

Assignment 2: Coding Basics

Jon Nichols

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.
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. creating a sequence from 1 to 100 increasing by four  
hundred_sequence <- seq(1,100,4)
```

```
#2. determining mean and median of sequence  
mean(hundred_sequence)
```

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

```
median(hundred_sequence)
```

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

```
#3. set up a true/false equation to determine if mean is bigger than median  
mean(hundred_sequence)>median(hundred_sequence)
```

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

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.

```
#Student names, character vector
names <- c("Bill", "Ashley", "Bob", "Sam")

#Test Scores, numerical vector
scores <- c(75,88,54,99)

#Passed, logical vector
passed <- c(T,T,F,T)

#combined vectors into a dataframe
df_class_testresults <- data.frame("Student_Name" = names,"Test_Score" = scores,"Result" = passed)
```

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

Answer: A data frame can contain different types of vectors

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.

```
# i tested both if...else and ifelse statements below
```

```
passing_grade <- function(scores) {
  if(scores>50) {
    print("TRUE")
  }
  else {
    print("FALSE")
  }
}
```

```
passing_grade(51)
```

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

```
# the ifelse statement below is the one that works when applying to my original vector
```

```
passing_grade_2 <- function(scores){
  ifelse(scores>50, "PASS", "FAIL")
}
```

```
passing_grade_2(55)
```

```
## [1] "PASS"
```

```
passing_grade_2(scores)
```

```
## [1] "PASS" "PASS" "PASS" "PASS"
```

```
passing_grade(scores)
```

```
## Warning in if (scores > 50) {: the condition has length > 1 and only the first
## element will be used
```

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

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

Answer: Only the `ifelse` worked. `if` and `else` can only respond to one logical element at a time