

How to Compare Dates in PySpark: A Step-by-Step Guide

Authored by
stats writer

February 6, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Compare Dates in PySpark: A Step-by-Step Guide*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=129517>

PySpark is a powerful tool used for processing and analyzing large datasets. In order to compare dates in PySpark, there are a few steps that need to be followed. Firstly, the dates need to be converted into a proper date format using the "to_date" function. Then, the "cast" function can be used to convert the dates into a specific data type, such as a timestamp. Once the dates are in the proper format, they can be compared using standard comparison operators such as ">", "<", "=", etc. An example of comparing dates in PySpark would be as follows:

```
df = spark.createDataFrame(, )  
  
df = df.withColumn('dates', to_date(df.dates))  
  
df = df.withColumn('dates', cast(df.dates, 'timestamp'))  
  
df.filter(df.dates > '2020-01-01').show()
```

This code snippet compares the dates in the "dates" column and filters out any dates that are greater than 2020-01-01. This is just one example of how dates can be compared in PySpark, and the same logic can be applied for other comparison scenarios as well.

Compare Dates in PySpark (With Example)

You can use the following syntax to compare dates in a PySpark DataFrame:

```
#create new column that compares dates in due_date  
and finish_date columns  
df_new = df.withColumn('met_due_date',  
when(df.finish_date <= df.due_date, 'yes')  
.otherwise('no'))
```

This particular example creates a new column called met_due_date that compares the dates in the due_date

and `finish_date` columns of a `DataFrame` and returns `yes` or `no` to indicate if the finish date was equal to or prior to the due date.

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

Example: How to Compare Dates in PySpark

Suppose we have the following PySpark `DataFrame` that contains information about the due date for various tasks along with the date the task was actually finished:

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
```

```
#define data
```

```
data = ,
```

```
,
```

```
,
```

```
,
```

```
]
```

```
#define column names
```

```
columns =
```

```
#create dataframe using data and column names
```

```
df = spark.createDataFrame(data, columns)
```

```
#view dataframe
```

```
df.show()
```

```
+----+-----+-----+
|task| due_date|finish_date|
+----+-----+-----+
| A|2023-04-11| 2023-04-03|
| B|2023-04-15| 2023-04-12|
| C|2023-04-24| 2023-04-25|
| D|2023-05-26| 2023-05-23|
| E|2023-07-18| 2023-08-10|
+----+-----+-----+
```

Suppose we would like to add a new column to the DataFrame that returns either yes or no to indicate if the finish date is equal to or prior to the due date.

We can use the following syntax to do so:

```
#create new column that compares dates in due_date  
and finish_date columns
```

```
df_new = df.withColumn('met_due_date',
```

```
when(df.finish_date <= df.due_date, 'yes')  
.otherwise('no'))
```

```
#view new DataFrame  
df_new.show()
```

```
+----+-----+-----+-----+  
|task| due_date|finish_date|met_due_date|  
+----+-----+-----+-----+  
| A|2023-04-11| 2023-04-03| yes|  
| B|2023-04-15| 2023-04-12| yes|  
| C|2023-04-24| 2023-04-25| no|  
| D|2023-05-26| 2023-05-23| yes|  
| E|2023-07-18| 2023-08-10| no|  
+----+-----+-----+-----+
```

The new `met_due_date` column returns either `yes` or `no` to indicate if each task was finished by the due date.

For example, we can see:

Task A was finished on 2023-04-03, which is before the due date of 2023-04-11, so the column returns `yes`. Task C was finished on 2023-04-25, which is not before the due date of 2023-04-24, so the column returns `no`.

And so on.

Note that we used the `withColumn` function to return a new `DataFrame` with the `met_due_date` column added and all other existing columns left the same.

You can find the complete documentation for the `PySpark withColumn` function .

ARABPSYCHOLOGY.COM