How can I sort my data in R?

Authored by

stats writer

June 30, 2024

Sorting data in R refers to arranging the values in a data set in a specific order. This can be achieved using the "sort" function in R. The user can specify the variable or column by which the data should be sorted, as well as the order in which it should be sorted (ascending or descending). The "order" function can also be used to retrieve the indices of the sorted data, providing more flexibility in data manipulation. Sorting data is an essential step in data analysis, as it allows for easier identification of patterns and trends within the data.

How can I sort my data in R? | R FAQ

When you think about sorting your data, you would probably first consider using a function called sort. There is a function in R that you can use (called the sort function) to sort your data in either ascending or descending order. The variable by which sort you can be a numeric, string or factor variable. You also have some options on how missing values will be handled: they can be listed first, last or removed. We will show several examples of sorting data in R using the hsb2 data frame. After reading in the data, we will attach it and then list out the first 10 cases. In doing this listing, please note the

argument before
the comma (in the square brackets) refers to the rows, while an argument after
the comma refers to columns.

```
hsb2 <- read.table("https://stats.idre.ucla.edu/wp-content/uploads/2016/02/hsb2-1.csv", header=T, sep=",")

attach(hsb2)

hsb2
```

id female race ses schtyp prog read write math science socst
1 70 0 4 1 1 1 57 52 41 47 57
2 121 1 4 2 1 3 68 59 53 63 61
3 86 0 4 3 1 1 44 33 54 58 31
4 141 0 4 3 1 3 63 44 47 53 56
5 172 0 4 2 1 2 47 52 57 53 61
6 113 0 4 2 1 2 44 52 51 63 61
7 50 0 3 2 1 1 50 59 42 53 61
8 11 0 1 2 1 2 34 46 45 39 36
9 84 0 4 2 1 1 63 57 54 58 51
10 48 0 3 2 1 2 57 55 52 50 51

Let's start by just using the sort function with the variable read, since we can see from the output above that the data set is not sorted on read.

sort(read)

28 31 34 34 34 34 34 34 35 36 36 36 37 37 39 39 39 39 39 39 39 39 41 41 42
42 42 42 42 42 42 42 42 42 42 42 43 43 44 44 44 44 44 44 44 44 44 44 44 44
44 44 45 45 46 47 47 47 47 47 47 47 47 47 47 47 47 47 47 47 47 47 47 47 47
47 47 47 47 47 47 47 48 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50 50
50 52 52 52 52 52 52 52 52 52 52 52 52 52 53 54 55 55 55 55 55 55 55 55 55
55 55 55 55 55 57 57 57 57 57 57 57 57 57 57 57 57 57 57 60 60 60 60 60 60
60 60 60 61 63 63 63 63 63 63 63 63 63 63 63 63 63 63 63 63 65 65 65 65 65
65 65 65 65 66 68 68 68 68 68 68 68 68 68 68 71 71 73 73 73 73 73 76 76

hsb2

id female race ses schtyp prog read write math science socst
1 70 0 4 1 1 1 57 52 41 47 57
2 121 1 4 2 1 3 68 59 53 63 61
3 86 0 4 3 1 1 44 33 54 58 31
4 141 0 4 3 1 3 63 44 47 53 56
5 172 0 4 2 1 2 47 52 57 53 61
6 113 0 4 2 1 2 44 52 51 63 61
7 50 0 3 2 1 1 50 59 42 53 61
8 11 0 1 2 1 2 34 46 45 39 36
9 84 0 4 2 1 1 63 57 54 58 51
10 48 0 3 2 1 2 57 55 52 50 51

Clearly, this is not what we wanted. The data in the data frame are not sorted
based on the value of read. Rather, only the variable read was sorted,
independently of the data frame. Let's try again. This time, we
will use the order function. For most applications, you will want
to use the order function and not the sort function to sort data
in a data frame. For the purposes of example, we will

save the newly
sorted data into a new data frame, so that we can do multiple sorting examples
without rearranging the original data frame. Also, note how ties are
handled: if there is only one variable to be sorted on, the cases with
tied values are left in their original order. As we will see in the next
example, if the data frame is sorted on multiple variables, the values of
later variables will be used to "break the tie".

```
sort1.hsb2 <- hsb2
sort1.hsb2
```

```
id female race ses schtyp prog read write math science socst
141 19 1 1 1 1 1 28 46 43 44 51
79 164 0 4 2 1 3 31 36 46 39 46
8 11 0 1 2 1 2 34 46 45 39 36
26 53 0 3 2 1 3 34 37 46 39 31
64 108 0 4 2 1 1 34 33 41 36 36
88 117 0 4 3 1 3 34 49 39 42 56
```

99 1 1 1 1 1 3 34 44 40 39 41

130 45 1 3 1 1 3 34 35 41 29 26

108 89 1 4 1 1 3 35 35 40 51 33

82 165 0 4 1 1 3 36 49 54 61 36

116 106 1 4 2 1 3 36 44 37 42 41

192 175 1 4 3 2 1 36 57 42 50 41

25 12 0 1 2 1 3 37 44 45 39 46

57 67 0 4 1 1 3 37 37 42 33 32

40 153 0 4 2 1 3 39 31 40 39 51

We can also sort the data frame by more than one variable.

