Week04 Lab: R Language Basics

Hannah Kim

4/25/23

Simple Calculations

Let's practice basic arithmetic functions and use R as a calculator:

```
5+3
```

```
[1] 8
```

```
5-3
```

```
[1] 2
```

```
5*3
```

```
[1] 15
```

```
5/3
```

```
[1] 1.666667
```

Saving Your Answers

Assigning values to an object.

```
# Form for creating objects: objectName <- value
x <- 3 * 4
x
```

```
[1] 12
```

```
this_is_a_really_long_name <- 2.5
r_rocks <- 2 ^ 3
this_is_a_really_long_name
```

```
[1] 2.5
```

```
#calling rrocks would not work because of a type
```

Calling functions

R has many basic functions that are built in such as seq to create a sequences of numbers.

```
#functionName(arg1 = val1, arg2 = val2, and so on)
seq(1,10)
```

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

You can also create sequences with different step sizes.

```
seq(1,10, by=2)
```

```
[1] 1 3 5 7 9
```

```
#not all functios require an argument
example(seq)
```

```
seq> seq(0, 1, length.out = 11)
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

seq> seq(stats::rnorm(20)) # effectively 'along'
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20

seq> seq(1, 9, by = 2)     # matches 'end'
[1] 1 3 5 7 9

seq> seq(1, 9, by = pi)    # stays below 'end'
[1] 1.000000 4.141593 7.283185

seq> seq(1, 6, by = 3)
[1] 1 4

seq> seq(1.575, 5.125, by = 0.05)
 [1] 1.575 1.625 1.675 1.725 1.775 1.825 1.875 1.925 1.975 2.025 2.075 2.125
[13] 2.175 2.225 2.275 2.325 2.375 2.425 2.475 2.525 2.575 2.625 2.675 2.725
[25] 2.775 2.825 2.875 2.925 2.975 3.025 3.075 3.125 3.175 3.225 3.275 3.325
[37] 3.375 3.425 3.475 3.525 3.575 3.625 3.675 3.725 3.775 3.825 3.875 3.925
[49] 3.975 4.025 4.075 4.125 4.175 4.225 4.275 4.325 4.375 4.425 4.475 4.525
[61] 4.575 4.625 4.675 4.725 4.775 4.825 4.875 4.925 4.975 5.025 5.075 5.125

seq> seq(17) # same as 1:17, or even better seq_len(17)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
```

```r
date()
```

```
[1] "Tue Apr 25 11:55:18 2023"
```

Getting Help in R

There is a built in help function to read the descriptions of a function that you know the name of and want more information about

```r
help(log)
?log
```

For example if we want to know what cross tabulate does:

```
help.search("cross tabulate")
??"cross tabulate"
```

Vectors, Vectoring, and Indexing

The length function returns the length of a vector. Unlike other coding languages there is no scalar type and instead values like ‘word’ or 3.1 are stored as a vector of length 1.

```
length(3.1)
```

```
[1] 1
```

To create longer vectors, we combine values with the function c():

```
x <- c(56, 95.3, 0.4)
x
```

```
[1] 56.0 95.3  0.4
```

```
#or
y <- c(3.2, 1.1, 0.2)
y
```

```
[1] 3.2 1.1 0.2
```

Vectorization

Vectorization lets us loop over the elements in a vector with writing an explicit loop:

```
x+y
```

```
[1] 59.2 96.4  0.6
```

```
x-y
```

```
[1] 52.8 94.2  0.2
```

```r
x/y
```

```
[1] 17.50000 86.63636  2.00000
```

In addition to operators like `+` and `*`, many of R's math functions (e.g., `sqrt()`, `round()`, `log()`, etc.) are all vectorized:

```r
sqrt(x)
```

```
[1] 7.4833148 9.7621719 0.6324555
```

```r
round(sqrt(x), 3)
```

```
[1] 7.483 9.762 0.632
```

```r
log(x)/2 + 1 # note how we can combined vectorized operations
```

```
[1] 3.0126758 3.2785149 0.5418546
```

Vector Indexing

We can use indexing to get a specific element in the vector to retrieve.

```r
x <- c(56, 95.3, 0.4)
x[2]
```

```
[1] 95.3
```

The index positions starts at 1. R's vectors are 1-indexed.

```r
x[1]
```

```
[1] 56
```

If you try to retrieve an element that does not exist in the vector it will return N/A

```
x[4]
```

```
[1] NA
```

We can also change elements by combining indexing and assignment:

```
x[3] <- 0.5
x
```

```
[1] 56.0 95.3  0.5
```

Version of R used

```
sessionInfo()
```

```
R version 4.2.3 (2023-03-15)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Big Sur ... 10.16

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
 [1] compiler_4.2.3  fastmap_1.1.1   cli_3.6.1       tools_4.2.3
 [5] htmltools_0.5.5 yaml_2.3.7      rmarkdown_2.21  knitr_1.42
 [9] xfun_0.38       digest_0.6.31   jsonlite_1.8.4  rlang_1.1.0
[13] evaluate_0.20
```