How can data cleaning be performed in R, with an example?

Authored by

stats writer

June 25, 2024

Data cleaning is a process of identifying and correcting inaccurate, incomplete, or irrelevant data in a dataset to ensure its accuracy and reliability for analysis. In R, data cleaning can be performed using various functions and packages such as "tidyverse" and "dplyr". For example, the "na.omit" function can be used to remove rows with missing values, the "select" function can be used to select specific columns, and the "mutate" function can be used to create new variables. Furthermore, the "gsub" function can be used to replace incorrect values in a dataset. Overall, data cleaning in R involves using a combination of functions and packages to identify and correct any errors or inconsistencies in the data.

Perform Data Cleaning in R (With Example)

Data cleaning refers to the process of transforming into data that is suitable for analysis or model-building.

In most cases, "cleaning" a dataset involves dealing with missing values and duplicated data.

Here are the most common ways to "clean" a dataset in R:

Method 1: Remove Rows with Missing Values

```
library(dplyr)

#remove rows with any missing values
df %>% na.omit()
```

Method 2: Replace Missing Values with Another Value

```
library(dplyr)
library(tidyr)

#replace missing values in each numeric column with median value of column
df %>% mutate(across(where(is.numeric), ~replace_na(., median(., na.rm=TRUE))))
```

Method 3: Remove Duplicate Rows

```
library(dplyr)

df %>% distinct(.keep_all=TRUE)
```

The following examples show how to use each of these methods in practice with the following data frame in R that contains information about various basketball players:

```
#create data frame
df <- data.frame(team=c('A', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I'),
points=c(4, 4, NA, 8, 6, 12, 14, 86, 13, 8),
rebounds=c(9, 9, 7, 6, 8, NA, 9, 14, 12, 11),
assists=c(2, 2, NA, 7, 6, 6, 9, 10, NA, 14))
```

#view data frame
df

team points rebounds assists
1 A 4 9 2
2 A 4 9 2
3 B NA 7 NA
4 C 8 6 7
5 D 6 8 6
6 E 12 NA 6
7 F 14 9 9
8 G 86 14 10
9 H 13 12 NA
10 I 8 11 14

Example 1: Remove Rows with Missing Values

We can use the following syntax to remove rows with missing values in any column:

```
library(dplyr)

#remove rows with missing values
new_df <- df %>% na.omit()

#view new data frame
```

```
new_df

team points rebounds assists
1 A 4 9 2
2 A 4 9 2
4 C 8 6 7
5 D 6 8 6
7 F 14 9 9
8 G 86 14 10
10 I 8 11 14
```

Notice that the new data frame does not contain any rows with missing values.

Example 2: Replace Missing Values with Another Value

We can use the following syntax to replace any missing values with the median value of each column:

```
library(dplyr)
library(tidyr)

#replace missing values in each numeric column with median value of column
new_df <-df %>%
mutate(across(where(is.numeric),~replace_na(.,median(
```

.,na.rm=TRUE))))

#view new data frame
new_df

team points rebounds assists
1 A 4 9 2.0
2 A 4 9 2.0
3 B 8 7 6.5
4 C 8 6 7.0
5 D 6 8 6.0
6 E 12 9 6.0
7 F 14 9 9.0
8 G 86 14 10.0
9 H 13 12 6.5
10 I 8 11 14.0

Notice that the missing values in each numeric column have each been replaced with the median value of the column.

Note that you could also replace median in the formula with mean to instead replace missing values with the mean value of each column.

Note: We also had to load the tidyr package in this example because the drop_na() function comes from this package.

Example 3: Remove Duplicate Rows

We can use the following syntax to replace any missing values with the median value of each column:

```
library(dplyr)

#remove duplicate rows
new_df <- df %>% distinct(.keep_all=TRUE)

#view new data frame
new_df
```

```
  team points rebounds assists
1    A      4       9       2
2    B     NA       7      NA
3    C      8       6       7
4    D      6       8       6
5    E     12      NA       6
6    F     14       9       9
7    G     86      14      10
8    H     13      12      NA
```

9 I 8 11 14

Notice that the second row has been removed from the data frame because each of the values in the second row were duplicates of the values in the first row.

Note: You can find the complete documentation for the dplyr distinct() function .