Assignment 2 - R Intermediate

James Strange

1/17/2019

1.) Describe the values stored in the object output. In other words what did the loops create?

```
output
```

```
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa            5.006       3.428        1.462       0.246
## versicolor        5.936       2.770        4.260       1.326
## virginica         6.588       2.974        5.552       2.026
```

```
class(output)
```

```
## [1] "matrix"
```

"output" is a matrix of mean trait values created by subsetting the iris data.frame on the basis of species and applying a mathematical operation to calculate column mean for each particular species subset

2. Describe using pseudo-code how `output` was calculated.

```
Loop from 1 to length of the species identity vector
   Take a subset of the iris data for each species
   Loop from 1 to number of columns in the current iris species subset
      set x = 0 and y = 0
      If the number of rows in current subset column > 0
        Then loop from 1 to number of rows in the current subset column
        x = sum of values in each successive row in the current column
        y = count of number of row elements in the current column
        Calculate each value in the output matrix using the operation "x / y"
```

3. The variables in the loop were named so as to be vague. How can the objects `output`, `x`, and `y` could be renamed such that it is clearer what is occurring in the loop?

```
rename 'output' - 'mean_traits'
rename 'x' - 'trait_sum'
rename 'y' - 'sample_size'
```

4. It is possible to accomplish the same task using fewer lines of code. Please suggest one other way to calculate `output` that decreases the number of loops by 1.

```
sp_ids = unique(iris$Species)

mean_traits = matrix(0, nrow=length(sp_ids), ncol=ncol(iris)-1)
rownames(mean_traits) = sp_ids
colnames(mean_traits) = names(iris[ , -ncol(iris)])

for(i in seq_along(sp_ids)) {
    iris_sp = subset(iris, subset=Species == sp_ids[i], select=-Species)
    for(j in 1:(ncol(iris_sp))) {
```

```
        mean_traits[i, j] = mean(iris_sp[, j])
    }
}
mean_traits
```

```
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa            5.006       3.428        1.462       0.246
## versicolor        5.936       2.770        4.260       1.326
## virginica         6.588       2.974        5.552       2.026
```

5. You have a vector `x` with the numbers 1:10. Write a for loop that will produce a vector `y` that contains the sum of `x` up to that index of `x`. So for example the elements of `x` are 1, 2, 3, and so on and the elements of `y` would be 1, 3, 6, and so on.

```
y = NULL
x = 1:10
for (i in 1:length(x)) {
  y[i] = sum(x[1:i])
}
y
```

```
##  [1]  1  3  6 10 15 21 28 36 45 55
```

6. Modify your for loop so that if the sum is greater than 10 the value of `y` is set to NA

```
y = NULL
x = 1:10
for (i in 1:length(x)) {
  y[i] =  sum(x[1:i])
  if (sum(x[1:i]) > 10) {
    y[i] =  NA
  }
}
y
```

```
##  [1]  1  3  6 10 NA NA NA NA NA NA
```

7. Place your for loop into a function that accepts as its argument any vector of arbitrary length and it will return `y`.

```
# addition function "m"

add_func = function(m) {
  y = NULL
  for (i in 1:length(m)) {
    y[i] = sum(m[1:i])
  }
  return(y)
}

add_func(1:10)
```

```
##  [1]  1  3  6 10 15 21 28 36 45 55
```

```
test_vector = c(1.5, 2.5, 3.5, 4.5, 5.5)
add_func(test_vector)
```

```
## [1]  1.5  4.0  7.5 12.0 17.5
```