

Fix ValueError: cannot set a row with mismatched columns?

Authored by
stats writer

November 30, 2025

RECOMMENDED CITATION

stats writer (2025). *Fix ValueError: cannot set a row with mismatched columns?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=102081>

Fix ValueError: cannot set a row with mismatched columns error occurs when a single row of data is being added to an existing Pandas dataframe and the number of columns in the dataframe and the data don't match. This can be fixed by ensuring that the data being added to the dataframe has the same number of columns as the original dataframe. Additionally, if there is missing data for certain columns, it should be filled in with NaN or an appropriate value.

One error you may encounter when using pandas is:

ValueError: cannot set a row with mismatched columns

This error occurs when you attempt to add a new row to a pandas DataFrame but the number of values in the new row doesn't match the number of columns in the existing DataFrame.

The following example shows how to fix this error in practice.

How to Reproduce the Error

Suppose we create the following pandas DataFrame:

```
import pandas as pd
```

```
#create DataFrame
```

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

```
#view DataFrame
```

```
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
```

```
8 I 22 8 9
```

Now suppose we try to append a new row to the end of the DataFrame:

#define new row to append

```
new_team =
```

```
#append row to DataFrame
```

```
df.loc = new_team
```

```
#view updated DataFrame
```

```
df
```

ValueError: cannot set a row with mismatched columns

We receive a **ValueError** because the new row we're trying to append only contains two values, but the existing DataFrame has four columns.

How to Fix the Error

The easiest way to fix this error is to use the **append()** function to add the new row to the end of the DataFrame, which will automatically fill in missing values with NaN:

The following syntax shows how to use this function in practice:

#define new row to append

```
new =
```

```
#append row to end of DataFrame
```

```
df = df.append(pd.Series(new, index=df.columns), ignore_index=True)
```

```
#view updated DataFrame
```

```
df
```

```
team points assists rebounds
```

```
0 A 18 5.0 11.0
```

```
1 B 22 7.0 8.0
```

```
2 C 19 7.0 10.0
```

```
3 D 14 9.0 6.0
```

```
4 E 14 12.0 6.0
```

```
5 F 11 9.0 5.0
```

```
6 G 20 9.0 9.0
```

```
7 H 28 4.0 12.0
```

```
8 I 22 8.0 9.0
```

9 J 30 NaN NaN

Notice that we don't receive any **ValueError** and the new row has been appended to the end of the DataFrame.

Also notice that both of the missing values in the new row were simply filled in with NaN values.

ARABPSYCHOLOGY.COM