

Introduction to Sorting Data with R

Revolution Analytics





1 Sort and Ordering operations

2 Using `which()`

3 Review





Overview

In this lab we'll cover data processing and manipulation. The objectives are:

- Sort rows of dataframes according to column values
- Use conditional criteria to modify values of dataframes and matrices





Outline

1 Sort and Ordering operations

2 Using `which()`

3 Review





Sorting and Ordering Operations

Two distinct functions

`sort()` sort a vector

`order()` Returns the *element order* that results in a sorted vector



sort()

```
(test <- sample(letters, 10))
```

```
## [1] "g" "j" "n" "u" "e" "s" "w" "m" "l" "b"
```

```
sort(test)
```

```
## [1] "b" "e" "g" "j" "l" "m" "n" "s" "u" "w"
```





order()

```
order(test)
```

```
## [1] 10 5 1 2 9 8 3 6 4 7
```

```
test[order(test)]
```

```
## [1] "b" "e" "g" "j" "l" "m" "n" "s" "u" "w"
```





Sorting direction

Reverse the sorting direction by using `decreasing` in either function

```
sort(test, decreasing = TRUE)
```

```
## [1] "w" "u" "s" "n" "m" "l" "j" "g" "e" "b"
```

```
order(test, decreasing = TRUE)
```

```
## [1] 7 4 6 3 8 9 2 1 5 10
```




Application of order()

- Useful for sorting something other than the values you want to sort by
- application to dataframes
- Useful when you want to break ties





Using `order()` to sort a data frame

```
mtcars[order(mtcars$mpg, decreasing = TRUE), ]
```

```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Toyota Corolla 33.9  4  71.1  65 4.22 1.835 19.90 1  1   4    1
## Fiat 128       32.4  4  78.7  66 4.08 2.200 19.47 1  1   4    1
## Honda Civic    30.4  4  75.7  52 4.93 1.615 18.52 1  1   4    2
## Lotus Europa   30.4  4  95.1 113 3.77 1.513 16.90 1  1   5    2
## Fiat X1-9      27.3  4  79.0  66 4.08 1.935 18.90 1  1   4    1
## Porsche 914-2  26.0  4 120.3  91 4.43 2.140 16.70 0  1   5    2
## ...
```



Exercise 1

- Sort the mtcars by from the highest horsepower car to the lowest horsepower car.





Tie breaks

`order()` can take multiple variables

It will break ties in earlier variables by later variables

```
mtcars[order(mtcars$mpg, mtcars$hp, decreasing = TRUE), ]
```

```
##           mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Fiat 128        32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Fiat X1-9      27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
... 
```



Exercise 2

- Sort the mtcars data set by automatic vs. manual transmission (am), and within each transmission type, sort from the most efficient to the least.
- ADVANCED: How could you have one variable sorted in a descending order, but another variable sorted in an ascending order?





Outline

1 Sort and Ordering operations

2 Using `which()`

3 Review





The function `which()`

Use `which()` to identify values in a vector or array that satisfy a list of criteria, e.g. which car in `mtcars` has the highest horse power:

```
which(mtcars$hp == max(mtcars$hp))
```

```
## [1] 31
```



Uses of which()

We can use this piece of information to extract the name of the car:

```
rownames(mtcars)[which(mtcars$hp == max(mtcars$hp))]
```

```
## [1] "Maserati Bora"
```

Or all the information about the car:

```
mtcars[which(mtcars$hp == max(mtcars$hp)), ]
```

```
##           mpg  cyl  disp  hp  drat    wt  qsec vs  am  gear  carb
## Maserati Bora  15    8   301 335  3.54  3.57 14.6  0   1    5     8
```





Exercise 3

- Which car in `mtcars` has the lowest hp? Hint: use `min()` to find the minimum value.





