

How can a Pandas Series be converted into a DataFrame?

Authored by
stats writer

May 5, 2024

RECOMMENDED CITATION

stats writer (2024). *How can a Pandas Series be converted into a DataFrame?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=142942>

A Pandas Series can be converted into a DataFrame by using the "to_frame()" function. This function will convert the Series into a one-column DataFrame, with the Series values as the values in the column. Alternatively, the "pd.DataFrame()" function can also be used to create a DataFrame from a Series, by passing the Series as an argument and specifying the column name. This will result in a DataFrame with the Series values as the values in the specified column. Additionally, multiple Series can be combined into a DataFrame by passing them as a dictionary to the "pd.DataFrame()" function, with each Series becoming a column in the DataFrame. Overall, the conversion of a Series into a DataFrame allows for easier manipulation and analysis of data, as DataFrames offer more functionality and flexibility.

Convert Pandas Series to DataFrame (With Examples)

You can use the following basic syntax to convert a pandas Series to a pandas DataFrame:

```
my_df = my_series.to_frame(name='column_name')
```

The following examples show how to use this syntax in practice.

Example 1: Convert One Series to Pandas DataFrame

Suppose we have the following pandas Series:

```
import pandas as pd
```

```
#create pandas Series
```

```
my_series = pd.Series()
```

```
#view pandas Series
```

```
print(my_series)
```

```
0 3
```

```
1 4
```

```
2 4
```

```
3 8
```

```
4 14
```

```
5 17
```

```
6 20
```

```
dtype: int64
```

```
#view object type
```

```
print(type(my_series))
```

```
<class 'pandas.core.series.Series'>
```

We can use the `to_frame()` function to quickly convert this pandas Series to a pandas DataFrame:

```
#convert Series to DataFrame and specify column name  
to be 'values'
```

```
my_df = my_series.to_frame(name='values')
```

```
#view pandas DataFrame print(my_df)
```

```
values
```

0 3

1 4

2 4

3 8

4 14

5 17

6 20

```
#view object type print(type(my_df))
```

```
<class 'pandas.core.frame.DataFrame'>
```

Example 2: Convert Multiple Series to Pandas DataFrame

Suppose we have three different pandas Series:

```
import pandas as pd
```

```
#define three Series
```

```
name = pd.Series()
```

```
points = pd.Series()
```

```
assists = pd.Series()
```

We can use the following syntax to convert each Series into a DataFrame and concatenate the three DataFrames into one final DataFrame:

```
#convert each Series to a DataFrame  
name_df = name.to_frame(name='name')  
points_df = points.to_frame(name='points')  
assists_df = assists.to_frame(name='assists')
```

```
#concatenate three Series into one DataFrame  
df = pd.concat(, axis=1)
```

```
#view final DataFrame  
print(df)
```

```
name points assists  
0 A 34 8  
1 B 20 12  
2 C 21 14  
3 D 57 9  
4 E 68 11
```

The final result is a pandas DataFrame where each Series represents a column.

The following tutorials explain how to perform other common data object conversions in pandas: