Icahn
School of
Medicine at
Mount
Sinai

Introduction to coding with R

Part I

Joselyn C. Chávez Fuentes

01/09/2024

Let's recap

- What's the difference between R and RStudio?
- What are the most common RStudio panels?
- What's the console for?
- How do we create a new script?
- Where do we find existing variables?
- How do we read/import a table?
- How do we visualize the table?
- How do we write/export a table?

Data structures in R

- Vectors
- Matrices
- Data frames
- Lists
- Functions

Choosing a good variable name

- Be clear and concise.
- When possible, use lowercase.
- Don't include special characters. Avoid dieresis and other accents (é, è, â, î or ô, tilde ñ, ü or ï)
- Use _ or Upper/Lower case to separate words, never space.
- Avoid conflicts with any R keywords (True, False, and, if, else, and other function names)

Let's try this

What of these variable names follow good practices?

a) MY_FIRST_VARIABLE

b) OxygenLevel

c) patient_name

d) final.value

e) mean

Vectors

Creating a vector

Using the assignment operator

Vector with a single value

```
my_vector <- 10
my_vector <- "a"
```

Using the combine function

```
my_vector <- c(1,10,25,30)
my_vector
```

```
## [1]  1 10 25 30
```

```
my_vector <- c("a","b","c")
my_vector
```

```
## [1] "a" "b" "c"
```

```
my_vector <- 1:5
my_vector
```

```
## [1] 1 2 3 4 5
```

Let's practice

- Create a variable called vector_1 that contains the number 300
- Create a variable called vector_2 that contains the numbers 100:500
- Create a variable called vector_3 that contains the letters "a" to "e"
- Create a variable called vector_4 that contains your name, age, and the city where you were born.

Using the seq() function

```
my_vector <- seq(1:10)
my_vector
```

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

```
my_vector <- seq(from = 0, to = 10, by = 2)
my_vector
```

```
## [1]  0  2  4  6  8 10
```

- Write a vector with the numbers 1 to 500 in steps of 10

Vector features

- Vectors have only one dimension (length)

```
my_vector <- c(1,2,3,4)
length(my_vector)
```

```
## [1] 4
```

- All vector components must be the same type
  - Numeric
  - Integer
  - Double
  - Character
  - Factor
  - Logical

- Numeric

```
x_num <- c(1, 2, 3)
class(x_num)
```

```
## [1] "numeric"
```

- Integer

```
x_int <- c(1L, 2L, 3L)
class(x_int)
```

```
## [1] "integer"
```

- Double

```
x_dbl <- c(1, 2.5, 3.1)
typeof(x_dbl)
```

```
## [1] "double"
```

- Character

```
x_chr <- c("a", "chair", "the window")
class(x_chr)
```

```
## [1] "character"
```

- Factor

```
x_fct <- factor("mouse_a","mouse_b","mouse_c")
class(x_fct)
```

```
## [1] "factor"
```

- Logical

```
x_log <- c(TRUE, FALSE, TRUE)
class(x_log)
```

```
## [1] "logical"
```

- R finds a way to unify the data type

```
x <- c(1, "a", TRUE)
x
```

```
## [1] "1"    "a"    "TRUE"
```

```
class(x)
```

```
## [1] "character"
```

What will be the class of these vectors?

```
y <- c(5, 7, "airplanes")

z <- c(10, 30.5, TRUE)
```

Converting one class to another using as. functions

```
x <- c(1.9, 2, 0, 0)
class(x)
```

```
## [1] "numeric"
```

```
as.double(x)
```

```
## [1] 1.9 2.0 0.0 0.0
```

```
as.integer(x)
```

```
## [1] 1 2 0 0
```

Converting one class to another using as. functions

```
as.character(x)
```

```
## [1] "1.9" "2"   "0"   "0"
```

```
as.factor(x)
```

```
## [1] 1.9 2   0   0
## Levels: 0 1.9 2
```

```
as.logical(x)
```

```
## [1]  TRUE  TRUE FALSE FALSE
```

Missing values

- NA

```
x <- c(1, "a", TRUE, NA)
x
```

```
## [1] "1"    "a"    "TRUE" NA
```

- NaN

```
x <- c(10, -1, NA)
log(x)
```

```
## [1] 2.302585      NaN       NA
```

How do we access the vector elements?

Using an integer as index

Vector index in R starts at 1

```
x <- c(10,20,30,40,50)
x
```

```
## [1] 10 20 30 40 50
```

```
x[3] # Extracts the third element
```

```
## [1] 30
```

```r
x <- c(10,20,30,40,50)

# Extracts index from 3 to 5
x[3:5]
```

```
## [1] 30 40 50
```

```r
x <- c("a","b","c","d","e")

# Extracts index 2 and 5
x[c(2,5)]
```

```
## [1] "b" "e"
```

Let's practice

- Create a vector with numbers from 50 to 100 in steps of 5.
  - Get the first 7 numbers.
  - Get the last 8 numbers.
- Create a vector with letters "a" to "k".
  - Get the letters c, d, and j

Thanks!