

Assignment #2

Chelsea Woodruff

1/15/2019

```
data(iris)
head(iris)

##      Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1           5.1           3.5           1.4           0.2  setosa
## 2           4.9           3.0           1.4           0.2  setosa
## 3           4.7           3.2           1.3           0.2  setosa
## 4           4.6           3.1           1.5           0.2  setosa
## 5           5.0           3.6           1.4           0.2  setosa
## 6           5.4           3.9           1.7           0.4  setosa

sp_ids = unique(iris$Species)

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

##              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
```

Exercises

Iris loops

1. Describe the values stored in the object `trait_avg` (originally called 'output'). In other words what did the loops create?

Created a matrix housing the averages of sepal length and width in addition to petal length and width for the species of setosa, versicolor, and virginica.

2. Describe using pseudo-code how `trait_avg` was calculated, for example,

```
Upload iris data. Call all unique species `sp_ids`
Create a matrix with row names being species identities and column names being names
Loop from 0 through all values of species identities
Take a subset of iris data.
i being row. j being column.
x and y start as zero or NULL
x is the sum of all values and y is the number of all values.
```

output is the averages of of sepal length and width **in** addition to petal length and width **for** the species of setosa, versicolor, and virginica.

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` can be renamed to `trait_avg`.

`x` can be renamed to `value_sum`.

`y` can be renamed to `num_values`.

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

Instead of:

```
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
  }
}
```

Substitute with:

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

Sum of a sequence

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 <- NULL
for (i in x){
  y[i] <- sum(x[1:i])
}
```

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 <- NULL
for (i in x){
  y[i] <- sum(x[1:i])
  if (y[i]>10) {
    print('NA')
  }
}
```

```

    }
    else {
      print(y[i])
    }
  }
}

```

```

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

```

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

```

chels_loop <- function(n)
  x <- c(1:n)
  y <- NULL
  for (i in x) {
    y[i] <- print('y')
  }

```

```

## [1] "y"
## [1] "y"
## [1] "y"
## [1] "y"
## [1] "y"
## [1] "y"
## [1] "y"
## [1] "y"
## [1] "y"
## [1] "y"

```