Assignment 2: Coding Basics

Caroline Watson

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.
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.

```
#creating a sequance from 1 to 100 increasing by 4
seq(1, 100, 4)
```

```
##  [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
```

```
#assigning a name to the sequence
one_to_100_by_4 <- seq(1, 100, 4)

#calling the sequence by it's name
one_to_100_by_4
```

```
##  [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.

```
#calculating the mean of the sequence
mean(one_to_100_by_4)
```

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

```
#calculating the median of the sequence
median(one_to_100_by_4)
```

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

3.

```
#seeing if the mean of the sequence is greater than the median of the sequence
mean(one_to_100_by_4) > median(one_to_100_by_4)
```

```
## [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.

5. & 6.

```
#creating a vector with names of students
names_students <- c("Jane", "John", "Anna", "Caroline") #character vector
names_students
```

```
## [1] "Jane"     "John"     "Anna"     "Caroline"
```

```
#creating a vector with test scores out of a total of 100 points
test_scores <- c("85", "90", "60", "70") #numeric vector
test_scores
```

```
## [1] "85" "90" "60" "70"
```

```
#did students pass the test with passing grade of 50?
passed_test <- test_scores > 50  #logical vector
passed_test
```

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

7.

```
#creating a data frame with all of the vectors above
dataframe_vectors <- data.frame(names_students, test_scores, passed_test)
dataframe_vectors
```

```
##   names_students test_scores passed_test
## 1           Jane          85        TRUE
## 2           John          90        TRUE
## 3           Anna          60        TRUE
## 4       Caroline          70        TRUE
```

8.

```
#renaming column headings for the data frame
names(dataframe_vectors) <- c("Names", "Test Score", "Passed")
dataframe_vectors
```

```
##      Names Test Score Passed
## 1     Jane         85   TRUE
## 2     John         90   TRUE
## 3     Anna         60   TRUE
## 4 Caroline         70   TRUE
```

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

   ANSWER: A data frame is different from a matrix because a matrix contains elements of the same type and a data frame can contain elements of different types (i.e. numeric, factors, etc.).

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.

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

10.

```
#creating a function with an 'if' and 'else' statement
passed_test_function <- function(passed_test){
  if(passed_test > 50) {
    return("TRUE")
  }
else {
  return("FALSE")
}
  }


#creating a function with an 'ifelse' statement
passed_test_function2 <- function(passed_test){
  ifelse(passed_test > 50, "TRUE", "FALSE")
}
```

11.

```
#testing the function to see if it works
passed_test_function(60)
```

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

```
passed_test_function(40)
```

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

```
#testing the function with an 'ifelse' statement to see if it works
passed_test_function2(60)
```

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

```
passed_test_function2(40)
```

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

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

    ANSWER: From the above code we can see that 'if' and 'else' and 'ifelse' both worked to determine whether the statement was true or false.