Assignment 2: Coding Basics

Enikoe Bihari

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. use seq() function to make a sequence starting and 1 and ending at 100, skipping every 4 numbers

num.list <- seq(1,100,4)

#2. use mean() function to find mean and median() function to find median

mean <- mean(num.list)
median <- median(num.list)

#3. test whether mean is greater than median

median < mean
```

```
## [1] 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.

```r
# make lists

# character
students <- c(’Elod’,’Tas’,’Almos’,’Tohotom’)
typeof(students)
```

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

```r
# numeric
scores <- c(40,100,100,90)
typeof(scores)
```

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

```r
# logical
passed <- c(F,T,T,T)
typeof(passed)
```

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

```r
# make a data frame with the lists as the columns
test.info <- data.frame(
  students = students,
  scores = scores,
  passed = passed
)
```

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

   Answer: Both matrices and data frames consist of data arranged in columns and rows of the same length. However, a matrix must have data of the same mode, while a data frame can have data in different modes (numeric, character, etc.).

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.

```r
# make a function that finds out whether students passed the test based on their scores
passed.test <- function(scores, students){
  # create an empty vector to store the results
  result = c()
  # iterate through every item in the list of scores
  for(i in 1:length(scores)){
    # find the name that corresponds to that score in the list of names
    name = students[i]
    # if the score is over 50, print the student’s name and result, add the result to the list of resul
    if (scores[i] >= 50) {
     print(paste(name, "passed"))
     result = c(result, ’passed’)
    }
    # if the score is not over 50, print the student’s name and result, add the result to the list of r
    else {
      print(paste(name, "did not pass"))
      result = c(result, ’not passed’)
    }
  }
  # resturn the list of results
  return(result)
}

result = passed.test(scores, students)
```

```
## [1] "Elod did not pass"
## [1] "Tas passed"
## [1] "Almos passed"
## [1] "Tohotom passed"
```

```
# add the vector of whether they passed or not as a new column "result" in the data frame
test.info$result = result
test.info
```

```
##   students scores passed     result
## 1     Elod     40  FALSE not passed
## 2      Tas    100   TRUE     passed
## 3    Almos    100   TRUE     passed
## 4  Tohotom     90   TRUE     passed
```

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

    Answer: I used separate ‘if’ and ‘else’ statements because they seem more explicit/easy to read. Also, because I wanted to include some extra steps where I actually created a new list with the pass/fail results (to later add on to my data frame), the separate “if” “else” statements helped me incorporate these steps into my function in a more organized manner. However, it seems that both type of statements would work, since there are only 2 conditions to consider: greater than or equal to 50, or less than 50.