How can I use R to calculate Z-scores?

Authored by

stats writer

April 17, 2024

R is a programming language and statistical software that can be used to calculate z-scores. Z-scores are a measure of how many standard deviations a data point is above or below the mean. To calculate z-scores in R, you can use the built-in function "scale()", which standardizes the data by subtracting the mean and dividing by the standard deviation. This function can be applied to a specific column or variable in a data set, allowing you to easily obtain z-scores for your data. Additionally, R offers various statistical packages that can also assist in calculating z-scores and interpreting the results. Overall, using R for z-score calculations offers a convenient and efficient way to analyze and understand your data.

Calculate Z-Scores in R

In statistics, a z-score tells us how many standard deviations away a value is from the mean. We use the following formula to calculate a z-score:

$z = (X - \mu) / \sigma$

where:

X is a single raw data valueμ is the population meanσ is the population standard deviation

This tutorial explains how to calculate z-scores for raw data values in R.

Example 1: Find Z-Scores for a Single Vector

The following code shows how to find the z-score for every raw data value in a vector:

```
#create vector of data
data <- c(6, 7, 7, 12, 13, 13, 15, 16, 19, 22)

#find z-score for each data value
z_scores <- (data-mean(data))/sd(data)

#display z-scores
z_scores
```

-1.3228757 -1.1338934 -1.1338934 -0.1889822 0.0000000 0.0000000
0.3779645 0.5669467 1.1338934 1.7008401

Each z-score tells us how many standard deviations away an individual value is from the mean. For example:

The first raw data value of "6" is 1.323 standard deviations *below* the mean.The fifth raw data value of "13" is 0 standard deviations away from the mean, i.e. it is equal to the mean.The last raw data value of "22" is 1.701 standard deviations *above* the mean.

Example 2: Find Z-Scores for a Single Column in a DataFrame

The following code shows how to find the z-score for every raw data value in a single column of a dataframe:

```
#create dataframe
df <- data.frame(assists = c(4, 4, 6, 7, 9, 13),
points = c(24, 29, 13, 15, 19, 22),
rebounds = c(5, 5, 7, 8, 14, 15))

#find z-score for each data value in the 'points' column
z_scores <- (df$points-mean(df$points))/sd(df$points)

#display z-scores
z_scores
```

```
0.6191904 1.4635409 -1.2383807 -0.9006405 -0.2251601
0.2814502
```

Each z-score tells us how many standard deviations away an individual value is from the mean. For example:

The first raw data value of "24" is 0.619 standard deviations *above* the mean.The second raw data value of "29" is 1.464 standard deviations *above* the mean.The third raw data value of "13" is 1.238 standard deviations *below* the mean.

And so on.

Example 3: Find Z-Scores for Every Column in a DataFrame

```
#create dataframe
df <- data.frame(assists = c(4, 4, 6, 7, 9, 13),
points = c(24, 29, 13, 15, 19, 22),
rebounds = c(5, 5, 7, 8, 14, 15))

#find z-scores of each column
sapply(df, function(df) (df-mean(df))/sd(df))

assists points rebounds
-0.92315712 0.6191904 -0.9035079
-0.92315712 1.4635409 -0.9035079
-0.34011052 -1.2383807 -0.4517540
-0.04858722 -0.9006405 -0.2258770
0.53445939 -0.2251601 1.1293849
1.70055260 0.2814502 1.3552619
```

The z-scores for each individual value are shown relative to the column they're in. For example:

The first value of "4" in the first column is 0.923 standard deviations below the mean value of its column.The first value of "24" in the second column is .619 standard deviations above the mean value of its

column.The first value of "9" in the third column is .904 standard deviations *below* the mean value of its column.

And so on.

You can find more R tutorials .