

Missão Prática – Mundo 05 – Nível 03

Gilvan Pereira de Oliveira – 2023.01.53256-61197

Polo Centro – São Lourenço Da Mata – PE

RPG0033 – TRATANDO A IMENSIDÃO DOS DADOS – 9001 – 2025.1<https://github.com/GilvanPOliveira/FullStack/tree/main/Mundo05/tratandoDados>**Contextualização**

Para resolução das micro atividades será necessário ter em mãos um conjunto de dados no formato CSV. Tais dados podem ser obtidos a partir de fontes gratuitas, disponíveis na Web (como, por exemplo, o dataset disponível em <https://archive.ics.uci.edu/dataset/352/online+retail>), assim como um ambiente contendo o interpretador da linguagem python.

Ainda em relação aos conjuntos de dados, a atividade “pico web” terá como base o seguinte conjunto de dados (que deverá ser copiado e salvo num arquivo “csv”, usando como separados de colunas o“,”):

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

```

dados.csv X
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
23 21;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;'2020/12/26';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
34

```

O uso do dataframe acima é imprescindível, uma vez que ele contém dados não válidos que deverão ser tratados posteriormente. Vide as linhas 18 e 28 (coluna Calories); 22 e 26 (coluna Date).

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

Criação do Script de leitura local

```
leitura_csv.py X
...
1 import pandas as pd
2
3 dadosInformados = './dados.csv'
4
5 dados = pd.read_csv(dadosInformados, sep=';', engine='python', encoding='utf-8')
6
7 print(dados)
8
```

- Importar a biblioteca pandas, necessária para a manipulação de dados;
- Definir o arquivo CSV que conterá os dados informados e a sua localização;
- Ler o arquivo CSV utilizando os seguintes parâmetros:
 - sep: define o separador de colunas, que é o ponto e vírgula;
 - engine: define a linguagem utilizada, que é o Python;
 - encoding: define a codificação do arquivo, como 'uft-8', que é a mais comum.
- Exibe o conteúdo lido do arquivo CSV, no terminal.

```
[Running] python -u "c:\microAtividade01\leitura_csv.py"
  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
4  4         45  '2020/12/05'   117      148    4060.0
5  5         60  '2020/12/06'   102      127    3000.0
6  6         60  '2020/12/07'   110      136    3740.0
7  7         450  '2020/12/08'   104      134    2533.0
8  8         30  '2020/12/09'   109      133    1951.0
9  9         60  '2020/12/10'    98      124    2690.0
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
26 26         60  20201226    100      120    2500.0
27 27         60  '2020/12/27'    92      118    2410.0
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

[Done] exited with code=0 in 8.576 seconds
```

Criação do Script de leitura remoto

```

csv_externo.py X
...
1  from ucimlrepo import fetch_ucirepo
2
3  # fetch dataset
4  online_retail = fetch_ucirepo(id=352)
5
6  # data (as pandas dataframes)
7  X = online_retail.data.features
8  y = online_retail.data.targets
9
10 # metadata
11 print(online_retail.metadata)
12
13 # variable information
14 print(online_retail.variables)
15

```

	name	role	...	units	missing_values
0	InvoiceNo	ID	...	None	no
1	StockCode	ID	...	None	no
2	Description	Feature	...	None	no
3	Quantity	Feature	...	None	no
4	InvoiceDate	Feature	...	None	no
5	UnitPrice	Feature	...	sterling	no
6	CustomerID	Feature	...	None	no
7	Country	Feature	...	None	no

[8 rows x 7 columns]

[Done] exited with code=0 in 5.484 seconds

- Necessário efetuar a instalação do pacote: ucimlrepo;
- Criar um novo arquivo: csv_externo.py;
- O código do exemplo foi fornecido pelo link, fornecido no sway da missao;
 - <https://archive.ics.uci.edu/dataset/352/online+retail>

No arquivo criado:

- Importar o pacote instalado para localizar os dados desejados;
- Selecionar o id dos dados a serem exibidos;
- Separar os dados de forma ampla e de forma específica;
- Exibir ambas as formas no terminal.

