Week 1 Discussion Exercises

Sami Cemek

The plan for week 1

- Goal: The basics of how R sees the world and how you interact with R
- Topics: Basic Building Blocks, Sequences of Numbers, Vectors
- All material from this week can be found in the Week 1 Learning Resources page.

Packages Exercises

In the console, try typing

```
install.packages("tidyverse")
```

to download and install the tidyverse package. This only needs to be done once.

Then use the `library()` command to open the package in your current R session. You need to do this every time you restart R.

```
library(tidyverse)
```

Data type exercises

DTE1. To R, `5` , `5L` , and `"5"` are very different things.

- 1. What happens if you add 5 and 5L? What is the resulting data type?
- 2. What happens if you add 5L and “5”?
- 3. What happens if you add 5 and `TRUE` ?
- 4. What happens if you add 5 and `NA` ?

Your answer: 1) It prints 10 on the console. I used typeof() function to find its type. The resulting data is a double. 2) It gives an error that says “Error in 5L +”5" : non-numeric argument to binary operator" 3) 5 + TRUE prints 6 on the console. 4) It prints NA which stands for “Not Available”.

DTE2 (Intermediate). R has an internal hierarchy of data types/modes. We will learn more about this later, but if you want to play around, see if you can predict the output of the following issues. What can you deduce about the rules of implicit coercion?

```
x <- c(4.2, "4.2")
typeof(x)
```

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

```
print(x)
```

```
## [1] "4.2" "4.2"
```

```
x <- c(4.2, 1L)
typeof(x)
```

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

```
print(x)
```

```
## [1] 4.2 1.0
```

```
x <- c(TRUE, 5L)
typeof(x)
```

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

```
print(x)
```

```
## [1] 1 5
```

```
x <- c(TRUE, FALSE + 1)
typeof(x)
```

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

```
print(x)
```

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

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

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

```
print(x)
```

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

Arithmetic Exercises

AE1. What happens if you type in “pi” to R? What about “PI”? Are there any other constants built into R?

Your answer: When I wrote pi, it printed 3.141593 on the console. When I wrote PI, it printed “Error: object ‘PI’ not found” on the console because R is case sensitive. pi and PI is different. The other constants built into R: LETTERS,letters, month.abb, month.name.

```
pi
```

```
## [1] 3.141593
```

```
#PI
```

AE2. Write code to multiply 3 by 2, then add 7, then square the sum, and then divide the overall result by 5.

Your answer: The result is: 33.8

```
x <- ((((3 * 2) + 7)^2)/5)
x
```

```
## [1] 33.8
```

AE3. Use the console to complete 5-10 simple calculations in a row. What are two ways to easily retrieve or re-run earlier calculations?

Your answer: Simple calculation examples: 1+1 = 2 2*5 = 10 3/6 = 0.5 9-3 = 6 The easier way to re-run earlier code is by pressing up button. That will bring the previous command.

Comparison Exercises (Relational and logical operators)

CE1. Write code to test whether an exam score `exam` is at least 80. Then replace `exam` with the student’s actual test score of 85, and verify that R tells you that the statement is `TRUE` .

Your answer:

```
exam <- 80
exam <- 85

#compare if exam score is equal to or less than 85, if yes print TRUE
exam >= 80
```

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

```
exam
```

```
## [1] 85
```

CE2. Suppose we type `5 = 5` in the console. What happens? Why?

Your answer: It prints “Error in 5 = 5 : invalid (do_set) left-hand side to assignment”. We use “=” to assign variables. We can’t assign an integer to an integer.

CE3. Bonus: Suppose we type `5L != 5.0` in the console. What happens? What about `5L != 5.00000000000000001` ? Why?

Your answer: It returns FALSE because 5L and 5.0 is the same thing even though their data type is different. This “!=” sign means not equal to. Again, it return FALSE for the second statement as well. That means they are equal.

CE4. Suppose we are studying the effectiveness of fertilizer in growing tomatoes. We have a variable containing the fruit weight, “my_fruit”. What would we use to find out if my_fruit weighed more than 100g?

Your answer: We can check if my_fruit is heavier more than 100g by using comparison signs. For test purposes I will give my_fruit a value such as 120g.

```
#Gave a value for test purposes
my_fruit <- 120
#Check if my_fruit is weight more than 100g, if yes print TRUE
my_fruit > 100
```

CE5. How could we test whether `my_fruit` weighs between 100g and 150g?

```
#Check if my_fruit is weighs between 100g and 150g, if yes print TRUE
between(my_fruit,100,150)
```

Your answer: We will compare my_fruit with 100g and 150g. See the code above.

CE6 (Intermediate). What are two different methods for testing whether my_fruit weighs less than 150g?

Your answer:

CE7 (Intermediate). Try running `5 | FALSE` and explain what happens.

Your answer:

Variable name exercises

VNE1. Which is an appropriate variable name for the average age of a class of 6th grade students? -1) average_age_in_years -2) Mean.Age -3) 6_grade_age -4) mean -5) mean-age

Your answer: 1) YES 2) You can use dot when naming variable but it is not recommended. It’s a bad practice. 3) NO, variable name can’t start with a number. 4) You can use mean as a variable name but there is a function called mean which might cause confusion. Therefore it is not recommended. It’s a bad practice. 5) NO, variable name should not include “-” character.

VNE2. Which tab in RStudio lets you see all the variables that exist in your current R session?

Your answer: The “Environment” tab in the top right window lists the variables and functions present in the current R session.

