

How can a cross join be performed in Pandas, and could you provide an example?”

Authored by
stats writer

June 27, 2024

RECOMMENDED CITATION

stats writer (2024). *How can a cross join be performed in Pandas, and could you provide an example?”*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=155524>

A cross join in Pandas is a type of merge operation that combines two data frames by creating all possible combinations of rows between the two data frames. This can be achieved by using the "cross" method in the Pandas library, which generates a new data frame with a row for every possible combination of rows from the two original data frames. An example of this would be merging a data frame containing customer information with a data frame containing product information, resulting in a new data frame with a row for every possible customer-product combination. This allows for a comprehensive analysis of data from multiple sources.

Perform a Cross Join in Pandas (With Example)

You can use the following basic syntax to perform a cross join in pandas:

```
#create common key
```

```
df1 = 0
```

```
df2 = 0
```

```
#outer merge on common key (e.g. a cross join)
```

```
df1.merge(df2, on='key', how='outer')
```

The following example shows how to use this function in practice.

Example: Perform Cross Join in Pandas

Suppose we have the following two pandas DataFrames:

```
import pandas as pd
```

```
#create first DataFrame
```

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

```
print(df1)
```

```
team points
```

```
0 A 18
```

```
1 B 22
```

```
2 C 19
```

```
3 D 14
```

```
#create second DataFrame
```

```
df2 = pd.DataFrame({'team': ,  
'assists': })
```

```
print(df2)
```

```
team assists
```

```
0 A 4
```

```
1 B 9
```

```
2 F 8
```

The following code shows how to perform a cross join on the two DataFrames:

```
#create common key
```

```
df1 = 0
```

```
df2 = 0
```

```
#perform cross join
```

```
df3 = df1.merge(df2, on='key', how='outer')
```

```
#drop key column
```

```
del df3
```

```
#view results
```

```
print(df3)
```

```
team_x points team_y assists
```

```
0 A 18 A 4
```

```
1 A 18 B 9
```

```
2 A 18 F 8
```

```
3 B 22 A 4
```

```
4 B 22 B 9
```

```
5 B 22 F 8
```

```
6 C 19 A 4
```

```
7 C 19 B 9
```

```
8 C 19 F 8
```

```
9 D 14 A 4
```

```
10 D 14 B 9
```

11 D 14 F 8

The result is one DataFrame that contains every possible combination of rows from each DataFrame.

For example, the first row of the first DataFrame contains team A and 18 points. This row is matched with every single row in the second DataFrame.

Next, the second row of the first DataFrame contains team B and 22 points. This row is also matched with every single row in the second DataFrame.

The end result is a DataFrame with 12 rows.

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