



## Missão Prática – Mundo 5 – Nível 3

Andrey Haertel Aires - Matricula: 2021.07.22851-2

Polo Centro - Palhoça –SC

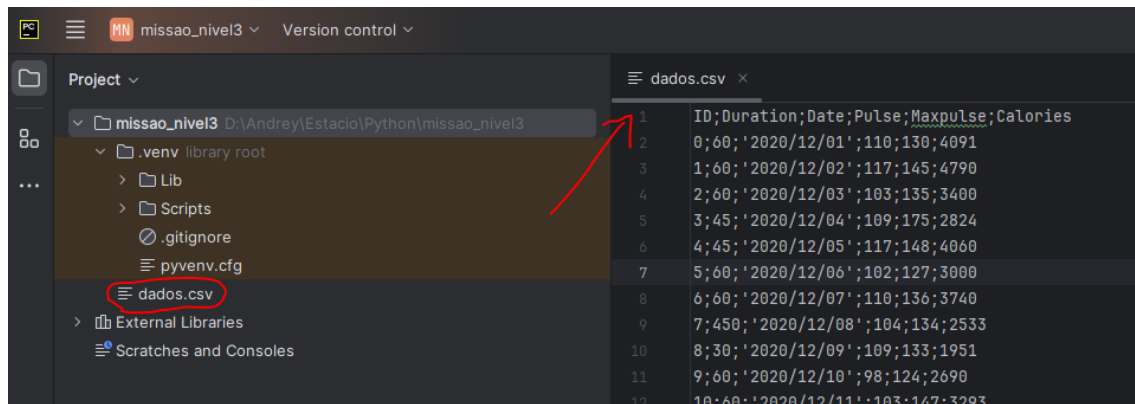
**RPG0033 - TRATANDO A IMENSIDÃO DOS DADOS**

**T 9001 – 5º Semestre Letivo**

Github: [https://github.com/AndreyHaires/MissaoPraticaMundo5\\_N3](https://github.com/AndreyHaires/MissaoPraticaMundo5_N3)

Utilizando o Pycharm, foi criado um projeto de nome “missão\_nivel3”.

Com o conjunto de dados da Contextualização foi criado um arquivo de nome “dados.csv”



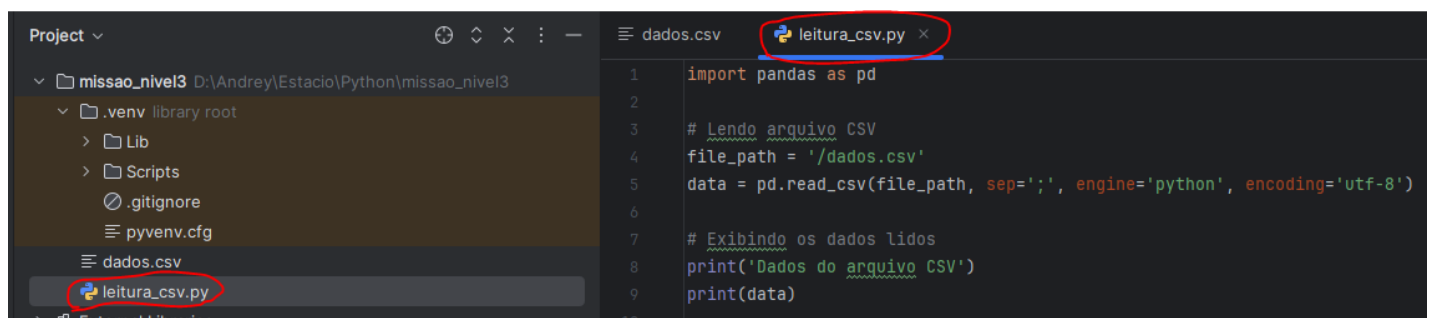
Foi criado um novo arquivo Python com o nome “leitura\_csv.py”

Iniciamos importando a biblioteca pandas e nomeando como pd.

