A Gentle Introduction to R

Jonathan Whiteley

2023-08-06

Pop Quiz

We will review these at the end, so you can see how much you have learned.

- What does 'CRAN' stand for?
- Why is it named 'R'?
- How can you use R interactively?
- How do you find out what a function does & how to use it?
- How do you store values to re-use later?
- True or False: Warnings can be ignored, but an Error means I made a mistake.
- True or False: Error messages will tell me how to fix the problem.

Learning Objectives

- Get familiar with the R interface
- Enter commands
 - ▶ input & output: using R interactively
 - use some common functions
- Understand Errors & Warnings
- Use technical terms for R concepts
- How to get Help

Why is it named \mathbb{R} ?

- R started as an open-source implementation of the S statistical computing language (S-PLUS)
 - S was created at Bell Laboratories in 1976
 - R was based on the S syntax (mostly v3), but works very differently "under the hood".
- R was created by Ross Ihaka and Robert Gentleman at the University of Aukland in the early 1990s.

A quick tour of 'base R'



A quick tour of RStudio



- ullet R has a slightly different interface for each $oldsymbol{O}$ perating $oldsymbol{S}$ ystem (OS)
 - ▶ GUI = Graphical User Interface
- In every case, you interact with R primarily using a command line
 - aka "Question-and-Answer Model"
 - You ask R to do something (a command), and R tells you the answer (result).
 - ▶ Instructions are given to R using the R language.

The R command-line

The command prompt normally looks like this:

>

- ▶ This is R's way of saying "I am ready to accept new commands".
- ▶ Type a new command on the line after this prompt (i.e., input).
- Press return/enter to run the current command
- If you can still edit the command next to the prompt, then it has not been submitted to R to execute (it is still waiting for input).
- If the last prompt is not empty (i.e., there is text beside it)
 and you cannot edit what is beside the prompt,
 it means R is still running the last command and is not ready to accept
 a new command yet.
 - Wait for a new empty prompt to appear before entering the next command.

The R command-line (cont'd)

• If the prompt looks like this:

+

it means the last command was incomplete and R is waiting for more input.

R will not do anything until the command is completed or cancelled.

- This usually means you forgot a closing quote ", parenthesis (, bracket [, or brace {
- You can cancel the current command at any time by pressing escape (esc)



In this presentation,

• commands that can be entered in the command-line look like this:

Input (commands)

Expected output (results) look like this:

Output (results)



demo(graphics)

• some plots and graphs that can be made with R

demo(image)

 \bullet image-like graphics and maps that can be produced with ${\tt R}$

demo(lm.glm)

ullet a demonstration of linear modelling & GLMs

demo()

a list of available demos

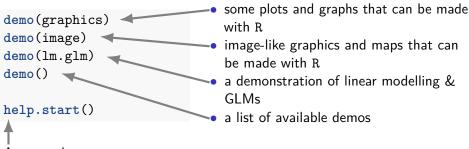
help.start()

← A great place to start, especially if you are comfortable reading documentation for a programming language. More on this later.

Note

R will not only show the output, but also the code used to produce it.





A great place to start, especially if you are comfortable reading documentation for a programming language. More on this later.

Note

R will not only show the output, but also the code used to produce it.

R is a calculator

- These are *expressions*
- Expressions are evaluated, and the value (result) is returned (sometimes invisibly)



- With the cursor next to the empty prompt (>),
 use the up & down arrow keys (↑↓) to re-produce previous commands
- This lets you "scroll through your command history"
- Press up (↑) once, and you get the last command you entered without having to copy & paste

Symbolic variables

• You can store values (*objects*) in symbolic variables (*names*) using an assignment operator

```
assign the value on the right to the name on the left
```

Names can include:

```
letters a-z A-Z numbers 0-9 periods . underscores _
```

```
A <- 10

B <- 10 * 10

A_log <- log(A)

B.seq <- 1:B

assign('x', 3)
```

 Names should begin with a letter

Retrieve values

When a variable name is evaluated, it returns the stored value.

A							В							
## [[1] 10						##	[1]	100					
A_log														
## [1] 2.303								[1]	3					
B.seq														
шш	[4]	4	0	2	4	_	_	7	0	0	10	4.4	10	10
##	[1]	1	2	3	4	5	6	7	8	9	10	11	12	13
##	[19]	19	20	21	22	23	24	25	26	27	28	29	30	31
##	[37]	37	38	39	40	41	42	43	44	45	46	47	48	49
##	[55]	55	56	57	58	59	60	61	62	63	64	65	66	67
##	[73]	73	74	75	76	77	78	79	80	81	82	83	84	85
##	[91]	91	92	93	94	95	96	97	98	99	100			

Built-in variables

Some words and letters already have values in R and should **never be used as variable names**.

[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "I ## [20] "T" "U" "V" "W" "X" "Y" "Z"

Reserved words

Some words and letters already have special meaning in the R language (*keywords*) and should **never be used as variable names**.

References & More Information help.start()

Available from the above screen:

- An Introduction to R
- The R Language Definition

Online:

- RStudio Education (education.rstudio.com)
 - tutorials, workshop materials, and other resources.
- R Manuals (https://cran.r-project.org/manuals.html)
- R Contributed Documentation
 - e.g., http://cran.r-project.org/doc/contrib/usingR.pdf
- Internet search
 - Stack Overflow (stackoverflow.com)