

How can we shuffle the rows of a Pandas DataFrame?

Authored by
stats writer

May 3, 2024

RECOMMENDED CITATION

stats writer (2024). *How can we shuffle the rows of a Pandas DataFrame?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=142253>

Shuffling the rows of a Pandas DataFrame refers to randomly rearranging the order of the rows in the DataFrame. This can be achieved by using the "sample" function with the "frac" parameter set to 1, which will randomly select and return all the rows in the DataFrame. Another option is to use the "shuffle" function, which will shuffle the rows in-place without returning a new DataFrame. These methods can be useful for creating randomized data samples or for implementing machine learning algorithms that require randomized data inputs.

Shuffle Rows in a Pandas DataFrame

You can use the following syntax to randomly shuffle the rows in a pandas DataFrame:

#shuffle entire DataFrame

```
df.sample(frac=1)
```

#shuffle entire DataFrame and reset index

```
df.sample(frac=1).reset_index(drop=True)
```

Here's what each piece of the code does:

The `sample()` function takes a sample of all rows without replacement. The `frac` argument specifies the fraction of rows to return in the sample. A `frac` value of 1 specifies to use all rows. The `reset_index(drop=True)` function specifies to reset the index of the rows.

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

Example 1: Shuffle Entire DataFrame

The following code shows how to shuffle all rows in a pandas DataFrame:

```
import pandas as pd

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

```
#view DataFrame
```

```
df
```

```
team points rebounds
```

```
0 A 77 19
```

```
1 A 82 22
```

```
2 A 86 15
```

```
3 B 88 28
```

```
4 B 80 33
```

```
5 C 95 29
```

```
#shuffle all rows of DataFrame
```

```
df.sample(frac=1)
```

team points rebounds

1 A 82 22

3 B 88 28

2 A 86 15

5 C 95 29

4 B 80 33

0 A 77 19

Notice that the rows are shuffled and each row retained its original index value.

Also note that each time you run this function, the order of the rows will change.

Example 2: Shuffle Entire DataFrame & Reset Index

The following code shows how to shuffle all rows in a pandas DataFrame and reset the index values:

```
import pandas as pd

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

```
#view DataFrame
```

```
df
```

```
team points rebounds
```

```
0 A 77 19
```

```
1 A 82 22
```

```
2 A 86 15
```

```
3 B 88 28
```

```
4 B 80 33
```

```
5 C 95 29
```

```
#shuffle all rows of DataFrame
```

```
df.sample(frac=1).reset_index(drop=True)
```

```
team points rebounds
```

```
0 A 77 19
```

```
1 C 95 29
```

```
2 A 82 22
```

```
3 B 88 28
```

```
4 A 86 15
```

```
5 B 80 33
```

Notice that the rows are shuffled and the index is also reset so that the first row has an index value of 0, the second row has an index value of 1, and so on.