Student: Mirrien Liang 301325351

Problem 2: Review of R

Consider the following for loops in R. For each for loop, list the values (in order) that the variable `i` takes on in the body of the loop. Briefly (in no more than a few sentences) explain why.

a) `for(i in 1+2:3.4*5) { }`

The variable `i` will take on `i=11` and `i=16` because it first runs `2:3.4`, which is `i=2` and `i=3`. It then times `i` by `5` and plus `1`, which gives `i=11` and `i=16`.

b) `for(i in dim(matrix(0, nr = 7, nc = 8))) { }`

The variable `i` will take on `i=7` and `i=8` because the code `matrix(0, nr=7, nr=8)` creates a 7×8 matrix with all entries equal to zero. Then the `dim()` returns the dimension of the matrix in a vector form which contains only 7 and 8. The above function is equivalent to `for(i in c(7,8)) { }`.

c) `for(i in rnorm(3)) { }`

The variable `i` will take on *three* random numbers given by the function `rnorm(n=3, mean=0, sd=1)`. For example, if we `set.seed(1)`, `rnorm(3)` will generate a vector with `-0.6264538, 0.1836433, -0.8356286`, which are the numbers `i` will take on.

d) `for(i in iris[1:3,3]) { }`

The `iris` is a famous data set that measures flowers' sepal and petal length/width for each of the three species of iris. The dataset has 5 columns and 150 rows. The `[1:3,3]` will return a vector containing the first, second, and third entries in the third column of the iris dataset, which are `1.4`, `1.4`, and `1.3` respectively. These three numbers are what the variable `i` will take on.

e) `for(j in c(1, 2, 3, 4, 5)) { }`

The variable `j` will take on exactly the five numbers given by the vector created via `c()`. The above for loop does not define a variable named `i`.

f) `for(i in (function(x) x*x)(c(1, 2, 3))) { }`

The variable `i` will take on `1`, `4`, and `9`, which are the results given by the function that takes `x=c(1,2,3)` and produces $\mathtt{x}^2$ for each of the three entries.

g) `for(i in NULL) { }`

The variable `i` does not take on any value because `NULL` is considered as an empty range/vector/list here.

h) `for(i in strsplit(as.character(4*atan(1)),'') [[1]][1:10]) { }`

The variable `i` will take on the first ten ***characters*** (i.e., including the decimal point) of π. The above for loop will first run `4*atan(1)` and return $4 \cdot \arctan(1) = \pi$. Then `as.character()` converts π into a string `"3.14159265358979"`, which is then split into single characters `"3" "." "1" "4" "1" "5"` ... via the function `strsplit()`. The `strsplit()` returns a nested vector (or a list) where the first element is a vector containing those single characters. The `[[1]][1:10}` then selects the first ten elements of the first vector in the nested vector, which are the ***characters*** `"3" "." "1" "4" "1" "5" "9" "2" "6" "5"`.