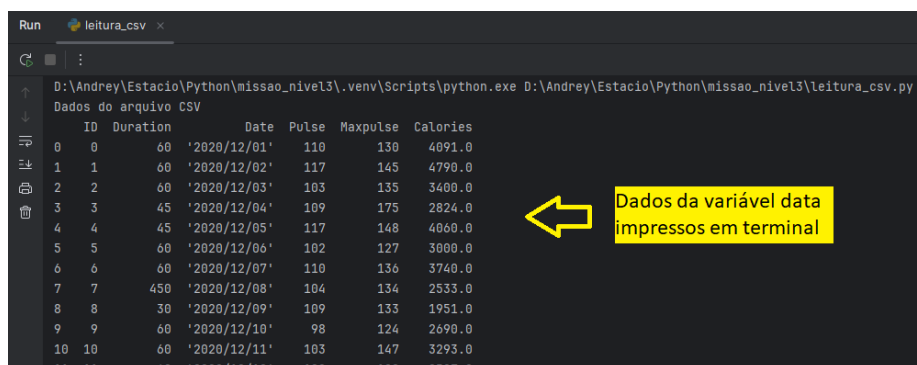
Em seguida em “file\_path” descrevemos o local que se encontra o arquivo csv que será lido.

Os commands a seguir definem o separador dos valores nesse caso o ponto e virgula “;”

Como engine para analisar o arquivo foi definido da biblioteca pandas o “python”, como encoding foi utilizado o ‘utf-8’ e a variável que irá armazenar esses dados será “data”.



Depois foi utilizado a função print para imprimir os dados da variável.



Verificando se os dados foram importados corretamente, foi utilizado o método “.info()”, que fornece uma visão geral dos dados com numero de entradas, nome de colunas, contagem dos valores nulos e tipos de dados das colunas.

```
11 print(data.info())
```

Resultado:

```
<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
None
```

Para imprimir as primeiras e ultimas N linhas do arquivo, utilizamos os métodos:

Para imprimir as “5” primeiras linhas: `print(data.head(5))`

Para imprimir as últimas “5” linhas: `print(data.tail(5))`

```
14 # Imprimindo as primeiras 5 linhas
15 print("Primeiras 5 linhas:")
16 print(data.head(5))
17
18 # Imprimindo as últimas 5 linhas
19 print("\nÚltimas 5 linhas:")
20 print(data.tail(5))
```

Resultado:

```
Primeiras 5 linhas:
   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

Últimas 5 linhas:
   ID  Duration      Date  Pulse  Maxpulse  Calories
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
```

Para criar uma nova variável e atribuir a ela a cópia do conjunto de dados original, sem que as alterações feitas na cópia não afetem o conjunto de dados original, foi utilizado o método: “.copy()”.

A Variável recebeu o nome de “data\_copia”:

Para substituir todos os valores nulos da coluna ‘Calories’ por 0 foi utilizado o método “.fillna(0)”

Foi utilizado o “.print(data\_copia)” para verificar se as mudanças foram realizadas.

```
27 # Alterando os dados de data_copia
28
29 #Substituindo todos os valores nulos da coluna 'Calories' por 0;
30 data_copia['Calories'] = data_copia['Calories'].fillna(0)
31
32 #verificando se a mudança acima foi aplicada com sucesso;
33 print(data_copia)
34
```

Resultado:

Imprimindo o conteúdo da variável data\_copy

	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

Process finished with exit code 0

Para substituir valores nulos na coluna 'Date' por '1900/01/01' foi usado novamente o método “.fillna()” e após isso utilizado o print para verificar se a substituição foi realizada corretamente.

```
36 # Substituindo valores nulos na coluna 'Date' por '1900/01/01'
37 data_copia['Date'] = data_copia['Date'].fillna('1900/01/01')
38
39 # Verificando se a substituição foi realizada
40 print("\n Coluna data com dados alterados:")
41 print(data_copia['Date'])
```

