Assignment 2: Coding Basics

Keith Bollt

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.

```
sequence_for_q1 <- seq(1,100)# I am naming and defining my sequence
sequence_for_q1# I am telling R to make my sequence now.
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
##  [18]  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
##  [35]  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51
##  [52]  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68
##  [69]  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
##  [86]  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
```

```
median (sequence_for_q1)
```

```
## [1] 50.5
```

```
mean(sequence_for_q1)# I am taking the median and mean, respectively, for the sequence I just created.
```

```
## [1] 50.5
```

```
mean (sequence_for_q1) > median(sequence_for_q1) # This is a logical function. Because the answer is fa
```

```
## [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.

```
vector_names <- c("Keith","Steven","Juan","Quentin") # character vector
vector_names
```

```
## [1] "Keith"   "Steven"  "Juan"    "Quentin"
```

```
vector_score <- c(49.998, 95, 90, 92) #numeric vector
vector_score
```

```
## [1] 49.998 95.000 90.000 92.000
```

```
vector_pass <- c(FALSE, TRUE, TRUE, TRUE) #logical vector
vector_pass
```

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

```
Gradebook <- data.frame(vector_names, vector_score, vector_pass)
Gradebook
names(Gradebook) <- c("Name", "Grade", "Did They Pass?")
View(Gradebook)
```

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

   ANSWER: A matrix has information of the same type, be it numerical, character or logical, in each column. A data frame can have more than one type of information 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.

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

```
Who_Should_I_Pass <- function(y) {
if (y>=50)
  {TRUE}
  else {FALSE}      #This function did not work
}
Who_Should_I_Pass (vector_score)
```

```
## Warning in if (y >= 50) {: the condition has length > 1 and only the first
## element will be used
```

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

```
Who_Should_I_Pass2 <- function(z) {
  ifelse(z>= 50, TRUE, FALSE)     # But this one did!
}
Who_Should_I_Pass2(vector_score)
```

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

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

ANSWER: Only ifelse worked. I was using the function on a vector, and the ‘if’ and ‘else’ function does not work on vectors. As the warning indicates, it was only able to solve the function for the first of the four scores.