Assignment 2: Coding Basics

Taro Katayama

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 first and last name into the file name (e.g., "FirstLast_A02_CodingBasics.Rmd") prior to submission.

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.

```r
#1.
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 93 97
```

```r
hundred_sequence<-seq(1,100,4)
#created sequence beginning from 1, ending in 100, increasing by four
#assigned this function a name hundred_sequence

#2.
mean(hundred_sequence)
```

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

```r
#calculated mean of function hundred_sequence

median(hundred_sequence)
```

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

```r
#calculated median of function hundred_sequence

#3.
mean(hundred_sequence)>median(hundred_sequence)
```

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

```
#created a conditional statement asking if mean of hundred_sequence is greater than median of hundred_s
```

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
test_score<- c(95,70,25,80)
#test scores uses numbers as the mode

student_name<-c("Taro", "Gitmo", "Goop", "EEEEEEEE")
#student_name uses characters as the mode

test_pass_fail<-c(TRUE,TRUE,FALSE,TRUE)
#test_pass_fail uses characters as the mode
#7/8
class_statistics<-data.frame("student name"=student_name, "test score"=test_score, "test results"=test_
class_statistics
```

```
##   student.name test.score test.results
## 1         Taro         95         TRUE
## 2        Gitmo         70         TRUE
## 3         Goop         25        FALSE
## 4     EEEEEEEE         80         TRUE
```

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

   Answer: This data frame is different from a matrix because it includes different modes. Matrices cannot have 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.

```
#10
#if else
pass_fail_function<-function(x) {
  if (x>=50) {x=TRUE}
  else {x=FALSE}
}
#ifelse
pass_fail_function_if_else<-function(x) {
ifelse(x>=50, TRUE, FALSE)
}

#11
passing_grade<-pass_fail_function(test_score)
```

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

```
test_function<-pass_fail_function_if_else(test_score)
print(test_function)
```

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

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

    Answer: 'ifelse' worked for me. It worked because it is a valid logical function, where the test scores met the function demands, where if x is greater than or equal to 50, it would output TRUE, and anything else would output FALSE