R_Intermediate_Assignment

Kyra Reisenfeld

January 14, 2020

```
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
```

```
##create unique vector of species names
sp_ids = unique(iris$Species)
##make an empty matrix that is 3x4
output = matrix(0, nrow=length(sp_ids), ncol=ncol(iris)-1)
##assign the species ids as the rownames of the output
rownames(output) = sp_ids
##assign measurements (Sepal.Length Sepal.Width Petal.Length Petal.Width) as the column
 names of the output
colnames(output) = names(iris[ , -ncol(iris)])
##create a loop
for(i in seq_along(sp_ids)) {
  ##subset the measurements so that "Species" is not listed
    iris_sp = subset(iris, subset=Species == sp_ids[i], select=-Species)
    ##identify each column
    for(j in 1:(ncol(iris_sp))) {
        x = 0
        y = 0
        ##for every value that is greater than 0 (everything)
        if (nrow(iris_sp) > 0) {
          ##identify sum of rows and sum of traits for each column
            for(k in 1:nrow(iris_sp)) {
                x = x + iris_sp[k, j]
                y = y + 1
            }
           ##the sum of the values divided by the number of observations
            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 loop created means for each trait for each species.
2. Describe using pseudo-code how `output` was calculated. See comments in code above
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 descibed as the “means” for each measurement. Y could be described as the sum of all of the observations (number of rows), this could be called “sum_rows”. X could be described as the sum of all the measurements for each trait, this could be called “sum_trait”
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.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
iris %>%
  group_by(Species) %>%
  summarise_if(is.numeric, mean)
```

```
## # A tibble: 3 x 5
##   Species    Sepal.Length Sepal.Width Petal.Length Petal.Width
##   <fct>             <dbl>       <dbl>        <dbl>       <dbl>
## 1 setosa             5.01        3.43         1.46       0.246
## 2 versicolor         5.94        2.77         4.26       1.33
## 3 virginica          6.59        2.97         5.55       2.03
```

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`

```
sum_seq <- function(p) {
    d <- NULL
    for(i in p) {
       d[i] = sum(p[1:i])
    }
  print(d)
}
sum_seq(x)
```

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