How can columns be converted to DateTime in Pandas?

Authored by

stats writer

April 17, 2024

"Columns in a Pandas dataframe can be converted to DateTime format by using the 'to_datetime()' function. This method allows for the conversion of various date and time formats to a standardized DateTime format, making it easier to perform time-based operations and analysis on the data. The 'to_datetime()' function also provides options to handle missing or incorrect values, ensuring accurate and consistent DateTime conversion. By utilizing this function, columns in a Pandas dataframe can be seamlessly converted to DateTime, allowing for efficient and precise manipulation and analysis of time-based data."

Convert Columns to DateTime in Pandas

Often you may be interested in converting one or more columns in a pandas DataFrame to a DateTime format. Fortunately this is easy to do using the to_datetime() function.

This tutorial shows several examples of how to use this function on the following DataFrame:

```
import numpy as np
import pandas as pd

#create DataFrame
df = pd.DataFrame({'event': ,
'start_date': ,
'end_date': })

#view DataFrame
df
```

event start_date end_date
0 A 20150601 20150608
1 B 20160201 20160209
2 C 20170401 201704161

#view column data types
df.dtypes

event object
start_date object
end_date object
dtype: object

Example 1: Convert a Single Column to DateTime

The following code shows how to convert the "start_date" column from a string to a DateTime format:

#convert start_date to DateTime format
df = pd.to_datetime(df)

#view DataFrame
df

event start_date end_date
0 A 2015-06-01 20150608

1 B 2016-02-01 20160209
2 C 2017-04-01 20170416

#view column date types
df.dtypes

event object
start_date datetime64
end_date object
dtype: object

Note that the to_datetime() function is smart and can typically infer the correct date format to use, but you can also specify the format to use with the format argument:

#convert start_date to DateTime format
df = pd.to_datetime(df, format='%Y%m%d')

#view DataFrame
df

event start_date end_date
0 A 2015-06-01 20150608
1 B 2016-02-01 20160209

2 C 2017-04-01 20170416

#view column date types
df.dtypes

event object
start_date datetime64
end_date object
dtype: object

Example 2: Convert Multiple Columns to DateTime

The following code shows how to convert both the "start_date" and "end_date" columns from strings to DateTime formats:

#convert start_date and end_date to DateTime formats
df] = df].apply(pd.to_datetime)

#view DataFrame
df

event start_date end_date
0 A 2015-06-01 2015-06-08
1 B 2016-02-01 2016-02-09
2 C 2017-04-01 2017-04-16

```
#view column date types
df.dtypes

event object
start_date datetime64
end_date datetime64
dtype: object
```

Example 3: Convert Columns to DateTime Format with Seconds

In some cases you may also have columns that include a date along with the hours, minutes and seconds, such as the following DataFrame:

```
#create DataFrame
df = pd.DataFrame({'event': ,
'start_date': ,
'end_date': })

#view DataFrame
df

event start_date end_date
0 A 20150601043000 20150608
1 B 20160201054500 20160209
2 C 20170401021215 20170416
```

Once again, the to_datetime() function is smart and can usually infer the correct format to use without us specifying it:

```
#convert start_date to DateTime format
df = pd.to_datetime(df)

#view DataFrame
df

event start_date end_date
0 A 2015-06-01 04:30:00 20150608
1 B 2016-02-01 05:45:00 20160209
2 C 2017-04-01 02:12:15 20170416

#view column date types
df.dtypes

event object
start_date datetime64
end_date object
dtype: object
```

Of course, in the wild you're likely to come across a variety of weird DateTime formats so you may have to

actually use the format argument to tell Python exactly what DateTime format to use.

How to Convert Datetime to Date in Pandas

How to Convert Strings to Float in Pandas