CS 251 Statistical Computing

HOP2B: R for statistical project

2/19/2020 Developed by Aya Khalil

04/02/2020 Reviewed by Apiwat Chuaphan

10/16/2020 Reviewed by Shanshan Yu

School of Technology and Computing @City University of Seattle (CityU)

**Before You Start**

* If you already finished this module through any CityU Technology Institute (TI) courses,  
  just skim this module and skip it.
* Version numbers may not match with the guide. But that should be fine.  
  If given the option to choose between stable release (long-term support) or most recent, please choose the stable release.
* This guide targets Windows OS users. So, MacOS users may have different commands to input in the shell/terminal.
* We cannot explain every step. **This cookbook always needs your own creative judgement.**
* **For your working directory, use your course number.** The hands-on tutorial may use a different course number as an example.

**Learning Outcomes**

* Logic statements
* Loops
* Functions

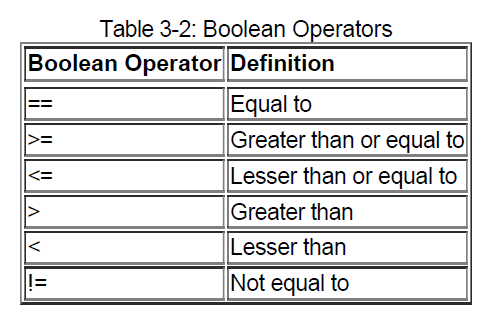
**Resource**

* Hui, E. G. M. (2019). [*Learn R for applied statistics: With data visualization, regressions, and statistics*](https://login.proxy.cityu.edu/sso/skillport?context=144516). Apress.

**Logical Statements**

if…else statements are usually the logical fragments of your code in R. They give your program some intelligence and decision making by specifying the if rules.

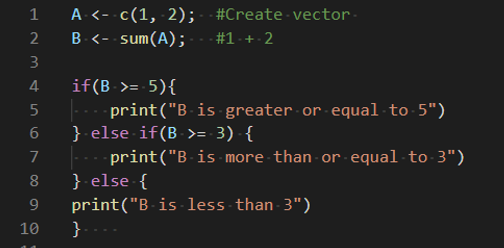
Table 3-2 shows the Boolean operators that can be used in writing the Boolean expressions of the if… else statements



**You should be in:**

* Desktop/CS251/hop02-YourUserName
* In Module2 project folder, create ifElseCondition.R

**Type the following code in ifElseCondition.R**



-Save your code, file>save or ctrl+s

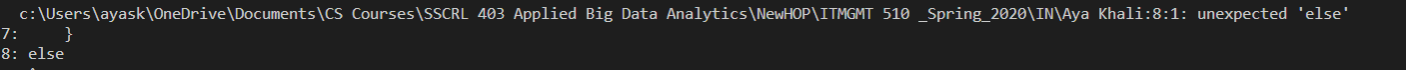
-Run your code:

- Select Run Source button:

**Output:**



**If you got this error:**



**Then, make sure to rearrange your curly brackets to look like the picture above.**

**Loops**

Loops are used to repeat certain fragments of code. For example, if you want to print the "This is R." message 100 times, it will be very tiresome to type print("This is R. "); 100 times. You can use loops to print the message 100 times more easily. Loops can usually be used to go through a set of vectors, lists, or data frames. In R, there are several loop options: both while loop,

for loop, and repeat loop.

**For Loop**

Let's start with the syntax for a for loop:

for (value in vector) {

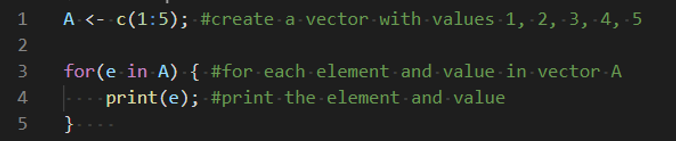
#statements

}

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file forLoop.R

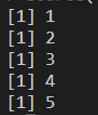
Type the following code in forLoop.R file



-Save your file, ctrl+s or file>save

Run your code

Output:



In the previous example, you create vector A, with values 1 to 5. For each element in the vector, you print the element in the console.

Explanation:

1. For(e in A), for first iteration e will be the first element in the vector A ,not the index, and second iteration, e will be the second element in the vector, and it will loop till the vector ends.
2. For each iteration, we print the value of e which is the vector element.

