Assignment 2: Coding Basics

Aurora McCollum

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.
onetohundred<-seq(1,100,4) #creating a sequence: from, to, by
onetohundred
```

```
##  [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
```

```
#2.
mean(onetohundred) #finding mean
```

```
## [1] 49
```

```
median(onetohundred) #finding median
```

```
## [1] 49
```

```
#3.
mean(onetohundred)>median(onetohundred) #is mean greater than median
```

```
## [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.

```
#5.
studentname<-c("Rory"=1, "Michelle"=2, "Sam"=3, "Corey"=4) #numeric
studentname
```

```
##     Rory Michelle      Sam    Corey
##        1        2        3        4
```

```
class(studentname)
```

```
## [1] "numeric"
```

```
testscores<-c(80, 70, 90, 49) #numeric
testscores
```

```
## [1] 80 70 90 49
```

```
class(testscores)
```

```
## [1] "numeric"
```

```
passed<-c(TRUE, TRUE, TRUE, FALSE) #logical
passed
```

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

```
class(passed)
```

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

```
#7.

studentscorepassed<-cbind(studentname,testscores,passed)
studentscorepassed
```

```
##          studentname testscores passed
## Rory               1         80      1
## Michelle           2         70      1
## Sam                3         90      1
## Corey              4         49      0
```

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

   Answer: Matrices can only hold one kind of data, data frames can contain multiple classes of data.

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.

```
exam_pass<-ifelse(testscores>=50,"pass","fail")
exam_pass
```

```
## [1] "pass" "pass" "pass" "fail"
```

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

Answer: ifelse worked because I wanted anything greater than 50 to count as a pass, and everything else to be a fail. So if greater than 50 pass, else fail.