How can I calculate the standard deviation by group in R, and what are some examples of how to do so?

Authored by

stats writer

June 26, 2024

Standard deviation is a statistical measure that indicates how much the data values in a group vary from the mean value of that group. In R, the standard deviation of a group can be calculated using the `sd()` function. This function takes in a vector of data values as input and returns the standard deviation of those values. To calculate the standard deviation of a group in R, the data values of that group must be first organized into a vector. This can be done using the `c()` function.

There are several ways to calculate the standard deviation of a group in R, depending on the type of data and the desired result. For example, if the data is organized into a data frame with multiple columns, the `apply()` function can be used to calculate the standard deviation of each column. Similarly, the `aggregate()` function can be used to calculate the standard deviation of a specific column within a data frame grouped by another column.

In addition, the `tapply()` function can be used to calculate the standard deviation of a group within a data frame based on a categorical variable. This is useful for comparing the standard deviation of different groups within the same data set. Another method is to use the `group_by()` function from the dplyr package, which allows for easy grouping of data and calculation of standard deviation within each group.

Overall, there are various methods to calculate the standard deviation of a group in R, depending on the type of data and the desired outcome. These functions provide a convenient and efficient way to analyze and compare the variability of data within different groups.

Calculate Standard Deviation by Group in R (With Examples)

You can use one of the following methods to calculate the standard deviation by group in R:

Method 1: Use base R

```
aggregate(df$col_to_aggregate,
list(df$col_to_group_by), FUN=sd)
```

Method 2: Use dplyr

```
library(dplyr)

df %>%
group_by(col_to_group_by) %>%
summarise_at(vars(col_to_aggregate), list(name=sd))
```

Method 3: Use data.table

```
library(data.table)

setDT(df)

dt
```

The following examples show how to use each of these methods in practice with the following data frame in R:

```
#create data frame
df <- data.frame(team=rep(c('A', 'B', 'C'), each=6),
points=c(8, 10, 12, 12, 14, 15, 10, 11, 12,
18, 22, 24, 3, 5, 5, 6, 7, 9))

#view data frame
```

df

team points

1 A 8

2 A 10

3 A 12

4 A 12

5 A 14

6 A 15

7 B 10

8 B 11

9 B 12

10 B 18

11 B 22

12 B 24

13 C 3

14 C 5

15 C 5

16 C 6

17 C 7

18 C 9

Method 1: Calculate Standard Deviation by Group Using Base R

The following code shows how to use the aggregate()

function from base R to calculate the standard deviation of points scored by team:

```
#calculate standard deviation of points by team
aggregate(df$points, list(df$team), FUN=sd)

Group.1 x
1 A 2.562551
2 B 6.013873
3 C 2.041241
```

Method 2: Calculate Standard Deviation by Group Using dplyr

The following code shows how to use the group_by() and summarise_at() functions from the dplyr package to calculate the standard deviation of points scored by team:

```
library(dplyr) #calculate standard deviation of points scored by team df %>%
group_by(team) %>%
summarise_at(vars(points), list(name=sd))

# A tibble: 3 x 2
team name
```

1 A 2.56

2 B 6.01

3 C 2.04

Method 3: Calculate Standard Deviation by Group Using data.table

The following code shows how to calculate the standard deviation of points scored by team using functions from the data.table package:

```
library(data.table) #convert data frame to data table
setDT(df)

#calculate standard deviation of points scored by team
df
```

```
   team sd
1:    A 2.562551
2:    B 6.013873
3:    C 2.041241
```

Notice that all three methods return the same results.

Note: If you're working with an extremely large data frame, it's recommended to use the dplyr or data.table

approach since these packages perform much faster than base R.