How can I calculate the difference between two dates using Pandas?

Authored by

stats writer

June 29, 2024

Pandas is a popular data analysis library in Python that offers a variety of functions for working with dates and times. One of its useful features is the ability to calculate the difference between two dates. This can be achieved by using the built-in function "timedelta" which allows for easy manipulation of time intervals. By simply subtracting one date from another, the timedelta function returns the difference in the specified units such as days, months, or years. This makes it a convenient tool for tasks such as tracking time durations, calculating age, or identifying time gaps in data. With its intuitive syntax and efficient processing, Pandas makes it simple to accurately calculate the difference between two dates for any data analysis or time-sensitive project.

Pandas: Calculate a Difference Between Two Dates

You can use the following syntax to calculate a difference between two dates in a pandas DataFrame:

df = (df - df) / np.timedelta64(1, 'D')

This particular example calculates the difference between the dates in the end_date and start_date columns in terms of days.

Note that we can replace the 'D' in the timedelta64() function with the following values to calculate the date difference in different units:

W: WeeksM: MonthsY: Years

The following examples show how to calculate a date difference in a pandas DataFrame in practice.

Example 1: Calculate Difference Between Two Dates with Datetime Columns

Suppose we have the following pandas DataFrame:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'start_date': pd.date_range(start='1/5/2020', periods=6, freq='W'), 'end_date': pd.date_range(start='6/1/2020', periods=6, freq='M')})

#view DataFrame
print(df)

  start_date end_date
0 2020-01-05 2020-06-30
1 2020-01-12 2020-07-31
2 2020-01-19 2020-08-31
3 2020-01-26 2020-09-30
4 2020-02-02 2020-10-31
5 2020-02-09 2020-11-30

#view dtype of each column in DataFrame
df.dtypes
```

start_date datetime64
end_date datetime64
dtype: object

Since both columns in the DataFrame already have a dtype of datetime64, we can use the following syntax to calculate the difference between the start and end dates:

```
import numpy as np

#create new columns that contains date differences
df = (df - df) / np.timedelta64(1, 'D')
df = (df - df) / np.timedelta64(1, 'W')
df = (df - df) / np.timedelta64(1, 'M')
df = (df - df) / np.timedelta64(1, 'Y')

#view updated DataFrame
print(df)
```

start_date end_date diff_days diff_weeks diff_months diff_years
0 2020-01-05 2020-06-30 177.0 25.285714 5.815314 0.484610
1 2020-01-12 2020-07-31 201.0 28.714286 6.603832

0.550319
2 2020-01-19 2020-08-31 225.0 32.142857 7.392349 0.616029
3 2020-01-26 2020-09-30 248.0 35.428571 8.148011 0.679001
4 2020-02-02 2020-10-31 272.0 38.857143 8.936528 0.744711
5 2020-02-09 2020-11-30 295.0 42.142857 9.692191 0.807683

The new columns contain the date differences between the start and end dates in terms of days, weeks, months, and years.

Example 2: Calculate Difference Between Two Dates with String Columns

Suppose we have the following pandas DataFrame:

```
import pandas as pd

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

#view dtype of each columnprint(df.dtypes)
```

start_date object
end_date object
dtype: object

Since neither column in the DataFrame has a dtype of datetime64, we will receive an error if we attempt to calculate the difference between the dates:

```
import numpy as np

#attempt to calculate date difference
df = (df - df) / np.timedelta64(1, 'D')
```

TypeError: unsupported operand type(s) for -: 'str' and 'str'

```
import numpy as np

#convert columns to datetime
df] = df].apply(pd.to_datetime)

#calculate difference between dates
df = (df - df) / np.timedelta64(1, 'D')

#view updated DataFrame
```

```
print(df)

start_date end_date diff_days
0 2020-01-05 2020-06-30 177.0
1 2020-01-12 2020-07-31 201.0
2 2020-01-19 2020-08-31 225.0
```

Since we first converted each column to a datetime format, we were able to successfully calculate the difference between the dates without any errors.

Additional Resources

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