

Can you explain how to use the Pandas apply() method with the inplace argument?

Authored by
stats writer

May 6, 2024

RECOMMENDED CITATION

stats writer (2024). *Can you explain how to use the Pandas apply() method with the inplace argument?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=143188>

The Pandas apply() method is a powerful tool for manipulating data in a DataFrame. It allows for the application of a function to each element in a column or row, making it easier to perform complex operations on large datasets. The inplace argument is an optional parameter that can be used with the apply() method to make changes to the original DataFrame. When set to True, the changes will be made directly to the DataFrame without creating a new copy. This can be especially useful when working with large datasets to avoid having to create duplicate copies. Overall, the Pandas apply() method with the inplace argument is a useful tool for efficient and effective data manipulation.

Use Pandas apply() inplace

The pandas function can be used to apply a function across rows or columns of a pandas DataFrame.

This function is different from other functions like drop() and replace() that provide an inplace argument:

```
df.drop(, inplace=True)
```

```
df.rename({'old_column' : 'new_column'}, inplace=True)
```

The apply() function has no inplace argument, so we must use the following syntax to transform a DataFrame inplace:

```
df = df.apply(lambda x: x*2)
```

The following examples show how to use this syntax in

practice with the following pandas DataFrame:

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'points': ,  
'assists': ,  
'rebounds': })
```

```
#view DataFrame
```

```
df
```

```
points assists rebounds
```

```
0 25 5 11
```

```
1 12 7 8
```

```
2 15 7 10
```

```
3 14 9 6
```

```
4 19 12 6
```

```
5 23 9 5
```

```
6 25 9 9
```

```
7 29 4 12
```

Example 1: Use apply() inplace for One Column

The following code shows how to use apply() to transform one data frame column inplace:

```
#multiply all values in 'points' column by 2 inplace  
df.loc = df.points.apply(lambda x: x*2)
```

```
#view updated DataFrame  
df
```

```
points assists rebounds
```

```
0 50 5 11
```

```
1 24 7 8
```

```
2 30 7 10
```

```
3 28 9 6
```

```
4 38 12 6
```

```
5 46 9 5
```

```
6 50 9 9
```

```
7 58 4 12
```

Example 2: Use apply() inplace for Multiple Columns

The following code shows how to use apply() to transform multiple data frame columns inplace:

```
multiply all values in 'points' and 'rebounds' column by  
2 inplace
```

```
df] = df].apply(lambda x: x*2)
```

```
#view updated DataFrame
```

df

points assists rebounds

0 50 5 22

1 24 7 16

2 30 7 20

3 28 9 12

4 38 12 12

5 46 9 10

6 50 9 18

7 58 4 24

Example 3: Use apply() inplace for All Columns

The following code shows how to use apply() to transform all data frame columns inplace:

#multiply values in all columns by 2

df = df.apply(lambda x: x*2)

#view updated DataFrame

df

points assists rebounds

0 50 10 22

1 24 14 16

2 30 14 20

3 28 18 12

4 38 24 12

5 46 18 10

6 50 18 18

7 58 8 24

ARABPSYCHOLOGY.COM