

# Assignment 2: Coding Basics

*Rachel Landman*

## 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.
seq1 <- seq (1,100,4) ### (from,to,by)

#2.
meanseq1 <- mean(seq1)
### asking r to take the mean of the above sequence, which I labeled "seq1"

medianseq1 <- median(seq1)
### asking r to take the median of the above sequence, which I labeled "seq1"

#3.

meanseq1 > medianseq1

## [1] FALSE
### I am writing a conditional statement and asking R if the mean is greater than
### the median. R says no because they are 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.

```
#5. #6.
students_vector <- c("Anna", "Alison", "Abel", "Alfred")
students_vector ### character vector

## [1] "Anna" "Alison" "Abel" "Alfred"

scores_vector <-c(40, 98, 26, 86)
scores_vector ### numeric vector

## [1] 40 98 26 86

test_vector <- c(FALSE, TRUE, FALSE, TRUE)
test_vector ### logical vector

## [1] FALSE TRUE FALSE TRUE

#7.
testscores_dataframe <-data.frame(students_vector, scores_vector, test_vector)

#8.

names(testscores_dataframe) <- c("Student Name","Score","Passed")
```

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

Answer: Data frames are more general than matrices. This data frame has multiple types of data (numeric, character, logical), which you couldn't do in a matrix. In a matrix the data type would have to be consistent (e.g. all numeric).

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.

```
#10.
passing_grade <- function(x) {if(x<50) (x=FALSE) else (x=TRUE)}

passing_grade(30) ###test the function for x<50

## [1] FALSE

passing_grade(98) ###test the fnection for x>50

## [1] TRUE

###the above outputs shows that the function works

passing_grade2 <- function(x){
  ifelse(x<50, FALSE, TRUE)}
### first value is condition, second is output if condition is met and
###third is output if condition is not met

passing_grade2(30) ###test the fnection for x<50

## [1] FALSE
```

```

passing_grade2(98) ###test the fnection for x>50

## [1] TRUE
###the above outputs show that the function works

#11.

passing_grade(scores_vector)

## Warning in if (x < 50) (x = FALSE) else (x = TRUE): the condition has
## length > 1 and only the first element will be used
## [1] FALSE
###apply the if and else function to the scores vector

passing_grade2(scores_vector)

## [1] FALSE TRUE FALSE TRUE
###apply the ifelse function to the scores vector

```

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

Answer: The `ifelse` function worked, but the `if` and `else` function did not work. The R help tab says `ifelse` is a vectorize `if`, and therefore, it can be applied to our scores vector. The `if` function is designed to work with things that have a length of 1, so when applied we get an error message that the condition has length > 1.