Assignment 2: Coding Basics

Laura Exar

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.

Sequence <- seq(1,100,4) #generated a sequence of numbers from 1 to 100, by 4
Sequence
```

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

```
#2.

Mean <- mean(Sequence) #computed the mean of the sequence
Mean
```

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

```
Median <- median(Sequence) #computed the median of the sequence
Median
```

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

```
#3.

Mean > Median #determined if the mean if greater than the median
```

```
## [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.

```
student_names <- c("Laura Exar", "Aileen Lavelle", "Maeve Arthur", "Allison Barbaro") #created a vector
typeof(student_names) #character
```

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

```
student_names
```

```
## [1] "Laura Exar"      "Aileen Lavelle"  "Maeve Arthur"    "Allison Barbaro"
```

```
test_scores <- c(100, 96, 98, 96) #created a vector of test scores
typeof(test_scores) #double
```

```
## [1] "double"
```

```
test_scores
```

```
## [1] 100  96  98  96
```

```
passing_test <- c("TRUE", "TRUE", "TRUE", "TRUE") #created a vector of whether or not the students pass
typeof(passing_test) #character
```

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

```
passing_test
```

```
## [1] "TRUE" "TRUE" "TRUE" "TRUE"
```

```
student_test_scores <- data.frame(student_names, test_scores, passing_test) #created a dataframe with a
student_test_scores
```

```
##     student_names test_scores passing_test
## 1      Laura Exar         100         TRUE
## 2  Aileen Lavelle          96         TRUE
## 3    Maeve Arthur          98         TRUE
## 4 Allison Barbaro          96         TRUE
```

```
colnames(student_test_scores) <- c("Names", "Scores", "Pass/Fail") #added column names to the dataframe
student_test_scores
```

```
##               Names Scores Pass/Fail
## 1      Laura Exar    100      TRUE
## 2  Aileen Lavelle     96      TRUE
## 3    Maeve Arthur     98      TRUE
## 4 Allison Barbaro     96      TRUE
```

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

   Answer: A data frame can be made up of several different classes, while a matrix is only composed of one class of data.

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.

```
#if/else function that determines if the students passed or failed
passed_test2 <- function(x){
  ifelse(x > 50, "TRUE", "FALSE")
}

grades2 <- passed_test2(test_scores); grades2
```

```
## [1] "TRUE" "TRUE" "TRUE" "TRUE"
```

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

    Answer: The 'ifelse' function worked, while the 'if' and 'else' function did not. The 'if' and 'else' function can only look at one vector element a time, while 'ifelse' can look at all of the vector elements at once. Our vector had multiple elements, which is why the 'ifelse' function worked, while the 'if' and 'else' function did not. The 'ifelse' function is also more efficient, as it uses one line of code rather than several.