

Assignment 2: Coding Basics

Jaimie Wargo

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.

```
sequence_1_to_30 <- seq(1,30,3)
```

#I created an object sequence_1_to_30 to describe the vector I am creating. The seq(x,y,z) function gen

#2.

```
mean_seq <- mean(sequence_1_to_30)
```

#I created the mean_seq object and assigned to it the mean of the sequence using the mean() function.

```
med_seq <- median(sequence_1_to_30)
```

#Med_seq is the object I assigned the median of the sequence to, using the median() function.

```
print('Mean')
```

```
## [1] "Mean"
```

```
print(mean_seq)
```

```
## [1] 14.5
```

```
print("Median:")
```

```
## [1] "Median:"
```

```
print(med_seq)
```

```
## [1] 14.5
```

#I am printing the mean and median to show these values rather than just storing them as objects.

#3.

```
print("Is mean greater than median?")
```

```
## [1] "Is mean greater than median?"
```

```
mean_seq > med_seq
```

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

#I just typed this logical equation into R, and it returns whether this statement is true or false.

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.

#5-6.

```
student_names <- c("Wendy", "Bagel", "Nova", "Minerva")
```

#character vector

```
test_scores <- c(45, 86, 95, 72)
```

#numeric vector

```
passed_test <- c(FALSE, TRUE, TRUE, TRUE)
```

#logical vector

#I checked the vector types using the class() function

#7-8.

```
test1_results <- data.frame("StudentNames" = student_names, "TestScores" = test_scores, "Passed" = passed_test)
```

#I used data.frame() to create a data frame from the vectors, and I named them informatively using caps

```
test1_results
```

```
## StudentNames TestScores Passed
## 1 Wendy 45 FALSE
## 2 Bagel 86 TRUE
## 3 Nova 95 TRUE
## 4 Minerva 72 TRUE
```

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

Answer: This data frame has multiple data types, whereas a matrix can only have one data type.

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.

#I first used a for loop to have the function cycle through all the values in my vector-- I know we tech

```
PassFail <- function(scores){
  for(x in scores)
    if(x>=50){
      print("TRUE")
    } else{
      print("FALSE")
    }
}

PassFail(test_scores)
```

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

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

Answer: The ifelse function does not print the values assigned; it just returns them. If and else allows for the values to be printed and not returned, showing that if and else has more flexibility than ifelse, but it requires more typing.