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

# Substitua 'nome_do_arquivo.csv' pelo nome exato do seu arquivo
file_path = 'smart_watches.csv'

# Leia o arquivo CSV
df = pd.read_csv('C:\\Users\\sandr\\.ipynb_checkpoints\\smart_watches.csv', sep=';', engine

# Verifique os dados importados
print("Informações gerais:")
print(df.info())
print("\nPrimeiras 10 linhas:")
print(df.head(10))
print("\nÚltimas 10 linhas:")
print(df.tail(10))
```

```
Informações gerais:
<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
```

Primeiras 10 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
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:

	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

[4]:
```
# Substitua valores nulos na coluna 'Calories' por 0
df_copy['Calories'] = df_copy['Calories'].fillna(0)

# Verifique se a mudança foi aplicada com sucesso
print("Conjunto de dados após substituir valores nulos na coluna 'C
print(df_copy)
```

```
Conjunto de dados após substituir valores nulos na coluna 'Calories
    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
```

```python
# Substitua valores nulos na coluna 'Date' por '1900/01/01'
df_copy['Date'].fillna('1900/01/01', inplace=True)

# Verifique se a mudança foi aplicada com sucesso
print("Conjunto de dados após substituir valores nulos na coluna 'Date':")
print(df_copy)
```

```
Conjunto de dados após substituir valores nulos na coluna '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    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
```

[6]:
```python
# Tente transformar a coluna 'Date' em datetime
try:
    df_copy['Date'] = pd.to_datetime(df_copy['Date'], format='%Y/%m/%d')
except Exception as e:
    print(f"Erro ao transformar a coluna 'Date' em datetime: {e}")
```

```
Erro ao transformar a coluna 'Date' em datetime: time data "'2020/12/01'" doesn't match format "%Y/%m/%d", at position 0. You might want to try:
    - passing `format` if your strings have a consistent format;
    - passing `format='ISO8601'` if your strings are all ISO8601 but not necessarily in exactly the same format;
    - passing `format='mixed'`, and the format will be inferred for each element individually. You might want to use `dayfirst` alongside this.
```

[7]:
```python
# Substitua '1900/01/01' por 'NaN'
df_copy['Date'].replace('1900/01/01', pd.NaT, inplace=True)

# Transforme a coluna 'Date' em datetime novamente
df_copy['Date'] = pd.to_datetime(df_copy['Date'], errors='coerce')

# Verifique as mudanças
print("Conjunto de dados após transformação dos dados da coluna 'Date':")
print(df_copy)
```

```python
# Tente transformar a coluna 'Date' em datetime
try:
    df_copy['Date'] = pd.to_datetime(df_copy['Date'], format='%Y/%m/%d')
except Exception as e:
    print(f"Erro ao transformar a coluna 'Date' em datetime: {e}")
```

```
Erro ao transformar a coluna 'Date' em datetime: time data "'2020/12/01'" doesn't match format "%Y/%m/%d", at position 0. You might want to try:
    - passing `format` if your strings have a consistent format;
    - passing `format='ISO8601'` if your strings are all ISO8601 but not necessarily in exactly the same format;
    - passing `format='mixed'`, and the format will be inferred for each element individually. You might want to use `dayfirst` alongside this.
```

```python
# Substitua '1900/01/01' por 'NaN'
df_copy['Date'].replace('1900/01/01', pd.NaT, inplace=True)

# Transforme a coluna 'Date' em datetime novamente
df_copy['Date'] = pd.to_datetime(df_copy['Date'], errors='coerce')

# Verifique as mudanças
print("Conjunto de dados após transformação dos dados da coluna 'Date':")
print(df_copy)
```

```
Conjunto de dados após transformação dos dados da coluna '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
```

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	NaT	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

```
# Corrija o valor "20201226" para o formato datetime
df_copy['Date'] = df_copy['Date'].astype(str).replace('20201226', '2020/12/26')
df_copy['Date'] = pd.to_datetime(df_copy['Date'], errors='coerce')

# Verifique as mudanças
print("Conjunto de dados após correção do valor '20201226':")
print(df_copy)
```

```
Conjunto de dados após correção do valor '20201226':
    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	NaT	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

