Assignment 2: Coding Basics

Sydney Williams

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. #Creating a sequence of numbers from one to 30, increasing by threes.
sequence_1 <- seq(1,30,3)
sequence_1
```

```
##  [1]  1  4  7 10 13 16 19 22 25 28
```

```
#2. #Computing the mean and median of the sequence with the mean and median functions
mean <- mean(sequence_1)
median <- median(sequence_1)

#3. #Creating a function to ask R whether mean is greather than median
mean > median
```

```
## [1] 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.
9. QUESTION: How is this data frame different from a matrix?

   Answer: A data frame can hold multiple types of data, while a matrix can only have one type of data. Data frames have column names, matrices do not, you refer to the data point using row and column indices.

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.

```
# Create vectors for names, test scores, and pass status
names_vector <- c("James", "Ben", "Jordan", "Betty")
scores_vector <- c(68, 82, 95, 45)
pass_status_vector <- scores_vector >= 50

# This is the character vector
names_vector <- c("James", "Ben", "Jordan", "Betty")

# This is the numerical vector out of a total of 100 points
scores_vector <- c(68, 82, 95, 45)

# This is the logical vector that shows TRUE if passed and FALSE if failed
pass_status_vector <-scores_vector >= 50

# Combine vectors into a data frame
students_test_data <- data.frame(
  Names = names_vector,
  TestScores = scores_vector,
  PassStatus = pass_status_vector)

# Labeling the columns with informative names
colnames(students_test_data) <- c("Student Name","Test Score (out of 100)", "Pass/Fail")
print(students_test_data)
```

```
##   Student Name Test Score (out of 100) Pass/Fail
## 1        James                      68      TRUE
## 2          Ben                      82      TRUE
## 3       Jordan                      95      TRUE
## 4        Betty                      45     FALSE
```

```
# 11a.  Function for passing score
passing_score <- function(test_scores) {
  result <- ifelse(test_scores >= 50, TRUE, FALSE)
  print(result)  }

# 11b. Function for vector data from Question 5
passing_score(scores_vector)
```

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

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

    Answer: ifelse worked. “if and else” gave an error that said condition had a length >1.