DOCUMENTAÇÃO TRABALHO TRATANDO A IMENSIDÃO DOS DADOS

Desenvolvedor: André Luis Gonçalves Carvalhal
Matrícula: 202203185403

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

2. Crie um novo arquivo/script;

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

4. Atribua os dados lidos a uma variável;

```
import pandas as pd
data = pd.read_csv('data.csv',sep=';',engine='python',encoding='UTF8')
```

5. Verifique se os dados foram importados adequadamente:
   a. Imprima as informações gerais sobre o conjunto de dados;

```
data.info()
```

```
<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
dtypes: int64(4), object(2)
memory usage: 1.6+ KB
```

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

```
In [183...
data.head(3)
```

Out[183...

	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

```
In [184...
data.tail(3)
```

Out[184...

	ID	Duration	Date	Pulse	Maxpulse	Calories
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

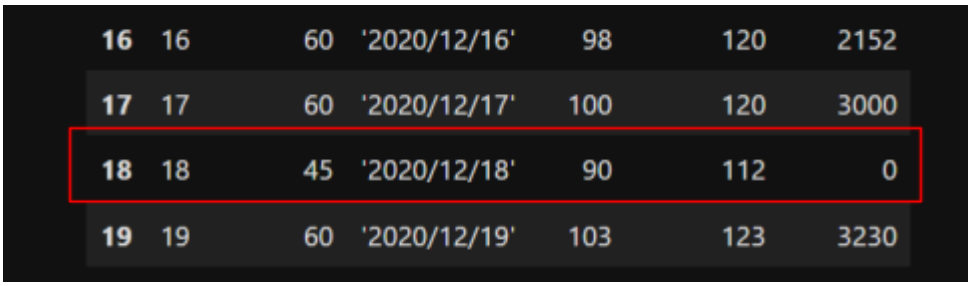
6. Crie uma nova variável e atribua a ela uma cópia do conjunto de dados original (variável criada no passo 4);

```
data2 = pd.DataFrame(data)
```

7. Nessa nova variável, contendo uma cópia dos dados:
    a. Substitua todos os valores nulos da coluna ‘Calories’ por 0;

```
data2['Calories'].fillna(0, inplace = True)
data2
```

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

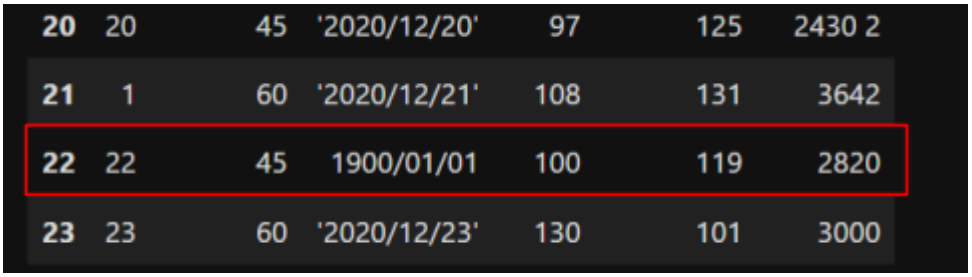


16	16	60	'2020/12/16'	98	120	2152
17	17	60	'2020/12/17'	100	120	3000
18	18	45	'2020/12/18'	90	112	0
19	19	60	'2020/12/19'	103	123	3230

8. Ainda na nova variável:
    a. Substitua os valores nulos da coluna ‘Date’ por 1900/01/01’;

```
data2['Date'].fillna('1900/01/01', inplace = True)
data2
```

    b. Imprima o conjunto de dados e confira se a mudança foi aplicada com sucesso;



20	20	45	'2020/12/20'	97	125	2430 2
21	1	60	'2020/12/21'	108	131	3642
22	22	45	1900/01/01	100	119	2820
23	23	60	'2020/12/23'	130	101	3000

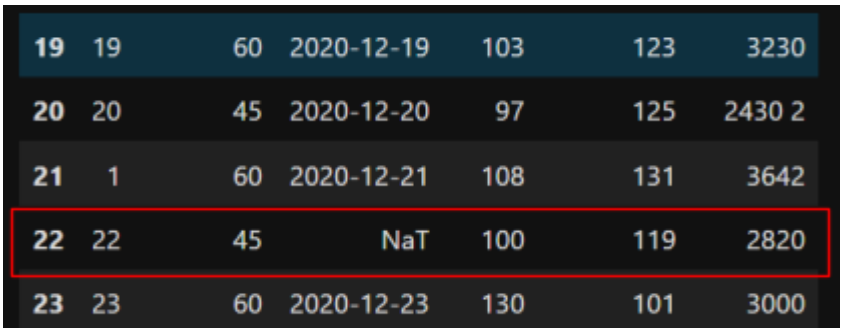
    c. Transforme os dados da coluna ‘Date’ em datetime usando o método ‘to_datetime’;

No arquivo do Jupyter Notebook existe uma observação explicando que os erros citados a seguir, não ocorrem em versões mais atualizadas do python e do kernel do Jupyter Notebook e da biblioteca pandas, então os passo foram "executados" como se os erros citados tivessem ocorrido e suas correções efetuadas.