**While Loop**

You can also use while loop to loop until you meet a specific Boolean expression:

While (Boolean Expression) {

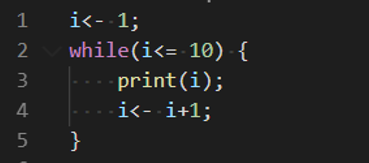
#Code to run or repeat until Bolean Expression is false

}

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file whileLoop.R

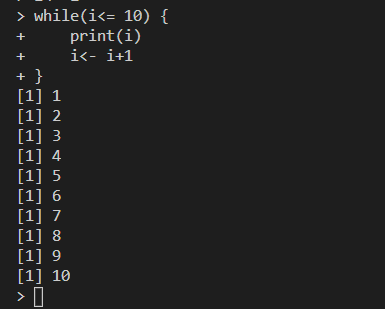
Type the following code in whileLoop.R file



-Save your file, ctrl+s or file>save

Run your code: ctrl+A, then ctrl+enter

Output:



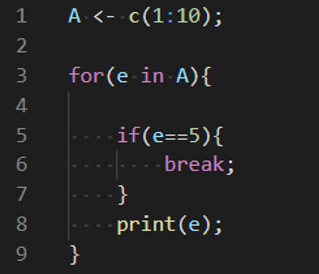
**Break and Next Keywords**

In loop statements, you can use the break keyword and the next keyword. The break keyword is to stop the iterations of the loop. The next keyword is to skip the current iteration of a loop.

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file breakKey.R

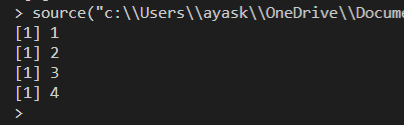
Type the following code in breakKey.R file



-Save your file, ctrl+s or file>save

Run your code: ctrl+A, then ctrl+enter

Output:



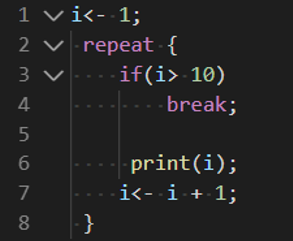
**Repeat Loop**

The repeat loop repeats the code many times, and there is no Boolean expression to meet. To stop the loop, you must use the break keyword.

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file repeatLoop.R

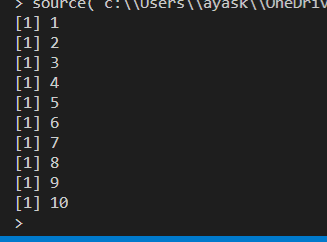
Type the following code in repeatLoop.R file



-Save your file, ctrl+s or file>save

Run your code: ctrl+A, then ctrl+enter

Output:



Explanation:

In this example, you create variable i with a value of 1. You repeat printing i and add 1 to i until i is more than 10, and then you break out from the loop. If you forget or don't add a condition statement and the break keyword, you can end up in an

infinite loop. An infinite loop is dangerous because it can consume your system resources and cause your program to keep on looping at the same place. The for loop is the preferable loop because the condition is defined in the first statement.

The while loop is the next preferred loop.

A condition statement is also stated in the first statement:

while(i<= …) …

If you forget to increment the i, I <- I + 1;, you will also have an infinite loop.

**Functions**

Functions help you organize your code and allow you to reuse code fragments whenever you need. To create a function, use the following syntax:

function\_name<- function(arg1, arg2, …) {

# Codes fragments

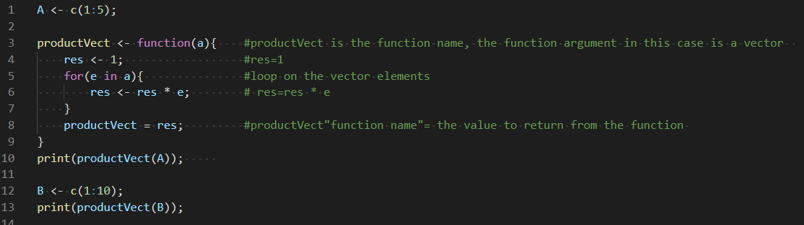
function\_name = #value to return

}

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file productFun.R

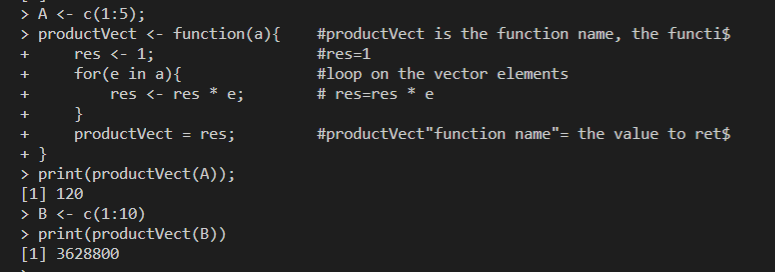
Type the following code in productFun.R file



