Lab 1: Getting started with R

Not graded, just practice

2024-08-29

To learn to program in R (or any language), you can read about how to do it, and watch someone else do it; but the only way to really learn is to do it yourself. Create some data structures, try some stuff, and see what happens! Here are some practice quiz questions to guide your learning. We will go over the solutions to these in lab.

0.1 Materials from lab

- Brittany's lab slides
- Wesley's lab slides

0.2 Google Colab

True or false? We can start a new R notebook in Google Colab with `File > New notebook`

- (A) True
- (B) True, but we need to change the Runtime environment too
- (C) False

For problem sets, how will you submit your colab notebook for grading?

- (A) `File > Download > Download .ipynb`, then upload to Gradescope
- (B) `File > Download > Download .ipynb`, then upload to Canvas
- (C) `File > Download > Download .py`, then upload to Gradescope
- (D) `File > Download > Download .py`, then upload to Canvas

What version of R is Google Colab running? Hint: use `sessionInfo()`.

What is the relationship between R and Google Colab?

- (A) R is a programming language. Google Colab is a smaller, specific version of R.
- (B) R is a programming language. Google Colab is a development environment where you can run R
- (C) R and Google Colab are both programming languages.
- (D) R is a paid (proprietary) programming language. Google Colab is a free service to run R.

What happens to files you upload to google colab when the Runtime environment is restarted?

- (A) They are saved in your google drive.
- (B) They are saved in Google Colab's cloud for future use
- (C) They are deleted
- (D) They are stored for 7 business days, then deleted.

0.3 R Basics

Which of the following are expressions?

- (A) 10
- (B) 5 + 10
- (C) x <- 5 + 10
- (D) x <- y + 10
- (E) mean(x)

Which of the following are valid variable names in R?

- (A) childAge
- (B) response_time
- (C) 1stPlaceWinner
- (D) 2fast2furious
- (E) pi

Suppose we open a new colab notebook and run the following code block. What will be returned?

```
x <- 1 + 2
y <- 0 + 3
ls()
```

- (A) 3
- (B) x=3 • y=3
- (C) 'x' • 'y'
- (D) mean(c(1,3,5)) • median(c(1,3,5))

Which of the following will load the `emo` package into the current environment?

- (A) install.packages('emo')
- (B) library(emo)
- (C) data(emo)
- (D) attributes(emo)

Which of the following occur in the code block below?

```
# compute the mean of x and y
mean(c(x,y))
```

- (A) a message
- (B) a function

- (C) a comment
- (D) an expression

0.4 Vectors

Which of the following returns the vector `20 22 24 26`

- (A) `20:26`
- (B) `seq(from=20, to=26, by =2)`
- (C) `rep(c(20, 22, 24, 26), times = 4)`
- (D) `c(20, 22, 24, 26)`

Suppose we construct a vector with `c(1, "two", 3, 4, 5, 6)` and assign it to `x`. What will the following code block return?

```
typeof(x)
```

What is the previous question an example of?

- (A) attribute addition
- (B) explicit coercion
- (C) implicit coercion
- (D) none of the above

What will the following code block return?

```
x <- 1:4
y <- matrix(x, ncol=2, nrow=2)
typeof(y)
```

What will the following code block return?

```r
x <- c()
length(x)
```

Given the following output from `str(x)`, what will `as.logical(x)` return?

```
 num [1:4] 1 0 1 0
```

- (A) an error
- (B) TRUE • FALSE • TRUE • FALSE
- (C) FALSE
- (D) FALSE • TRUE • FALSE • TRUE

Given the following vector, what will `as.double(x)` return?

```r
x <- c("one", "two", "three")
```

- (A) an error
- (B) 1 • 2 • 3
- (C) 2 • 4 • 6
- (D) 'one' • 'two' • 'three'

What happens if you add a vector of numbers to a single number in the following expression?

```r
c(1, 3, 5) + 1
```

- (A) 2 • 3 • 5
- (B) Error: length mismatch
- (C) 1 • 3 • 5 • 1
- (D) 2 • 4 • 6

What happens if you multiply a vector times another vector?

```r
c(1, 3, 5) * c(10, 100, 1000)
```

- (A) 10 • 300 • 5000
- (B) Error: length mismatch
- (C) 10 • 30 • 50 • 100 • 300 • 500 • 1000 • 3000 • 5000
- (D) a 2 x 3 matrix
- (E) Error: cannot multiply vectors

Suppose we run the following code. What will `any(x)` return?

```
x <- c(1, 5, 11) > 10
```

- (A) TRUE
- (B) FALSE
- (C) Error: vector is double but requires logical

0.5 Subsetting

Which of the following code subsets the vector `x <- c("blue", "pink", "red")` to return just the first element?

- (A) x[1]
- (B) x[[1]]
- (C) x[0]
- (D) x[-c(2, 3)]
- (E) x[“blue”]

Suppose we run the following code. What will `x[[2]]` return?

```
x <- seq(from = 2, to =8, by=2)
```

Suppose we run the following code. What will `m[1, 2]` return?

```
m <- matrix(c(10,20,30,40), nrow=2, ncol=2)
```

Suppose we run the following code. What will `df$y[4]` return?

```
df <- data.frame(
    x = c(2, 4, 6, 8),
    y = c("l", "m", "n", "o")
)
```

0.6 Missing Values

Suppose we run the following code. What will `is.na(y)` return?

```
y <- c(25, 25, NA, 36)
```

- (A) 3
- (B) Error: Non-atomic vector is invalid type
- (C) TRUE
- (D) FALSE FALSE TRUE FALSE

Suppose we run the following code. What will `is.null(y)` return?

```
y <- c()
```

Suppose we run the following code. What will `mean(y)` return?

```
y <- c()
```
