

How to Easily Create a 3D Pandas DataFrame from a 2D DataFrame

Authored by
stats writer

November 24, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Create a 3D Pandas DataFrame from a 2D DataFrame*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=100430>

To create a 3D Pandas DataFrame, you must first create a 2D data frame, and then use the "stack" method to convert it into a 3D data frame. To illustrate this, let's take a look at an example. We'll create a 2D data frame with the numbers 1 to 6 in the columns A, B, and C and then use the 'stack' method to make it 3D. The result will be a 3D data frame with the numbers 1 to 6 in the columns A, B, C, A1, B1, and C1.

You can use the module to quickly create a 3D pandas DataFrame.

This tutorial explains how to create the following 3D pandas DataFrame using functions from the xarray module:

```
product_A product_B product_C
year quarter
2021 Q1 1.624345 0.319039 50
      Q2 -0.611756 0.319039 50
      Q3 -0.528172 0.319039 50
      Q4 -1.072969 0.319039 50
2022 Q1 0.865408 -0.249370 50
      Q2 -2.301539 -0.249370 50
      Q3 1.744812 -0.249370 50
      Q4 -0.761207 -0.249370 50
```

Example: Create 3D Pandas DataFrame

The following code shows how to create a 3D dataset using functions from **xarray** and **NumPy**:

```
import numpy as np
import xarray as xr

#make this example reproducible
np.random.seed(1)

#create 3D dataset
xarray_3d = xr.Dataset(
{"product_A": (("year", "quarter"), np.random.randn(2, 4))},
coords={
"year": ,
"quarter": ,
"product_B": ("year", np.random.randn(2)),
"product_C": 50,
```

```
},  
)  
  
#view 3D dataset  
print(xarray_3d)  
  
Dimensions: (year: 2, quarter: 4)  
Coordinates:  
* year (year) int32 2021 2022  
* quarter (quarter) <U2 'Q1' 'Q2' 'Q3' 'Q4'  
product_B (year) float64 0.319 -0.2494  
product_C int32 50  
Data variables:  
product_A (year, quarter) float64 1.624 -0.6118 -0.5282 ... 1.745 -0.7612
```

Note: The NumPy function returns sample values from the .

We can then use the `to_dataframe()` function to convert this dataset to a pandas DataFrame:

```
#convert xarray to DataFrame  
df_3d = xarray_3d.to_dataframe()
```

```
#view 3D DataFrame  
print(df_3d)  
  
product_A product_B product_C  
year quarter  
2021 Q1 1.624345 0.319039 50  
Q2 -0.611756 0.319039 50  
Q3 -0.528172 0.319039 50  
Q4 -1.072969 0.319039 50  
2022 Q1 0.865408 -0.249370 50  
Q2 -2.301539 -0.249370 50  
Q3 1.744812 -0.249370 50  
Q4 -0.761207 -0.249370 50
```

The result is a 3D pandas DataFrame that contains information on the number of sales made of three different products during two different years and four different quarters per year.

We can use the `type()` function to confirm that this object is indeed a pandas DataFrame:

```
#display type of df_3d  
type(df_3d)
```

```
pandas.core.frame.DataFrame
```

The object is indeed a pandas DataFrame.

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

ARABPSYCHOLOGY.COM