How can I remove outliers in R?

Authored by

stats writer

April 18, 2024

The process of removing outliers in R involves identifying and removing any data points that are significantly different from the majority of the data. This can be achieved by using statistical methods such as calculating the mean and standard deviation, or by using visual techniques such as box plots. Once the outliers have been identified, they can be removed from the dataset to improve the accuracy and reliability of the data analysis. This process is commonly used in data preprocessing and is essential in ensuring the validity of statistical analyses in R.

Remove Outliers in R

An outlier is an observation that lies abnormally far away from other values in a dataset. Outliers can be problematic because they can affect the results of an analysis.

This tutorial explains how to identify and remove outliers in R.

How to Identify Outliers in R

Before you can remove outliers, you must first decide on what you consider to be an outlier. There are two common ways to do so:

1. Use the interquartile range.

The interquartile range (IQR) is the difference between the 75th percentile (Q3) and the 25th percentile (Q1) in a dataset. It measures the spread of the middle 50% of

values.

You could define an observation to be an outlier if it is 1.5 times the interquartile range greater than the third quartile (Q3) or 1.5 times the interquartile range less than the first quartile (Q1).

Outliers = Observations > Q3 + 1.5*IQR or < Q1 - 1.5*IQR

2. Use z-scores.

A z-score tells you how many standard deviations a given 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

You could define an observation to be an outlier if it has a z-score less than -3 or greater than 3.

Outliers = Observations with z-scores > 3 or < -3

How to Remove Outliers in R

Once you decide on what you consider to be an outlier, you can then identify and remove them from a dataset. To illustrate how to do so, we'll use the following data frame:

```
#make this example reproducible
set.seed(0)

#create data frame with three columns A', 'B', 'C'
df <- data.frame(A=rnorm(1000, mean=10, sd=3),
B=rnorm(1000, mean=20, sd=3),
C=rnorm(1000, mean=30, sd=3))

#view first six rows of data frame
head(df)
A B C
1 13.78886 19.13945 31.33304
2 9.02130 25.52332 30.03579
3 13.98940 19.52971 29.97216
4 13.81729 15.83059 29.09287
5 11.24392 15.58069 31.47707
6 5.38015 19.79144 28.19184
```

We can then define and remove outliers using the z-score method or the interquartile range method:

Z-score method:

The following code shows how to calculate the z-score of each value in each column in the data frame, then remove rows that have at least one z-score with an absolute value greater than 3:

```
#find absolute value of z-score for each value in each column
z_scores <- as.data.frame(sapply(df, function(df) (abs(df-mean(df))/sd(df))))

#view first six rows of z_scores data frame
head(z_scores)

A B C
1 1.2813403 0.25350805 0.39419878
2 0.3110243 1.80496734 0.05890232
3 1.3483190 0.12766847 0.08112630
4 1.2908343 1.32044506 0.38824414
5 0.4313316 1.40102642 0.44450451
6 1.5271674 0.04327186 0.70295309
```

#only keep rows in dataframe with all z-scores less than absolute value of 3
no_outliers <- z_scores

#view row and column count of new data frame
dim(no_outliers)

994 3

The original data frame had 1,000 rows and 3 columns. The new data frame has 994 rows and 3 columns, which tells us that 6 rows were removed because they had at least one z-score with an absolute value greater than 3 in one of their columns.

Interquartile range method:

In some cases we may only be interested in identifying outliers in one column of a data frame. For example, suppose we only want to remove rows that have an outlier in column 'A' of our data frame.

The following code shows how to remove rows from the data frame that have a value in column 'A' that is 1.5 times the interquartile range greater than the third

quartile (Q3) or 1.5 times the interquartile range less than the first quartile (Q1).

```
#find Q1, Q3, and interquartile range for values in column A
Q1 <- quantile(df$A, .25)
Q3 <- quantile(df$A, .75)
IQR <- IQR(df$A)

#only keep rows in dataframe that have values within 1.5*IQR of Q1 and Q3
no_outliers <- subset(df, df$A> (Q1 - 1.5*IQR) & df$A< (Q3 + 1.5*IQR))

#view row and column count of new data frame
dim(no_outliers)

994 3
```

The original data frame had 1,000 rows and 3 columns. The new data frame has 994 rows and 3 columns, which tells us that 6 rows were removed because they had at least one outlier in column A.

When to Remove Outliers

If one or more outliers are present, you should first verify that they're not a result of a data entry error. Sometimes an individual simply enters the wrong data value when recording data.

If the outlier turns out to be a result of a data entry error, you may decide to assign a new value to it such as the mean or the median of the dataset.

If the value is a true outlier, you may choose to remove it if it will have a significant impact on your overall analysis. Just make sure to mention in your final report or analysis that you removed an outlier.

In this tutorial we used rnorm() to generate vectors of normally distributed random variables given a vector length *n*, a population mean μ and population standard deviation σ. You can read more about this function here.

We also used sapply() to apply a function across each column in a data frame that calculated z-scores. You can read more about that function here.