Assignment 2: Coding Basics

Emma Brentjens

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 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 object "seq_100": sequence from 1 to 100 increasing by 4
seq_100 <- seq(1, 100, 4)
seq_100 ##output of 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.
mean(seq_100) ##calculating mean of sequence
```

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

```
median(seq_100) ##calculating median of sequence
```

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

```
#3.
##command to see if the mean of the sequence is greater than the median
isTRUE(mean(seq_100)>median(seq_100))
```

```
## [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.

```
#5.
##creating vectors of student names, test scores, and whether they passed
student_names <- c("Anne", "Brad", "Chad", "Devin") ##character vector
test_scores <- c(92, 75, 48, 83) ##numeric vector
pass_test <- c(T, T, F, T) ##logical vector

#7.
##combining vectors into dataframe
student_test_scores <- data.frame(student_names, test_scores, pass_test)
##creating column names for dataframe
colnames(student_test_scores) <- c("Student Name", "Test Score", "Pass?")
##dataframe output
student_test_scores
```

```
##   Student Name Test Score Pass?
## 1         Anne         92  TRUE
## 2         Brad         75  TRUE
## 3         Chad         48 FALSE
## 4        Devin         83  TRUE
```

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

   Answer: A data frame can have different types of classes or modes (e.g. can combine numbers and dates), while matrices cannot.

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.

```
#10.
##if/else function to determine whether score is passing
passing_grade <- function(x){
  if(x>50){y=TRUE}
     else{y=FALSE}
  print(y)}

passing_grade(92) ## if/else function cannot run a vector of values
```

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

```
## must use ifelse function to assess all scores in the vector
passing_grade_2 <- function(x) {
  ifelse(x>50, TRUE, FALSE)}

passing_grade_2(test_scores)
```

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

```
passing_grade_2(80) ##ifelse function also works for single values
```

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

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

    Answer: The 'if' and 'else' function could show the output for single numbers, but not the whole vector. The ifelse function could print the results for the entire vector because it is designed handle vectors.