-Save your file, ctrl+s or file>save

Run your code: ctrl+A, then ctrl+enter

Output:



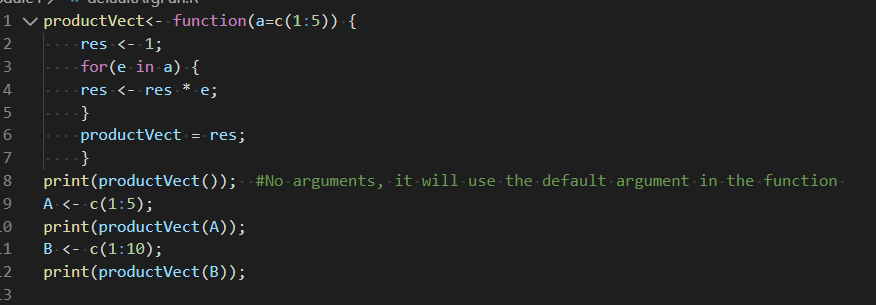
Explanation:

A is a vector with values from 1 to 5. You call the function by using productVect(A). The argument, a, in the function declaration is the formal argument. The argument, A, you passed to the function while calling the productVect() function is called the actual argument. When you call the function using productVect(A), a, the formal argument, is assigned with A, the actual argument. We can call the function more than one time, so we defined another vector and when we call the function using productVect(B), a, the formal argument, is assigned with B, the actual argument.

**You should be in:**

* Desktop/CS251/hop02-YourUserName/
* In Module2 project folder, create new file defaultArgFun.R

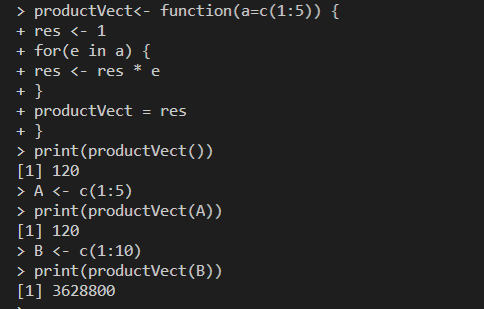
Type the following code in defaultArgFun.R file



-Save your file, ctrl+s or file>save

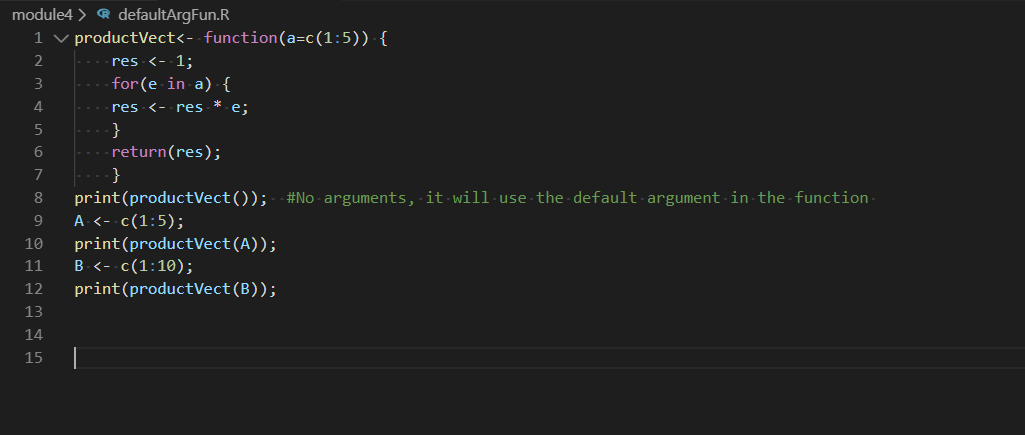
Run your code: ctrl+A, then ctrl+enter

Output:



