R Stats Bootcamp

1.6 - Data subsetting

Megan Lewis

2024-12-11

R stats bootcamp - Module 1

Schedule:

- ~~Session 1: An introduction and script workflow~~
- ~~Session 2: R language~~
- ~~Session 3: R functions~~
- ~~Session 4: Data objects~~
- ~~Session 5: Data frames~~
- **Session 6: Data sub setting**

1.0 Sub-setting and Manipulation

Command data and you are powerful

With a good basic set of moves for sub-setting and manipulating data, you can overpower any data set no matter how large and powerful they may be. Then you will have a strong data Sumo.

HARUG R Stats Bootcamp by Ed Harris

1.1 Session 6 objectives:

- Indexing concept
- Using `which()` and sub-setting
- Selection on data.frame objects
- using `aggregate()`
- Practice exercises

2.0 Indexing concept

Indexing

If you would like to slice and dice your data, you will need to learn about indexing!

2.0 Indexing concept

- Basics of indexing is simple in R
- Data storage objects like:
  - Vectors
  - Matrices
  - Arrays
- Accessed via “addresses”

2.1 How indexing works

- Numeric vector called `my_vector` with 10 values
- Index values will be 1 to 10.

```
my_vector <- c(11.3, 11.2, 10.4, 10.4, 8.7,
               10.8, 10.5, 10.3, 9.7, 11.2)

my_vector
```

```
[1] 11.3 11.2 10.4 10.4  8.7 10.8 10.5 10.3  9.7 11.2
```

2.1 How indexing works

```
1  my_vector <- c(11.3, 11.2, 10.4, 10.4, 8.7,
2                 10.8, 10.5, 10.3, 9.7, 11.2)
3
4  my_vector
```

```
[1] 11.3 11.2 10.4 10.4  8.7 10.8 10.5 10.3  9.7 11.2
```

- Notice the `[1]` in the R console output?
- Indicates the index of the value right next to it

2.1 How indexing works

- If we could see actual index values…

```
> my_vector
 [1] 11.3 11.2 10.4 10.4  8.7 10.8 10.5 10.3  9.7 11.2
       ^    ^    ^    ^    ^    ^    ^    ^    ^    ^
      [1]  [2]  [3]  [4]  [5]  [6]  [7]  [8]  [9] [10]
```

2.2 Vectors

- Can create vector subsets by manipulating the index.
- Indexes of one dimension
- For example, `my_vector[1:i]`
  - `i` is length of vector

Demo

HARUG R Stats Bootcamp by Ed Harris

2.3 Matrices

- Two dimensions instead of one
- For example, `my_matrix[1:i, 1:j]`
  - `i` is number of rows
    - `j` is number of columns

