Assignment 2: Coding Basics

Livia Hoxha

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.

```
threes_sequence <- seq(1, 30, by = 3) # here I inserted the command to generate the sequence as request
threes_sequence
```

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

```
mean_value <- mean(threes_sequence) # here I computed the mean of the sequence
median_value <- median(threes_sequence) # here I computed the median of the sequence
mean_greater_than_median <- mean_value > median_value # here the mean and the median are compared

cat("Mean:", mean_value, "\n")
```

```
## Mean: 14.5
```

```
cat("Median:", median_value, "\n")
```

```
## Median: 14.5
```

```
cat("Is mean greater than median?", mean_greater_than_median, "\n")
```

```
## Is mean greater than median? FALSE
```

```
#1.

#2.

#3.
```

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.

```
# In Vector 1 I have assigned the names of the students
student_names <- c("Ana", "Ben", "Chase", "Dan")

# In Vector 2 I assigned the test scores out of 100 points
test_scores <- c(75, 62, 90, 42)

# In Vector 3 I set Whether they passed the test (TRUE or FALSE) with a passing grade of 50
pass_status <- test_scores >= 50

# Now I combine the vectors into a data frame and by giving it a name
student_data <- data.frame(Name = student_names, Score = test_scores, Passed = pass_status)

# I specify the command to the data frame
student_data
```

```
##    Name Score Passed
## 1   Ana    75   TRUE
## 2   Ben    62   TRUE
## 3 Chase    90   TRUE
## 4   Dan    42  FALSE
```

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

   Answer: Data frames in general are more versatile for diverse data, while matrices are more specialized for numerical data. For this assignment a data frame is the more appropriate choice, because it allows us to have columns with different data types. Whereas, if we were to use a matrix, we would be limited to a single data type for all elements. Furthermore, matrices lack column names, making it not clear what each column represents.

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
# Here I define the function to check if a test score is passing or not by using the if else
check_passing_score <- function(score) {
  if (score >= 50) {
    return(TRUE)
  } else {
    return(FALSE)
  }
}
# I specify the test scores created in the above question
test_scores <- c(75, 62, 90, 42)

# Here I apply the function to each test score in the vector
cat("Passing Grades:\n")
```

```
## Passing Grades:
```

```r
for (score in test_scores) {
  cat("Score:", score, "Passing:", check_passing_score(score), "\n")
}
```

```
## Score: 75 Passing: TRUE
## Score: 62 Passing: TRUE
## Score: 90 Passing: TRUE
## Score: 42 Passing: FALSE
```

```r
# Now we define a function to check if a test score is passing using ifelse
check_and_print_passing_score_ifelse <- function(scores) {
  result <- ifelse(scores >= 50, "TRUE", "FALSE")
  cat(result, "\n")
}

# This is the vector of test scores
test_scores <- c(75, 62, 90, 42)

# Here I apply the function to the vector of test scores
cat("Passing Grades:\n")
```

```
## Passing Grades:
```

```r
check_and_print_passing_score_ifelse(test_scores)
```

```
## TRUE TRUE TRUE FALSE
```

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

Answer: I tried both `if` and `else` and `ifelse`options. The `if` and `else` statement worked because the aim of this question was to check each test score individually and see whether it's passing or not passing, as we are dealing with individual elements.`ifelse` on the other hand is used for vectorized operations where a condition to each element of a vector is applied and a new vector of results. It's not the right choice for this specific task of checking individual test scores and printing the results. Disclaimer: I used the help of online sources and tools to guide me through this assignment.