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:
    a. Substitua, na coluna 'Date', o valor '1900/01/01' por 'NaN';

```
data2.replace('1900/01/01','NaN', inplace = True)
data2
```

    b. Utilizando o método 'to_datetime', repita o passo de transformação dos dados da coluna 'Date' para datetime;
    c. Imprima o conjunto de dados para verificar se as mudanças acima foram aplicadas com sucesso;

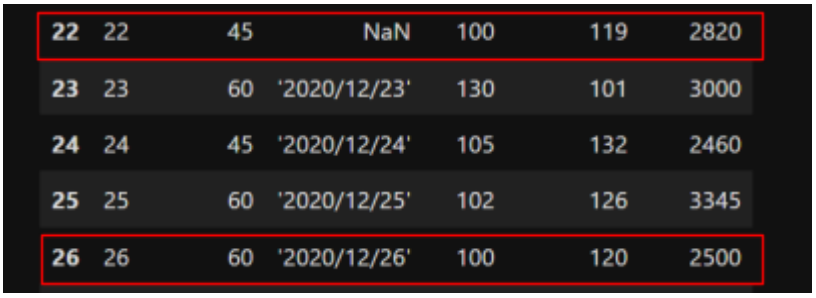


19	19	60	2020-12-19	103	123	3230
20	20	45	2020-12-20	97	125	2430 2
21	1	60	2020-12-21	108	131	3642
22	22	45	NaT	100	119	2820
23	23	60	2020-12-23	130	101	3000

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';

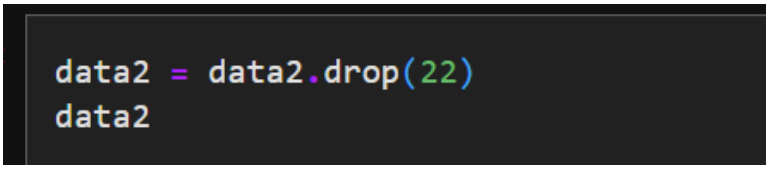
```
data2.replace({'Date' : {
    '20201226' : "'2020/12/26'"
}}, inplace = True)
data2
```

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;
12.



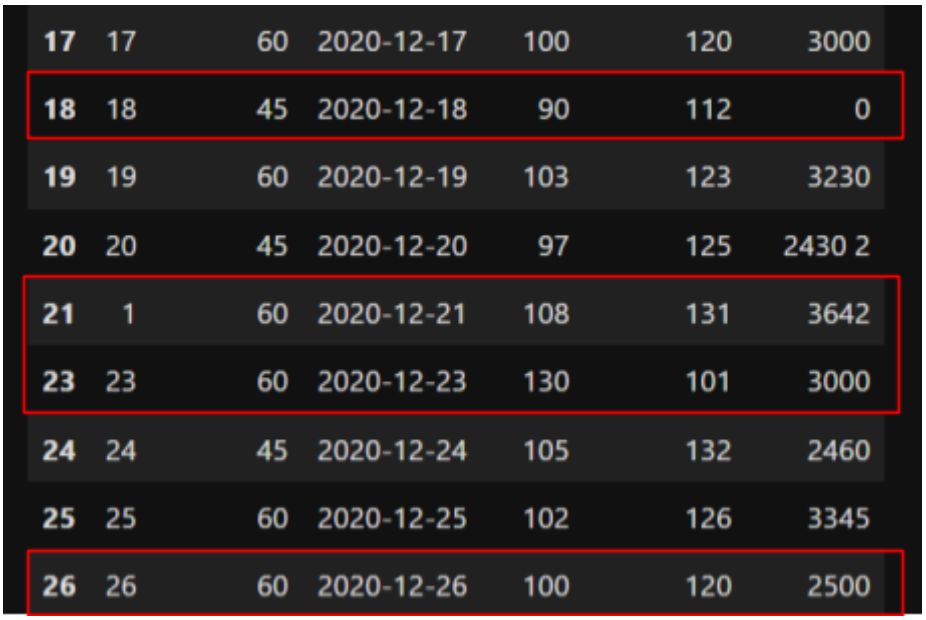
22	22	45	NaN	100	119	2820
23	23	60	'2020/12/23'	130	101	3000
24	24	45	'2020/12/24'	105	132	2460
25	25	60	'2020/12/25'	102	126	3345
26	26	60	'2020/12/26'	100	120	2500

13. 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;
14.



```
data2 = data2.drop(22)
data2
```

15. Imprima o dataframe e verifique se todas as transformações foram executadas conforme solicitado nos passos anteriores



17	17	60	2020-12-17	100	120	3000
18	18	45	2020-12-18	90	112	0
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
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	2020-12-26	100	120	2500