Up-to-Date-tRicks

Yaniv

2019-01-19

Contents

Idea

I will continue updating this throughout the course. The goal here is a quick cheatsheet for the `R` we have learned or are learning.

NOTE 1: From now, this will grow when

1. You provide a usable example OR
2. You ask a question about a function that you want clarified.

NOTE 2: You should Also aim to build the skills if googling / finding answers for how to do things in R and how to use `help` files. After all, you don't want to limit yourself to knowing only what I taught. We will work on those skills together.

NOTE 3: There are numerous excellent R cheat sheets.

- The RStudio environment https://github.com/rstudio/cheatsheets/raw/master/rstudio-ide.pdf
- Transformation w dplyr https://github.com/rstudio/cheatsheets/raw/master/data-transformation.pdf
- Visualization w ggplot2 https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf
- Using RMarkdown https://github.com/rstudio/cheatsheets/raw/master/rmarkdown-2.0.pdf
- Importing data https://github.com/rstudio/cheatsheets/raw/master/data-import.pdf

Intro to R / Fundamentals

This is associated with work for 1/23 and the introduction to `R` in datacamp https://www.datacamp.com/courses/free-introduction-to-r

Commenting

```
# Commenting. Use the hash aka # to comment.
# This means R will not try to run these words.
```

Arithmetic

```
1 + 1        # Addition ..................
2 * 2        # Multiplication .............
```
yields: 2

yields: 4

```
3 ^ 3        # Powers ..................... yields: 27
sqrt(4)      # Square roots ............... yields: 2
5 - 5        # Subtraction ................ yields: 0
6 / 6        # Division ................... yields: 1
factorial(7) # Factorial ................... yields: 5040
4 %% 8       # Modulo [example 1] ......... yields: 4
8 %% 4       # Modulo [example 4] ......... yields: 0
```

Assignment

```
# To keep values in R's head, we need to assign them
# Assignments should show up in RStudio's Environment tab.
# Or you can see all assignments with the ls function
x <- 2       # Assign "x" the value 2
y <- 2
```

```
x * 4        # Multiply x by 4 ............ yields: 8
ls()         # Show whats in Rs head ...... yields: x, y [in this case]
```

```
z <- "a"
```

```
ls()         # Show whats in Rs head ...... yields: x, y, z [in this case]
```

Data types

- Numerics (numbers)
- Logical (TRUE / FALSE)
- Character (words)

- Factor (words etc that have been stored as integers [this can be convenient or frustrating]).

Check the classes of different `R` objects:

```
class(1) ................................ yields: numeric
class(1 > 2) ............................ yields: logical
class("a") .............................. yields: character
class(factor("a")) ...................... yields: factor
```

Logical operators

We can ask logical questions, these can be quite useful (for example when we hope to subset our data)

```
x * 4        # Multiply x by 4? ........... yields: 8
1   == 2     # Does one equal two? ........ yields: FALSE
"y" == "x"   # Does "y" equal "x"? ........ yields: FALSE
y   == x     # Does y equal x? ............ yields: TRUE
1   >  2     # Is one greater than two? ... yields: FALSE
1   <  2     # Is one less than two? ...... yields: TRUE
1   >= 1     # Is one > than or = to one? . yields: TRUE
1   <= 1     # Is one < than or = to one? . yields: TRUE
1:4 %in% 4:5 # for each value in the vector
1, 2, 3, 4, which are in the vector 4, 5? . yields: FALSE, FALSE, FALSE, TRUE
```

Data structures

Vectors, Matrices, Arrays, Data frames, Tibbles and Lists
We will focus on vectors and data frames (tibbles are a special kind of data frame). We may come back to lists later. We will almost never use matrices of arrays.

```
a   <- c(1, 8, 5, 3) # Assign a to a vector w. elements one, eight,  five, and three
b   <- c(john = 1, jen = 8)
```

```
a[3]          # Find the 3rd element in a . yields: 5
b["jen"]      # Find jen's value in b ..... yields: 8
```