

Assignment 2: Coding Basics

John Alcorn

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.  
one_to_100_by_4 <- seq(1,100,4)
```

```
#2.  
mean(one_to_100_by_4)>median(one_to_100_by_4)
```

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

```
#3.
```

#I created a sequence from 1 to 100 by 4's for question one. For question 2, I created a statement to s

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 <- c("John", "Mike", "Lisa", "Sara") #character (names of students)  
student_scores <- c(85, 43, 56, 20) #numeric (scores on test)  
student_pass_fail <- function(x){
```

```

ifelse(x>50, TRUE, FALSE)
}
pass_or_fail <- student_pass_fail(student_scores) #character (pass/fail of test scores)
df_student_scores <- as.data.frame(cbind("student names" = student_names,"student scores" = student_scores))
#data frame of student scores, names, and pass/fail

```

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

Answer: A matrix consists of rows and columns of the same type of data (numeric). A data frame consists of rows and columns but can have different types of data (character, numeric, integer...etc.).

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.

```

#ifelse function
student_pass_fail <- function(x){
  test_pass_fail<-ifelse(x>50, TRUE, FALSE)
  print(test_pass_fail)
}
student_pass_fail(student_scores)

## [1] TRUE FALSE TRUE FALSE

#if and else function
student_pass_fail2 <- function(x){
  if(x>50){
    print(TRUE)
  }
  else{
    print(FALSE)
  }
}
student_pass_fail2(student_scores)

## Warning in if (x > 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: ‘`ifelse`’ worked. It seems that the ‘`if`’ and ‘`else`’ cannot work for a vector.