Assignment 2: Coding Basics

Iddrisu Sharu Deen

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.

```r
num_seq <- seq(from = 1, to = 30)
num_seq
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30
```

2. Compute the mean and median of this sequence.

```r
num_seq.mean <- mean(num_seq)
num_seq.mean
```

```
## [1] 15.5
```

```r
num_seq.median <- median(num_seq)
num_seq.median
```

```
## [1] 15.5
```

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

```r
num_seq.mean > num_seq.median
```

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

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

```r
#1. I used the seq() function to generate the sequence of numbers with the from, to arguments denoting

#2. I passed num_seq to the mean() funtion to generate the mean, and saved the result in the variable n

#3. I passed num_seq to the median() funtion to generate the mean, and saved the result in the variable
```

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.

```
student_names <- c("John", "Jane", "Jill", "Jim")
test_scores <- c(70, 49, 65, 45)
passed <- ifelse(test_scores > 50, TRUE, FALSE)
```

6. Label each vector with a comment on what type of vector it is.

```
# student_names is a character vector
class(student_names)
```

```
## [1] "character"
```

```
# test_scores is a numeric vector
class(test_scores)
```

```
## [1] "numeric"
```

```
#passed is a logical vector
class(passed)
```

```
## [1] "logical"
```

7. Combine each of the vectors into a data frame. Assign the data frame an informative name.

```
student_performance <- data.frame(student_names, test_scores, passed)
student_performance
```

```
##   student_names test_scores passed
## 1          John          70   TRUE
## 2          Jane          49  FALSE
## 3          Jill          65   TRUE
## 4           Jim          45  FALSE
```

8. Label the columns of your data frame with informative titles.

```
names(student_performance) <- c("Student Names", "Test Scores", "Passed")
student_performance
```

```
##   Student Names Test Scores Passed
## 1          John          70   TRUE
## 2          Jane          49  FALSE
## 3          Jill          65   TRUE
## 4           Jim          45  FALSE
```

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

   Answer: Data frames can contain multpile datatypes whereas matrices can only contain one datatype.

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.

```
myfunc <- function(x){
  ifelse(x > 50, TRUE, FALSE)}
```

11. Apply your function to the vector with test scores that you created in number 5.

```
myfunc(test_scores)
```

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

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

    Answer: The -ifelse- option is better because it uses fewer lines of code compared to the -if- and -else-