VNE3 (Intermediate). Consider creating a variable called `x` and setting it equal to 5:

```
x <- 5
x = 5
```

Discuss why we should use the assignment operator `<-` instead of an equal sign `=`, or argue that we should use `=` to create variables. Both technically work… (this is a philosophical question more than a practical question)

Your answer:

Sequences of Numbers

SN1. Show how to use seq() and : to print the numbers from 10 down to 1.

```
seq(10,1)
```

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

SN2: Show how to use rep to print each of the following sequences:

a. “orange” “orange” “orange” “blue” “blue” “blue”

```
rep(c("orange","blue"), each = 3)
```

```
## [1] "orange" "orange" "orange" "blue"   "blue"   "blue"
```

b. “orange” “blue” “orange” “blue” “orange” “blue”

```
rep(c("orange","blue"), times = 3)
```

```
## [1] "orange" "blue"   "orange" "blue"   "orange" "blue"
```

SN3. A common neurological exercise is to count backward from 100 by multiples of 7.

a. Show how to use R to calculate and print this sequence (100, 93, …, 2) with a single line of code.

```
#This took for a while to figure out :))
seq.int(from = 100, to = 1, by= -7)
```

```
##  [1] 100  93  86  79  72  65  58  51  44  37  30  23  16   9   2
```

b. (Intermediate) Modify your previous code so that both the starting number (100) and the increment (7) are variables you can change. You will be able to define those variables, and then count backwards from START by multiples of INCREMENT. This will involve 3 lines of code.

Vector exercises:

A local animal rescue group is trying to track the effectiveness of their social media presence; they are currently interested in tracking follower growth. The table below summarizes the number of page likes or new followers each day:

	Sun	Mon	Tues	Wed	Thurs	Fri	Sat
FB	30	43	55	89	71	52	42
Twitter	60	32	86	44	21	30	28

VE1. Create 2 vectors named `fb` and `twitter` containing the daily counts for each site. If possible, verify using code that each vector has 7 values in it.

Your answer:

```
#Create vectors
fb <- c(30,43,55,89,71,52,42)
twitter <- c(60,32,86,44,21,30,28)

#Verify each vector has 7 values
length(fb)
```

```
## [1] 7
```

```
length(twitter)
```

```
## [1] 7
```

VE2. Create a vector called `daysofweek` containing the names of the days of the week (“Sunday”, “Monday”, “Tuesday”, etc.), then use the vector to assign names to the social media vectors using the `names()` function.

Your answer: See the code below.

```
# I created two different vectors for days of week to display both of the social medias separate
ly

#For facebook
daysofweek1 <- c("Sunday", "Monday", "Tuesday", "Wednesday" , "Thursday", "Friday", "Saturday")

names(daysofweek1)<-fb[1:7]

#For twitter
daysofweek2 <- c("Sunday", "Monday", "Tuesday", "Wednesday" , "Thursday", "Friday", "Saturday")

names(daysofweek2)<-twitter[1:7]

daysofweek1
```

```
##          30          43          55          89          71          52 
##    "Sunday"    "Monday"   "Tuesday" "Wednesday"  "Thursday"    "Friday" 
##          42 
##  "Saturday"
```

```
daysofweek2
```

```
##          60          32          86          44          21          30 
##    "Sunday"    "Monday"   "Tuesday" "Wednesday"  "Thursday"    "Friday" 
##          28 
##  "Saturday"
```

VE3. Have R calculate the total new followers/likes combined on fb and twitter for each day. You should have one vector with 7 values, where each value represents the total number of new followers on FB and twitter combined. For instance, the first value equal 90, which is the total for Sunday.

Your code should work even if we change the original values in `fb` or `twitter`.

Your answer: See the code below.

```
#This vector for total new followers/likes combined on fb and twitter
daysofweek3 <- c("Total_Sunday", "Monday", "Tuesday", "Wednesday" , "Thursday", "Friday", "Satur
day")

daysofweek3[1] <- fb[1] + twitter[1] #Sunday
daysofweek3[2] <- fb[2] + twitter[2] #Monday
daysofweek3[3] <- fb[3] + twitter[3] #Tuesday
daysofweek3[4] <- fb[4] + twitter[4] #Wednesday
daysofweek3[5] <- fb[5] + twitter[5] #Thursday
daysofweek3[6] <- fb[6] + twitter[6] #Friday
daysofweek3[7] <- fb[7] + twitter[7] #Saturday

#print the values to see
daysofweek3
```

```
## [1] "90"  "75"  "141" "133" "92"  "82"  "70"
```

VE4. Save the total shares from the previous question in a new variable called `dailytotals` then print it out.

Your answer:

```
dailytotals <- daysofweek3
dailytotals
```

```
## [1] "90"  "75"  "141" "133" "92"  "82"  "70"
```

VE5. Bonus: Suppose we have instagram followers for Monday through Wednesday only. If we try to add this vector to our facebook reactions, what will happen? Why?

Your answer:

VE6. Have R calculate the total number of new followers & page likes on Facebook all week. This should be a single value.

Your answer: 382

```
#facebook
fb_total_like <-  fb[1] + fb[2] + fb[3] + fb[4] + fb[5] + fb[6] + fb[7]
fb_total_like
```

```
## [1] 382
```

```
#twitter (for the next question)
twitter_total_like <-  twitter[1] + twitter[2] + twitter[3] + twitter[4] + twitter[5] + twitter[
6] + twitter[7]
twitter_total_like
```

```
## [1] 301
```

VE7. What about the total combined new followers on both FB and Twitter all week? You should have a single number that is the result of adding up all 14 values in the original table. Give at least 2 ways to have R calculate this using the appropriate vectors from VE1-VE6.

Your answer: 683

```
#Used the vectors from the previous question
fb_total_like + twitter_total_like
```

```
## [1] 683
```