

# How can I use the describe() function in Pandas to get specific percentiles for my data?

Authored by  
**stats writer**

June 24, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I use the describe() function in Pandas to get specific percentiles for my data?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151240>

The describe() function in Pandas is a useful tool for obtaining descriptive statistics of a dataset. It provides information such as mean, standard deviation, and quartile values. To obtain specific percentiles for your data using the describe() function, you can specify the desired percentile values in the parameter list. This will allow you to get precise information on the distribution of your data and identify any potential outliers. By using the describe() function in Pandas, you can easily analyze your data and make informed decisions based on the specific percentiles obtained.

## **Pandas: Use describe() with Specific Percentiles**

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

**By default, pandas calculates the 25th, 50th and 75th percentiles for variables.**

**However you can use the percentiles argument within the describe() function to specify the exact percentiles to calculate.**

**The following examples show how to use this argument in practice with the following pandas DataFrame:**

```
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
```

**Example 1: Use describe() with Default Percentiles**

The following code shows how to use the describe() function to 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 8.00000 8.000000
mean 18.250000 7.75000 8.375000
std 5.365232 2.54951 2.559994
min 11.000000 4.00000 5.000000
25% 14.000000 6.50000 6.000000
50% 18.500000 8.00000 8.500000
75% 20.500000 9.00000 10.250000
max 28.000000 12.00000 12.000000
```

Notice that the describe() function calculates the 25th, 50th and 75th percentiles for each variable by default.

Example 2: Use describe() with Custom Percentiles

The following code shows how to use the describe() function with the percentiles argument to calculate the 30th, 60th and 90th percentiles for each numeric variable in the DataFrame:

```
#calculate custom percentiles for each numeric variable
df.describe(percentiles=)
```

```
points assists rebounds
```

```
count 8.000000 8.00000 8.000000
mean 18.250000 7.75000 8.375000
```

```
std 5.365232 2.54951 2.559994
min 11.000000 4.00000 5.000000
30% 14.400000 7.00000 6.200000
50% 18.500000 8.00000 8.500000
60% 19.200000 9.00000 9.200000
90% 23.800000 9.90000 11.300000
max 28.000000 12.00000 12.000000
```

Notice that the describe() function returns the 30th, 60th and 90th percentiles for each numeric variable.

Note: The describe() function also returns the 50th percentile because this represents the median value for each variable and it is one of the default metrics calculated by the describe() function.

Example 3: Use describe() with No Percentiles

The following code shows how to use the describe() function with the argument percentiles= to calculate no percentiles for each numeric variable in the DataFrame:

```
#calculate no percentiles for each numeric variable
```

```
df.describe(percentiles=)
```

```
points assists rebounds
```

```
count 8.000000 8.00000 8.000000  
mean 18.250000 7.75000 8.375000  
std 5.365232 2.54951 2.559994  
min 11.000000 4.00000 5.000000  
50% 18.500000 8.00000 8.500000  
max 28.000000 12.00000 12.000000
```

Note that the 50th percentile is still included in the output because it represents the for each variable.

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