Basic Course on R: Programming Structures 1 Practical Answers

Elizabeth Ribble*

20-24 May 2019

Contents

*emcclel3@msudenver.edu

1 Part A: `if()`, `else`, and `ifelse()` and Vectorization

1. Write a function `evenOrOdd()` involving `if()` and `else` that takes an argument `x` and returns `"Even"` or `"Odd"` depending on whether or not `x` is divisible by 2. (*Do not* use the `ifelse()` function).

```
evenOrOdd <- function(x) {
   if(x %% 2 ==0) {
       return("Even")
   } else {
       return("Odd")
   }
}
```

2. Is your function `evenOrOdd()` *vectorized*? Check by passing it the vector:

```
w <- c(3, 6, 6, 4, 7, 9, 11, 6)
```

```
w <- c(3, 6, 6, 4, 7, 9, 11, 6)
evenOrOdd(w)

## Warning in if (x%%2 == 0) {: the condition has length > 1 and only
the first element will be used

## [1] "Odd"
```

The function is not vectorized because it only runs on the first element!

3. Another way to determine if each element of a vector is even or odd is to use the `ifelse()` function, which serves as a vectorized version `if()` and `else`. Use `ifelse()` to obtain `"Even"` or `"Odd"` for each element of `w`.

```
w <- c(3, 6, 6, 4, 7, 9, 11, 6)
ifelse(w %% 2 ==0, "Even", "Odd")

## [1] "Odd"  "Even" "Even" "Even" "Odd"  "Odd"  "Odd"  "Even"
```

2 Part B: Loops

1. How many times will `"Frisbee Sailing"` be printed to the screen in each of the following sets of commands? Try to answer without using R.

   a)
   ```
   i <- 5
   while(i < 1) {
     print("Frisbee Sailing")
     i <- i + 1
   }
   ```

   It will not print because the original i is not less than 1.

   b)
   ```
   ## not run
   i <- 0
   while(i < 5) {
     print("Frisbee Sailing")
   }
   ```

   It will print indefinitely - until you hit escape on the keyboard or click stop in RStudio because i is always less than 5.

   c)
   ```
   i <- 0
   while(i < 5) {
     print("Frisbee Sailing")
     i <- i + 1
   }
   ```

   ```
   ## [1] "Frisbee Sailing"
   ## [1] "Frisbee Sailing"
   ## [1] "Frisbee Sailing"
   ## [1] "Frisbee Sailing"
   ## [1] "Frisbee Sailing"
   ```

   The phrase will print five times; it stops once the i+1 value reaches 5.

2. How many times will `"Masked Marvel"` be printed to the screen in the following set of commands? Try to answer without using R.

   ```
   i <- 1
   repeat {
     if(i > 5) break
   ```

```
  print("Masked Marvel")
  i <- i + 1
}

## [1] "Masked Marvel"
## [1] "Masked Marvel"
## [1] "Masked Marvel"
## [1] "Masked Marvel"
## [1] "Masked Marvel"
```

The phrase will print five times; it stops once the i+1 value reaches 5.

3. The file **kennedys.txt** has a command to create a list containing two generations of the famous Kennedy family:

```
Kennedys <- list(
    JosephJr = character(0),
    John = c("Caroline", "JohnJr", "Patrick"),
    Rosemary = character(0),
    Kathleen = character(0),
    Eunice = c("RobertIII", "Maria", "Timothy", "Mark", "Anthony"),
    Patricia = c("Christopher", "Sydney", "Victoria", "Robin"),
    Robert = c("Kathleen", "JosephII", "RobertJr", "David",
               "MaryC", "Michael", "MaryK", "Christopher",
               "Matthew", "Douglas", "Rory"),
    Jean = c("Stephen", "William", "Amanda", "Kym"),
    Edward = c("Kara", "EdwardJr", "Patrick")
)
```

Read in the file with the use of `source()` and type `ls()` to see if the list was created (type `Kennedys` to view the object).

```
source("kennedys.txt")
```

Loop over the list of the first generation of Kennedys, keeping track of how many children each one has in a vector.

```
children <- NULL
for(i in Kennedys){
  children <- c(children, length(i))
```

```
}
children
```

```
## [1]  0  3  0  0  5  4 11  4  3
```

3 Part C: Logical Operators &, |, and !

1. What will be the result of the following:

```
(10 < 20 && 15 < 16) || 9 == 10
```

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

TRUE because the first statement (in parentheses) is TRUE and the second is FALSE.

2. One of the following evaluates to TRUE, the other to FALSE. Which is which?

```
4 < 3 && (5 < 6 || 8 < 9)
```

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

```
(4 < 3 && 5 < 6) || 8 < 9
```

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

The first one FALSE because the first statement before && is FALSE. The second one is TRUE because one of the two statements to the left and right of || is TRUE.

3. The data set below contains the systolic and diastolic blood pressure readings for 22 patients (and can be found in the file **BPressure.txt**).

   a) Read the data from **BPressure.txt** into a data frame called `bp` using `read.table()`.

```
bp <- read.table("BPressure.txt", header=TRUE)
```

b) A person's blood pressure is classified as normal if the systolic level is below 120 and the diastolic level is below 80.Use square brackets `[ ]` to extract from `bp` the rows corresponding to patients with normal blood pressures.

```
bp[(bp$Systolic < 120 & bp$Diastolic < 80), ]
```

```
##    PatientID Systolic Diastolic
## 2         SS       96        60
## 3         FR      100        70
## 8         JI      110        40
## 9         MC      119        66
## 12        KD      108        54
## 13        DS      110        50
## 17        SB      118        76
## 21        EC      112        62
```

c) Now use square brackets `[ ]` to extract the rows corresponding to patients whose blood pressures *aren't* normal.

```
bp[!(bp$Systolic < 120 & bp$Diastolic < 80), ]
```

```
##    PatientID Systolic Diastolic
## 1         CK      120        50
## 4         CP      120        75
## 5         BL      140        90
## 6         ES      120        70
## 7         CP      165       110
## 10        FC      125        76
## 11        RW      133        60
## 14        JW      130        80
## 15        BH      120        65
## 16        JW      134        80
## 18        NS      122        78
## 19        GS      122        70
## 20        AB      122        78
## 22        HH      122        82
```