



Estácio

Missão Prática | Nível 3 | Mundo 5

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 encoding relativo aos dados constantes no arquivo lido (esse último parâmetro pode ser opcional, dependendo do encoding existente);
 - d. Atribua os dados lidos do CSV à variável criada anteriormente; Salve as alterações;
 - e. Imprima/exiba em tela os dados da variável.

0s



```
import pandas as pd

dados_exercicio = None

dados_exercicio = pd.read_csv(
    '/content/trabalho_03.csv',
    sep=';',
    engine='python',
    encoding='utf-8'
)

print(dados_exercicio)
```



| | 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 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).



```
subconjunto_dados = dados_exercicio[['Date', 'Pulse', 'Calories']]  
print(subconjunto_dados)
```



| | Date | Pulse | Calories |
|----|--------------|-------|----------|
| 0 | '2020/12/01' | 110 | 4091 |
| 1 | '2020/12/02' | 117 | 4790 |
| 2 | '2020/12/03' | 103 | 3400 |
| 3 | '2020/12/04' | 109 | 2824 |
| 4 | '2020/12/05' | 117 | 4060 |
| 5 | '2020/12/06' | 102 | 3000 |
| 6 | '2020/12/07' | 110 | 3740 |
| 7 | '2020/12/08' | 104 | 2533 |
| 8 | '2020/12/09' | 109 | 1951 |
| 9 | '2020/12/10' | 98 | 2690 |
| 10 | '2020/12/11' | 103 | 3293 |
| 11 | '2020/12/12' | 100 | 2507 |
| 12 | '2020/12/12' | 100 | 2507 |
| 13 | '2020/12/13' | 106 | 3453 |
| 14 | '2020/12/14' | 104 | 3793 |
| 15 | '2020/12/15' | 98 | 2750 |
| 16 | '2020/12/16' | 98 | 2152 |
| 17 | '2020/12/17' | 100 | 3000 |
| 18 | '2020/12/18' | 90 | NaN |
| 19 | '2020/12/19' | 103 | 3230 |
| 20 | '2020/12/20' | 97 | 2430 2 |
| 21 | '2020/12/21' | 108 | 3642 |
| 22 | NaN | 100 | 2820 |
| 23 | '2020/12/23' | 130 | 3000 |
| 24 | '2020/12/24' | 105 | 2460 |
| 25 | '2020/12/25' | 102 | 3345 |
| 26 | 20201226 | 100 | 2500 |
| 27 | '2020/12/27' | 92 | 2410 |
| 28 | '2020/12/28' | 103 | NaN |
| 29 | '2020/12/29' | 100 | 2800 |
| 30 | '2020/12/30' | 102 | 3803 |
| 31 | '2020/12/31' | 92 | 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)

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()”.

```
pd.options.display.max_rows = 9999

dados_exercicio = pd.read_csv('trabalho_03.csv', sep=';', engine='python', encoding='utf-8')

print(dados_exercicio.to_string())
```

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

```
pd.options.display.max_rows = 20

dados_exercicio = pd.read_csv('trabalho_03.csv', sep=';', engine='python', encoding='utf-8')

print("Primeiras 10 linhas:")
print(dados_exercicio.head(10))

print("\nÚltimas 10 linhas:")
print(dados_exercicio.tail(10))
```

↩ 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 |
|----|----|----------|--------------|-------|----------|----------|
| 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 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:
 - 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.

```
dados_exercicio = pd.read_csv('trabalho_03.csv', sep=';', engine='python', encoding='utf-8')

print("Informações gerais sobre o conjunto de dados:")
dados_exercicio.info()
```

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

Missão Prática | Tratando a imensidão dos dados

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 de incluir parâmetros adicionais como os relativos ao separador dos dados, a engine e o encoding;
4. Atribua os dados lidos a uma variável;
5. Verifique se os dados foram importados adequadamente:
 - i. Imprima as informações gerais sobre o conjunto de dados;
 - ii. 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:
 - i. Substitua todos os valores nulos da coluna ‘Calories’ por 0;
 - ii. Imprima o conjunto de dados para verificar se a mudança acima foi aplicada com sucesso;
8. Ainda na nova variável:
 - i. Substitua os valores nulos da coluna ‘Date’ por ‘1900/01/01’;
 - ii. Imprima o conjunto de dados e confira se a mudança foi aplicada com sucesso;
 - iii. Transforme os dados da coluna ‘Date’ em datetime usando o método ‘to_datetime’;

