

Assignment 2: Coding Basics

Nicole Eastman

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. Use the sequence function using the format: (<1 number in sequence>, <last number in sequence>, by=  
seq(1,100, by=4)
```

```
## [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
```

```
sequence1<- seq(1,100, by=4)
```

```
#2. Use the mean() and the median() functions to solve and assign them a name
```

```
mean1<- mean(sequence1)
```

```
mean1
```

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

```
median1<- median(sequence1)
```

```
median1
```

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

```
#3. ask R if the mean is greater than, less than, or equal to
```

```
mean1>median1
```

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

```
mean1<median1
```

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

```
mean1==median1
```

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

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.

```
names<- c("Nicole","Andrew","Judy","Chris")
names
```

```
## [1] "Nicole" "Andrew" "Judy"   "Chris"
```

```
scores<- c(98,97,96,95)
scores
```

```
## [1] 98 97 96 95
```

```
passed<-c(TRUE, TRUE, TRUE, TRUE)
passed
```

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

```
testresults<- data.frame(names,scores,passed)
testresults
```

```
##   names scores passed
## 1 Nicole     98   TRUE
## 2 Andrew     97   TRUE
## 3  Judy      96   TRUE
## 4  Chris      95   TRUE
```

```
colnames(testresults)<- c("Student", "Test Score", "Passing")
testresults
```

```
##   Student Test Score Passing
## 1  Nicole         98     TRUE
## 2  Andrew         97     TRUE
## 3   Judy          96     TRUE
## 4   Chris          95     TRUE
```

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

Answer: By using a data frame you can have different classes of data such as a numeric column and a column containing names. Matrices only contain numerical values and they do not have titles for the rows and columns.

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(scores>=50, print("True"), print("False"))
```

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

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

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

Answer: The `ifelse` function worked because the `if, else` option gives an error stating the length of the result does not match the length of the conditional.