

How can I create a 3D Pandas DataFrame, and what are some examples of how it can be used?

Authored by
stats writer

June 27, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I create a 3D Pandas DataFrame, and what are some examples of how it can be used?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=154502>

A 3D Pandas DataFrame is a data structure that allows for the storage and manipulation of three-dimensional data in a tabular format. This type of DataFrame can be created by importing the Pandas library in Python and using the "Panel" function. It can also be created by reshaping a 2D DataFrame using the "pivot" or "melt" functions.

There are various ways in which a 3D Pandas DataFrame can be utilized. For example, it can be used for storing and analyzing time series data, where the rows represent different time points, the columns represent different variables, and the third dimension represents different categories or groups. It can also be used for spatial data analysis, where the rows and columns represent different locations, and the third dimension represents different time periods.

Additionally, a 3D Pandas DataFrame can be useful for creating and managing multi-indexed data, where each index represents a different level of categorization. This allows for efficient data organization and retrieval. It can also be used for merging and joining multiple datasets that have different dimensions, making it a versatile tool for data integration.

Overall, a 3D Pandas DataFrame is a powerful tool for data analysis and can be beneficial in various applications, such as finance, marketing, and research. Its flexibility and ability to handle complex data make it a valuable asset for any data scientist or analyst.

Create a 3D Pandas DataFrame (With Example)

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