Conjunto de dados apos correçao do valor '20201226':

	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	NaT	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

```
# Remova registros com valores nulos
df_cleaned = df_copy.dropna()

# Verifique o DataFrame limpo
print("Conjunto de dados após remover registros com valores nulos:")
print(df_cleaned)
```

```
Conjunto de dados após remover registros com valores nulos:
    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
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
```

[76]:
```
import pandas as pd
```

[2]:
```
print(pd.__version__)
```

```
2.3.0
```

[23]:
```
dados = None
```

[77]:
```
    caminho_arquivo = 'online_retail.csv'
    try:
        dados = pd.read_csv('C:\\Users\\sandr\\.ipynb_checkpoints\\online_retail.csv', sep=';', encoding='utf-8', engine='python')
    except FileNotFoundError:
        print(f"Erro: Arquivo não encontrado em {caminho_arquivo}")
        exit()
    except Exception as e:
        print(f"Ocorreu um erro ao ler o arquivo: {e}")
        exit()
```

[9]:
```
print(dados)
```

```
       InvoiceNo StockCode                          Description  Quantity  \
0         536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER         6
1         536365     71053                  WHITE METAL LANTERN         6
2         536365    84406B       CREAM CUPID HEARTS COAT HANGER         8
3         536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE         6
4         536365    84029E       RED WOOLLY HOTTIE WHITE HEART.         6
...          ...       ...                                  ...       ...
541904    581587     22613          PACK OF 20 SPACEBOY NAPKINS        12
541905    581587     22899         CHILDREN'S APRON DOLLY GIRL          6
541906    581587     23254        CHILDRENS CUTLERY DOLLY GIRL          4
541907    581587     23255      CHILDRENS CUTLERY CIRCUS PARADE         4
541908    581587     22138        BAKING SET 9 PIECE RETROSPOT          3

            InvoiceDate UnitPrice  CustomerID         Country
0       01/12/2010 08:26      2,55     17850.0  United Kingdom
1       01/12/2010 08:26      3,39     17850.0  United Kingdom
2       01/12/2010 08:26      2,75     17850.0  United Kingdom
3       01/12/2010 08:26      3,39     17850.0  United Kingdom
4       01/12/2010 08:26      3,39     17850.0  United Kingdom
...                  ...       ...         ...             ...
```

```
...                   ...       ...          ...            ...
541904  09/12/2011 12:50      0,85      12680.0         France
541905  09/12/2011 12:50       2,1      12680.0         France
541906  09/12/2011 12:50      4,15      12680.0         France
541907  09/12/2011 12:50      4,15      12680.0         France
541908  09/12/2011 12:50      4,95      12680.0         France

[541909 rows x 8 columns]
```

```
analise = dados['Country']
analise1 = dados['CustomerID']
analise2 = dados['UnitPrice']
```

```
print(analise)
print(analise1)
print(analise2)
```

```
0         United Kingdom
1         United Kingdom
2         United Kingdom
3         United Kingdom
4         United Kingdom
               ...
541904           France
541905           France
541906           France
541907           France
541908           France
Name: Country, Length: 541909, dtype: object
0         17850.0
1         17850.0
2         17850.0
3         17850.0
4         17850.0
           ...
541904    12680.0
541905    12680.0
541906    12680.0
541907    12680.0
541908    12680.0
Name: CustomerID, Length: 541909, dtype: float64
0         2,55
1         3,39
```

```
4          3,39
           ...
541904     0,85
541905      2,1
541906     4,15
541907     4,15
541908     4,95
Name: UnitPrice, Length: 541909, dtype: object
```

```
pd.set_option('display.max_rows', 9999)
```

```
df = pd.DataFrame(dados)
```

```
df = pd.DataFrame(dados)
```

```
last_n_rows = df.tail(10)
print(last_n_rows)
```

```
       InvoiceNo StockCode                       Description  Quantity  \
541899    581587     22726         ALARM CLOCK BAKELIKE GREEN         4
541900    581587     22730         ALARM CLOCK BAKELIKE IVORY         4
541901    581587     22367  CHILDRENS APRON SPACEBOY DESIGN           8
541902    581587     22629               SPACEBOY LUNCH BOX          12
541903    581587     23256        CHILDRENS CUTLERY SPACEBOY         4
541904    581587     22613        PACK OF 20 SPACEBOY NAPKINS        12
541905    581587     22899       CHILDREN'S APRON DOLLY GIRL         6
541906    581587     23254      CHILDRENS CUTLERY DOLLY GIRL         4
541907    581587     23255  CHILDRENS CUTLERY CIRCUS PARADE          4
541908    581587     22138      BAKING SET 9 PIECE RETROSPOT         3

             InvoiceDate UnitPrice  CustomerID Country
541899  09/12/2011 12:50      3,75     12680.0  France
541900  09/12/2011 12:50      3,75     12680.0  France
541901  09/12/2011 12:50      1,95     12680.0  France
541902  09/12/2011 12:50      1,95     12680.0  France
541903  09/12/2011 12:50      4,15     12680.0  France
541904  09/12/2011 12:50      0,85     12680.0  France
541905  09/12/2011 12:50       2,1     12680.0  France
541906  09/12/2011 12:50      4,15     12680.0  France
541907  09/12/2011 12:50      4,15     12680.0  France
541908  09/12/2011 12:50      4,95     12680.0  France
```

