Assignment 2: Coding Basics

Jonathan Joyner

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.

```r
#1.
# forming sequence 1 to 30 in intervals of 3 and assigning it the name "Series_1"
Series_1 <- seq(1,30,3)
#2.
# calculating mean and median of Series_1
mean(Series_1)
```

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

```r
median(Series_1)
```

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

```r
#3.
# using an if statement to determine if the mean of Series_1 is greater than the median of Series_1
ifelse(mean(Series_1)>median(Series_1),"greater","not greater")
```

```
## [1] "not greater"
```

```
#4.
#Comments can be seen above
```

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.
#Assigning each vector with relevant names
student_names <- c("Ashley","Bryan","Charlie","Diana")
test_scores <- c(83,41,67,33)
pass_or_fail <- c("TRUE","FALSE","TRUE","FALSE")
#6.
#explaining assigned values
student_names #names of 4 students
```

```
## [1] "Ashley"  "Bryan"   "Charlie" "Diana"
```

```
test_scores #test score percentages for each student
```

```
## [1] 83 41 67 33
```

```
pass_or_fail #whether each student passed or failed
```

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

```
#7 and #8.
#creating a data frame from values and naming each column appropriately
school_report <- data.frame(names=student_names,scores=test_scores,pass=pass_or_fail)
print(school_report)
```

```
##     names scores  pass
## 1  Ashley     83  TRUE
## 2   Bryan     41 FALSE
## 3 Charlie     67  TRUE
## 4   Diana     33 FALSE
```

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

   Answer:A matrix is arranged in a fixed grid whereas a dataframe is listed in rows and columns that can vary. A dataframe can also include data of varying types whereas a matrix has to have homogeneous data, be it numerical, character, or other.

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.

```
#10.
#ifelse statement created which measures vector of 3 scores against value of 50 and determines if great
ifelse(c(40,63,75)>50,"TRUE","FALSE")
```

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

11. Apply your function to the vector with test scores that you created in number 5.

```
#11.
#ifelse statement created which measures assigned vector "test_scores" against value of 50 and determin
ifelse(test_scores>50,"TRUE","FALSE")
```

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

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

    Answer: ifelse worked because it returns vectors rather than just single components like if.