Assignment 2: Coding Basics

Emma Kaufman

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.
Sequence = seq(1, 30, 3) #from 1, to 30, by 3 to create the sequence of numbers called Sequence
Sequence
```

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

```r
#2.
Mean = mean(Sequence) #calculating the mean of Sequence
Mean
```

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

```r
Median = median(Sequence) #calculating the median of Sequence
Median
```

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

```
#3.
Mean>Median #is mean greater than median? will return false if false, return true if true
```

```
## [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.

```
#5.
Names <- c('Emma', 'Jenn', 'Tim', 'Cara') #character vector
Names
```

```
## [1] "Emma" "Jenn" "Tim"  "Cara"
```

```
#class(Names)

Scores <- c(80, 40, 50 ,90) #numeric vector
Scores
```

```
## [1] 80 40 50 90
```

```
#class(Scores)

Passing <- c('TRUE', 'FALSE', 'TRUE', 'TRUE') #character vector
Passing
```

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

```
#class(Passing)

#7.
#creating a dataframe from the Names, Scores, and Passing vectors
df_test_results <- as.data.frame(cbind(Names,Scores,Passing))
df_test_results
```

```
##   Names Scores Passing
## 1  Emma     80    TRUE
## 2  Jenn     40   FALSE
## 3   Tim     50    TRUE
## 4  Cara     90    TRUE
```

```r
#8.
#Titles of the vectors used to create the df were informative, as a result the column
#names are informative, but can change them using colnames()

#changing the names of the column titles
colnames(df_test_results) <- c('Student Names', 'Test Scores out of 100', 'Above 50%?')

df_test_results
```

```
##   Student Names Test Scores out of 100 Above 50%?
## 1          Emma                     80       TRUE
## 2          Jenn                     40      FALSE
## 3           Tim                     50       TRUE
## 4          Cara                     90       TRUE
```

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

   Answer: They are different because a matrix can only contain one class type, but this data frame contains multiple class types. It has a column of numeric data as well as columns of character data. A matrix would only contain one class type.

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.

```r
is_it_passing <- function(x){
  ifelse(x>=50,TRUE,FALSE)
}
```

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

```r
is_it_passing(Scores)
```

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

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

    Answer: ifelse worked because we only need to know if the score is greater than or equal to 50, and can determine whether a score passes or fails accordingly. If the score is greater than or equal to 50, the function returns TRUE. In any other scenario (the test score being less than 50), the function returns FALSE.