

Assignment 2: Coding Basics

Student Name

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. using the sequence function to go from 1 to 100 by 4s. storing the sequence in seq1.  
seq1 = seq(1, 100, 4)
```

```
#2. using the mean function to find the mean of seq1. storing the mean in mean1. producing the output.  
mean1 = mean(seq1)  
mean1
```

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

```
median1 = median(seq1)  
median1
```

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

```
#3. asking r if mean1 is greater than median1. output is false because they are equal.  
mean1 > median1
```

```
## [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.

```
vector.a = c("Ango Gabloggian", "Aquaman", "Jane Goodall", "Alfred Hitchcock")
# typeof(vector.a) -- character

vector.b = c(23, 72, 100, 65)
# typeof(vector.b) -- double

vector.c = c(FALSE, TRUE, TRUE, TRUE)
# typeof(vector.c) -- logical

df.vector.a = as.data.frame(vector.a)
df.vector.b = as.data.frame(vector.b)
df.vector.c = as.data.frame(vector.c)

grades.df = cbind(df.vector.a, df.vector.b, df.vector.c)
grades.df
```

```
##           vector.a vector.b vector.c
## 1 Ango Gabloggian      23     FALSE
## 2      Aquaman       72      TRUE
## 3   Jane Goodall     100      TRUE
## 4 Alfred Hitchcock     65      TRUE
```

```
grades1.df = data.frame("name"=vector.a, "score"=vector.b, "pass"=vector.c)
```

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

Answer: A data frame only has columns while a matrix also categorizes data based on rows.

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.

```
passing.grade <- function(x) {
  ifelse(x>50, print(TRUE), print(FALSE))
}

#passing.grade <- function(x) {
#  if(x > 50) {
#    print('TRUE')
#  }
#}
```

```
# }  
# else {  
#   print('FALSE')  
# }  
#}
```

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

```
passing.grade(vector.b)
```

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

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

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

Answer: 'ifelse' worked because 'if' and 'else' would only allow one value to print from vector.b.