How can I create a date column in Pandas from separate columns for year, month, and day?

Authored by

stats writer

June 26, 2024

To create a date column in Pandas from separate columns for year, month, and day, the pd.to_datetime() function can be used. This function converts the three separate columns into a single datetime column, which can then be easily manipulated and used for analysis. The year, month, and day columns should be in numeric format, and the datetime column can be named and added to the existing dataframe using the df.insert() function. This method allows for efficient and accurate creation of a date column in Pandas.

Pandas: Create Date Column from Year, Month and Day

You can use the following basic syntax to create a date column from year, month, and day columns in a pandas DataFrame:

df = pd.to_datetime(dict(year=df.year, month=df.month, day=df.day))

The following example shows how to use this syntax in practice.

Example: Create Date Column from Year, Month and Day in Pandas

Suppose we have the following pandas DataFrame that shows the sales made by some company on various dates:

import pandas as pd

#create DataFrame

```
df = pd.DataFrame({'year': ,
'month': ,
'day': ,
'sales': })

#view DataFrame
print(df)

year month day sales
0 2021 7 4 140
1 2022 1 15 200
2 2022 1 25 250
3 2022 2 27 180
4 2022 5 27 130
5 2022 10 24 87
6 2022 11 10 90
7 2022 12 18 95
```


We can use the following syntax to create a new column called date that combines the values from the year, month, and day columns in the DataFrame to create a date for each row:

```
#create date column from year, month, and day columns
```

df = pd.to_datetime(dict(year=df.year, month=df.month, day=df.day))

#view updated DataFrame
print(df)

year month day sales date
0 2021 7 4 140 2021-07-04
1 2022 1 15 200 2022-01-15
2 2022 1 25 250 2022-01-25
3 2022 2 27 180 2022-02-27
4 2022 5 27 130 2022-05-27
5 2022 10 24 87 2022-10-24
6 2022 11 10 90 2022-11-10
7 2022 12 18 95 2022-12-18

Notice that the date column contains date values based on the values from the year, month, and day columns in each row.

If we use df.info() to get information about each column in the DataFrame, we can see that the new date column has a data type of datetime64:

#display information about each column in DataFrame

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 5 columns):
# Column Non-Null Count Dtype
--- ------ -------------- -----
0 year 8 non-null int64
1 month 8 non-null int64
2 day 8 non-null int64
3 sales 8 non-null int64
4 date 8 non-null datetime64
dtypes: datetime64(1), int64(4)
memory usage: 388.0 bytes
```

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