

How can I drop a column in Pandas only if it exists?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I drop a column in Pandas only if it exists?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=154023>

The process of dropping a column in Pandas can be accomplished by using the "drop" function. However, it is important to first check if the column exists before attempting to drop it. To do so, the "in" operator can be used to check if the column name is present in the data frame. If the column exists, the "drop" function can be used to remove it. If the column does not exist, an error can be handled to avoid any disruptions in the code. This approach ensures that a column is only dropped if it exists in the data frame, preventing any potential errors or issues in the data analysis process.

Pandas: Drop Column if it Exists

You can use the following basic syntax to drop one or more columns in a pandas DataFrame if they exist:

```
df = df.drop(, axis=1, errors='ignore')
```

Note: If you don't use the argument `errors='ignore'` then you'll receive an error if you attempt to drop a column that doesn't exist.

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

Example: Drop Column if it Exists in Pandas

Suppose we have the following pandas DataFrame that contains information about various basketball players:

```
import pandas as pd
```

```
#create DataFrame
```

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

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

```
team points assists minutes all_star  
0 A 18 5 10.1 True  
1 B 22 7 12.0 False  
2 C 19 7 9.0 False  
3 D 14 9 8.0 True  
4 E 14 12 8.4 True  
5 F 11 9 7.5 True
```

Now suppose we attempt to drop the columns with the names `minutes_played` and `points`:

```
#drop minutes_played and points columns  
df = df.drop(, axis=1)
```

KeyError: " not found in axis"

We receive an error because the column `minutes_played` does not exist as a column name in the `DataFrame`.

Instead, we need to use the `drop()` function with the `errors='ignore'` argument:

```
#drop minutes_played and points columns  
df = df.drop(, axis=1, errors='ignore')
```

```
#view updated DataFrameprint(df)
```

```
team assists minutes all_star
```

```
0 A 5 10.1 True
```

```
1 B 7 12.0 False
```

```
2 C 7 9.0 False
```

```
3 D 9 8.0 True
```

```
4 E 12 8.4 True
```

```
5 F 9 7.5 True
```

Notice that the `points` column has been dropped from the `DataFrame`.

Also notice that we don't receive any error even though we attempted to drop a column called `minutes_played`,

which does not exist.

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

ARABPSYCHOLOGY.COM