações;
4. Imprima na tela o conjunto de dados original (criado na microatividade 1) usando o método “to_string()”.

	ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	60	'2020/12/01'	110	130	4091.0
1	1	60	'2020/12/02'	117	145	4790.0
2	2	60	'2020/12/03'	103	135	3400.0
3	3	45	'2020/12/04'	109	175	2824.0

Microatividade – 4

- 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 (criado na microatividade 1);
3. Imprima na tela as apenas as últimas 10 linhas do conjunto de dados original (criado na microatividade 1).

```
[ ]  df = pd.read_csv('/content/drive/MyDrive/MP N3.csv', sep=';')
     display(df)
     display(df.head(4))
     df.tail(4)
```

10	10	60	'2020/12/11'	103	147	3293.0
11	11	60	'2020/12/12'	100	120	2507.0
12	12	60	'2020/12/12'	100	120	2507.0
13	13	60	'2020/12/13'	106	128	3453.0
14	14	60	'2020/12/14'	104	132	3793.0
15	15	60	'2020/12/15'	98	123	2750.0
16	16	60	'2020/12/16'	98	120	2152.0
17	17	60	'2020/12/17'	100	120	3000.0
18	18	45	'2020/12/18'	90	112	NaN
19	19	60	'2020/12/19'	103	123	3230.0
20	20	45	'2020/12/20'	97	125	2430.0
21	21	60	'2020/12/21'	108	131	3642.0
22	22	45	NaN	100	119	2820.0
23	23	60	'2020/12/23'	130	101	3000.0
24	24	45	'2020/12/24'	105	132	2460.0
25	25	60	'2020/12/25'	102	126	3345.0

Microatividade – 5

- Procedimentos

1. Abra o arquivo/script utilizado nas microatividades anteriores;
2. Tendo como base o conjunto de dados original:

a. Imprima as informações gerais sobre o conjunto – suas colunas, linhas e dados;

b. Descubra a partir do comando acima:

   i. O total de linhas;

   ii. O total de colunas;

   iii. A quantidade de dados nulos, caso existam;

   iv. O tipo de dado de cada coluna;

   v. A quantidade de memória utilizada pelo conjunto de dados.

Missão Prática

Tratando a imensidão dos dados

Contextualização

- Procedimentos:

1- Para essa atividade você deverá, obrigatoriamente, utilizar o conjunto de dados (fornecido anteriormente, na seção “Contextualização”) composto pelas colunas ID;Duration;Date;Pulse;Maxpulse;Calories

2 - Crie um novo arquivo/script;

3 - Leia o conteúdo do CSV fornecido, atentando-se para a necessidade ou não deincluir parâmetros adicionais como os relativos ao separador dos dados, a engine e o enconding;

4 - Atribua os dados lidos a uma variável;

5 - Verifique se os dados foram importados adequadamente:

a. Imprima as informações gerais sobre o conjunto de dados;

b. Imprima as primeiras e últimas N linhas do arquivo.

6 - Crie uma nova variável e atribua a ela uma cópia do conjunto de dados original (variável criada no passo 4);

7 - Nessa nova variável, contendo uma cópia dos dados:

a.Substitua todos os valores nulos da coluna ‘Calories’ por 0;

b.Imprima o conjunto de dados para verificar se a mudança acima foi aplicada com sucesso;

	ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	60	'2020/12/01'	110	130	4091.0
1	1	60	'2020/12/02'	117	145	4790.0
2	2	60	'2020/12/03'	103	135	3400.0
3	3	45	'2020/12/04'	109	175	2824.0

	ID	Duration	Date	Pulse	Maxpulse	Calories
28	28	60	'2020/12/28'	103	132	NaN
29	29	60	'2020/12/29'	100	132	2800.0
30	30	60	'2020/12/30'	102	129	3803.0
31	31	60	'2020/12/31'	92	115	2430.0

```
[ ]  df = pd.read_csv('/content/drive/MyDrive/MP N3.csv', sep=';')
     display(df)
     display(df.head(4))
     df.tail(4)
```

