

# How can I use the describe() function in Pandas to only calculate the mean and standard deviation of my data?

Authored by  
**stats writer**

June 24, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I use the describe() function in Pandas to only calculate the mean and standard deviation of my data?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151250>

The describe() function in Pandas is a useful tool for obtaining descriptive statistics of a dataset. By default, it calculates various statistics such as count, mean, standard deviation, minimum and maximum values, and quartiles of all the numerical columns in the dataset. However, if you only want to calculate the mean and standard deviation of your data, you can specify the desired statistics by passing the 'include' parameter with a list of the desired statistics, such as . This will limit the output of the describe() function to only the mean and standard deviation of your data, providing a concise summary of these two important measures. This feature of the describe() function allows for customizable and efficient data exploration and analysis.

## **Pandas: Use describe() for Only Mean and Std**

**You can use the describe() function to generate for variables in a pandas DataFrame.**

**By default, the describe() function calculates the following metrics for each numeric variable in a DataFrame:**

**count (number of values) mean (mean value) std (standard deviation) min (minimum value) 25% (25th percentile) 50% (50th percentile) 75% (75th percentile) max (max value)**

**However you can use the following syntax to only calculate the mean and standard deviation for each numeric variable:**

**df.describe().loc]**

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

Example: Use describe() in Pandas to Only Calculate Mean and Std

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': ,
'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**

If we use the describe() function, we can calculate descriptive statistics for each numeric variable in the DataFrame:

**#calculate descriptive statistics for each numeric variable**

**df.describe()**

**points assists rebounds**

**count 8.000000 8.000000 8.000000**

**mean 18.250000 7.750000 8.375000**

**std 5.365232 2.54951 2.559994**

**min 11.000000 4.000000 5.000000**

**25% 14.000000 6.500000 6.000000**

**50% 18.500000 8.000000 8.500000**

**75% 20.500000 9.000000 10.250000**

**max 28.000000 12.000000 12.000000**

However, we can use the following syntax to only calculate the and for each numeric variable:

**#only calculate mean and standard deviation of each**

**numeric variable**

**df.describe().loc]**

**points assists rebounds**

**mean 18.250000 7.75000 8.375000**

**std 5.365232 2.54951 2.559994**

**Notice that the output only includes the mean and standard deviation for each numeric variable.**

**Note that the describe() function still calculated each descriptive statistic as earlier but we used the loc function to select only the rows with the names mean and std in the output.**