

How can I use the describe() function in Pandas to suppress scientific notation?

Authored by
stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use the describe() function in Pandas to suppress scientific notation?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151235>

The describe() function in Pandas is a useful tool for summarizing and analyzing data. It provides valuable statistical information such as mean, standard deviation, and quartile values. However, when dealing with large numbers, the function may display them in scientific notation, making it difficult to interpret the results. To suppress scientific notation, one can use the set_option() function in Pandas and specify the desired format for displaying numbers. By setting the option to 'float_format' and providing a string with the desired format, the describe() function will display numbers in the specified format instead of scientific notation. This allows for easier interpretation and analysis of the data.

Pandas: Use describe() and Suppress Scientific Notation

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

To suppress scientific notation in the output of the describe() function, you can use the following methods:

Method 1: Suppress Scientific Notation When Using describe() with One Column

```
df.describe().apply(lambda x: format(x, 'f'))
```

Method 2: Suppress Scientific Notation When Using describe() with Multiple Columns

```
df.describe().apply(lambda x: x.apply('{0:.5f}'.format))
```

The following examples show how to use each method in practice with the following pandas DataFrame:

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'store': ,  
'sales': ,  
'returns':})
```

```
#view DataFrame
```

```
print(df)
```

```
store sales returns
```

```
0 A 8450550 2212200
```

```
1 A 406530 145200
```

```
2 A 53000 300
```

```
3 A 6000 2500
```

```
4 B 2000 700
```

```
5 B 4000 600
```

```
6 B 5400 800
```

```
7 B 6500 1200
```

Example 1: Suppress Scientific Notation When Using describe()

with One Column

If we use the describe() function to calculate descriptive statistics for the sales column, the values in the output will be displayed in scientific notation:

```
#calculate descriptive statistics for sales column
```

```
df.describe()
```

```
count 8.000000e+00
```

```
mean 1.116748e+06
```

```
std 2.966552e+06
```

```
min 2.000000e+03
```

```
25% 5.050000e+03
```

```
50% 6.250000e+03
```

```
75% 1.413825e+05
```

```
max 8.450550e+06
```

```
Name: sales, dtype: float64
```

Notice that each of the values in the output are displayed using scientific notation.

We can use the following syntax to suppress scientific notation in the output:

```
#calculate descriptive statistics for sales column and
```

suppress scientific notation

```
df.describe().apply(lambda x: format(x, 'f'))
```

count 8.000000

mean 1116747.500000

std 2966551.594104

min 2000.000000

25% 5050.000000

50% 6250.000000

75% 141382.500000

max 8450550.000000

Name: sales, dtype: object

Notice that the values in the output are now shown without scientific notation.

Example 2: Suppress Scientific Notation When Using describe() with Multiple Columns

If we use the describe() function to calculate descriptive statistics for each numeric column, the values in the output will be displayed in scientific notation:

#calculate descriptive statistics for each numeric column

df.describe()

sales returns

```
count 8.000000e+00 8.000000e+00
mean 1.116748e+06 2.954375e+05
std 2.966552e+06 7.761309e+05
min 2.000000e+03 3.000000e+02
25% 5.050000e+03 6.750000e+02
50% 6.250000e+03 1.000000e+03
75% 1.413825e+05 3.817500e+04
max 8.450550e+06 2.212200e+06
```

Notice that each of the values in the output are displayed using scientific notation.

We can use the following syntax to suppress scientific notation in the output:

```
#calculate descriptive statistics for numeric columns
and suppress scientific notation
df.describe().apply(lambda x: x.apply('{0:,.5f}'.format))
```

sales returns

```
count 8.00000 8.00000
mean 1116747.50000 295437.50000
```

```
std 2966551.59410 776130.93692
min 2000.00000 300.00000
25% 5050.00000 675.00000
50% 6250.00000 1000.00000
75% 141382.50000 38175.00000
max 8450550.00000 2212200.00000
```

Notice that the values in the output are now shown without scientific notation.

Note that in this example we used `0:.5f` to display 5 decimal places in the output.

Feel free to change the 5 to a different number to display a different number of decimal places.

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