

Can the `.str` accessor only be used with string values?

Authored by
stats writer

July 2, 2024

RECOMMENDED CITATION

stats writer (2024). *Can the `.str` accessor only be used with string values?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165583>

The .str accessor is a method used to access and manipulate string values in programming languages such as Python. It allows for various string operations to be performed, such as replacing, splitting, and capitalizing characters within a string. However, it is important to note that the .str accessor can only be used with string values. Attempting to use it with other data types, such as integers or lists, will result in an error. Therefore, it is essential to ensure that the data being accessed with the .str accessor is a string in order for it to function properly.

Fix: Can only use .str accessor with string values

One error you may encounter when using Python is:

AttributeError: Can only use .str accessor with string values!

This error usually occurs when you attempt to replace a pattern in a string column of a pandas DataFrame, but the column you're working with isn't actually a string.

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

How to Reproduce the Error

Suppose we have 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 6.5 5 11
```

```
1 A 7.8 7 8
```

```
2 A 8.0 7 10
```

```
3 A 9.0 9 6
```

```
4 B 7.5 12 6
```

```
5 B 3.4 9 5
```

```
6 B 6.6 9 9
```

```
7 B 6.8 4 12
```

Now suppose we attempt to replace every decimal in the "points" column with a blank:

```
#attempt to replace decimal in "points" column with a  
blank
```

```
df = df.str.replace('.', '')
```

AttributeError: Can only use .str accessor with string

values!

We receive an error because the "points" column is not a string column.

How to Fix the Error

The easiest way to get around this error is to use the .astype(str) function before attempting to replace any values in the "points" column:

```
#replace decimal in "points" column with a blank
```

```
df = df.astype(str).str.replace('.', '')
```

```
#view updated DataFrame
```

```
df
```

```
team points assists rebounds
```

```
0 A 65 5 11
```

```
1 A 78 7 8
```

```
2 A 80 7 10
```

```
3 A 90 9 6
```

```
4 B 75 12 6
```

```
5 B 34 9 5
```

```
6 B 66 9 9
```

7 B 68 4 12

Notice that each decimal in the "points" column has been replaced and we don't receive any error since we used the `.astype(str)` function before attempting to replace any values in the "points" column.

Note: You can find the complete documentation for the `replace()` function .

Additional Resources

The following tutorials explain how to fix other common errors in Python: