

HW2-Intermediate

Sarah Wiegreffe

January 20, 2016

Exercises- Iris Loops

```
data(iris)
#head(iris)
sp_ids = unique(iris$Species)

output = matrix(0, nrow=length(sp_ids), ncol=ncol(iris)-1)
rownames(output) = sp_ids
colnames(output) = 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))) {
    x = 0
    y = 0
    if (nrow(iris_sp) > 0) {
      for(k in 1:nrow(iris_sp)) {
        x = x + iris_sp[k, j]
        y = y + 1
      }
      output[i, j] = x / y
    }
  }
}
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

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

The loops created a matrix storing the mean values of each of the four descriptive factors of the irises for each species.

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

```
#get the unique species names from the iris dataset and store them in a vector
#create a matrix of zeros as placeholders with as many rows as there are unique species and as many columns as there are traits
#name the rows after the unique species names
#name the columns after the column names of the iris dataframe minus the last column
#loop through each species name and subset the iris dataset to the rows of that species, removing the Species column
#loop through each trait (column) of the subset
#if the subset has rows, loop through each row of the subset to sum up the values of each row for each trait
#sum up the number of rows
#calculate the mean of each trait for each species by dividing the sum of values by the number of rows
```

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.

‘output’ could be renamed to ‘means’. ‘x’ could be renamed to ‘sumVals’ and ‘y’ could be renamed to ‘numRows’.

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.

The innermost loop could be removed, and the mean could be calculated by instead using a function such as `mean(col)`.

```
data(iris)
#head(iris)
sp_ids = unique(iris$Species)

means = matrix(0, nrow=length(sp_ids), ncol=ncol(iris)-1)
rownames(means) = sp_ids
colnames(means) = 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))) {
    if (nrow(iris_sp) > 0) {
      means[i, j] = mean(iris_sp[,j])
    }
  }
}
means
```

```
##           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.

```
x = c(1:10)
y = c()
for (i in 1:10) {
  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

```

x = c(1:10)
y = c()
for (i in 1:10) {

  sum = sum(x[1:i])

  if (sum <= 10) {
    y[i] = sum
  }

  else {
    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.

```

calculate_mean = function(x) {
  y = c()
  for (i in 1:length(x)) {

    sum = sum(x[1:i])

    if (sum <= 10) {
      y[i] = sum
    }

    else {
      y[i] = NA
    }
  }
  return(y)
}

#test example
x = c(4, 2, 1, 2, 3, 9, 6, 7, 8, 1, 13)
calculate_mean(x)

```

```
## [1] 4 6 7 9 NA NA NA NA NA NA
```