How do I convert a string to a datetime format in Pandas?

Authored by

stats writer

June 25, 2024

Converting a string to a datetime format in Pandas allows for efficient processing and manipulation of time-related data. This can be achieved by using the built-in to_datetime() function, which converts a string representation of a date or time into a datetime object. This enables users to perform various operations such as sorting, filtering, and grouping based on specific time intervals. The process of converting a string to a datetime format in Pandas is simple and can be done with just a few lines of code, making it an essential tool for data analysis and manipulation.

Convert String to Datetime in Pandas

You can use the following methods to convert a string column to a datetime format in a pandas DataFrame:

Method 1: Convert One String Column to Datetime

df = pd.to_datetime(df)

Method 2: Convert Multiple String Columns to Datetime

df] = df].apply(pd.to_datetime)

The following examples show how to use each of these methods in practice with the following pandas DataFrame:

import pandas as pd

#create DataFrame
df = pd.DataFrame({'task': ,

'due_date': ,
'comp_date': })

#view DataFrame
print(df)

task due_date comp_date
0 A 2022-04-15 2022-04-14
1 B 2022-05-19 2022-05-23
2 C 2022-06-14 2022-06-24
3 D 2022-10-24 2022-10-07

#view data type of each columnprint(df.dtypes)

task object
due_date object
comp_date object
dtype: object

We can see that each column in the DataFrame currently has a data type of object, i.e. a string.

Example 1: Convert One String Column to Datetime

We can use the following syntax to convert the due_date column from a string to a datetime:

```
#convert due_date column to datetime
df = pd.to_datetime(df)

#view updated DataFrame
print(df)

task due_date comp_date
0 A 2022-04-15 4-14-2022
1 B 2022-05-19 5-23-2022
2 C 2022-06-14 6-24-2022
3 D 2022-10-24 10-7-2022

#view data type of each column
print(df.dtypes)

task object
due_date datetime64
comp_date object
dtype: object
```

We can see that the due_date column has been converted to a datetime while all other columns have remain unchanged.

Example 2: Convert Multiple String Columns to Datetime

We can use the following syntax to convert both the due_date and comp_date columns from a string to a datetime:

```
#convert due_date and comp_date columns to datetime
df] = df].apply(pd.to_datetime)

#view updated DataFrame
print(df)

task due_date comp_date
0 A 2022-04-15 2022-04-14
1 B 2022-05-19 2022-05-23
2 C 2022-06-14 2022-06-24
3 D 2022-10-24 2022-10-07

#view data type of each column
print(df.dtypes)

task object
due_date datetime64
comp_date datetime64
dtype: object
```

We can see that the due_date and comp_date columns

have both been converted from a string to a datetime.

Note: You can find the complete documentation for the pandas to_datetime() function .

The following tutorials explain how to perform other common operations in pandas: