How do you use the describe() function in Pandas and what are some examples of its usage?

Authored by

stats writer

May 5, 2024

The describe() function in Pandas is used to generate descriptive statistics for a given data set, such as mean, median, and standard deviation. This function provides a quick overview of the data, allowing users to identify any potential outliers or anomalies. It is particularly useful for exploring numeric data and can be applied to both series and data frames.

To use the describe() function, simply call it on the desired data set, either by itself or as part of a larger data manipulation process. For example, if we have a data frame called "df" containing information on sales figures, we can use the describe() function as follows: df.describe(). This will output a summary table with the descriptive statistics for each column in the data frame.

Some examples of the describe() function's usage include identifying the minimum and maximum values in a data set, determining the mean or median of a column, or checking for any missing values. It can also be used to compare different data sets or subsets within a larger data frame.

In summary, the describe() function is a useful tool for gaining a quick understanding of a data set and can aid in the data exploration and analysis process.

Use describe() Function in Pandas (With Examples)

You can use the describe() function to generate descriptive statistics for a pandas DataFrame.

This function uses the following basic syntax:

df.describe()

The following examples show how to use this syntax in practice with 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 25 5 11

1 A 12 7 8

2 B 15 7 10

3 B 14 9 6

4 B 19 12 6

5 C 23 9 5

6 C 25 9 9

7 C 29 4 12

Example 1: Describe All Numeric Columns

By default, the describe() function only generates descriptive statistics for numeric columns in a pandas DataFrame:

#generate descriptive statistics for all numeric columns

df.describe()

points assists rebounds
count 8.000000 8.00000 8.000000
mean 20.250000 7.75000 8.375000
std 6.158618 2.54951 2.559994
min 12.000000 4.00000 5.000000
25% 14.750000 6.50000 6.000000
50% 21.000000 8.00000 8.500000
75% 25.000000 9.00000 10.250000
max 29.000000 12.00000 12.000000

Descriptive statistics are shown for the three numeric columns in the DataFrame.

Note: If there are missing values in any columns, pandas will automatically exclude these values when calculating the descriptive statistics.

Example 2: Describe All Columns

To calculate descriptive statistics for every column in the DataFrame, we can use the include='all' argument:

#generate descriptive statistics for all columns

df.describe(include='all')

team points assists rebounds
count 8 8.000000 8.00000 8.000000
unique 3 NaN NaN NaN
top B NaN NaN NaN
freq 3 NaN NaN NaN
mean NaN 20.250000 7.75000 8.375000
std NaN 6.158618 2.54951 2.559994
min NaN 12.000000 4.00000 5.000000
25% NaN 14.750000 6.50000 6.000000
50% NaN 21.000000 8.00000 8.500000
75% NaN 25.000000 9.00000 10.250000
max NaN 29.000000 12.00000 12.000000

Example 3: Describe Specific Columns

The following code shows how to calculate descriptive statistics for one specific column in the pandas DataFrame:

#calculate descriptive statistics for 'points' column only
df.describe()

count 8.000000

mean 20.250000

std 6.158618

min 12.000000

25% 14.750000

50% 21.000000

75% 25.000000

max 29.000000

Name: points, dtype: float64

The following code shows how to calculate descriptive statistics for several specific columns:

#calculate descriptive statistics for 'points' and 'assists' columns only

df].describe()

points assists

count 8.000000 8.00000

mean 20.250000 7.75000

std 6.158618 2.54951

min 12.000000 4.00000

25% 14.750000 6.50000

50% 21.000000 8.00000

75% 25.000000 9.00000

max 29.000000 12.00000

You can find the complete documentation for the describe() function .

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