Assignment 2: Coding Basics

Andrea Gonzalez Natera

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 first and last name into the file name (e.g., "FirstLast_A02_CodingBasics.Rmd") prior to submission.

Basics Day 1

1. Generate a sequence of numbers from one to 100, increasing by fours. Assign this sequence a name.

```
#1. I created a sequence from 1 to 100 in intervals of 4  using the seq() function and called this Seq1
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. Compute the mean and median of this sequence.

```
#2. Used the mean() and median() functions on Seq1
mean(Seq1)
```

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

```
median(Seq1)
```

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

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

```
#3. Determined if mean of Seq1 was bigger than median by using the > sign
mean(Seq1)>median(Seq1)
```

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

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

```
#1. I created a sequence from 1 to 100 in intervals of 4  using the seq() function and called this Seq1

#2. Used the mean() and median() functions on Seq1

#3. Determined if mean of Seq1 was bigger than median by using the > sign
```

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.

```
sName <- c('Ale','Sabine','Mariana','Andrea')
#sName is a character vector
sName
```

```
## [1] "Ale"     "Sabine"  "Mariana" "Andrea"
```

```
sScore <- c(90,85,50,95)
sScore
```

```
## [1] 90 85 50 95
```

```
#sScore is a numeric vector
sPassed <- c(TRUE,TRUE,FALSE,TRUE)
sPassed
```

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

```
#sPassed is a logical vector
```

6. Label each vector with a comment on what type of vector it is. Answer: See above

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

```
TestResults <- data.frame(sName,sScore,sPassed)
TestResults
```

```
##     sName sScore sPassed
## 1     Ale     90    TRUE
## 2  Sabine     85    TRUE
## 3 Mariana     50   FALSE
## 4  Andrea     95    TRUE
```

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

```
TestResults <- data.frame(Name = sName, Score = sScore, Result = sPassed)
TestResults
```

```
##      Name Score Result
## 1     Ale    90   TRUE
## 2  Sabine    85   TRUE
## 3 Mariana    50  FALSE
## 4  Andrea    95   TRUE
```

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

   Answer: This dataframe has three different types of data (numeric,character,logistic) a matrix can only have one 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.

```
PassFailIfElse <- function(x){
  ifelse(x>50,TRUE,FALSE)
}
PassFailIf <- function(x) {
```

```
  if (x[1]>50 & x[2]>50 & x[3]>50 & x[4]>50) {c(TRUE,TRUE,TRUE,TRUE)} else {c(x[1]>50,x[2]>50,x[3]>50,x
}
```

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

```
StudentPassedIfElse <- as.data.frame(PassFailIfElse(sScore))
StudentPassedIfElse
```

```
##   PassFailIfElse(sScore)
## 1                   TRUE
## 2                   TRUE
## 3                  FALSE
## 4                   TRUE
```

```
StudentPassedIf <- as.data.frame(PassFailIf(sScore))
StudentPassedIf
```

```
##   PassFailIf(sScore)
## 1               TRUE
## 2               TRUE
## 3              FALSE
## 4               TRUE
```

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

    Answer: I think ifelse worked better because it evaluated the every item in the vector, in the 'ife' and 'else' example I had to input multiple conditions so that the lenght would be bigger than 1, in this case that all of the grades where above 50, and in the else statment I simply asked it to return a vectors whose values where the logic result of that item in the vector's relationship with 50 (bigger = TRUE, smaller = False) This is not an elegant solutions and in terms of time it is the same as creating a logistical vector>: c(TestResults$Score > 50)