Assignment 2: Coding Basics

Reino Hyyppa

OVERVIEW

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

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your first and last name into the file name (e.g., "FirstLast_A02_CodingBasics.Rmd") prior to submission.

Basics Day 1

1. Generate a sequence of numbers from one to 100, increasing by fours. 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.

# create sequence of numbers, ranging from 1 to 100, increasing by 4.
numb <- seq(1, 100, by=4)

#2.

# determine mean and median of sequence of numbers created in step 1.
mean <- mean(numb)
med <- median(numb)

#3.

# determine if mean is greater than the median.
mean > med
```

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

```
# returns false, therefore, mean is not greater than the median
```

Basics Day 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.

```
# vector type = character
name <- c("Hannah", "Julia", "Cisco", "Ben", "Reino", "Maddie", "Kacey", "Rachel", "Nadia", "Devon")

# vector type = numeric
scores <- c(65, 45, 100, 55, 95, 40, 60, 75, 90, 20)

# vector type = character
pass <- ifelse(scores>50, "TRUE", "FALSE")

# create data frame, combining all of the vectors, and assigning informative column names
df_student_pass <- data.frame("Names"=name, "Test_Score"=scores, "Pass"=pass)
class(df_student_pass)
```

```
## [1] "data.frame"
```

```
# check data frame
df_student_pass
```

```
##     Names Test_Score  Pass
## 1  Hannah         65  TRUE
## 2   Julia         45 FALSE
## 3   Cisco        100  TRUE
## 4     Ben         55  TRUE
## 5   Reino         95  TRUE
## 6  Maddie         40 FALSE
## 7   Kacey         60  TRUE
## 8  Rachel         75  TRUE
## 9   Nadia         90  TRUE
## 10  Devon         20 FALSE
```

```
# check for column names
colnames(df_student_pass)
```

```
## [1] "Names"      "Test_Score" "Pass"
```

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

Answer: A data frame stores multiple types of data into rows and columns, whereas data stored in a matrix must be the same data type.

10. Create a function with an if/else statement. Your function should determine whether a 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. Hint: Use `print`, not `return`. The name of your function should be informative.

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

```
# create formula
grade_check <- function(grade) {
  pass_fail <- ifelse(grade > 50, "TRUE", "FALSE")
  print(pass_fail)
}

# check formula
grade_check(99)
```

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

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

Answer: The ifelse statement worked the best to check both whether the input value is above or below a specified value.