Lab 8: Control Structures

Sunghwu Song

March 26, 2021

Introduction

The main purpose of this lab is to practice control structures in R:

- `if` and `else`: testing a condition and acting on it
- `for`: execute a loop a fixed number of times
- `while`: execute a loop while a condition is true
- `repeat`: execute an infinite loop (must break out of it to stop)
- `break`: break the execution of a loop
- `next`: skip an iteration of a loop

You will need to modify the code chunks so that the code works within each of chunk (usually this means modifying anything in ALL CAPS). You will also need to modify the code outside the code chunk. When you get the desired result for each step, change `Eval=F` to `Eval=T` and knit the document to HTML to make sure it works. After you complete the lab, you should submit your HTML file of what you have completed to Sakai before the deadline.

Part 1: Vector and Control Structures

a) (2 points) Write code that creates a vector `x` that contains `100` random observations from the standard normal distribution (this is the normal distribution with the mean equal to `0` and the variance equal to `1`).

```
x = c()
for (i in 1 : 100) {
x[i] = rnorm(1)
}
```

b) (2 points) Write code that replaces the observations in the vector `x` that are greater than or equal to `0` with a string of characters `"non-negative"` and the observations that are smaller than `0` with a string of characters `"negative"`. Hint: try `ifelse()` funtion.

```
x <- ifelse(x>=0, "non-negative", "negative")
x
```

```
##   [1] "negative"     "non-negative" "non-negative" "non-negative" "negative"
##   [6] "non-negative" "non-negative" "non-negative" "negative"     "negative"
```

```
##  [11] "negative"     "negative"     "negative"     "non-negative" "negative"
##  [16] "negative"     "non-negative" "non-negative" "non-negative" "non-negative"
##  [21] "non-negative" "non-negative" "non-negative" "non-negative" "non-negative"
##  [26] "non-negative" "negative"     "negative"     "negative"     "negative"
##  [31] "negative"     "non-negative" "negative"     "non-negative" "negative"
##  [36] "non-negative" "non-negative" "non-negative" "non-negative" "negative"
##  [41] "negative"     "non-negative" "negative"     "non-negative" "negative"
##  [46] "negative"     "negative"     "negative"     "non-negative" "non-negative"
##  [51] "negative"     "non-negative" "non-negative" "non-negative" "non-negative"
##  [56] "non-negative" "non-negative" "negative"     "negative"     "negative"
##  [61] "negative"     "negative"     "negative"     "non-negative" "negative"
##  [66] "non-negative" "non-negative" "non-negative" "non-negative" "non-negative"
##  [71] "negative"     "negative"     "negative"     "negative"     "non-negative"
##  [76] "negative"     "non-negative" "negative"     "negative"     "non-negative"
##  [81] "non-negative" "non-negative" "non-negative" "non-negative" "non-negative"
##  [86] "non-negative" "non-negative" "non-negative" "negative"     "negative"
##  [91] "non-negative" "negative"     "negative"     "negative"     "negative"
##  [96] "negative"     "negative"     "non-negative" "negative"     "non-negative"
```

c) (2 points) Write `for`-Loop to count how many observations in the vector `x` are non-negative and how many observations are negative. (There are many easier ways to solve this problem. Please use `for`-Loop to practice the things learned in the lecture.)

```
neg = 0
for (i in 1:100) {
  if (x[i] == "negative") {
    neg = neg + 1
  }
}
neg
```

```
## [1] 47
```

```
nonneg = 100-neg
nonneg
```

```
## [1] 53
```

Part 2: Matrix and Control Structures

a) (4 points) Create a 100000 by 10 matrix `A` with the numbers 1 : 1000000. Create a `for`-loop that calculates the sum for each row of the matrix and save the results to a vector `sum_row`. (Don't print the whole matrix in your submission as the matrix is very large. Otherwise, you'll lose scores for it.)

```
A <- matrix(1:1000000, nrow = 100000, ncol = 10)
sum_row <- vector()
for (i in 1:100000) {
    sum_row[i] <- sum(A[i,])
}
head(sum_row)
```

```
## [1] 4500010 4500020 4500030 4500040 4500050 4500060
```

Verify that your results are consistent with what you obtain with the built-in `rowSums` function.

```
sum_row_rowSums = as.integer(rowSums(A))
head(sum_row_rowSums)
head(sum_row)
```

b) (4 points) Another common loop structure that is used is the `while` loop, which functions much like a `for` loop, but will only run as long as a test condition is `TRUE`. Modify your `for` loop from exercise (a) and make it into a `while` loop. Write code to check if the results from `for` loop are the same as the results from `while` loop.

```
B <- matrix(1:1000000, nrow = 100000, ncol = 10)
sum_row2 <- vector()
i = 100000
while (i>0) {
    sum_row2[i] <- sum(B[i,])
    i = i - 1
}
head(sum_row2)
```

```
## [1] 4500010 4500020 4500030 4500040 4500050 4500060
```

Part 3: Data Frame and Control Structures

a) (4 points) Write a `for` loop to compute the mean of every column in `mtcars` and save the results to a vector `col_mean`. (Ignore missing values)

```
col_mean = vector("double", ncol(mtcars))
names(col_mean) = names(mtcars)
for (i in names(mtcars)) {
  col_mean[i] = mean(mtcars[[i]])
}
col_mean
```

```
##        mpg        cyl       disp         hp       drat         wt       qsec
##  20.090625   6.187500 230.721875 146.687500   3.596563   3.217250  17.848750
##         vs         am       gear       carb
##   0.437500   0.406250   3.687500   2.812500
```

b) (2 points) Compute the number of unique values in each column of `iris` and print the results during the loop.

```
data(iris)
p3b <- vector("double", ncol(iris))
names(p3b) <- names(iris)
for (i in names(iris)) {
  p3b[i] <- length(unique(iris[[i]]))
}
p3b
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
##           35           23           43           22            3
```