Assignment 2: Coding Basics

Emily McNamara

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. sequence of numbers 1 to 100, increasing by fours
seq(1, 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 93 97
```

```
seq1 <-seq( 1, 100, by = 4)
seq1
```

```
##  [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
```

```
## I first created a sequence from 1 to 100 by fours. Then I named the sequence "seq1" and called up "se

#2. Mean and Median of Seq1
mean(seq1)
```

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

```
median(seq1)
```

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

```
## I found the mean and median of seq1 which are both 49.

#3. Is the mean greater than the median?
mean(seq1) > median(seq1)
```

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

```
## I asked R if the mean of seq1 is greater than the median of seq1 and it said False because both are
```

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.

```
# Q5 & Q6: Vectors with Vector Types

vectorstudentnames <- c("bily", "bobby", "sally", "emily") #character vector
vectorstudentnames
```

```
## [1] "bily"  "bobby" "sally" "emily"
```

```
vectortestscores <- c(45, 70, 80, 90) #numeric vector
vectortestscores
```

```
## [1] 45 70 80 90
```

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

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

```
# Q7: Data Frame

dataframetestscores <- data.frame (vectorstudentnames, vectortestscores, vectorpass)
dataframetestscores
```

```
##   vectorstudentnames vectortestscores vectorpass
## 1               bily               45      FALSE
## 2              bobby               70       TRUE
## 3              sally               80       TRUE
## 4              emily               90       TRUE
```

```
# Q8: Labeled Data Frame

names(dataframetestscores) <- c("Student Names", "Test Scores", "Pass/Fail"); View(dataframetestscores)
```

```
## Warning in system2("/usr/bin/otool", c("-L", shQuote(DSO)), stdout = TRUE):
## running command ''/usr/bin/otool' -L '/Library/Frameworks/R.framework/Resources/
## modules/R_de.so'' had status 1
```

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

   Answer: A data frame creates a separate table outside of the console. It is a cleaner format and allows you to add specific information into each column and row rather instead of just one type of element like a matrix. These different types of elements in a data fram allow you to include different modes like numeric and factor.

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.

```r
# Q10: Both versions of if/else statement for whether students passed or failed the test

## Version 1: "if" and "else" statements
x <- 50
x
```

```
## [1] 50
```

```r
testscore <- function(x) {
  if(x < 50) { x = FALSE}
  else {
    x = TRUE
  }
}

# Version 2: "ifelse" statement
y <- 50
y
```

```
## [1] 50
```

```r
testscore2 <- function(y) {
  ifelse(y<50, FALSE, TRUE)
}
testscore2
```

```
## function(y) {
##   ifelse(y<50, FALSE, TRUE)
## }
```

```r
# Q11: Determining which students passed or failed the test
billy <- testscore(45); billy
```

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

```r
billyifelse <- testscore2(45); billyifelse
```

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

```r
bobby <- testscore(70); bobby
```

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

```r
bobbyifelse <-testscore2(70); bobbyifelse
```

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

```r
sally <- testscore(80); sally
```

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

```r
sallyifelse <- testscore2(80); sallyifelse
```

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

```r
emily <- testscore(90); emily
```

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

```
emilyifelse <- testscore2(90); emilyifelse
```

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

```
studenttestscores <-testscore(vectortestscores)
```

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

```
studenttestscores2 <-testscore2(vectortestscores)
studenttestscores2
```

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

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

    Answer: When I broke it down by individual students, both versions worked. However, when I used the vector with the test scores, only the "ifelse" statement worked. I think this is because in the "if" and "else" version, only the first element can be used, whereas, in the "ifelse" statement, you can use a list of elements.