Assignment 2: Coding Basics

marisa fajardo

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.
by4_sequence<- seq(1,100,4) #creating sequence of numbers up to 100 going by 4's
by4_sequence
```

```
##  [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_by4_sequence<- mean(by4_sequence) #naming mean of sequence
mean_by4_sequence
```

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

```
med_by4_sequence<- median(by4_sequence) #naming median of sequence
med_by4_sequence
```

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

```
#3.
mean_by4_sequence > med_by4_sequence #checking if mean is larger 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.

```
students<- c('Ziggy', 'Clementine', 'Cleve', 'Cornelius', 'Toad') #creating vector using categorical da
scores<- c(49, 89, 76, 93, 84) #creating vector of numerical data - scores
passing<- c(scores>49) #creating categorical vector of pass or failing scores

dfclassscores<- cbind(students,scores,passing) #combining vectors into data frame
dfclassscores
```

```
##      students     scores passing
## [1,] "Ziggy"      "49"   "FALSE"
## [2,] "Clementine" "89"   "TRUE"
## [3,] "Cleve"      "76"   "TRUE"
## [4,] "Cornelius"  "93"   "TRUE"
## [5,] "Toad"       "84"   "TRUE"
```

```
dfclassscores_complete <- data.frame("student"=students,"test score"=scores, "pass or fail grade" = pas
dfclassscores_complete #viewing data frame
```

```
##      student test.score pass.or.fail.grade
## 1      Ziggy         49              FALSE
## 2 Clementine         89               TRUE
## 3      Cleve         76               TRUE
## 4  Cornelius         93               TRUE
## 5       Toad         84               TRUE
```

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

   Answer: a data frame can include different forms of data, such as categorical and numerical, while a matrix must contain the same data forms.

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.

```r
passinggrade<- function(x) {ifelse(scores>49, "pass", "fail")} #Creating function to determine pass of

print(passinggrade(scores)) #applying function to scores vector and printing out results
```

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

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

    Answer: ifelse worked better than if and else. When I attempted to use if and else, my vector length was restricted to one and it would not print out full results of the vector. ifelse had no limitations on vector length, therefore it worked best for my set of test scores.