w3schools Tutorials ▾ Exercises ▾ Services ▾ Sign Up Log in

≡ . CSS JAVASCRIPT SQL PYTHON JAVA PHP HOW TO W3.CSS C

Pandas - Cleaning Data of Wrong Format

❮ Previous Next ❯

Data of Wrong Format

Cells with data of wrong format can make it difficult, or even impossible, to analyze data.

To fix it, you have two options: remove the rows, or convert all cells in the columns into the same format.

Convert Into a Correct Format

In our Data Frame, we have two cells with the wrong format. Check out row 22 and 26, the 'Date' column should be a string that represents a date:

```
      Duration          Date  Pulse  Maxpulse  Calories
  0         60  '2020/12/01'    110       130     409.1
  1         60  '2020/12/02'    117       145     479.0
  2         60  '2020/12/03'    103       135     340.0
  3         45  '2020/12/04'    109       175     282.4
  4         45  '2020/12/05'    117       148     406.0
  5         60  '2020/12/06'    102       127     300.0
  6         60  '2020/12/07'    110       136     374.0
  7        450  '2020/12/08'    104       134     253.3
  8         30  '2020/12/09'    109       133     195.1
  9         60  '2020/12/10'     98       124     269.0
  10        60  '2020/12/11'    103       147     329.3
  11        60  '2020/12/12'    100       120     250.7
```

```
16        60  '2020/12/16'     98       120      215.2
17        60  '2020/12/17'    100       120      300.0
18        45  '2020/12/18'     90       112        NaN
19        60  '2020/12/19'    103       123      323.0
20        45  '2020/12/20'     97       125      243.0
21        60  '2020/12/21'    108       131      364.2
22        45           NaN    100       119      282.0
23        60  '2020/12/23'    130       101      300.0
24        45  '2020/12/24'    105       132      246.0
25        60  '2020/12/25'    102       126      334.5
26        60      20201226    100       120      250.0
27        60  '2020/12/27'     92       118      241.0
28        60  '2020/12/28'    103       132        NaN
29        60  '2020/12/29'    100       132      280.0
30        60  '2020/12/30'    102       129      380.3
31        60  '2020/12/31'     92       115      243.0
```

Let's try to convert all cells in the 'Date' column into dates.

Pandas has a `to_datetime()` method for this:

Example

Get your own Python Server

Convert to date:

```
import pandas as pd

df = pd.read_csv('data.csv')

df['Date'] = pd.to_datetime(df['Date'])

print(df.to_string())
```

Try it Yourself »

Result:

```
  2        60  '2020/12/03'   103     135    340.0
  3        45  '2020/12/04'   109     175    282.4
  4        45  '2020/12/05'   117     148    406.0
  5        60  '2020/12/06'   102     127    300.0
  6        60  '2020/12/07'   110     136    374.0
  7       450  '2020/12/08'   104     134    253.3
  8        30  '2020/12/09'   109     133    195.1
  9        60  '2020/12/10'    98     124    269.0
  10       60  '2020/12/11'   103     147    329.3
  11       60  '2020/12/12'   100     120    250.7
  12       60  '2020/12/12'   100     120    250.7
  13       60  '2020/12/13'   106     128    345.3
  14       60  '2020/12/14'   104     132    379.3
  15       60  '2020/12/15'    98     123    275.0
  16       60  '2020/12/16'    98     120    215.2
  17       60  '2020/12/17'   100     120    300.0
  18       45  '2020/12/18'    90     112      NaN
  19       60  '2020/12/19'   103     123    323.0
  20       45  '2020/12/20'    97     125    243.0
  21       60  '2020/12/21'   108     131    364.2
  22       45           NaT   100     119    282.0
  23       60  '2020/12/23'   130     101    300.0
  24       45  '2020/12/24'   105     132    246.0
  25       60  '2020/12/25'   102     126    334.5
  26       60  '2020/12/26'   100     120    250.0
  27       60  '2020/12/27'    92     118    241.0
  28       60  '2020/12/28'   103     132      NaN
  29       60  '2020/12/29'   100     132    280.0
  30       60  '2020/12/30'   102     129    380.3
  31       60  '2020/12/31'    92     115    243.0
```

As you can see from the result, the date in row 26 was fixed, but the empty date in row 22 got a NaT (Not a Time) value, in other words an empty value. One way to deal with empty values is simply removing the entire row.

Removing Rows

The result from the converting in the example above gave us a NaT value, which can be handled as a NULL value, and we can remove the row by using the `dropna()` method.

Example

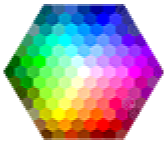
Remove rows with a NULL value in the "Date" column:

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

Try it Yourself »

❮ Previous Log in to track progress Next ❯

COLOR PICKER



SPACES UPGRADE AD-FREE

NEWSLETTER GET CERTIFIED REPORT ERROR

Top Tutorials

HTML Tutorial
CSS Tutorial
JavaScript Tutorial
How To Tutorial
SQL Tutorial

Top References

HTML Reference
CSS Reference
JavaScript Reference
SQL Reference
Python Reference

jQuery Tutorial

jQuery Reference

Top Examples

HTML Examples
CSS Examples
JavaScript Examples
How To Examples
SQL Examples
Python Examples
W3.CSS Examples
Bootstrap Examples
PHP Examples
Java Examples
XML Examples
jQuery Examples

Get Certified

HTML Certificate
CSS Certificate
JavaScript Certificate
Front End Certificate
SQL Certificate
Python Certificate
PHP Certificate
jQuery Certificate
Java Certificate
C++ Certificate
C# Certificate
XML Certificate

FORUM ABOUT

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning.
Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness
of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.