count\_row = df.shape[0] # Gives number of rows

count\_col = df.shape[1] # Gives number of columns

Or, more succinctly,

r, c = df.shape

y = np.array([0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1])

y.tolist().count(1)

my\_array = np.array([0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1])

sum(1 for val in my\_array if val==0)

Out: 8

This funktion returns the number of occurences of a variable in an array:

def count(array,variable):

number = 0

for i in range(array.shape[0]):

for j in range(array.shape[1]):

if array[i,j] == variable:

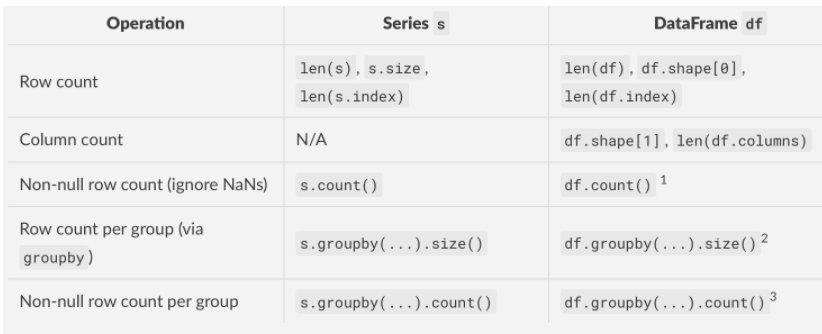
number += 1

return number

len(df.index)

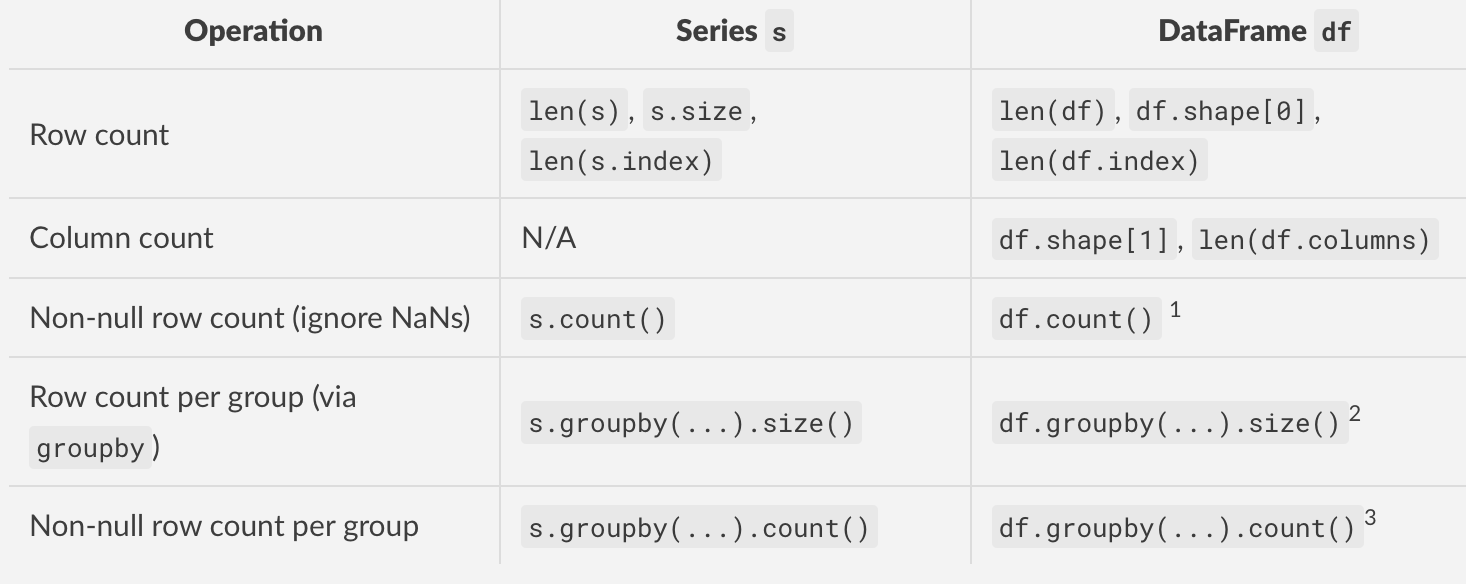
df.shape[0]

df[df.columns[0]].count() (== number of non-NaN values in first column)



**How do I get the row count of a Pandas DataFrame?**

This table summarises the different situations in which you'd want to count something in a DataFrame (or Series, for completeness), along with the recommended method(s).

[](https://i.stack.imgur.com/3FXuI.png)

**Footnotes**

1. DataFrame.count returns counts for each column as a Series since the non-null count varies by column.
2. DataFrameGroupBy.size returns a Series, since all columns in the same group share the same row-count.
3. DataFrameGroupBy.count returns a DataFrame, since the non-null count could differ across columns in the same group. To get the group-wise non-null count for a specific column, use df.groupby(...)['x'].count() where "x" is the column to count.

#**Minimal Code Examples**

Below, I show examples of each of the methods described in the table above. First, the setup -

df = pd.DataFrame({

'A': list('aabbc'), 'B': ['x', 'x', np.nan, 'x', np.nan]})

s = df['B'].copy()

df

A B

0 a x

1 a x

2 b NaN

3 b x

4 c NaN

s

0 x

1 x

2 NaN

3 x

4 NaN

Name: B, dtype: object

https://stackoverflow.com/questions/15943769/how-do-i-get-the-row-count-of-a-pandas-dataframe