Assignment 2: Coding Basics

Samantha White-Murillo

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on coding basics.

Directions

1. Rename this file `<Samantha White-Murillo>_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 <- seq(1,30,3) #creating a sequence and labeling it

#2.
mean(sequence) # operation with the sequence
```

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

```
median(sequence)
```

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

```
#3.
mean(sequence) > median(sequence) #comparison of results
```

```
## [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.

```
#Student
names_of_students <- c("Samantha","Ivy","Juan")

#Student_scores
score_of_students <- seq(30,100,length.out = 3)

#Passed_the_test
passed <- score_of_students > 50

#Making_data_frame
is.data.frame(names_of_students)
```

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

```
df_name_of_students <- is.data.frame(names_of_students)
df_name_of_students
```

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

```
class(df_name_of_students)
```

```
## [1] "logical"
```

```
#adding_frame
df <- cbind(df_name_of_students, score_of_students, passed)
class(df)
```

```
## [1] "matrix" "array"
```

```
df
```

```
##      df_name_of_students score_of_students passed
## [1,]                   0                30      0
## [2,]                   0                65      1
## [3,]                   0               100      1
```

```
#Other option of data frame creation
create_df <- data.frame(names_of_students,score_of_students,passed)
create_df
```

```
##   names_of_students score_of_students passed
## 1          Samantha                30  FALSE
## 2               Ivy                65   TRUE
## 3              Juan               100   TRUE
```

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

   Answer: matrices can only contain a single class of data, while data frames can consist of many 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.

```
#If, else metod
Test_scores <- function(score_of_students){
  if(score_of_students > 50) {
    score_of_students <- 'pass'
  }
  else {
    score_of_students <- 'fails'
  }
  print(Test_scores)
}

#ifelse metod
Test_scores <- function(score_of_students) {
  ifelse(score_of_students > 50, 'pass', 'fails')

  print(Test_scores)
}
```

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

    Answer: ifelde worked. Because if else is stricter than ifelse, and can generate error eadyly if the code is not well executed.