How can I create a Pandas DataFrame with random data?

Authored by

stats writer

June 30, 2024

Creating a Pandas DataFrame with random data involves using the built-in functions of the Pandas library to generate a DataFrame with randomly generated values. This can be achieved by using the "rand" or "randn" functions to generate random values and then passing them into the DataFrame constructor. Additionally, the "numpy" library can also be utilized to generate arrays of random numbers which can then be converted into a DataFrame using the Pandas library. This allows for the creation of a DataFrame with a specified number of rows and columns, filled with randomly generated data for analysis and manipulation.

Create Pandas DataFrame with Random Data

You can use the following basic syntax to create a pandas DataFrame that is filled with random integers:

df = pd.DataFrame(np.random.randint(0,100,size=(10, 3)), columns=list('ABC'))

This particular example creates a DataFrame with 10 rows and 3 columns where each value in the DataFrame is a random integer between 0 and 100.

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

Example 1: Create Pandas DataFrame with Random Data

The following code shows how to create a pandas DataFrame with 10 rows and 3 columns where each value in the DataFrame is a random integer between 0

and 100:

```
import pandas as pd
import numpy as np

#create DataFrame
df  =  pd.DataFrame(np.random.randint(0,100,size=(10, 3)), columns=list('ABC'))

#view DataFrame
print(df)
```

```
A B C
0 72 70 27
1 87 85 7
2 4 42 84
3 85 87 63
4 79 72 30
5 96 99 79
6 26 47 90
7 35 69 56
8 42 47 0
9 97 4 59
```

Note that each time you run this code, the random

integers in the DataFrame will be different.

If you'd like to create a reproducible example where the random integers are the same each time, you can use the following piece of code immediately before you create the DataFrame:

```
np.random.seed(0)
```

Now each time you run the code, the random integers in the DataFrame will be the same.

Example 2: Add Column of Random Data to Existing DataFrame

Suppose we have the following existing pandas DataFrame:

```
import pandas as pd

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

#view DataFrame
```

```
print(df)

team points assists rebounds
0 A 18 5 11
1 B 22 7 8
2 C 19 7 10
3 D 14 9 6
4 E 14 12 6
5 F 11 9 5
6 G 20 9 9
7 H 28 4 12
```

We can use the following code to add a new column called "rand" that contains random integers between 0 and 100:

```
import numpy as np

#add 'rand' column that contains 8 random integers between 0 and 100
df = np.random.randint(0,100,size=(8, 1))

#view updated DataFrame
print(df)
```

team points assists rebounds rand

0 A 18 5 11 47

1 B 22 7 8 64

2 C 19 7 10 82

3 D 14 9 6 99

4 E 14 12 6 88

5 F 11 9 5 49

6 G 20 9 9 29

7 H 28 4 12 19

Notice that the new column "rand" has been added to the existing DataFrame.

Additional Resources

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