Em cima como foi exibido no VsCode, abaixo como consta no site:

Variable Name	Role	Type	Description	Units	Missing Values
InvoiceNo	ID	Categorical	a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation		no
StockCode	ID	Categorical	a 5-digit integral number uniquely assigned to each distinct product		no
Description	Feature	Categorical	product name		no
Quantity	Feature	Integer	the quantities of each product (item) per transaction		no
InvoiceDate	Feature	Date	the day and time when each transaction was generated		no
UnitPrice	Feature	Continuous	product price per unit	sterling	no
CustomerID	Feature	Categorical	a 5-digit integral number uniquely assigned to each customer		no
Country	Feature	Categorical	the name of the country where each customer resides		no

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

```
leitura_csv.py X
```

```

...
1  import pandas as pd
2
3  dadosInformados = './dados.csv'
4
5  dados = pd.read_csv(dadosInformados, sep=';', engine='python', encoding='utf-8')
6
7  subConjunto = dados[['ID', 'Date', 'Calories']]
8
9  print(subConjunto)
10

```

```
[Running] python -u "...\microAtividade02\leitura_csv.py"
```

	ID	Date	Calories
0	0	'2020/12/01'	4091.0
1	1	'2020/12/02'	4790.0
2	2	'2020/12/03'	3400.0
3	3	'2020/12/04'	2824.0
4	4	'2020/12/05'	4060.0
5	5	'2020/12/06'	3000.0
6	6	'2020/12/07'	3740.0
7	7	'2020/12/08'	2533.0
8	8	'2020/12/09'	1951.0
9	9	'2020/12/10'	2690.0
10	10	'2020/12/11'	3293.0
11	11	'2020/12/12'	2507.0
12	12	'2020/12/12'	2507.0
13	13	'2020/12/13'	3453.0
14	14	'2020/12/14'	3793.0
15	15	'2020/12/15'	2750.0
16	16	'2020/12/16'	2152.0
17	17	'2020/12/17'	3000.0
18	18	'2020/12/18'	NaN
19	19	'2020/12/19'	3230.0
20	20	'2020/12/20'	2430.0
21	21	'2020/12/21'	3642.0
22	22	NaN	2820.0
23	23	'2020/12/23'	3000.0
24	24	'2020/12/24'	2460.0
25	25	'2020/12/25'	3345.0
26	26	20201226	2500.0
27	27	'2020/12/27'	2410.0
28	28	'2020/12/28'	NaN
29	29	'2020/12/29'	2800.0
30	30	'2020/12/30'	3803.0
31	31	'2020/12/31'	2430.0

```
[Done] exited with code=0 in 1.287 seconds
```

- Utilizando-se o mesmo arquivo da micro atividade anterior;
- Foi criado um subconjunto contendo apenas as colunas 'ID', 'Date' e 'Calories';
- Foi exibido esse subconjunto.

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

```
leitura_csv.py x
...
1  import pandas as pd
2
3  dadosInformados = './dados.csv'
4
5  dados = pd.read_csv(dadosInformados, sep=';', engine='python', encoding='utf-8')
6
7  pd.set_option('display.max_rows', 9999)
8
9  print(dados.to_string())
10
```

```
[Running] python -u "C:\microAtividade03\leitura_csv.py"
|  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
4  4      45  '2020/12/05'  117      148    4060.0
5  5      60  '2020/12/06'  102      127    3000.0
6  6      60  '2020/12/07'  110      136    3740.0
7  7     450  '2020/12/08'  104      134    2533.0
8  8      30  '2020/12/09'  109      133    1951.0
9  9      60  '2020/12/10'   98      124    2690.0
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
26 26     60  20201226    100      120    2500.0
27 27     60  '2020/12/27'   92      118    2410.0
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

[Done] exited with code=0 in 0.592 seconds
```

- Utilizando-se o mesmo arquivo da micro atividade 01;
- Foi solicitado o uso de uma propriedade de exibição do Pandas:
 - `display.max_rows` definindo seu valor para 9999;
- Porém, para exibir os valores de 'dados', foi necessário utilizar o método `to_string()` para converter o DataFrame em uma string completa.