Resultado:

20	'2020/12/20'
21	'2020/12/21'
22	1900/01/01
23	'2020/12/23'
24	'2020/12/24'
25	'1900/01/01'

Para substituir o valor da linha 22 da coluna Date o valor “1900/01/01” por “NaN” foi utilizado o método “.replace()”.

```
## Substituindo '1900/01/01' por NaN na coluna 'Date'
data_copia['Date'] = data_copia['Date'].replace('1900/01/01', np.nan)
print("\n Substituindo de '1900/01/01' por NaN:")
print(data_copia['Date'])
```

Resultado:

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

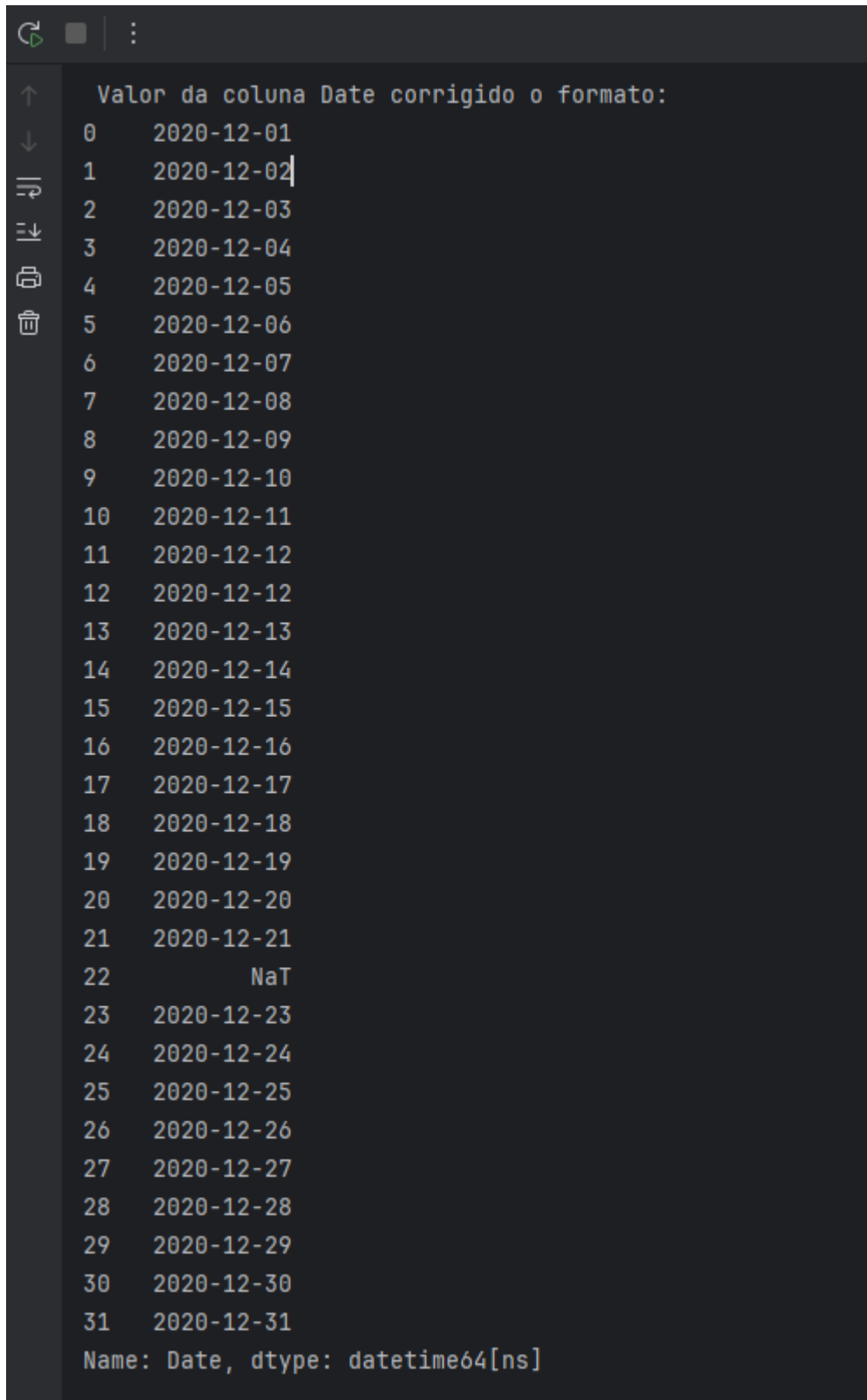
Para corrigir o valor especificamente da linha 26 na coluna Date, onde o valor estava em formato string e fora do padrão para formato de data, foi utilizado o método “.replace()”

```
# Substituindo o valor que não está no formato correto
data_copia['Date'] = data_copia['Date'].replace({"20201226": "2020/12/26"})
```

