

# How do I create a new DataFrame from an existing DataFrame in Pandas?

Authored by  
**stats writer**

June 28, 2024

## RECOMMENDED CITATION

stats writer (2024). *How do I create a new DataFrame from an existing DataFrame in Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=156427>

To create a new DataFrame from an existing DataFrame in Pandas, one can use the "copy()" method to make a copy of the original DataFrame. This will allow for any modifications or manipulations to be made to the new DataFrame without affecting the original. Additionally, the "DataFrame()" constructor can also be used to create a new DataFrame by passing in the data and columns from the existing DataFrame. This will create a new DataFrame with the same data and column names as the original. Both methods provide a convenient way to create a new DataFrame from an existing one in Pandas.

## **Pandas: Create New DataFrame from Existing DataFrame**

**There are three common ways to create a new pandas DataFrame from an existing DataFrame:**

### **Method 1: Create New DataFrame Using Multiple Columns from Old DataFrame**

```
new_df = old_df.copy()
```

### **Method 2: Create New DataFrame Using One Column from Old DataFrame**

```
new_df = old_df.copy()
```

### **Method 3: Create New DataFrame Using All But One Column from Old DataFrame**

```
new_df = old_df.drop('col1', axis=1)
```

The following examples show how to use each method with the following pandas DataFrame:

```
import pandas as pd

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

#view DataFrame
print(old_df)
```

Example 1: Create New DataFrame Using Multiple Columns from Old DataFrame

The following code shows how to create a new DataFrame using multiple columns from the old DataFrame:

```
#create new DataFrame from existing DataFrame
new_df = old_df].copy()

#view new DataFrame
print(new_df)
```

## points rebounds

0 18 11

1 22 8

2 19 10

3 14 6

4 14 6

5 11 7

6 20 9

7 28 12

## #check data type of new DataFrame

```
type(new_df)
```

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

Notice that this new DataFrame only contains the points and rebounds columns from the old DataFrame.

**Note:** It's important to use the `copy()` function when creating the new DataFrame so that we avoid any if we happen to modify the new DataFrame in any way.

**Example 2: Create New DataFrame Using One Column from Old DataFrame**

The following code shows how to create a new DataFrame using one column from the old DataFrame:

```
#create new DataFrame from existing DataFrame
```

```
new_df = old_df].copy()
```

```
#view new DataFrame
```

```
print(new_df)
```

```
points
```

```
0 18
```

```
1 22
```

```
2 19
```

```
3 14
```

```
4 14
```

```
5 11
```

```
6 20
```

```
7 28
```

```
#check data type of new DataFrame
```

```
type(new_df)
```

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

**Example 3: Create New DataFrame Using All But One Column from Old DataFrame**

**The following code shows how to create a new DataFrame using all but one column from the old**

## DataFrame:

```
#create new DataFrame from existing DataFrame
```

```
new_df = old_df.drop('points', axis=1)
```

```
#view new DataFrame
```

```
print(new_df)
```

```
team assists rebounds
```

```
0 A 5 11
```

```
1 A 7 8
```

```
2 A 7 10
```

```
3 A 9 6
```

```
4 B 12 6
```

```
5 B 9 7
```

```
6 B 9 9
```

```
7 B 4 12
```

```
#check data type of new DataFrame
```

```
type(new_df)
```

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

**Notice that this new DataFrame contains all of the columns from the original DataFrame *except* the points**

**column.**

### **Additional Resources**

**The following tutorials explain how to perform other common tasks in Python:**

ARABPSYCHOLOGY.COM