OBS: Foram exibidos apenas 31 linhas devido a quantidade de linhas que o arquivo 'dados' possui, porém o código foi ajustado para exibir até 9999 linhas.

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

```

leitura_csv.py X
...
1  import pandas as pd
2
3  dadosInformados = './dados.csv'
4
5  dados = pd.read_csv(dadosInformados, sep=';', engine='python', encoding='utf-8')
6
7  print("Primeiras 10 linhas do DataFrame:")
8  print(dados.head(10))
9
10 print("\nÚltimas 10 linhas do DataFrame:")
11 print(dados.tail(10))
12

[Running] python -u "C:\microAtividade04\leitura_csv.py"
Primeiras 10 linhas do DataFrame:
|  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
4  4      45  '2020/12/05'   117      148     4060.0
5  5      60  '2020/12/06'   102      127     3000.0
6  6      60  '2020/12/07'   110      136     3740.0
7  7     450  '2020/12/08'   104      134     2533.0
8  8      30  '2020/12/09'   109      133     1951.0
9  9      60  '2020/12/10'    98      124     2690.0

Últimas 10 linhas do DataFrame:
|  ID  Duration      Date  Pulse  Maxpulse  Calories
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
26 26      60  20201226    100      120     2500.0
27 27      60  '2020/12/27'    92      118     2410.0
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

[Done] exited with code=0 in 0.684 seconds

```

- Utilizando-se o mesmo arquivo da micro atividade 01;
- Foi solicitado exibir as 10 primeiras e as 10 últimas linhas do arquivo DataFrame: 'dados';

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

Utilizando-se o mesmo arquivo da micro atividade 01, foi solicitado:

```
leitura_csv.py X
...
1  import pandas as pd
2
3  dadosInformados = './dados.csv'
4
5  dados = pd.read_csv(dadosInformados, sep=';', engine='python', encoding='utf-8')
6
7  print("Informacoes gerais sobre o conjunto de dados:")
8  dados.info()
9
10 total_linhas, total_colunas = dados.shape
11 print(f"\nTotal de linhas: {total_linhas}")
12 print(f"Total de colunas: {total_colunas}")
13
14 print("\nQuantidade de dados nulos por coluna:")
15 print(dados.isnull().sum())
16
17 print("\nTipo de dado de cada coluna:")
18 print(dados.dtypes)
19
20 print("\nMemoria utilizada pelo conjunto de dados:")
21 print(dados.memory_usage(deep=True))
22
```

Exibir informações gerais sobre o conjunto de dados: suas colunas, linhas e dados

```
[Running] python -u "c:\microAtividade05\leitura_csv.py"
Informacoes gerais sobre o conjunto de dados:
<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    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 1.6+ KB
```

Total de Linhas e Colunas

```
Total de linhas: 32
Total de colunas: 6
```

Verificar a quantidade de dados nulos, caso existam

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

O tipo de dado de cada coluna

```
Tipo de dado de cada coluna:  
ID          int64  
Duration    int64  
Date        object  
Pulse       int64  
Maxpulse    int64  
Calories    float64  
dtype: object
```

A quantidade de memória utilizada pelo conjunto de dados

```
Memoria utilizada pelo conjunto de dados:  
Index      132  
ID         256  
Duration   256  
Date       1919  
Pulse      256  
Maxpulse   256  
Calories   256  
dtype: int64  
  
[Done] exited with code=0 in 0.701 seconds
```


Missão Prática | Tratando a imensidão dos dados

Contextualização

Como Analista de Dados, você recebeu, em um novo projeto, um conjunto de dados. Sua principal tarefa é tratar os dados desse conjunto a fim de que possam ser utilizados para a descoberta de conhecimento através de sua posterior análise e interpretação.

```
tratando_dados.py X
...
1  import pandas as pd
2  import numpy as np
3
4  # 04
5  dados = pd.read_csv('dados.csv', sep=';', engine='python', encoding='utf-8')
6
7  # 05
8  print("Informacoes Gerais do Dataset:")
9  dados.info()
10 print("\nPrimeiras Linhas:")
11 print(dados.head())
12 print("\nUltimas Linhas:")
13 print(dados.tail())
14
15 # 06
16 dados_copia = dados.copy()
17
18 # 07
19 dados_copia['Calories'] = dados_copia['Calories'].fillna(0)
20 print("\nApos substituir nulos em 'Calories' por 0:")
21 print(dados_copia)
22
23 # 08
24 dados_copia['Date'] = dados_copia['Date'].fillna('1900/01/01')
25 print("\nApos substituir nulos em 'Date' por '1900/01/01':")
26 print(dados_copia)
27 dados_copia['Date'] = (
28     dados_copia['Date']
29     .astype(str)
30     .str.strip("")
31     .str.strip("'")
32 )
33
```

```

34
35 dados_copia['Date'] = dados_copia['Date'].replace('20201226', '2020/12/26')
36 try:
37     dados_copia['Date'] = pd.to_datetime(
38         dados_copia['Date'],
39         format='%Y/%m/%d'
40     )
41 except Exception as e:
42     print("\nErro na conversao de 'Date':", e)
43
44 # 09
45 dados_copia['Date'] = dados_copia['Date'].replace('1900/01/01', np.nan)
46 dados_copia['Date'] = pd.to_datetime(
47     dados_copia['Date'],
48     format='%Y/%m/%d',
49     errors='coerce'
50 )
51 print("\nApos substituir '1900/01/01' por NaN e converter 'Date':")
52 print(dados_copia)
53
54 # 10
55 dados_copia = dados_copia.dropna(subset=['Date'])
56
57 # 11
58 dados_copia['Date'] = pd.to_datetime(
59     dados_copia['Date'], format='%Y/%m/%d', errors='coerce')
60 print("\nApos correcao do valor '20201226' e conversao final de 'Date':")
61 print(dados_copia)
62
63 # 12
64 dados_copia = dados_copia.dropna(subset=['Date'])
65 print("\nDataset final apos remocao de registros com 'Date' nulo:")
66 print(dados_copia)
67

```

4. Atribua os dados lidos a uma variável;
5. Verifique se os dados foram importados adequadamente:
 1. Imprima as informações gerais sobre o conjunto de dados;
 2. 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);

```

[Running] python -u " \missaoPratica\tratando_dados.py"
Informacoes Gerais do Dataset:
<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    float64
dtypes: float64(1), int64(4), object(1)
memory usage: 1.6+ KB

```

Primeiras Linhas:							Últimas Linhas:						
	ID	Duration	Date	Pulse	Maxpulse	Calories		ID	Duration	Date	Pulse	Maxpulse	Calories
0	0	60	'2020/12/01'	110	130	4091.0	27	27	60	'2020/12/27'	92	118	2410.0
1	1	60	'2020/12/02'	117	145	4790.0	28	28	60	'2020/12/28'	103	132	NaN
2	2	60	'2020/12/03'	103	135	3400.0	29	29	60	'2020/12/29'	100	132	2800.0
3	3	45	'2020/12/04'	109	175	2824.0	30	30	60	'2020/12/30'	102	129	3803.0
4	4	45	'2020/12/05'	117	148	4060.0	31	31	60	'2020/12/31'	92	115	2430.0

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

1. Substitua todos os valores nulos da coluna 'Calories' por 0;
2. Imprima o conjunto de dados para verificar se a mudança acima foi aplicada com sucesso;

Apos substituir nulos em 'Calories' por 0:						
	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
4	4	45	'2020/12/05'	117	148	4060.0
5	5	60	'2020/12/06'	102	127	3000.0
6	6	60	'2020/12/07'	110	136	3740.0
7	7	450	'2020/12/08'	104	134	2533.0
8	8	30	'2020/12/09'	109	133	1951.0
9	9	60	'2020/12/10'	98	124	2690.0
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	0.0
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
26	26	60	20201226	100	120	2500.0
27	27	60	'2020/12/27'	92	118	2410.0
28	28	60	'2020/12/28'	103	132	0.0
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

8. Ainda na nova variável:

1. Substitua os valores nulos da coluna 'Date' por '1900/01/01';
2. Imprima o conjunto de dados e confira se a mudança foi aplicada com sucesso;
3. Transforme os dados da coluna 'Date' em datetime usando o método 'to_datetime';