Após o passo anterior, foi executado novamente a transformação de todos os dados da coluna ‘Date’ para o formato datetime (usando o to\_datetime).

```
# Convertendo a coluna 'Date' para o formato datetime
data_copia['Date'] = pd.to_datetime(data_copia['Date'].str.replace("", ""), format='%Y/%m/%d')
print("\n Valor da coluna Date com formato corrigido:")
print(data_copia['Date'])
```

Resultado:



A screenshot of a Jupyter Notebook interface. The top bar shows a green refresh icon, a grey square, and a vertical ellipsis. The left sidebar contains icons for navigating between cells (up, down, up+down, down+down), a copy icon, and a delete icon. The main area displays a pandas Series of dates. The header is 'Valor da coluna Date corrigido o formato:'. The data consists of 32 rows, indexed from 0 to 31. Rows 0-21 contain dates from '2020-12-01' to '2020-12-21'. Row 22 contains 'NaT'. Rows 23-31 contain dates from '2020-12-23' to '2020-12-31'. At the bottom, the text 'Name: Date, dtype: datetime64[ns]' is displayed.

	Valor da coluna Date corrigido o formato:
0	2020-12-01
1	2020-12-02
2	2020-12-03
3	2020-12-04
4	2020-12-05
5	2020-12-06
6	2020-12-07
7	2020-12-08
8	2020-12-09
9	2020-12-10
10	2020-12-11
11	2020-12-12
12	2020-12-12
13	2020-12-13
14	2020-12-14
15	2020-12-15
16	2020-12-16
17	2020-12-17
18	2020-12-18
19	2020-12-19
20	2020-12-20
21	2020-12-21
22	NaT
23	2020-12-23
24	2020-12-24
25	2020-12-25
26	2020-12-26
27	2020-12-27
28	2020-12-28
29	2020-12-29
30	2020-12-30
31	2020-12-31

Name: Date, dtype: datetime64[ns]