Next, we will sort by read and then by prog.

```
sort2.hsb2 <- hsb2
sort2.hsb2
```

id female race ses schtyp prog read write math science socst

141 19 1 1 1 1 1 28 46 43 44 51

79 164 0 4 2 1 3 31 36 46 39 46

64 108 0 4 2 1 1 34 33 41 36 36

8 11 0 1 2 1 2 34 46 45 39 36

26 53 0 3 2 1 3 34 37 46 39 31

88 117 0 4 3 1 3 34 49 39 42 56

99 1 1 1 1 1 3 34 44 40 39 41

130 45 1 3 1 1 3 34 35 41 29 26

108 89 1 4 1 1 3 35 35 40 51 33

192 175 1 4 3 2 1 36 57 42 50 41

82 165 0 4 1 1 3 36 49 54 61 36

116 106 1 4 2 1 3 36 44 37 42 41

25 12 0 1 2 1 3 37 44 45 39 46

57 67 0 4 1 1 3 37 37 42 33 32

128 28 1 2 2 1 1 39 53 54 50 41

We can also sort in reverse order by using a minus sign (-) in front of the variable that we want sorted in reverse order. In
this example, the data are sorted on prog, and within each category of
prog, the variable read is sorted in reverse order.

sort3.hsb2 <- hsb2

sort3.hsb2

id female race ses schtyp prog read write math science socst

74 157 0 4 2 1 1 68 59 58 74 66

78 123 0 4 3 1 1 68 59 56 63 66

45 62 0 4 3 1 1 65 65 48 63 66

9 84 0 4 2 1 1 63 57 54 58 51

21 167 0 4 2 1 1 63 49 35 66 41

92 149 0 4 1 1 1 63 49 49 66 46

36 144 0 4 3 1 1 60 65 58 61 66

121 35 1 1 1 2 1 60 54 50 50 51

1 70 0 4 1 1 1 57 52 41 47 57

18 195 0 4 2 2 1 57 57 60 58 56

155 71 1 4 2 1 1 57 62 56 58 66

198 187 1 4 2 2 1 57 41 57 55 52

20 85 0 4 2 1 1 55 39 57 53 46

46 169 0 4 1 1 1 55 59 63 69 46

196 31 1 2 2 2 1 55 59 52 42 56

Now let's consider the various ways that missing data can be handled. We do not have any
missing data in our data file, so we will create some. Remember that
missing data are denoted as NA in R, regardless of the type of variable (even in numeric variables). In our example, we will create missing values for the variable science for
cases 2 through 5. Also note that we need to indicate

the data set in
which the variable science is to be found, followed by the dollar sign
($), even though we have attached the hsb2 data set.

```
hsb2$science <- NA

hsb2
```

```
id female race ses schtyp prog read write math science socst
1 70 0 4 1 1 1 57 52 41 47 57
2 121 1 4 2 1 3 68 59 53 NA 61
3 86 0 4 3 1 1 44 33 54 NA 31
4 141 0 4 3 1 3 63 44 47 NA 56
5 172 0 4 2 1 2 47 52 57 NA 61
6 113 0 4 2 1 2 44 52 51 63 61
7 50 0 3 2 1 1 50 59 42 53 61
8 11 0 1 2 1 2 34 46 45 39 36
9 84 0 4 2 1 1 63 57 54 58 51
10 48 0 3 2 1 2 57 55 52 50 51
```

Now that we have created some missing data, we can sort the data frame such that the missing
data are at the top, the bottom, or deleted from the

frame. Note that
once you create the missing values with the hsb2$science notation, you need to
use exactly that in the order function. Also, the argument after
na.last = has to be in upper case letters (TRUE, FALSE and NA). In the following
examples, we will use the head and tail functions to display the
first few and last few cases of the data frame, respectively, to show the
effect of the argument of na.last = .

```
sort5.hsb2 <- hsb2


head(sort5.hsb2)


id female race ses schtyp prog read write math science socst
2 121 1 4 2 1 3 68 59 53 NA 61
3 86 0 4 3 1 1 44 33 54 NA 31
4 141 0 4 3 1 3 63 44 47 NA 56
5 172 0 4 2 1 2 47 52 57 NA 61
56 15 0 1 3 1 3 39 39 44 26 42
```

130 45 1 3 1 1 3 34 35 41 29 26

sort6.hsb2 <- hsb2

tail(sort6.hsb2)

id female race ses schtyp prog read write math science socst
34 150 0 4 2 1 3 42 41 57 72 31
74 157 0 4 2 1 1 68 59 58 74 66
2 121 1 4 2 1 3 68 59 53 NA 61
3 86 0 4 3 1 1 44 33 54 NA 31
4 141 0 4 3 1 3 63 44 47 NA 56
5 172 0 4 2 1 2 47 52 57 NA 61

For this last example of removing the missing cases, we will create many missing values for
the variable id, making it easier to see that the cases with missing
values are not in the new data frame.

hsb2$id <- NA

sort8.hsb2 <- hsb2

sort8.hsb2

id female race ses schtyp prog read write math science socst

196 31 1 2 2 2 1 55 59 52 42 56

1 70 0 4 1 1 1 57 52 41 47 57

199 118 1 4 2 1 1 55 62 58 58 61

200 137 1 4 3 1 2 63 65 65 53 61

197 145 1 4 2 1 3 42 46 38 36 46

198 187 1 4 2 2 1 57 41 57 55 52

detach(hsb2)