Assignment 2: Coding Basics

Shiqi Zheng

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, Part 1

1. Generate a sequence of numbers from one to 30, increasing by threes. 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.
seq1 <- seq(1, 30, 3) # generate sequence of numbers
seq1 # print the sequence
```

```
##  [1]  1  4  7 10 13 16 19 22 25 28
```

```
#2.
mean <- mean(seq1) # compute mean of sequence
median <- median(seq1) # compute median

#3.
if (mean > median) {    #comparing median and mean
  print("mean is greater than the median")
} else if (mean < median) {
  print("mean is smaller than the median")
} else {
  print("mean is equal to the median")
}
```

```
## [1] "mean is equal to the median"
```

Basics, Part 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.

```r
#5.
name <- c("Alice","Bruce","Cindy","David") # character
score <- c(60,80,90,40) # numeric
pass <- c(TRUE,TRUE,TRUE,FALSE) # logical

#6.
 print(paste0("name is a ", class(name), " vector."))
```

```
## [1] "name is a character vector."
```

```r
 print(paste0("score is a ", class(score), " vector." ))
```

```
## [1] "score is a numeric vector."
```

```r
 print(paste0("pass is a ", class(pass), " vector."))
```

```
## [1] "pass is a logical vector."
```

```r
#7/8.
student_test_results <- data.frame("Name"=name,"Score"=score, "Pass"=pass)
student_test_results
```

```
##    Name Score  Pass
## 1 Alice    60  TRUE
## 2 Bruce    80  TRUE
## 3 Cindy    90  TRUE
## 4 David    40 FALSE
```

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

   Answer: Data frame can store different data types, while matrix can only store the same data type. The data frame also has column and row name.

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.

```
# `ifelse`
ifelse(score >= 50, TRUE, FALSE)
```

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

```
# if want to use `if` and `else`
for(i in 1:4) {
if (score[i] >= 50) {
  pass[i] <- TRUE
} else {
  pass[i] <- FALSE
}
  return(pass)
}
pass
```

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

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

    Answer: `ifelse` works for this situation. Because the if() statement can only check one element in a vector at one time, but here we have a vector of score that has more then one element.