How can I unpivot a Pandas DataFrame?

Authored by

stats writer

June 29, 2024

Unpivoting a Pandas DataFrame refers to the process of transforming a wide or "spreadsheet-like" table into a long or "tidy" format. This involves converting columns into rows and rearranging the data to have a single column for the variable names and a single column for the corresponding values. This can be achieved in Pandas by using the `melt()` function, which allows for the reshaping of data frames by specifying the set of columns to be used as identifiers and the set of columns to be unpivoted. This process is useful for data analysis and visualization as it allows for easier manipulation and interpretation of data.

Unpivot a Pandas DataFrame (With Example)

In pandas, you can use the function to unpivot a DataFrame - converting it from a wide format to a .

This function uses the following basic syntax:

df_unpivot = pd.melt(df, id_vars='col1', value_vars=)

where:

id_vars: The columns to use as identifiersvalue_vars: The columns to unpivot

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

Example: Unpivot a Pandas DataFrame

Suppose we have the following pandas DataFrame:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'team': ,
'points': ,
'assists': ,
'rebounds': })

#view DataFrame
print(df)

team points assists rebounds
0 A 18 5 11
1 B 22 7 8
2 C 19 7 10
3 D 14 9 6
4 E 14 12 6
```

We can use the following syntax to "unpivot" the DataFrame:

```
#unpivot DataFrame from wide format to long format
df_unpivot = pd.melt(df, id_vars='team', value_vars=)

#view updated DataFrame
```

```
print(df_unpivot)

team variable value
0 A points 18
1 B points 22
2 C points 19
3 D points 14
4 E points 14
5 A assists 5
6 B assists 7
7 C assists 7
8 D assists 9
9 E assists 12
10 A rebounds 11
11 B rebounds 8
12 C rebounds 10
13 D rebounds 6
14 E rebounds 6
```


We used the team column as the identifier column and we chose to unpivot the points, assists, and rebounds columns.

The result is a DataFrame in a long format.

Note that we can also use the var_name and value_name arguments to specify the names of the columns in the unpivoted DataFrame:

```
#unpivot DataFrame from wide format to long format
df_unpivot = pd.melt(df, id_vars='team', value_vars=,
var_name='metric', value_name='amount')

#view updated DataFrame
print(df_unpivot)
```

```
team metric amount
0 A points 18
1 B points 22
2 C points 19
3 D points 14
4 E points 14
5 A assists 5
6 B assists 7
7 C assists 7
8 D assists 9
9 E assists 12
10 A rebounds 11
11 B rebounds 8
12 C rebounds 10
```

13 D rebounds 6

14 E rebounds 6

Notice that the new columns are now labeled metric and amount.

Additional Resources