Assignment 2: Coding Basics

Alicia Zhao

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on coding basics.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Salk_A02_CodingBasics.Rmd") prior to submission.

The completed exercise is due on Tuesday, January 21 at 1:00 pm.

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.
#Generating a sequence from 1 to 100 by 4
seq_hundred=seq(1,100,4); seq_hundred
```

```
##  [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.
#Generating the mean of seq_hundred
mean(seq_hundred)
```

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

```
#Generating the median of seq_hundred
median(seq_hundred)
```

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

```
#3.
#Seeing if the mean is greater than the median of seq_hundred
mean(seq_hundred)>median(seq_hundred)
```

```
## [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.

```
#creating vectors for name of students, test scores, and whether or not they pass the test
student_name <- c("Stacy", "Tom", "Jillian", "Roger") #character vector
test_score <- c(77, 85, 43, 90) #numeric vector
test_pass <- c(TRUE, TRUE, FALSE, TRUE) #logical vector

#creating a data frame containing the vectors above
testscores_df <- data.frame (student_name, test_score, test_pass)
names(testscores_df) <- c("Student", "Score", "Pass"); testscores_df
```

```
##   Student Score  Pass
## 1   Stacy    77  TRUE
## 2     Tom    85  TRUE
## 3 Jillian    43 FALSE
## 4   Roger    90  TRUE
```

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

   Answer: A matrix is a two-dimensional structure that contains elements of the same type, whereas a data frame is more general and can have columns with different modes.

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. Hint: Use `print`, not `return`. The name of your function should be informative.
11. Apply your function to the vector with test scores that you created in number 5.

```
#creating function to determine whether a test score is passing
passtest <- function(x){
    ifelse(x>=50, TRUE, FALSE)
}

passtest2 <- function(x) {
  if(x>=50) {
    TRUE
  }
  else {
    FALSE
  }
}

#appying function to test_scores
passtest(test_score)
```

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

```
passtest2(test_score)
```

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

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

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

    Answer: ifelse worked. 'if' and 'else' did not work because the function requires a length-one logical vector. However, the vector has length >1 and thus only the first element will be used.