Assignment 2: Coding Basics

Lucy Wang

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. Generate a sequence of numbers named "num_30" from 1 to 30 that increase by 3.
num_30 <- seq(1, 30, 3)
num_30
```

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

```
#2. Find the mean and the median for num_30.
mean_num <- mean(num_30)
median_num <- median(num_30)
mean_num
```

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

```
median_num
```

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

```r
#3. Ask if the mean of num_30 is greater than the median.
mean_num > median_num
```

```
## [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.

```r
# Assign data to vectors
stu_name <- c("Lucy", "Wilson", "Anne", "Tim") #character
score <- c(100, 80, 60, 45) #numeric
pass_test <- score >= 50 #logical

# Combine vectors to a data frame
df <- data.frame("student_name" = stu_name, "test_score" = score, "pass_test" = pass_test)
# or, names(df) <- c("student_name", "test_score", "pass_test")
print(df)
```

```
##   student_name test_score pass_test
## 1         Lucy        100      TRUE
## 2       Wilson         80      TRUE
## 3         Anne         60      TRUE
## 4          Tim         45     FALSE
```

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

   Answer: A matrix can only contain one class of data, while a dataframe can contain different classes of data.

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.

```r
# Create a function to print whether a given score has pass the test using "ifelse" statement
f_pass <- function(test_score){
  ifelse(test_score >= 50, print(TRUE), print(FALSE) )
}

# Test whether the scores in question 5 have pass the test.
for (i in score){
  f_pass(i)
}
```

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

```r
# Create a function to print whether a given score has pass the test using "if" and "else statements
f_pass <- function(test_score){
  if (test_score >= 50){print(TRUE)}
  else{print(FALSE)}
}

# Test whether the scores in question 5 have pass the test.
for (i in score){
  f_pass(i)
}
```

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

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

    Answer: Both worked. Because the logic behind the two statements are necessrily the same. "ifelse" statement condenses the "if…else…" statements into one line.