Assignment 2: Coding Basics

Samantha Burch

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. seq(1, 100, 4) # 1-100 by 4 ans: [1]  1  5  9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81
seq1 <- seq(1, 100, 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
```

```
#2.
mean(seq(1, 100, 4)) # ans: 49
```

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

```
median(seq(1, 100, 4)) # ans: 49
```

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

```
#3.
mean(seq(1, 100, 4))>median(seq(1, 100, 4)) # ans: FALSE
```

```
## [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 and 6
vectorstudents <- c("Sam", "Carlen", "Rachel", "Ian") # character vector
vectorstudents
```

```
## [1] "Sam"    "Carlen" "Rachel" "Ian"
```

```
vectorgrades <- c(100,98,92,85) # numeric vector
vectorgrades
```

```
## [1] 100  98  92  85
```

```
vectorpassfail <- c(TRUE, TRUE, TRUE, TRUE) # logical vector
vectorpassfail
```

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

```
#7
dataframegrades <- data.frame(vectorstudents, vectorgrades, vectorpassfail)
dataframegrades #created data frame using each of my named vectors
```

```
##   vectorstudents vectorgrades vectorpassfail
## 1            Sam          100           TRUE
## 2         Carlen           98           TRUE
## 3         Rachel           92           TRUE
## 4            Ian           85           TRUE
```

```
#8
names(dataframegrades) <- c("Students", "Grades", "Pass/Fail"); View (dataframegrades)
```

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

   Answer: A data frame is different from a matrix in that each column can be assigned a different mode (i.e. numbers in one, characters in another). Matrices only show one element/the same type of data.

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 and else)
x <- 50
x
```

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

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

#ifelse
```

```
y <- 50
y
```

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

```
grade2 <- function(y) {
  ifelse(y<50, FALSE, TRUE)
}
grade2 #TRUE, TRUE, TRUE, TRUE
```

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

```
#11: results
Sam <- grade(100); Sam
```

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

```
Samifelse <- grade(100); Samifelse
```

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

```
Carlen <- grade(98); Carlen
```

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

```
Carlenifelse <- grade(98); Carlenifelse
```

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

```
Rachel <- grade(92); Rachel
```

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

```
Rachelifelse <- grade(92); Rachelifelse
```

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

```
Ian <- grade(85); Ian
```

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

```
Ianifelse <- grade(85); Ianifelse
```

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

```
grade <- grade(vectorgrades)
```

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

```
grade2 <- grade2(vectorgrades)
grade2
```

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

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

    Answer: The ‘if’ and ‘else’ option worked, as I was able to apply my vector function to each individual student to determine a passing grade - these outputs appeared in both the console and the ‘Environment’ under values. When I ran the ifelse option, the following message appeared: “Warning message: In if (x < 50) { : the condition has length > 1 and only the first element will be used.” However, when I run ‘grade2’ on its own, I do see the following output: “[1] TRUE TRUE TRUE TRUE.” That said, it allows me to see all answers at once.