Assignment 2: Coding Basics

Danlei Zou

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.
#creating sequence of numbers from 1 to 100, increasing by 4
seqIncrease4 <- seq(1, 100, 4)
print(seqIncrease4)
```

```
##  [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.

#computing the mean and median of this sequence
mean(seqIncrease4)
```

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

```
median(seqIncrease4)
```

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

```
#3 asking R to determine if mean is greater than median
mean(seqIncrease4)<median(seqIncrease4)
```

```
## [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 creating series of vectors, each with four components consisting of (a) names of students, (b) test
#6 label each vector with a comment on what type of vector it is, shown next to each vector

names <- c('Marianne', 'Felix', 'Ingrid', 'Claude') #character vector
names
```

```
## [1] "Marianne" "Felix"    "Ingrid"   "Claude"
```

```
scores <- c(90, 60, 45, 70) #numeric vector
scores
```

```
## [1] 90 60 45 70
```

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

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

```
#7 combining each vectors into a data frame, assigning data frame informative name
#8 labelling columns of data frame with informative titles, shown below

#creating the dataframe
testScoreResults <- data.frame(names,scores,passed)
names(testScoreResults) <- c("Name", "Score", "Passed?"); View(testScoreResults)

#printing the dataframe
testScoreResults
```

```
##       Name Score Passed?
## 1 Marianne    90    TRUE
## 2    Felix    60    TRUE
## 3   Ingrid    45   FALSE
## 4   Claude    70    TRUE
```

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

   Answer: In a matrix, all of the elements have to be the same type of data but in a data frame the columns can have different types of data.

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.

```r
#creating 'if' 'else' function to indicate TRUE or FALSE for test scores
#testScorePass <- function(score) {
# if (score < 50) {FALSE}
#  else {TRUE}
#  }
#applying 'if' 'else' function to test score vector from number 5
#testScorePass(scores)

#'if' 'else' function above doesn't print a result, can't check conditions with more than one element

#creating 'ifelse' function to indicate TRUE or FALSE for test scores
testScorePass2 <- function(score)
  {ifelse(score < 50, "FALSE", "TRUE")}

#applying 'ifelse' function to test score vector from number 5
testScorePass2(scores)
```

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

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

    Answer: 'ifelse' worked because unlike 'if' and 'else', it can check more than one element for the conditions set. 'if' and 'else' can only check conditions for one element, so it couldn't check the conditions for the entire vector.