How can data be standardized in R? Provide examples.

Authored by

stats writer

April 21, 2024

Data standardization refers to the process of transforming data to a common scale or format in order to make it easier to compare and analyze. In R, there are various methods and functions that can be used to standardize data.

One method is through scaling, where the data is transformed to have a mean of 0 and a standard deviation of 1. This can be achieved using the scale() function. For example, if we have a vector of numerical data called "vector1", we can standardize it using the following code:

standardized_vector1 <- scale(vector1)

Another method is through normalization, where the data is transformed to a range between 0 and 1. This can be done using the normalize() function. For instance, if we have a data frame called "df" with multiple columns, we can normalize it using the following code:

normalized_df <- normalize(df)

Data standardization can also be achieved through centering, where the data is transformed to have a mean of 0. This can be done using the center() function. For example, if we have a matrix called "matrix1", we can center it using the following code:

centered_matrix1 <- center(matrix1)

In addition to these methods, there are other functions in R that can be used for data standardization, such as standardize(), which allows for custom scaling and centering. Overall, data standardization in R is a useful process for preparing data for analysis and comparison.

Standardize Data in R (With Examples)

To standardize a dataset means to scale all of the values in the dataset such that the mean value is 0 and the standard deviation is 1.

The most common way to do this is by using the z-score standardization, which scales values using the following formula:

(xi - x) / s

where:

xi: The ith value in the datasetx: The sample means: The sample standard deviation

The following examples show how to use the function along with the dplyr package in R to scale one or more variables in a data frame using the z-score standardization.

Standardize a Single Variable

The following code shows how to scale just one variable in a data frame with three variables:

```
library(dplyr)

#make this example reproducible
set.seed(1)

#create original data frame
df <- data.frame(var1= runif(10, 0, 50),
var2= runif(10, 2, 23),
var3= runif(10, 5, 38))
```

#view original data frame

df

var1 var2 var3

1 13.275433 6.325466 35.845273

2 18.606195 5.707692 12.000703

3 28.642668 16.427480 26.505234

4 45.410389 10.066178 9.143318

5 10.084097 18.166670 13.818282

6 44.919484 12.451684 17.741765

7 47.233763 17.069989 5.441881

8 33.039890 22.830028 17.618803

9 31.455702 9.980739 33.699798

10 3.089314 18.326350 16.231517

#scale *var1* to have mean = 0 and standard deviation = 1

df2 <- df %>% mutate_at(c('var1'), ~(scale(.) %>% as.vector))

df2

var1 var2 var3

1 -0.90606801 6.325466 35.845273

2 -0.56830963 5.707692 12.000703

3 0.06760377 16.427480 26.505234

4 1.13001072 10.066178 9.143318

5 -1.10827188 18.166670 13.818282

6 1.09890684 12.451684 17.741765

7 1.24554014 17.069989 5.441881

8 0.34621281 22.830028 17.618803

9 0.24583830 9.980739 33.699798

10 -1.55146305 18.326350 16.231517

Notice that just the first variable was scaled while the other two variables remained the same. We can quickly confirm that the new scaled variable has a mean value of 0 and a standard deviation of 1:

#calculate mean of scaled variable
mean(df2$var1)

-4.18502e-18 #basically zero#calculate standard deviation of scaled variable
sd(df2$var1)

1

Standardize Multiple Variables

The following code shows how to scale several variables in a data frame at once:

```
library(dplyr)

#make this example reproducible
set.seed(1)

#create original data frame
df <- data.frame(var1= runif(10, 0, 50),
var2= runif(10, 2, 23),
var3= runif(10, 5, 38))

#scale var1 and var2 to have mean = 0 and standard deviation = 1
df3 <- df %>% mutate_at(c('var1', 'var2'), ~(scale(.) %>% as.vector))
df3

var1 var2 var3
1 -0.90606801 -1.3045574 35.845273
2 -0.56830963 -1.4133223 12.000703
3 0.06760377 0.4739961 26.505234
4 1.13001072 -0.6459703 9.143318
5 -1.10827188 0.7801967 13.818282
6 1.09890684 -0.2259798 17.741765
7 1.24554014 0.5871157 5.441881
8 0.34621281 1.6012242 17.618803
```

9 0.24583830 -0.6610127 33.699798

10 -1.55146305 0.8083098 16.231517

Standardize All Variables

The following code shows how to scale *all* variables in a data frame using the mutate_all function:

```
library(dplyr)

#make this example reproducible
set.seed(1)

#create original data frame
df <- data.frame(var1= runif(10, 0, 50),
var2= runif(10, 2, 23),
var3= runif(10, 5, 38))

#scale all variables to have mean = 0 and standard deviation = 1
df4 <- df %>% mutate_all(~(scale(.) %>% as.vector))
df4

var1 var2 var3
1 -0.90606801 -1.3045574 1.6819976
2 -0.56830963 -1.4133223 -0.6715858
```

3 0.06760377 0.4739961 0.7600871

4 1.13001072 -0.6459703 -0.9536246

5 -1.10827188 0.7801967 -0.4921813

6 1.09890684 -0.2259798 -0.1049130

7 1.24554014 0.5871157 -1.3189757

8 0.34621281 1.6012242 -0.1170501

9 0.24583830 -0.6610127 1.4702281

10 -1.55146305 0.8083098 -0.2539824

How to Normalize Data in R

How to Calculate Standard Deviation in R

How to Impute Missing Values in R

How to Transform Data in R (Log, Square Root, Cube Root)