Assignment 2: Coding Basics

Gaby Czarniak

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. Functions: sequence
# specifying from and to
# specifying by intervals of three
seq(1, 30, 3)
```

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

```
# assigning name to sequence
thirty_sequence <- seq(1, 30, 3)

#2. Summary statistics
# computing mean of sequence
mean(thirty_sequence)
```

```
## [1] 14.5
```

```r
# computing median of sequence
median(thirty_sequence)
```

```
## [1] 14.5
```

```r
#3. Conditional statements
# asking R to determine whether the mean of the sequence is greater than the median
(mean(thirty_sequence)) > median((thirty_sequence))
```

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

```r
# returning "false", R determines that mean is not greater than 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 & #6 Creating vectors and labeling them
# creating object as vector
# consisting of (a) names of students
student_names <- as.vector(c("ali", "ben", "cara", "diego"))

# creating object as vector
# consisting of (b) test scores out of a total 100 points
test_scores <- as.vector(c(40, 70, 80, 90))

# creating object as vector
# consisting of (c) whether the student has passed
# with a passing grade of 50
passed_test <- as.vector(c(FALSE, TRUE, TRUE, TRUE))

#7 Combining vectors into a data frame
# using as.data.frame() to convert something to a data frame
# versus start data frame from scratch
# using names as first column for data frame
# so I can look up students by name
df_student_names <- as.data.frame(student_names)
df_student_names
```

```
##   student_names
## 1           ali
## 2           ben
## 3          cara
## 4         diego
```

```r
class(df_student_names)
```

```
## [1] "data.frame"
```

```r
# Adding the other vectors using Option 1 in 02Lab_Intro_DataFrame.Rmd by Luana found here:
# https://warpwire.duke.edu/w/SxsFAA/
# binding the test_scores and passed_test vectors to the df
# labeling the all-up df such that it represents all columns, not just student names
df_class_score_status <- cbind(df_student_names, test_scores, passed_test)
class(df_class_score_status)
```

```
## [1] "data.frame"
```

```r
df_class_score_status
```

```
##   student_names test_scores passed_test
## 1           ali          40       FALSE
## 2           ben          70        TRUE
## 3          cara          80        TRUE
## 4         diego          90        TRUE
```

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

   Answer: As per the class notes for Week 2 (https://env872.github.io/assets/files/M2/EDA_W3_Coding_Reproducibility_Lab.pdf), we started by making vectors, which are one-dimensional sequences of data elements of the same data type. Then we made a data frame using those one-dimensional vectors of the same length, but the vectors differed from one another in data type. Matrices are two-dimensional sequences of data elements of the same type. While both this data frame and matrices store data two-dimensionally, in matrices, all rows and columns must be of the same data type, whereas in this data frame, the data within vectors is of the same type, but from vector to vector (or column to column) the data type differs (character, numeric, etc.).

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.

```r
#10 Creating function with if/else statement
true_false_fxn <- function(x) {
  ifelse(x<50, FALSE, TRUE)
  # including logical expression -- if x < 50
  # then FALSE should happen -- they failed
  # otherwise TRUE should happen -- they passed
}

# printing function results of whether it's true/false a given student passed and
#11 Applying function to test scores vector created in #5
fxn_passed_test <- print(true_false_fxn(test_scores))
```

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

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

    Answer: The option of 'if' and 'else' did work successfully on a single test score inputted (as the number value), but did not work when run over the entire test_scores vector as input. The 'ifelse' option worked for both. Then I tested making a vector with just one value stored in it and running both 'if' and 'else' and 'ifelse' options on that vector. In that case, both options worked, which which must mean that 'if' and 'else' cannot take vectors as inputs when multiple values are stored in that vector, whereas 'ifelse' can.