[62]:
```
num_linhas = len(df)
print(f"Numero de linhas (len): {num_linhas}")
```

```
Numero de linhas (len): 541909
```

[67]:
```
num_colunas = df.shape
print(f"O numero de colunas e {num_colunas}")
```

```
O numero de colunas e (541909, 8)
```

[53]:
```
print(df.head(10))
```

```
  InvoiceNo StockCode                          Description  Quantity  \
0    536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER         6
1    536365     71053                  WHITE METAL LANTERN         6
2    536365    84406B       CREAM CUPID HEARTS COAT HANGER         8
3    536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE         6
4    536365    84029E       RED WOOLLY HOTTIE WHITE HEART.         6
5    536365     22752         SET 7 BABUSHKA NESTING BOXES         2
6    536365     21730    GLASS STAR FROSTED T-LIGHT HOLDER         6
7    536366     22633               HAND WARMER UNION JACK         6
8    536366     22632            HAND WARMER RED POLKA DOT         6
9    536367     84879        ASSORTED COLOUR BIRD ORNAMENT        32

       InvoiceDate UnitPrice  CustomerID         Country
0  01/12/2010 08:26      2,55     17850.0  United Kingdom
1  01/12/2010 08:26      3,39     17850.0  United Kingdom
2  01/12/2010 08:26      2,75     17850.0  United Kingdom
3  01/12/2010 08:26      3,39     17850.0  United Kingdom
4  01/12/2010 08:26      3,39     17850.0  United Kingdom
5  01/12/2010 08:26      7,65     17850.0  United Kingdom
6  01/12/2010 08:26      4,25     17850.0  United Kingdom
7  01/12/2010 08:28      1,85     17850.0  United Kingdom
8  01/12/2010 08:28      1,85     17850.0  United Kingdom
9  01/12/2010 08:34      1,69     13047.0  United Kingdom
```

```
total_nulos =df.isnull().sum().sum()
print("nTotal de valores nulos:", total_nulos)
```

```
nTotal de valores nulos: 136534
```

```
tipo_dados = df.dtypes
print(tipo_dados)
```

```
InvoiceNo       object
StockCode       object
Description     object
Quantity         int64
InvoiceDate     object
UnitPrice       object
CustomerID     float64
Country         object
dtype: object
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   InvoiceNo    541909 non-null  object 
 1   StockCode    541909 non-null  object 
 2   Description  540455 non-null  object 
 3   Quantity     541909 non-null  int64  
 4   InvoiceDate  541909 non-null  object 
 5   UnitPrice    541909 non-null  object 
 6   CustomerID   406829 non-null  float64
 7   Country      541909 non-null  object 
dtypes: float64(1), int64(1), object(6)
memory usage: 33.1+ MB
```