

What is the difference between Pandas join and merge?

Authored by
stats writer

May 5, 2024

RECOMMENDED CITATION

stats writer (2024). *What is the difference between Pandas join and merge?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=142947>

Pandas join and merge are two methods used for combining data frames in Python's Pandas library. While both methods essentially achieve the same result, there are some key differences between them.

Join is a method for combining two data frames based on their index, which is a unique identifier for each row. It is similar to a SQL join operation where data is matched based on a common key. This means that the resulting data frame will contain only the rows that have matching index values in both data frames.

On the other hand, merge is a more flexible method that allows for combining data frames based on any specified column or columns. It can perform various types of joins such as inner, outer, left, and right merges, depending on the desired output. This method also allows for handling duplicate values and missing data in a more customizable manner.

In summary, join is more limited in its functionality as it is solely based on index matching, while merge offers more options and control over how data is combined from multiple data frames. It is important to understand the differences between these methods in order to choose the most suitable one for a particular data analysis task.

Pandas Join vs. Merge: What's the Difference?

Both the join() and the merge() functions can be used to combine two pandas DataFrames.

Here's the main difference between the two functions:

The join() function combines two DataFrames by index. The merge() function combines two DataFrames by whatever column you specify.

These functions use the following basic syntax:

#use join() to combine two DataFrames by index

df1.join(df2)

#use merge() to combine two DataFrames by specific column name

```
df1.merge(df2, on='column_name')
```

In cases where you know that you want to join two DataFrames by index, the join() function can be used to save some typing.

The following examples show how to use each function in practice.

Example 1: How to Use the join() Function

The following code shows how to use the join() function to combine two DataFrames:

```
import pandas as pd
```

```
#create two DataFrames
```

```
df1 = pd.DataFrame({'name': , 'points':  
}).set_index('name')
```

```
df2 = pd.DataFrame({'name': , 'steals':  
}).set_index('name')
```

```
#view two DataFrames
```

```
print(df1); print(df2)
```

```
points steals
```

```
name name
```

```
A 8 A 4
```

```
B 12 B 5
```

```
C 19 C 2
```

```
#use join() function to join together two DataFrames
```

```
df1.join(df2)
```

```
points steals
```

```
name
```

```
A 8 4
```

```
B 12 5
```

```
C 19 2
```

By default, the join() function joined together the two DataFrames using the index column.

Example 2: How to Use the merge() Function

The following code shows how to use the merge() function to combine two DataFrames:

```
import pandas as pd
```

#create two DataFrames

```
df1 = pd.DataFrame({'name': , 'points':  
}).set_index('name')  
df2 = pd.DataFrame({'name': , 'steals':  
}).set_index('name')
```

#view two DataFrames

```
print(df1); print(df2)
```

```
points steals
```

```
name name
```

```
A 8 A 4
```

```
B 12 B 5
```

```
C 19 C 2
```

#use join() function to join together two DataFrames

```
df1.merge(df2, on='name')
```

```
points steals
```

```
name
```

```
A 8 4
```

```
B 12 5
```

```
C 19 2
```

Notice that the merge() function returned the exact same result, but we had to explicitly tell pandas to join

the DataFrames using the 'name' column.

You can find the complete online documentation for the `join()` and `merge()` functions here:

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

ARABPSYCHOLOGY.COM