Assignment 2: Coding Basics

Jinglin Zhang

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on coding basics.

Directions

1. Rename this file `<FirstLast>_A02_CodingBasics.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.
6. After Knitting, submit the completed exercise (PDF file) to Sakai.

Basics Day 1

1. Generate a sequence of numbers from one to 100, increasing by fours. Assign this sequence a name.

2. Compute the mean and median of this sequence.

3. Ask R to determine whether the mean is greater than the median.

4. Insert comments in your code to describe what you are doing.

```
#1.
sequ <- seq(1,100,4) # generate sequence

#2.
mean(sequ) # calculate the mean of sequence
```

```
## [1] 49
```

```
median(sequ) # calculate the median of sequence
```

```
## [1] 49
```

```
#3.
judger <- function (x){       # create a function to compare the mean and median
  if (mean(x) > median(x)){
    TRUE                      # if mean is greater than median, return 'TRUE'
  }
  else {
    FALSE                     # otherwise return 'FALSE'
  }
}
judger(sequ)
```

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

Basics Day 2

5. Create a series of vectors, each with four components, consisting of (a) names of students, (b) test scores out of a total 100 points, and (c) whether or not they have passed the test (TRUE or FALSE) with a passing grade of 50.
6. Label each vector with a comment on what type of vector it is.
7. Combine each of the vectors into a data frame. Assign the data frame an informative name.
8. Label the columns of your data frame with informative titles.

```
#install.packages('randomNames')
library(randomNames)
a <- randomNames(20) # character

b <- rnorm(20,60,15)
b <- round(b) # numeric

test <- function(x){
  ifelse(x<50,FALSE,TRUE)
}
c <- test(b) # logical

score <- data.frame(a,b,c)
colnames(score) <- c('Name','Score','Result')
score
```

```
##                           Name Score Result
## 1               Valdoria, John    43  FALSE
## 2                Minor, Mariah    45  FALSE
## 3                Bellamy, Reva    55   TRUE
## 4                    Lee, Jade    67   TRUE
## 5            Kargar, Stephanie    65   TRUE
## 6  Arellano Rodriguez, Antonio    37  FALSE
## 7               White, Shannon    68   TRUE
## 8               Wood, Danielle    61   TRUE
## 9              Aebischer, Anne    77   TRUE
## 10               Ezell, Isioma    57   TRUE
## 11             Medina, Jessica    72   TRUE
## 12           al-Abood, Sameera    56   TRUE
## 13           al-Azimi, Nabeeha    53   TRUE
## 14               Casias, Sofia    52   TRUE
## 15      Garcia-Wideman, Unwana    68   TRUE
## 16       el-Ozer, Abdul Kareem    48  FALSE
## 17               Percell, Sage    78   TRUE
## 18              Abrams, Teshawn    63   TRUE
## 19        Alexander, Dominique    72   TRUE
## 20              Howard, Jordan    64   TRUE
```

9. QUESTION: How is this data frame different from a matrix?

   Answer: both data frame and matrix can store two-dimensinal data in R, but matrix can only store one data type, while data frame can store multiple data types.

10. Create a function with an if/else statement. Your function should take a **vector** of test scores and print (not return) whether a given test score is a passing grade of 50 or above (TRUE or FALSE). You will need to choose either the `if` and `else` statements or the `ifelse` statement.
11. Apply your function to the vector with test scores that you created in number 5.

```
test <- function(x){
  ifelse(x<50,FALSE,TRUE)
}
c <- test(b)
```

12. QUESTION: Which option of `if` and `else` vs. `ifelse` worked? Why?

    Answer: ifelse worked, using if and else will reture the error "Error in if (x < 50) { : the condition has length > 1".the error occures because the if() function can only check 1 element in a vector, but we attemped to check the whole 20 elements in the vector.