# Taylor Miller

# Date: 1/21/2019

# Output: Word

# Intermediate R

# Contributors: Sam Norton, Kayla Pehl, JulieAnn, Cassie Evanchuk

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)  
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?

output[i, j] = x / y

#The object output is the means

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

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

#Loop from 1 to length of species identities Take a subset of iris data Loop from 1 to number of columns of the iris data If the iris data row is greater than 0 occurs then loop from 1 to the number of rows in the iris data x equals 0 plus the row and column of the iris data y equals y plus 1

# 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

#To be clearer 'output' could   
  
  
 #X is the sum of the observations  
 #Y is the number of observations  
 #The output is the mean

# 4. 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.

len = tapply(iris$Sepal.Length,iris$Species,mean)  
wid = tapply(iris$Sepal.Width,iris$Species,mean)  
plen = tapply(iris$Petal.Length,iris$Species,mean)  
pwid = tapply(iris$Petal.Width,iris$Species,mean)  
  
iris\_dfs<-data.frame(len,wid,plen,pwid)  
iris\_dfs

## len wid plen pwid  
## 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

#aggregate can be used as well

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

df <- data.frame(cbind(c(1), c(2), c(3), c(4), c(5), c(6), c(7), c(8), c(9), c(10), c(11)))  
  
  
  
 sum\_seq <- function(x) {  
 output <- NULL  
 for(i in 1:ncol(df))  
 output[i] = sum(x[1:i])  
 {print (output)}  
 }  
sum\_seq(x)

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