Assignment 2: Coding Basics

Erin Ansbro

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.

```
#1. creating a sequence with from to by, also naming the sequence seq
seq(1,30,3)
```

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

```
seq <- seq(1,30,3)
#2. calculating mean and median of sequence
mean(seq)
```

```
## [1] 14.5
```

```
median(seq)
```

```
## [1] 14.5
```

```r
#3. determining if median is greater than median
mean(seq) > median (seq)
```

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

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.

```r
Names <- c('Emma','John','Luana','Luke') #characters
Names
```

```
## [1] "Emma"  "John"  "Luana" "Luke"
```

```r
Grades <- c(81,99,99,49) #integer
Grades
```

```
## [1] 81 99 99 49
```

```r
Pass <- c(Grades > 50) #logical
Pass
```

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

```r
df_creation <- as.data.frame(Names) #creating data frame
df_creation
```

```
##   Names
## 1  Emma
## 2  John
## 3 Luana
## 4  Luke
```

```r
df_NickSchoolClassGrades <- cbind(df_creation, Grades, Pass) #adding columns to data frame
df_NickSchoolClassGrades
```

```
##   Names Grades  Pass
## 1  Emma     81  TRUE
## 2  John     99  TRUE
## 3 Luana     99  TRUE
## 4  Luke     49 FALSE
```

```
#renaming columns to something informative
names(df_NickSchoolClassGrades)
```

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

```
Names_of_Students <- Names
Names_of_Students
```

```
## [1] "Emma"  "John"  "Luana" "Luke"
```

```
Grades_of_Students <- Grades
Grades_of_Students
```

```
## [1] 81 99 99 49
```

```
Did_Students_Pass <- Pass
Did_Students_Pass
```

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

```
df_NickSchoolClassGrades #how do i chnage column names in the dataframe?
```

```
##   Names Grades  Pass
## 1  Emma     81  TRUE
## 2  John     99  TRUE
## 3 Luana     99  TRUE
## 4  Luke     49 FALSE
```

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

   Answer: This data frame is different because it has different types of data, e.g. numeric, logical, and characters. A matric needs to have the same type of data.

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.

```
ifelse(Grades<50, "False", "True")
```

```
## [1] "True"  "True"  "True"  "False"
```

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

    Answer: The ifelse worked for me, I had trouble putting the Grades into the function, there would be an error in the "if" statement because the condition had a length > 1. This was the if else i created: #grade <- function(Grades) { # if(Grades < 50) { # "False" # } # else { # "True" # } #} #if i entered grade(4), it would print back "False"), but if I submitted grade(Grades), I got the error 'Error in if (Grades < 50) { : the condition has length > 1'