Assignment 2: Coding Basics

Logan Loadholtz

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.

```
#1.
mysequence <- seq(1,100,4)
mysequence
```

```
##  [1]  1  5  9 13 17 21 25 29 33 37 41 45 49 53 57 61 65 69 73 77 81 85 89 93 97
```

```
#Here I first created a name for the sequence "mysequence". I assigned this name to the
#sequence. I typed seq to tell R I am creating a sequence. I typed 1 to tell R where
#I want the sequence to begin, and 100 to tell R where I want the sequence to end.
#Finally, the 4 indicates how often to display values in the sequence, here indicates
#increasing every fourth value.

#2.
mymean <- mean(mysequence)
#Here, I am telling R to calculate the mean of the sequence, and I am assigning it
#to the name mymean.

mymedian <- median(mysequence)
#Here, I am telling R to calculate the median of the sequence, and I am assigning
#it to the name mymedian.

#3.
mymean > mymedian
```

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

```
#Here, I am using a conditional statement where mean is greater than median. The output of
#this says that this is FALSE"
```

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.

```
#5
studentnames <- c("Logan", "Carly", "Hannah", "Helen", "Elise", "Erik", "Tim", "Henry")
studentnames
```

```
## [1] "Logan"  "Carly"  "Hannah" "Helen"  "Elise"  "Erik"   "Tim"    "Henry"
```

```
typeof(studentnames)
```

```
## [1] "character"
```

```
#This indicates that this is a "character" vector.


testscore <- c(97, 82, 99, 93, 85, 100, 48, 36)
testscore
```

```
## [1]  97  82  99  93  85 100  48  36
```

```
typeof(testscore)
```

```
## [1] "double"
```

```
#This indicates that this is a "double" vector


passingscores <- testscore > 50
passingscores
```

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

```
typeof(passingscores)
```

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

```
#This indicates that this is a "logical" vector.

studentstestgrades <- data.frame(studentnames, testscore, passingscores)
studentstestgrades
```

```
##   studentnames testscore passingscores
## 1        Logan        97          TRUE
## 2        Carly        82          TRUE
## 3       Hannah        99          TRUE
## 4        Helen        93          TRUE
## 5        Elise        85          TRUE
## 6         Erik       100          TRUE
```

```
## 7          Tim        48         FALSE
## 8        Henry        36         FALSE
```

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

   Answer: A matrix containes one type of data. This dataframe can contain multiple types of data, such as character, double, and logical.

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.

```
didstudentpassexam <- ifelse(testscore >50, TRUE, FALSE)

didstudentpassexam
```

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

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

    Answer: I utilized the "ifelse" option. This worked because the output shows scores that are less than 50 as "FALSE". Scores above 50 show up as "TRUE". FOr this, I used the vector "testscore", which is the vector of exam scores. Then, I typed >50 to tell R that if there is an exam score, to mark it as "TRUE". Anything less than this will create an output of "FALSE".