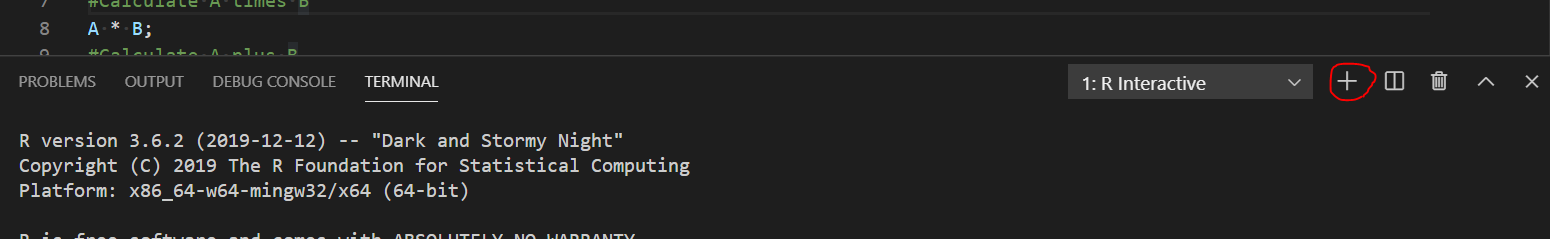
**Instead of using productVect = res, you can also use the return() function to return the results**

* **Type the following code to update defaultArgFun.R file**

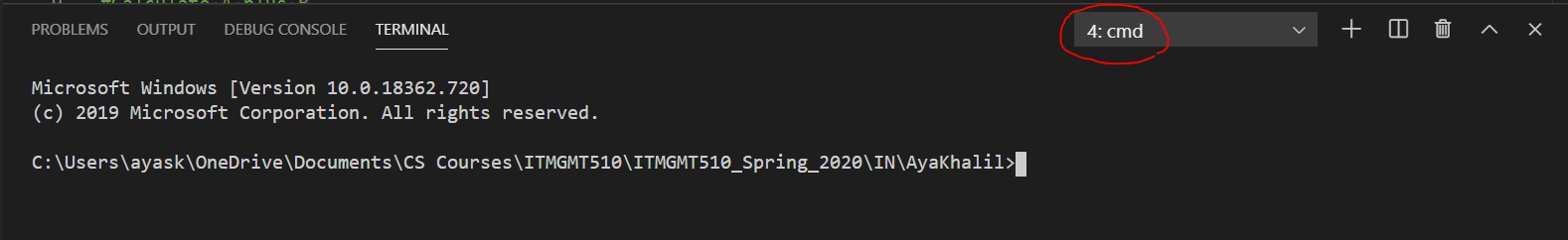


**Push your work to GitHub**

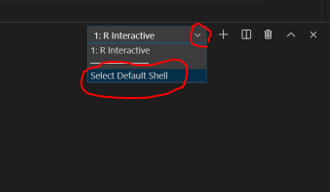
Open the cmd terminal,



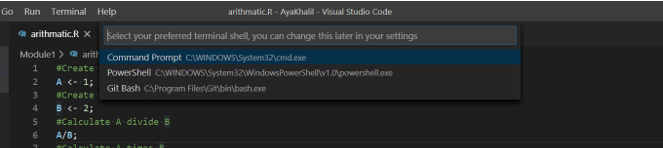
After selecting the + button, make sure the terminal is cmd



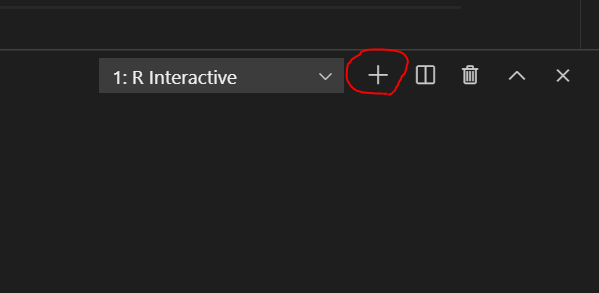
If the terminal not cmd, then you need to add it.



Then select command prompet



Then select the + button





**Make sure you are in**

* Desktop/CS251/hop02-YourUserName

Run the following commands to push your work to the GitHub repository:

Open the terminal from the VSCode by hit the control + ~ key and type the following command:

>>> git add .

>>> git commit -m “Submission for Module 2”

>>> git push origin master