Assignment 2: Coding Basics

Ina Liao

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.
seq1 <- seq(1,30,3)

#2.
seq1_mean<-mean(seq1); seq1_mean
```

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

```
seq1_median<-median(seq1); seq1_median
```

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

```
#3.
ans<-ifelse(seq1_mean>seq1_median,"True","False")
#if seq1_mean>seq1_median, then True
#if seq1_mean<seq1_median, then False
ans
```

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

```
#print the result
```

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.

```
names<-c("Ann","Bob", "Cindy", "David") #character
scores<-c(40,60,66,45) #numeric
pass<-c(FALSE,TRUE,TRUE,FALSE) #logical
```

```
class(names)
```

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

```
class(scores)
```

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

```
class(pass)
```

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

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.

```
stu_scores<-as.data.frame(cbind(names,scores,pass))
stu_scores
```

```
##   names scores  pass
## 1   Ann     40 FALSE
## 2   Bob     60  TRUE
## 3 Cindy     66  TRUE
## 4 David     45 FALSE
```

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

   Answer: In a matrix, all columns should have the same data type and length, while a data frame is a list of vectors that allows columns to contain different data types. In addition, unlike data frames containing each column's name, matrices do not have built-in column or row names.

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.

```
pass_or_fail<-function(x){
  ifelse(x>=50, "True", "False")
}
#if scores>=50, then True (pass)
#if scores<50, then False (fail)

result<-pass_or_fail(80); result
```

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

```
pass_or_fail_2<-function(x){
  if(x>=50){"True"}
  else{"False"}
}
```

```
result2<-pass_or_fail_2(50);result2
```

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

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

```
result2<-pass_or_fail(scores); result2
```

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

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

    Answer: Both of them work, but the arguments are different. As expressed in the above chunk, the argument of if else is "ifelse(condition, if True, if False)", while the argument of if and else is "if(condition){True} else{False}."