

How can I insert a row at a specific index position in a Pandas dataframe?

Authored by
stats writer

June 27, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I insert a row at a specific index position in a Pandas dataframe?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=155405>

To insert a row at a specific index position in a Pandas dataframe, the "loc" function can be used. This function allows for specifying the index position where the new row should be inserted, along with the necessary data for the new row. Additionally, the "iloc" function can also be used to insert a row at a specific numerical index position. Both of these methods will insert the new row while also shifting any existing rows below it to make space. This provides a convenient and efficient way to add new data to a specific location within a Pandas dataframe.

Pandas: Insert Row at Specific Index Position

You can use the following basic syntax to insert a row into a a specific index position in a pandas DataFrame:

```
#insert row in between index position 2 and 3  
df.loc = value1, value2, value3, value4
```

```
#sort index  
df = df.sort_index().reset_index(drop=True)
```

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

Example: Insert Row at Specific Index Position in Pandas

Suppose we have the following 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 syntax to insert a row in between index position 2 and 3:

```
#insert row in between index position 2 and 3  
df.loc = 'Z', 10, 5, 7
```

```
#sort index  
df = df.sort_index().reset_index(drop=True)
```

```
#view updated DataFrame  
print(df)
```

```
team points assists rebounds
```

```
0 A 18 5 11
```

```
1 B 22 7 8
```

```
2 C 19 7 10
```

```
3 Z 10 5 7
```

```
4 D 14 9 6
```

```
5 E 14 12 6
```

```
6 F 11 9 5
```

```
7 G 20 9 9
```

```
8 H 28 4 12
```

Notice that a row has been inserted in between the previous index position 2 and 3 with the following information:

```
team: Zpoints: 10assists: 5rebounds: 7
```

By using the `sort_index()` and `reset_index()` functions, we were then able to reassign values to the index ranging from 0 to 8.

Note that the new row must contain the same number of

values as the number of existing columns.

For example, if we attempted to insert a new row with only three values, we would receive an error:

```
#attempt to insert row with only three values  
df.loc = 10, 5, 7
```

ValueError: cannot set a row with mismatched columns

We receive a because the number of values in the new row does not match the number of existing columns in the DataFrame.

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