

Assignment 2: Coding Basics

Emily Wood

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 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.
seq1 <- seq(1, 100, 4)

# Seq1 is a sequence form 1 to 100 that increases by four

# 2.

meanseq1 <- mean(seq1)
meanseq1

## [1] 49

medianseq1 <- median(seq1)
medianseq1

## [1] 49

# Here I am calculating the mean and median of seq1 and assigning them a name.
# Both the mean and median are 49.

# 3.

meanseq1 > medianseq1
```

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

```
# Here I am asking R to determin if the mean is greater than the mdeian. It is  
# not so the result 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.
```

```
names_of_students <- c("Kelly", "Emily", "Sam", "Meech")  
class(names_of_students)
```

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

```
names_of_students
```

```
## [1] "Kelly" "Emily" "Sam"   "Meech"
```

```
# This is the vector of the names of each student. It is a character vector.
```

```
test_scores <- c(11, 22, 77, 88)  
class(test_scores)
```

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

```
test_scores
```

```
## [1] 11 22 77 88
```

```
# This is the vector of the test scores of each student. It is a numeric  
# vector.
```

```
Pass_Fail <- c(FALSE, FALSE, TRUE, TRUE)  
class(Pass_Fail)
```

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

```
Pass_Fail
```

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

```
# This is the vector of whether each student passed or not. It is a logical  
# vector.
```

```
# 7.
```

```
df_Pass_Fail <- as.data.frame(Pass_Fail)
```

```
dataFrame_TestResults <- cbind(df_Pass_Fail, test_scores, names_of_students)  
dataFrame_TestResults
```

```
##   Pass_Fail test_scores names_of_students
```

```
## 1    FALSE      11      Kelly
## 2    FALSE      22      Emily
## 3     TRUE      77       Sam
## 4     TRUE      88      Meech
```

```
is.data.frame(dataFrame_TestResults)
```

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

```
# 8.
```

```
dataFrame_TestResults_Final <- data.frame(Name = names_of_students, Grade = test_scores,
      Passed = Pass_Fail)
dataFrame_TestResults_Final
```

```
##      Name Grade Passed
## 1 Kelly     11  FALSE
## 2 Emily     22  FALSE
## 3 Sam       77   TRUE
## 4 Meech     88   TRUE
```

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

Answer: Matrices only contain one type of data. Dataframes like this one can contain multiple types such as numeric, character and logical verticies.

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.

```
gradingrecipe <- function(x) {
  ifelse(x > 50, TRUE, FALSE)
}

Gradingmeal <- gradingrecipe(test_scores)

Gradingmeal
```

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

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

Answer: 'ifelse' worked because it checks every element of a vector. The 'if' and 'else' staments only deal with a single value. Because of this, R showed an error when I tried using a function with 'if' and 'else' statements with the test scores vector.