Assignment 2: Coding Basics

Isaac Benaka

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.

```
count4 <- seq(1,100,4)
```

2. Compute the mean and median of this sequence.

```
mean(count4)
```

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

```
median(count4)
```

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

3. Ask R to determine whether the mean is greater than the median.

```
mean(count4) > median(count4)
```

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

4. Insert comments in your code to describe what you are doing.

```
#1. Use sequence function to list numbers 1-100, increasing by fours at each step. Assigned this sequen

#2. Calculated the mean and median values of the count4 sequence.

#3. Used the conditional statement ">" to determine whether the mean of count4 was greater than the med
```

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.

```
studentname <- c("John","Lisa","Jordan","Hana")
testscore <- c(42,83,75,100)
passed <- c(FALSE,TRUE,TRUE,TRUE)
```

6. Label each vector with a comment on what type of vector it is.

studentname is a character vector, testscore is a numeric vector, and passed is a logical vector.

7. Combine each of the vectors into a data frame. Assign the data frame an informative name.

```
gradebook <- data.frame(studentname,testscore,passed)
```

8. Label the columns of your data frame with informative titles.

```
names(gradebook) <- c("Students","Grade","Passed")
```

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

   Answer: My dataframe contains multiple classes of data — like words, integers, and booleans — which cannot be done in a matrix.

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.

```
# version with if and else
checkgrade1 <- function(testscore) {
  if(testscore > 50) {
    TRUE
  }
  else {
    FALSE
  }
}
```

```r
# version with ifelse
checkgrade2 <- function(testscore){
  ifelse(testscore > 50, TRUE, FALSE)
}
```

11. Apply your function to the vector with test scores that you created in number 5.

```r
# checkgrade1(testscore) This command resulted in an error
checkgrade2(testscore)
```

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

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

    Answer: 'ifelse' worked and 'if' and 'else' did not. 'if' and 'else' did not work because it tried to run the function for each value in the vector when it can only handle one value at a time.