Assignment 2: Coding Basics

Felipe Raby Amadori

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics (ENV872L) on coding basics in R.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Use the lesson as a guide. It contains code that can be modified to complete the assignment.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document. Space for your answers is provided in this document and is indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">". You should notice that the answer is highlighted in green by RStudio.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file. You will need to have the correct software installed to do this (see Software Installation Guide) Press the `Knit` button in the RStudio scripting panel. This will save the PDF output in your Assignments folder.
6. After Knitting, please submit the completed exercise (PDF file) to the dropbox in Sakai. Please add your last name into the file name (e.g., "Salk_A02_CodingBasics.pdf") prior to submission.

The completed exercise is due on Thursday, 24 January, 2019 before class begins.

Basics Day 1

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

```
hundred_sequence_byfour <- seq(1, 100, 4) # from, to, by
hundred_sequence_byfour # printing the result
```

```
##  [1]  1  5  9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89
## [24] 93 97
```

2. Compute the mean and median of this sequence.

```
mean_sequ <- mean(hundred_sequence_byfour) # computing the mean and assigning it to an object
mean_sequ # printing the mean
```

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

```
median_sequ <- median(hundred_sequence_byfour) # computing the median and assigning it to an object
median_sequ # printing the median
```

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

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

```
mean_sequ > median_sequ # conditional statement
```

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

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

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.

```
#creating vectors
names_students <- c("Elias","Gabriel","Tamara", "Felipe") # character vector, names of students
names_students
```

```
## [1] "Elias"   "Gabriel" "Tamara"  "Felipe"
```

```
test_scores <- c(98, 97, 99, 40) # numeric vector, test scores
test_scores
```

```
## [1] 98 97 99 40
```

```
pass_status <- c(TRUE,TRUE,TRUE,FALSE) #logical vector, passed the test
pass_status
```

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

```
Testscores_data <- data.frame(names_students, test_scores, pass_status) # creates a dataframe
                                                                        # called Testscores_data with
                                                                        # each of the vectors combined


names(Testscores_data) <- c("Name","Score","Passed")  # Labels the columns of the data frame
                                                      # with informative titles

Testscores_data  #shows the dataframe in a window
```

```
##      Name Score Passed
## 1   Elias    98   TRUE
## 2 Gabriel    97   TRUE
## 3  Tamara    99   TRUE
## 4  Felipe    40  FALSE
```

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

   ANSWER: The data frame is different from a matrix because a matrix only contains elements of the same type. The Testscores_data data frame has character, numeric, and logical elements in it.

10. Create a function with an if/else statement. Your function should determine whether a 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. The name of your function should be informative.

```
Passing_Function <- function(x) { #creates a function that determine whether a
                                  #test score is a passing grade of 50 or above (TRUE or FALSE)
  if(x < 50) {
    result <- FALSE
  }
  else {
    result <- TRUE
```

```
  }
}
```

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

```
pass_status_2 <- sapply(test_scores, Passing_Function)  # sapply applies the Passing_Function
                                                        # to the test_scores vector and creates
                                                        # a vector with the logic results.
pass_status_2
```

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

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

    ANSWER: It worked with if and else because we had two options, TRUE and FALSE, and two ranges of values, from 0 to 50 not including 50, and from 50 to 100.