Apos substituir nulos em 'Date' por '1900/01/01':

	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
4	4	45	'2020/12/05'	117	148	4060.0
5	5	60	'2020/12/06'	102	127	3000.0
6	6	60	'2020/12/07'	110	136	3740.0
7	7	450	'2020/12/08'	104	134	2533.0
8	8	30	'2020/12/09'	109	133	1951.0
9	9	60	'2020/12/10'	98	124	2690.0
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	0.0
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	1900/01/01	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
26	26	60	20201226	100	120	2500.0
27	27	60	'2020/12/27'	92	118	2410.0
28	28	60	'2020/12/28'	103	132	0.0
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

9. Tendo seguido todas as instruções anteriores, ao executar o passo anterior você deverá ter encontrado um erro informando que o valor '1900/01/01' não corresponde ao formato '%Y/%m/%d'. Para resolver esse problema:
1. Substitua, na coluna 'Date', o valor '1900/01/01' por 'NaN';
 2. Utilizando o método 'to_datetime', repita o passo de transformação dos dados da coluna 'Date' para datetime;
 3. Imprima o conjunto de dados para verificar se as mudanças acima foram aplicadas com sucesso;

Apos substituir '1900/01/01' por NaN e converter 'Date':

	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
4	4	45	2020-12-05	117	148	4060.0
5	5	60	2020-12-06	102	127	3000.0
6	6	60	2020-12-07	110	136	3740.0
7	7	450	2020-12-08	104	134	2533.0
8	8	30	2020-12-09	109	133	1951.0
9	9	60	2020-12-10	98	124	2690.0
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	0.0
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	NaT	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
26	26	60	2020-12-26	100	120	2500.0
27	27	60	2020-12-27	92	118	2410.0
28	28	60	2020-12-28	103	132	0.0
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

OBS: o valor está como NaT, pois no Pandas, é atribuído para date: NaT (not a time), e não, NaN (not a number).

10. Nesse ponto, você deverá ter esbarrado em outro erro, informando agora que o valor "20201226" não corresponde ao formato "%Y/%m/%d". Você precisará, agora, na coluna 'Date', transformar especificamente esse valor, atualmente uma string, para o formato datetime. Para isso você deverá combinar os métodos 'replace' e 'to_datetime';
11. Após o passo anterior, execute novamente a transformação de todos os dados da coluna 'Date' para o formato datetime (usando o to_datetime). Imprima o conjunto de dados atual para verificar se todas as transformações foram executadas com sucesso;

Apos correcao do valor '20201226' e conversao final de 'Date':

	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
4	4	45	2020-12-05	117	148	4060.0
5	5	60	2020-12-06	102	127	3000.0
6	6	60	2020-12-07	110	136	3740.0
7	7	450	2020-12-08	104	134	2533.0
8	8	30	2020-12-09	109	133	1951.0
9	9	60	2020-12-10	98	124	2690.0
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	0.0
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
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
26	26	60	2020-12-26	100	120	2500.0
27	27	60	2020-12-27	92	118	2410.0
28	28	60	2020-12-28	103	132	0.0
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

12. Por fim, remova os registros contendo valores nulos. Nesse ponto, apenas a coluna 'Date' possui um registro que atende a essa premissa (linha 22). Logo, utilize-a como base para realizar a transformação solicitada;
13. Imprima o dataframe e verifique se todas as transformações foram executadas conforme solicitado nos passos anteriores.

Dataset final apos remocao de registros com 'Date' nulo:

	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
4	4	45	2020-12-05	117	148	4060.0
5	5	60	2020-12-06	102	127	3000.0
6	6	60	2020-12-07	110	136	3740.0
7	7	450	2020-12-08	104	134	2533.0
8	8	30	2020-12-09	109	133	1951.0
9	9	60	2020-12-10	98	124	2690.0
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	0.0
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
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
26	26	60	2020-12-26	100	120	2500.0
27	27	60	2020-12-27	92	118	2410.0
28	28	60	2020-12-28	103	132	0.0
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

[Done] exited with code=0 in 0.733 seconds