Assignment 2: Coding Basics

Li Jia Go

## 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.  
seq(1, 100, 4) # using sequence function to generate sequence of numbers from 1 to 100 increasing by fours. i.e. seq (from, to, by)

## [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

hundredseq <- seq(1, 100, 4) # generating sequence from 1 to 100, increasing by 4. naming sequence 'hundredseq'  
hundredseq # print result

## [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.  
  
hundredseq\_mean <- mean(hundredseq) # computing the mean of the sequence using the mean function and assigning a name to the result  
hundredseq\_mean # print result

## [1] 49

hundredseq\_median <- median(hundredseq) # computing the median of the sequence using the median function and assigning a name to the result  
hundredseq\_median # print result

## [1] 49

# 3.  
hundredseq\_mean > hundredseq\_median # asking R if mean is greater than median

## [1] FALSE

hundredseq\_mean < hundredseq\_median # asking R if mean is less than median

## [1] FALSE

hundredseq\_mean == hundredseq\_median # asking R if mean and median are equal

## [1] TRUE

hundredseq\_mean != hundredseq\_median # asking R if mean and median are not equal

## [1] FALSE

## Basics Day 2

1. 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.
2. Label each vector with a comment on what type of vector it is.
3. Combine each of the vectors into a data frame. Assign the data frame an informative name.
4. Label the columns of your data frame with informative titles.

# 5.  
student\_names <- c("John", "Peter", "Sarah", "Jill") #creating a vector of student names  
student\_names #print result

## [1] "John" "Peter" "Sarah" "Jill"

student\_scores <- c(80, 90, 40, 48) #creating a vector of student's scores  
student\_scores #print result

## [1] 80 90 40 48

student\_pass <- c(TRUE, TRUE, FALSE, FALSE)  
student\_pass #print result

## [1] TRUE TRUE FALSE FALSE

# 6.  
class(student\_names) #character

## [1] "character"

class(student\_scores) #numeric

## [1] "numeric"

class(student\_pass) #logical

## [1] "logical"

# 7.  
df\_classnames <- as.data.frame(student\_names) # transforming vector into a data frame  
df\_classscores <- as.data.frame(student\_scores) # transforming vector into a data frame  
df\_classpass <- as.data.frame(student\_pass) # transforming vector into a data frame  
  
df\_classtestscores <- cbind(df\_classnames, df\_classscores, df\_classpass) # combining all the columns of the data frame  
df\_classtestscores #print result

## student\_names student\_scores student\_pass  
## 1 John 80 TRUE  
## 2 Peter 90 TRUE  
## 3 Sarah 40 FALSE  
## 4 Jill 48 FALSE

# 8.  
colnames(df\_classtestscores) <- c("names", "scores", "passed") # renaming names of columns of data frame  
df\_classtestscores #print result

## names scores passed  
## 1 John 80 TRUE  
## 2 Peter 90 TRUE  
## 3 Sarah 40 FALSE  
## 4 Jill 48 FALSE

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

Answer: Matrices can only store a single data type, while this data frame consists of different classes of data (i.e. characters, numbers, logic etc)

1. 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.
2. Apply your function to the vector with test scores that you created in number 5.

passfifty <- function(x) {  
 ifelse(x > 50, "TRUE", "FALSE")  
}  
  
answer <- passfifty(student\_scores)  
print(answer) #print result

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

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

Answer: Only ‘ifelse’ worked. The if() statement can only work with elements that are length 1, i.e. a single student. However, by using the ‘ifelse’ expression, we are able to check if the entire class of students passed the test at once.