

Assignment2_Enriquez

Alejandra Enriquez

January 21, 2019

```
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
    }
  }
}
```

1. The objects stored in output originally created a matrix with the same number of rows as sp_ids and one less column than the ones on sp_ids with values of 0. The loop took the values stored in the matrix of output and created a set of values in the matrix that make a ratio of x to y, which are calculated from the data in iris.
2. create a matrix to store values,
assign row names and column names from iris
for every iteration i in each Species of iris,
create a subset.
loop from 1 to number of columns of the iris data
if the number of rows in the subset is greater than 0,
then add all the rows for that column of the subset,
divide by the number of rows,
and save the values for each column in the matrix,
print the matrix.
3. The matrix named output could be renamed average_spcstrait, x and y can be renamed to sum_allrowcol and total_spcrow.

4. you can calculate mean with the function `mean()` rather than having another loop to calculate the averages.

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

	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. Here is a chunk of code to sum up x up to the index of x.

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

```
## [1] 1 3 6 10 15 21 28 36 45 55 #this is the output
```

- 6.

Here is a chunk of code that will create "NA" values in y for numbers > 10

```
y=NULL
x=c(1:10)
sumy=0
for (i in 1:10) {
  sumy=sumy + x[i]
  if (sumy>10){
    y[i]="NA"
  }
  else {
    y[i]=sumy
  }
}
y
```

```
## [1] "1" "3" "6" "10" "NA" "NA" "NA" "NA" "NA" "NA" #this is the output
```

7. This function will take any vector that I make and return y. in this case I made another vector, x.

```
x = c(1,3,4)#this is a vector I made up
cum_of_x=function(x){
  num_col=length(x)
  y=matrix(0,1,num_col)
  sumy=0
  for (i in 1:num_col) {
    sumy=sumy + x[i]
    if (sumy >10){
      y[i]="NA"
    }
    else {
      y[i]=sumy
    }
  }
}
y
## [1] "1" "3" "6" "10" "NA" "NA" "NA" "NA" "NA" "NA" #this is the output
```