Assignment 2: Coding Basics

Kathleen Mason

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]  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
```

```
#creating the sequence from 1 to 100 and counting by 4's
NumberList<- c(seq(1, 100, 4))
#Naming the sequence
NumberList
```

```
##  [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
```

```
#calling up the number list to see if it is right
#2.
mean(NumberList) #calculating the mean of the sequence
```

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

```
median(NumberList) #calculating the median of the sequence
```

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

```
#3.
mean(NumberList) > median(NumberList)
```

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

```r
#Asking if the mean is greater than the median,
#this came out as FALSE which means it is not
#greater than the mean, which is true becasue they are equal
mean (NumberList) == median(NumberList)
```

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

```r
#practicing my coding to make sure it is
#true that they are actually equal
```

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.

```r
vectorNames<- c ("Rachel", "Jake", "Jack", "MC")
#Character vector
vectorScores<- c (99, 100, 98, 48)
#Numerical Vector
vectorPass<- c(TRUE, TRUE, TRUE, FALSE)
#Logical vector

ClassTestResults<- data.frame(vectorNames, vectorScores, vectorPass)
#Creating dataframe with names, scores, and if they passed or not
names(ClassTestResults)<- c("Name", "Score", "Pass?")
#giving columns new names
```

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

   Answer: Data frames can have different modes/forms of information like numbers, text, while matrices have to have it all be the same form.

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.

```r
Testresults<- function(x) {if(x> 50) (x=TRUE) else (x=FALSE)}
Testresults2<-function(x) {ifelse(x>50,TRUE, FALSE)}
# ifelse(test, yes, no)

Testresults2(vectorScores)
```

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

```r
lapply(vectorScores, Testresults2 )
```

```
## [[1]]
## [1] TRUE
##
## [[2]]
```

```
## [1] TRUE
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] FALSE
```

```
lapply(vectorScores, Testresults)
```

```
## [[1]]
## [1] TRUE
##
## [[2]]
## [1] TRUE
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] FALSE
```

```
#lapply(what you want x to be(a vector), the function to use)
#Gives results of the function to the vector
```

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

    Answer: The 'ifelse' statement works best and directly applies the function to a vector. if and else provides an error message saying that the condition has a length greater than 1, therefore it doesn't work.I tried 'lapply' on the if and else function however, and the function does work so I am interested to learn why.