

Assignment 2: Coding Basics

Meg Manning

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, Part 1

1. Generate a sequence of numbers from one to 30, increasing by threes. 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. #create a sequence from 1-30 increasing by 3 and assign name
seq1 <- seq(1,30,3)
View(seq1)

#2. #Compute mean and median of seq1 using mean() and median() functions
mean(seq1)

## [1] 14.5

median(seq1)

## [1] 14.5

#3. #ask if mean is greater than median
mean(seq1) > median(seq1)

## [1] FALSE
```

#since they are the same, it gives the result FALSE, meaning the mean is not greater than the median

Basics, Part 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.

```
#create a vector for names using c()
names <- c("Meg", "Josh", "Nick", "Melissa") #character or string vector
#Create a vector for test scores out of 100
test_results <- c(99, 75, 50, 92) #numeric
#create a vector for if they passed
passed_or_not <- c(TRUE, TRUE, FALSE, TRUE) #logical vector

#combine vectors into a data frame and assign a name
student_results <- cbind(names, test_results, passed_or_not)

#create vector of labels
student_results_labels <- c("Student Names", "Test Result of out 100", "If Student Passed")
View(student_results_labels)

#Option 2
student_results <- cbind("Student Names" = names, "Test Result out of 100" = test_results, "If Student Pa

#create labels
colnames(student_results) <- student_results_labels
```

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

Answer: This data frame is different from a matrix because it contains multiple classes of data within it, like a spreadsheet.

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.

```
#create function with if/else statement - if test score above 50 is passing'
#Vector of scores
test_results
```

```
## [1] 99 75 50 92
```

```
#create function for passing/failing results using ifelse
passing_grade <- function(x){
  print(ifelse(x>50, TRUE, FALSE))
}
passing_grade(test_results)
```

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

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

Answer: `ifelse` worked better because it was a simple function to use because regardless of what you set the function to do with your input, it will either result in a “TRUE” or “FALSE” statement which is what we are trying to figure out.