10	10	60	'2020/12/11'	103	147	3293.0
11	11	60	'2020/12/12'	100	120	2507.0
12	12	60	'2020/12/12'	100	120	2507.0
13	13	60	'2020/12/13'	106	128	3453.0
14	14	60	'2020/12/14'	104	132	3793.0
15	15	60	'2020/12/15'	98	123	2750.0
16	16	60	'2020/12/16'	98	120	2152.0
17	17	60	'2020/12/17'	100	120	3000.0
18	18	45	'2020/12/18'	90	112	NaN
19	19	60	'2020/12/19'	103	123	3230.0
20	20	45	'2020/12/20'	97	125	2430.0
21	21	60	'2020/12/21'	108	131	3642.0
22	22	45	NaN	100	119	2820.0
23	23	60	'2020/12/23'	130	101	3000.0
24	24	45	'2020/12/24'	105	132	2460.0
25	25	60	'2020/12/25'	102	126	3345.0

Tratando a Imensidao Dos Dados

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

+ Código + Texto

```
[35] try:
         dados = pd.read_csv('/content/drive/MyDrive/dados.csv', sep=",", engine="python", encoding="utf-8")
     except FileNotFoundError:
         print("File not found. Please check the file name and path.")
```

```
File not found. Please check the file name and path.
```

```
[36] dados_cod_table = dados.copy()
```

```
[37] print(dados_cod_table.columns)
```

```
Index(['ID; Duration; Date; Pulse; Maxpulse; Calories'], dtype='object')
```

```
[38] if 'Calorie_burned' in dados_cod_table.columns:
         dados_cod_table["Calorie_burned"].fillna(0, inplace=True)
         print("\nDados após a substituição dos valores nulos na coluna 'Calorie_burned': ")
         print(dados_cod_table)
     else:
         print("Calorie_burned not found in DataFrame. Please check your column names.")
```

```
Calorie_burned not found in DataFrame. Please check your column names.
```

```
dados_cod_table.fillna("1900/01/01", inplace=True)
print("\nDados após a substituição dos valores nulos na coluna 'Date':")
print(dados_cod_table)
```

0s cor

```
dados_cod_table.fillna("1900/01/01", inplace=True)
print("\nDados após a substituição dos valores nulos na coluna 'Date':")
print(dados_cod_table)
```

```
Dados após a substituição dos valores nulos na coluna 'Date':
   ID; Duration; Date; Pulse; Maxpulse; Calories
0           00; 60; '2020/12/01'; 110; 130; 4091
1           01; 60; '2020/12/02'; 117; 145; 4790
2           02; 60; '2020/12/03'; 103; 135; 3400
3           03; 45; '2020/12/04'; 109; 175; 2824
4           04; 45; '2020/12/05'; 117; 148; 4060
5           05; 60; '2020/12/06'; 102; 127; 3000
6           06; 60; '2020/12/07'; 110; 136; 3740
7           07; 45; '2020/12/08'; 104; 134; 2533
8           08; 30; '2020/12/09'; 109; 133; 1951
9           09; 60; '2020/12/10'; 098; 124; 2690
10          10; 60; '2020/12/11'; 103; 147; 3293
11          11; 60; '2020/12/12'; 100; 120; 2507
12          13; 60; '2020/12/13'; 106; 128; 3453
13          14; 60; '2020/12/14'; 104; 132; 3793
14          15; 60; '2020/12/15'; 098; 123; 2750
15          16; 60; '2020/12/16'; 098; 120; 2152
16          17; 60; '2020/12/17'; 100; 120; 3000
17          18; 45; '2020/12/18'; 090; 112; 3506
18          19; 60; '2020/12/19'; 103; 123; 3230
19          20; 45; '2020/12/20'; 097; 125; 2430
20          21; 60; '2020/12/21'; 108; 131; 3642
21          22; 45; '2020/12/22'; 100; 119; 2820
22          23; 60; '2020/12/23'; 130; 101; 3000
```

```
[19] pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.22.4 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

```
[2] import pandas as pd
```

```
[5] from google.colab import drive
    drive.mount('/content/drive')
```