Exercise 4

Order and identify some data

- Order `mtcars` by the columns `am`, `gear`, and `carb`.
 - Explore what happens when you reorder `mtcars` by the same variables, but in a different order (try `gear`, `am`, and `carb`).
 - How does changing the order of their inclusion change the output?
- Which rows in `mtcars` have `hp > 100` but fewer than 6 cylinders (`cyl`)?





which() and NA values

- Can index with a logical vector
- Slightly different behavior than using `which()`

```
vec
```

```
## [1] 1 2 3 4 NA NA 7 8 NA 10 NA 12 13 14 15 16 NA 18 19 20 21 22 23  
## [24] NA 25 26
```

```
vec%%2 == 0
```

```
## [1] FALSE TRUE FALSE TRUE NA NA FALSE TRUE NA TRUE NA  
## [12] TRUE FALSE TRUE FALSE TRUE NA TRUE FALSE TRUE FALSE TRUE  
## [23] FALSE NA FALSE TRUE
```

```
which(vec%%2 == 0)
```

```
## [1] 2 4 8 10 12 14 16 18 20 22 26
```



which() vs. logical

```
letters[vec%%2 == 0]
```

```
## [1] "b" "d" NA NA "h" NA "j" NA "l" "n" "p" NA "r" "t" "v" NA "z"
```

```
letters[which(vec%%2 == 0)]
```

```
## [1] "b" "d" "h" "j" "l" "n" "p" "r" "t" "v" "z"
```

`which()` removes NA values in the logical vector: it only returns the indices where the logical vector is TRUE



Using which() to assign data

We've used `which()` up to this point to select data out of a larger data set.

Another use is to set values conditionally.

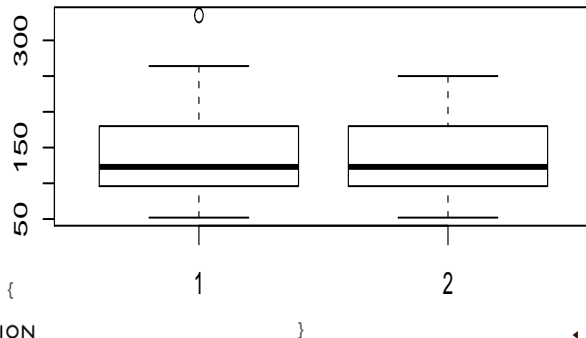




Example:

- 250 hp should be more than enough for anyone!
- So, let's treat all cars with > 250 hp as having only 250 hp

```
mtcars2 <- mtcars  
mtcars2$hp[mtcars2$hp > 250] <- 250
```





Exercise 5

Your turn:

- Store cars in a cars1 dataframe, replace all distance values greater than 40 with a value of 40





Extracting Columns rather than rows

- Prior examples: `which()` to select and assign *rows* in a dataset
- We can also use `which()` to select and assign *columns* in a dataset
- Create a new data set that is just `hp` and `mpg` from the `mtcars` data set.

```
head(mtcars)
```

```
##           mpg cyl  disp  hp  drat    wt  qsec vs  am  gear  carb
## Mazda RX4      21.0   6  160 110  3.90  2.620 16.46  0   1    4    4
## Mazda RX4 Wag  21.0   6  160 110  3.90  2.875 17.02  0   1    4    4
## Datsun 710      22.8   4  108  93  3.85  2.320 18.61  1   1    4    1
## Hornet 4 Drive  21.4   6  258 110  3.08  3.215 19.44  1   0    3    1
## Hornet Sportabout 18.7   8  360 175  3.15  3.440 17.02  0   0    3    2
## Valiant         18.1   6  225 105  2.76  3.460 20.22  1   0    3    1
```




Extracting Columns rather than rows

```
mtcars2 <- mtcars[, which(names(mtcars) %in% c("hp", "mpg"))]  
head(mtcars2)
```

```
##           mpg    hp  
## Mazda RX4      21.0 110  
## Mazda RX4 Wag  21.0 110  
## Datsun 710      22.8  93  
## Hornet 4 Drive  21.4 110  
## Hornet Sportabout 18.7 175  
## Valiant        18.1 105
```





Can select both rows and columns

- Extract hp and mpg for all automatic transmission cars (am == 0).

```
mtcars[which(mtcars$am == 0), which(names(mtcars) %in% c("hp", "mpg"))]
```

```
##                mpg  hp
## Hornet 4 Drive  21.4 110
## Hornet Sportabout 18.7 175
## Valiant        18.1 105
## Duster 360     14.3 245
## Merc 240D      24.4  62
## Merc 230       22.8  95
... 
```





Exercise 6

- Extract mpg, wt and disp for all manual transmission cars in the mtcars data set.





Outline

1 Sort and Ordering operations

2 Using `which()`

3 Review





Review

- Sorting with `sort()` and `order()`
- Selecting and assigning subsets of a dataset



Thank you

Revolution Analytics is the leading commercial provider of software and support for the popular open source R statistics language.

www.revolutionanalytics.com

1.855.GET.REVO

Twitter: @RevolutionR