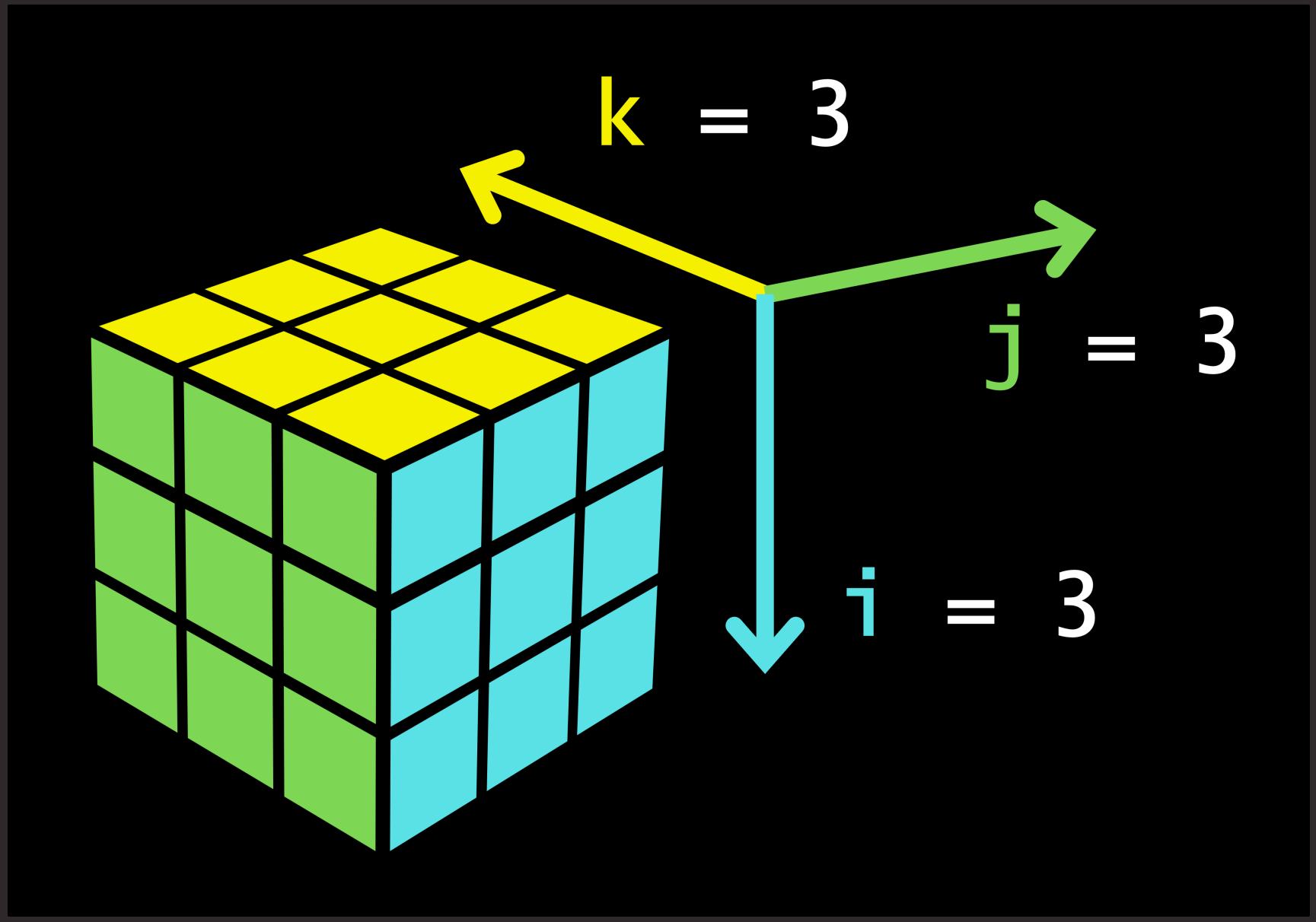
Demo

HARUG R Stats Bootcamp by Ed Harris

2.4 Arrays

- Data objects with more than 2 dimensions
- For example, `my_array[1:i, 1:j, 1:k]`
  - `i` is number of rows
  - `j` is number of columns
  - `k` is “depth” of `i * j`

2.4 Arrays



Demo

HARUG R Stats Bootcamp by Ed Harris

3.0 `which()` and sub-setting

- Sub-setting exploits index system
- Specifying index values explicitly
- By constructing queries that choose subset based on particular data values
- `which()` function is powerful tool here

Demo

HARUG R Stats Bootcamp by Ed Harris

4.0 Selection on `data.frame` objects

Data frames

Data frames are the ultimate data object for getting, storing, organizing and analyzing data. A good scientist must learn to communicate the subtlety of data. A good statistician must learn not to underestimate the subtlety of data. A good student must learn that subtlety may exist, even in simple data.

4.0 Selection on `data.frame` objects

- Vectors, matrices and arrays can only store one type of data
- Data frames can store several data types
- Though each column must be same data type

4.0 Selection on `data.frame` objects

- Few ways to think about selecting values in a data frame
  - Accessing values through variable names
    - Use either the `data_frame_name$variable_name` syntax or the `data_frame_name[ , ]` syntax
  - Access values by selecting particular rows of a data frame
    - Using `which()` function, the `[ , ]` syntax and Boolean phrases

4.1 `OrchidSprays` data

- Classic data set based on an experiment looking at how chemical additives could be used to deter honeybees from being attracted to crops a subsequently killed by pesticides.

4.1 `OrchidSprays` data

- Involves a `treatment` consisting of “lime sulfur emulsion” in increasing concentrations to a sucrose solution.
- `treatment` variable had 8 levels:

```
-   A: the highest amount of sulfur
-   B: Second highest
-   C: ...
-   D: ...
```

```
-   E: ...
-   F: ...
-   G: Lowest amount of sulfur
-   H: Control - no sulfur
```

4.1 `OrchidSprays` data

- The `decrease` variable was a measure of the amount of sucrose solution taken by honeybees
- Prediction: The higher the concentration of sulfur (deterrent), the lower the decrease in sucrose solution.

4.1 `OrchidSprays` data

- Experiment was a Latin Square design.
- Eight treatments arranged randomly in an array of eight columns
  - To randomize any effect of the treatment *order* or *position* on response variable.
- Response measured after placing 100 honeybees into an experimental chamber with the 64 containers of sucrose solution.

Demo

HARUG R Stats Bootcamp by Ed Harris

4.2 Practice selecting parts of a data frame

- Selecting particular parts of a data frame based on the values of one variable is a common and extremely useful task.

Demo

HARUG R Stats Bootcamp by Ed Harris

4.3 Selection based on more than one variable value

- Using basic building blocks of Boolean selection, more complex rules for selecting data can be made.

Demo

HARUG R Stats Bootcamp by Ed Harris

5.0 `aggregate()` function

- Useful and convenient tool to summarize parts of a data set according to some index of variable values.

Demo

HARUG R Stats Bootcamp by Ed Harris

6 Practice exercises

For the following exercises, use the `trees` data set built into R, which has `Girth`, `Height` and `Volume` variables for 31 Black Cherry trees.

- Examine the data
  - `help(trees)`
  - `data(trees)`
  - `str(trees)`

Practice exercise 01

- Show code to calculate the mean `Girth` of Black Cherry trees with a height less than 75 ft.

Practice exercise 02

- Use `help(cut)` and then use the `cut()` function to create a new factor variable based on the `Height` numeric variable in the `trees` data set.
- Try setting the `breaks` argument to 2 or 3.
- Rename the levels of your new factor to something meaningful

Practice exercise 03

- Using the new factor from question 2, use `aggregate()` to calculate the mean and standard deviation of all three variables in the `trees` data set.
- Show code and report results to 2 decimal points of accuracy

Practice exercise 04

- Show the code using `which()` and Boolean phrases as appropriate o find the rows in the `trees` data set where `Girth` is higher than 11 and `Height` is higher than 75.

Practice exercise 05

- Run the following code:

- Use `aggregate()` to calculate the mean `volume` for each `population`
- Hint: You many need to use help for the functions involved and pay close attention to your data frame…)

Practice exercise 06

- Write a plausible practice question involving any aspect of using `which()`, Boolean phrases and/or `aggregate()` involving the in-built R data set `iris`.