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:
 - i. Substitua, na coluna 'Date', o valor '1900/01/01' por 'NaN';
 - ii. Utilizando o método 'to_datetime', repita o passo de transformação dos dados da coluna 'Date' para datetime;
 - iii. Imprima o conjunto de dados para verificar se as mudanças acima foram aplicadas com sucesso;
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;
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.

```
[ ] import pandas as pd
import numpy as np

dados_exercicio = pd.read_csv('trabalho_03.csv', sep=';', engine='python', encoding='utf-8')

print("Informações gerais sobre o conjunto de dados:")
dados_exercicio.info()

print("\nPrimeiras 5 linhas:")
print(dados_exercicio.head(5))

print("\nÚltimas 5 linhas:")
print(dados_exercicio.tail(5))

dados_exercicio_copia = dados_exercicio.copy()

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

print("\nDados após substituir valores nulos em 'Calories' por 0:")
print(dados_exercicio_copia)

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

print("\nDados após substituir valores nulos em 'Date' por '1900/01/01':")
print(dados_exercicio_copia)

dados_exercicio_copia['Date'] = dados_exercicio_copia['Date'].str.replace("", "")

try:
    dados_exercicio_copia['Date'] = pd.to_datetime(dados_exercicio_copia['Date'], format='%Y/%m/%d', errors='coerce')
except Exception as e:
    print("\nErro ao converter a coluna 'Date' para datetime:", e)

dados_exercicio_copia['Date'].replace('1900/01/01', np.nan, inplace=True)

dados_exercicio_copia['Date'] = pd.to_datetime(dados_exercicio_copia['Date'], errors='coerce')

print("\nDados após substituir '1900/01/01' por NaN e converter para datetime:")
print(dados_exercicio_copia)

dados_exercicio_copia['Date'] = dados_exercicio_copia['Date'].replace('20201226', '2020/12/26')
dados_exercicio_copia['Date'] = pd.to_datetime(dados_exercicio_copia['Date'], errors='coerce')

print("\nDados após corrigir o valor '20201226' e converter para datetime:")
print(dados_exercicio_copia)

dados_exercicio_copia.dropna(subset=['Date'], inplace=True)

print("\nDataFrame final após todas as transformações:")
print(dados_exercicio_copia)
```

```

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

Primeiras 5 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

Últimas 5 linhas:
   ID  Duration      Date  Pulse  Maxpulse  Calories
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

Dados após substituir valores nulos em 'Calories' por 0:
   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
..  ..         ...   ...   ...   ...   ...
27  27         60 '2020/12/27'    92      118    2410
28  28         60 '2020/12/28'   103      132      0
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

[32 rows x 6 columns]

```

f

Dados após substituir valores nulos em 'Date' por '1900/01/01':

| | 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 |
| .. | .. | ... | ... | ... | ... | ... |
| 27 | 27 | 60 | '2020/12/27' | 92 | 118 | 2410 |
| 28 | 28 | 60 | '2020/12/28' | 103 | 132 | 0 |
| 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 |

[32 rows x 6 columns]

Dados após substituir '1900/01/01' por NaN e converter para datetime:

| | 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 |
| .. | .. | ... | ... | ... | ... | ... |
| 27 | 27 | 60 | 2020-12-27 | 92 | 118 | 2410 |
| 28 | 28 | 60 | 2020-12-28 | 103 | 132 | 0 |
| 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 |

[32 rows x 6 columns]

Dados após corrigir o valor '20201226' e converter para datetime:

| | 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 |
| .. | .. | ... | ... | ... | ... | ... |
| 27 | 27 | 60 | 2020-12-27 | 92 | 118 | 2410 |
| 28 | 28 | 60 | 2020-12-28 | 103 | 132 | 0 |
| 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 |

[32 rows x 6 columns]

DataFrame final após todas as transformações:

| | 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 |
| .. | .. | ... | ... | ... | ... | ... |
| 27 | 27 | 60 | 2020-12-27 | 92 | 118 | 2410 |
| 28 | 28 | 60 | 2020-12-28 | 103 | 132 | 0 |
| 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 |

[30 rows x 6 columns]