ApplyRsm

Javier Fong

11/5/2021

apply() Family

The apply() family is a group of functions contained in the R base package.Their main functionality is to manage slices of data in a repetitive way. They were created to avoid the use of loop functions and can slices the R structure in a multitude of different ways.
This family is made up of:

- apply()
- lapply()
- sapply()
- vapply()
- mapply()

apply()

For the purpose of this resume, well store the iris data set in the varaible data

```
data = iris
```

The apply() function has 3 main parameter:

1. X - Array, Matrix, Dataframe
2. MARGIN - This defines how the function is applied:

- MARGIN = 1 if we want the function be applied over the rows
- MARGIN = 2 if we want the function be applied over the columns

3. FUN - The function we want applied over a slice of data

```
#Note data[,1:4] delimits the data to the columns 1 to 4, which are the
#numerical columns

apply(X= data[,1:4], MARGIN = 2, FUN = sum )
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##        876.5        458.6        563.7        179.9
```

In this example, we summed (FUN = sum) all the values in the columns (margin = 2) of data (x = data).

lapply()

lapply() function works almost identically to the apply(), The main diffence between this 2 functions is that lappy returns a list object of size x after applying the function to the data.

```
lapply(
  X= data[,1:4]
  , MARGIN = 2
  , FUN = sum
)
```

```
## $Sepal.Length
## [1] 878.5
##
## $Sepal.Width
## [1] 460.6
##
## $Petal.Length
## [1] 565.7
##
## $Petal.Width
## [1] 181.9
```

If we compare the outputs of both functions, we get the following results:

```
class(apply(X= data[,1:4], MARGIN = 2, FUN = sum))
```

```
## [1] "numeric"
```

```
class(lapply(X= data[,1:4], MARGIN = 2, FUN = sum))
```

```
## [1] "list"
```

Although, the out and calculation may be the same in both implementations. We notice that the diference is in type of object each function return. Apply() in this case returns a numeric type vector, meanwhile lapply always return a list type obejct.

sapply()

The scope of sapply() is a little bigger than the one of lapply(). sapply() makes the iterative application of the desired function, but then proceeds to simplify the out as much as possible.

> Lets note that lapply() can acchive this simplification as well, but you must set the parameter *simplify* as true.

```
sapply(
  data[,1:4]
  , margin = 2
  , FUN = sum
)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##        878.5        460.6        565.7        181.9
```

mapply()

The m in mapply() standas for multivariate. Its functions is to vectorize the arguments of functions that dont tipically accept vectors as arguments. In other word, it simplifies the repetition of a function multiple times by vectorizing the input parameters.

```
# Create a 4x4 matrix
Q1 <- matrix(c(rep(1, 4), rep(2, 4), rep(3, 4), rep(4, 4)),4,4)
print(Q1)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    1    2    3    4
## [3,]    1    2    3    4
## [4,]    1    2    3    4
```

```r
# Or use `mapply()`
Q2 <- mapply(rep,1:4,4)
print(Q2)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    1    2    3    4
## [3,]    1    2    3    4
## [4,]    1    2    3    4
```