How can dates be compared in Pandas, and what are some examples of doing so?

Authored by

stats writer

June 25, 2024

Pandas is a popular data analysis library in Python that offers various functions for manipulating and comparing dates. Dates in Pandas are represented as datetime objects, which contain information about the year, month, day, hour, minute, and second. These objects can be easily compared using the comparison operators such as ">", " df

This will return a series of boolean values, where "True" indicates that date1 is greater than date2 and "False" indicates otherwise.

Another example of comparing dates in Pandas is to filter a dataframe based on a specific date range. This can be done by using the comparison operators in conjunction with the loc function.

For instance, to filter a dataframe for all rows with dates between January 1, 2020 and December 31, 2020, the following code can be used:

df.loc > '2020-01-01') & (df

Compare Dates in Pandas (With Examples)

You can use the following methods to compare dates between two columns in a pandas DataFrame:

Method 1: Add New Column to DataFrame that Shows Date Comparison

df = df < df

This particular example adds a new column called met_due_date that returns True or False depending on whether the date in the comp_date column is before the date in the due_date column.

Method 2: Filter DataFrame Based on Date Comparison

df_met_due_date = df < df]

This particular example filters the DataFrame to only keep rows where the date in the comp_date column is before the date in the due_date column.

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': })

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

#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

Example 1: Add New Column to DataFrame that Shows Date Comparison

The following code shows how to add a new column called met_due_date that returns True or False depending on whether the date in the comp_date column is before the date in the due_date column.

```
import pandas as pd

#create new column that shows if completion date is before due date
df = df < df

#view updated DataFrame
print(df)
```

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

For each row in the DataFrame, the new met_due_date column shows whether the date in the comp_date column is before the date in the due_date column.

For example, we can see that task A had a due date of 4/15/2022 and a completion date of 4/14/2022.

Since the completion date was before the due date, the value in the met_due_date column is True.

Example 2: Filter DataFrame Based on Date Comparison

The following code shows how to filter the DataFrame to only contain rows where the date in the comp_date column is before the date in the due_date column.

```
import pandas as pd

#filter for rows where completion date is before due date
df_met_due_date = df < df]

#view results
print(df_met_due_date)

task due_date comp_date
0 A 2022-04-15 2022-04-14
```

3 D 2022-10-24 2022-10-07

The new DataFrame has been filtered to only contain rows where the date in the comp_date column is before the date in the due_date column.

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