

Assignment 2: Coding Basics

Jack Carpenter

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.  
#create a sequence from 1 to 100 by 4s  
seq(1,100,4)  
  
## [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  
  
#name the sequence  
fours_sequence <- seq(1,100,4)  
fours_sequence  
  
## [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.  
#run median and mean of fours_sequence  
median(fours_sequence)  
  
## [1] 49  
  
mean(fours_sequence)  
  
## [1] 49  
  
#3.  
#compare the mean and median values - is mean or median greater?  
mean(fours_sequence)>median(fours_sequence)  
  
## [1] FALSE
```

#the mean is NOT greater than the median (equal) so this comes back as 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.

#5-8.

#create a vector with student names

```
student_name<- c("Ted", "Beard", "Keeley", "Roy")
student_name
```

```
## [1] "Ted"      "Beard"     "Keeley"    "Roy"
```

#create a vector with test scores

```
test_score <- c(98,87,69,48)
test_score
```

```
## [1] 98 87 69 48
```

#create a vector whether they passed or not

```
passed <- c(test_score>50)
passed
```

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

#combine vectors into dataframe

```
student_scores <- data.frame(student_name, test_score, passed)
student_scores
```

```
##   student_name test_score passed
## 1         Ted         98    TRUE
## 2        Beard         87    TRUE
## 3       Keeley         69    TRUE
## 4         Roy         48   FALSE
```

#column names already appear to be labeled usefully

```
class(student_scores)
```

```
## [1] "data.frame"
```

#double-check that its actually a dataframe

9. QUESTION: How is this data frame different from a matrix?
Answer: Matrices need to have all the same type of data (eg. all numbers or all names) whereas data frames can have multiple types of data (eg. names, dates, and numbers) in the same table.
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.
#trying to create a function named passinggrade with a true/false output
passing_grade <- function(x) {
  if(x > 50) {
    TRUE
  }
  else {
    FALSE
  }
}
passinggrade <- function(x) {
  ifelse(x>50, TRUE, FALSE)
}
#I like this version better - its shorter and cleaner
```

```
#11.
#passinggrade(test_score)
#this function works, but only with individual values
#how do I get it to read the whole vector?
passinggrade(student_scores[, "test_score"])
```

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

```
#ok there we go, have to name the dataframe AND the vector not just the vector
#I think I understand, it has to be specified where to look for the data
print(passinggrade(student_scores[, "test_score"]))
```

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

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

Answer: Both options work as long as they're worded correctly. The 'if' and 'else' combination works, but is much easier to make a mistake in because there are more sections and pieces to it. The 'ifelse' function works the same way - it reads the same basic test - but is cleaner and easier for small tests like this.