Assignment 2: Coding Basics

Katryna Niva

OVERVIEW

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

Directions

1. Rename this file `<FirstLast>_A02_CodingBasics.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.
6. After Knitting, submit the completed exercise (PDF file) to Sakai.

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.
FirstExercise <- seq(1,100,4)
# Here I am creating a sequence using the seq function where the first number is the starting value, th

#2.
mean(FirstExercise) -> MeanFirstExercise
median(FirstExercise) -> MedianFirstExercise
# Here I am determining the mean and median of the previously created sequence using the mean and media

#3.
MeanFirstExercise>MedianFirstExercise
```

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

```
# Here I am testing the notion that the mean is greater than the mdedium. The output was False because
```

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.

```
students <- c(a = "anna", b = "brian", c = "cat", d = "dave")
#character vector
str(students)
```

```
##  Named chr [1:4] "anna" "brian" "cat" "dave"
##  - attr(*, "names")= chr [1:4] "a" "b" "c" "d"
```

```
grades <- c("anna" = 60, "brian" = 40, "cat" = 80, "dave" = 70)
#number vector
str(grades)
```

```
##  Named num [1:4] 60 40 80 70
##  - attr(*, "names")= chr [1:4] "anna" "brian" "cat" "dave"
```

```
passing <- grades > 50
#logical vector
str(passing)
```

```
##  Named logi [1:4] TRUE FALSE TRUE TRUE
##  - attr(*, "names")= chr [1:4] "anna" "brian" "cat" "dave"
```

```
test_outcome <- data.frame(students, grades, passing)

print(test_outcome)
```

```
##   students grades passing
## a     anna     60    TRUE
## b    brian     40   FALSE
## c      cat     80    TRUE
## d     dave     70    TRUE
```

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

   Answer: All data within a matrix must be of the same data type. In contrast, a data frame can have different collumns with different data types. Both are two dimensional.

10. Create a function with an if/else statement. Your function should take a **vector** of test scores and print (not return) whether a given 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.
11. Apply your function to the vector with test scores that you created in number 5.

```
gradebook <- function(x)
{
ifelse(x>=50, TRUE, FALSE)
}

gradebook(grades)
```

```
##  anna brian   cat  dave
##  TRUE FALSE  TRUE  TRUE
```

```
print(gradebook(grades))
```

```
##  anna brian   cat  dave
##  TRUE FALSE  TRUE  TRUE
```

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

print(badoption(20))
```

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

```
print(badoption(70))
```

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

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

    Answer: Only the 'ifelse' option worked. That is because with this function you can input an entire vector which is what we are looking to do. The 'if' and 'else' option only works with the input of individual values.