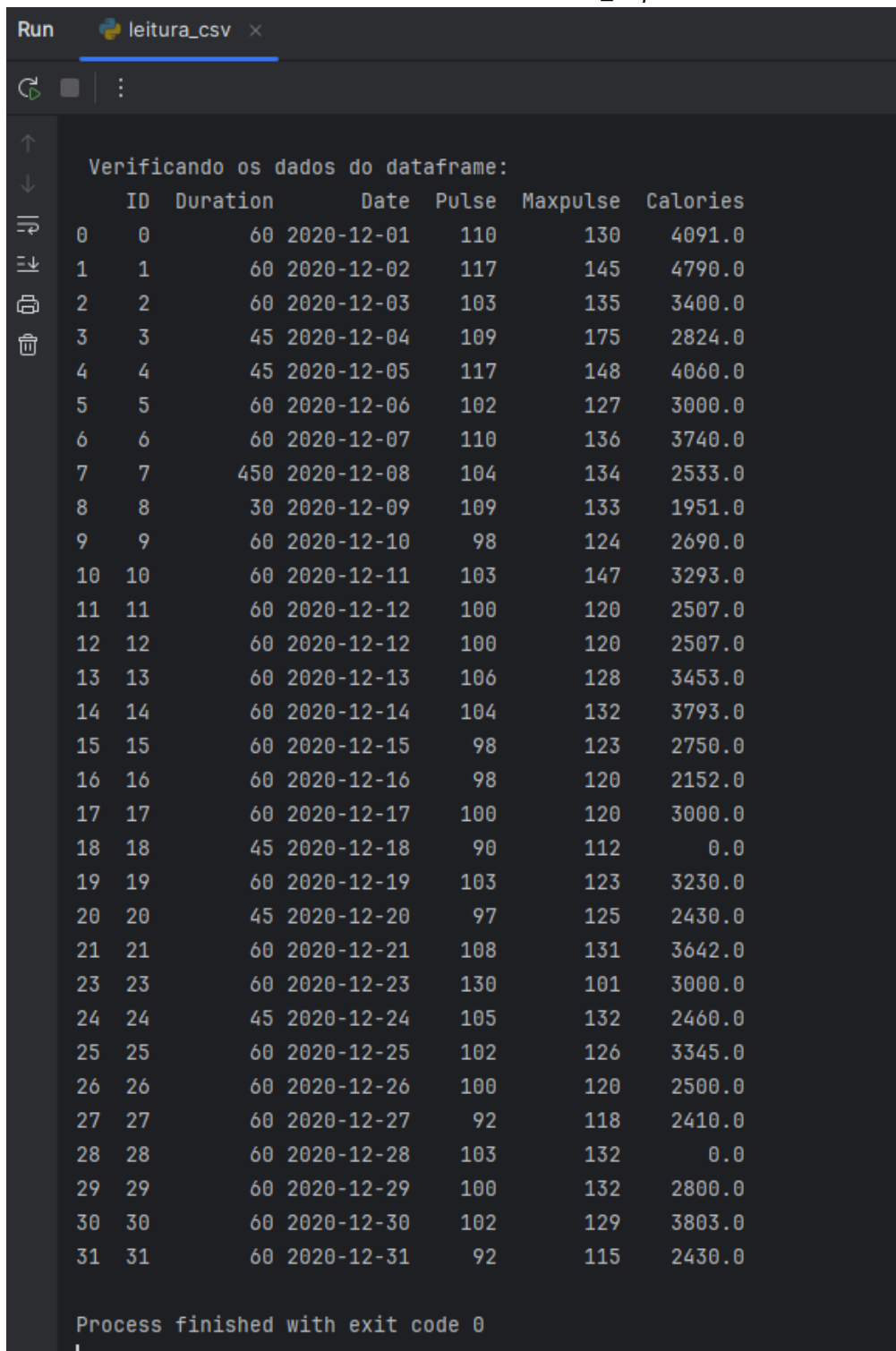
Para remover os registros contendo valores nulos, foi utilizado o método “.dropna()”

```
58  
59 # Removendo registros que possuem valores nulos na coluna 'Date'  
60 data_copia = data_copia.dropna(subset=['Date'])  
61
```

Foi utilizado o print para mostrar o dataframe alterado por completo.

```
62 #Encerrando com impressão de resultados.  
63 print("\n Verificando os dados do dataframe:")  
64 print(data_copia)
```

Resultado final – Dataframe “data\_copia”



The screenshot shows a Jupyter Notebook terminal window with the title 'Run' and a file named 'leitura\_csv'. The output of the print statement is displayed, showing the text 'Verificando os dados do dataframe:' followed by a table of data. The table has 7 columns: ID, Duration, Date, Pulse, Maxpulse, and Calories. The rows are indexed from 0 to 31. The data represents various metrics over time, with some values being 0.0, likely indicating missing or zero values.

	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

Process finished with exit code 0