

Intro to the Apply Function Family

Seung Yub Yoo, 24444673

12/03/17

Intro to the Apply Function Family

Purpose

Remember lab 08 - Intro to Loops? We learned the various ways in which we iterate operations, or perform the same procedure a number of times: `for`, `repeat`, `while`. While understanding the process of loops is essential to programming in general, loops in R in particular can be slow and inefficient. Many users will advise avoiding loops for R, and instead use the `apply()` functions that are much more efficient at carrying out iterations. In this blog post, we will discuss how to use the family of apply functions in R as an alternative to loops.

The `apply` function family is found in the R base package. If you type “`??apply`” in your R console, you will find the following functions:

```
base::apply Apply Functions
             Over Array Margins
base::subset Internal Objects in
             Package 'base'
base::by Apply a Function to
          a Data Frame Split
          by Factors
base::eapply Apply a Function
              Over Values in an
              Environment
base::lapply Apply a Function
              over a List or Vector
base::mapply Apply a Function to
              Multiple List or
              Vector Arguments
base::rapply Recursively Apply a
              Function to a List
base::tapply Apply a Function
              Over a Ragged
              Array
```

apply functions

These functions can be used as a simple alternative method to using loops. They can manipulate data entries from matrices, arrays, lists, and dataframes by calling a function to perform repetitive operations. For the scope of this intro, we will discuss `apply()`, `lapply()`, `mapply()`, and `tapply()`.

1) `apply()`

`> apply(X, MARGIN, FUN, ...)`

The `apply()` function is a function applied to margins of an array or matrix and returns a vector, array, or a list of values. The margins that the function operates on are the margins of the array: rows denoted by `1`, columns denoted by `2`, or both denoted by `1:2`. Argument `x` is an array that the function is operated on. Argument `MARGIN` is a vector that indicates which margin to be operated on. Argument `FUN` is the function to be applied.

ex)

```
m <- matrix(c(1:5, 51:55, 101:105), nrow = 5, ncol = 3)
m
```

```
##      [,1] [,2] [,3]
## [1,]    1   51 101
## [2,]    2   52 102
## [3,]    3   53 103
## [4,]    4   54 104
## [5,]    5   55 105
```

```
apply(m, 1, mean) #apply the mean function to each row in x
```

```
## [1] 51 52 53 54 55
```

```
apply(m, 2, mean) #apply the mean function to each column in x
```

```
## [1]    3   53 103
```

```
apply(m, c(1, 2), sqrt) #apply the sqrt function to each entry in x
```

```
##           [,1]      [,2]      [,3]
## [1,] 1.000000 7.141428 10.04988
## [2,] 1.414214 7.211103 10.09950
## [3,] 1.732051 7.280110 10.14889
## [4,] 2.000000 7.348469 10.19804
## [5,] 2.236068 7.416198 10.24695
```

```
apply(m, c(1, 2), function(x) sqrt(x[x > 50])) #apply the sqrt function to each entry in x that is greater than 50
```

```
##           [,1]      [,2]      [,3]
## [1,] Numeric,0 7.141428 10.04988
## [2,] Numeric,0 7.211103 10.09950
## [3,] Numeric,0 7.280110 10.14889
## [4,] Numeric,0 7.348469 10.19804
## [5,] Numeric,0 7.416198 10.24695
```

Note: When using user-defined functions, we must wrap up the function and give the data a name so that R recognizes the object for the function.

2) `lapply()`

`> lapply(X, FUN, ...)`

The `lapply()` function is similar to `apply`, except it inputs and outputs lists.

ex)

```
m <- list(x = 1:10, y = 11:20, z = 21:30)
m
```

```
## $x
## [1] 1 2 3 4 5 6 7 8 9 10
##
## $y
## [1] 11 12 13 14 15 16 17 18 19 20
##
## $z
## [1] 21 22 23 24 25 26 27 28 29 30
```

```
lapply(m, sum)
```

```
## $x
## [1] 55
##
## $y
## [1] 155
##
## $z
## [1] 255
```

Note: `sapply()` is a wrapper of `lapply()` by returning a vector instead of a list by default. `vapply()` is similar to `sapply()` but has another argument `FUN.VALUE` that can be used to specify a type of return value from `FUN`.

ex)

```
sapply(m, sum)
```

```
##   x   y   z
## 55 155 255
```

3) `mapply()`

`> mapply(FUN, ...)`

The `mapply()` function is a multivariate version of `sapply()`. It applies the specified function to the first elements of each argument, the second elements, the third elements, and so on.

ex)

```
a <- 1:10
b <- 11:20
mapply(sum, a, b) #apply the sum function to each index of the arguments
```

```
## [1] 12 14 16 18 20 22 24 26 28 30
```

Note: `mapply()` will not work when dealing with arguments of different lengths or when the function only accepts one input.

4) `tapply()`

```
> tapply(X, INDEX, FUN = NULL, ...)
```

The `tapply()` function splits the array based on specified data such as factors and then applies the function to each cell.

ex)

```
library(datasets)
tapply(mtcars$wt, mtcars$cyl, median)
```

```
##      4      6      8
## 2.200 3.215 3.755
```

```
# tapply() groups the cars together based on the number of cylinders and calculates the median weight for each group.
```

5) Summary

The apply function family is an efficient and simplified way of iterating functions repeatedly across rows, columns, and individual cells. As we have learned, the functions can also be used with lists, multiple lists, arrays, and matrices, all the while being able to specify our desired output by using `sapply()` and `vapply()`. Needless to say, using the apply family function becomes more seamless when we, as users, know which apply function to use by observing input data and having a desired type of output. Here is a convenient table to refer to when deciding which apply function to use:

Function Name	Objects the Function Works On	What the Function Sees as Elements	Result Type
apply	Matrix	Rows or columns	Vector, matrix, array, or list
	Array	Rows, columns, or any dimension	Vector, matrix, array, or list
	Data frame	Rows or columns	Vector, matrix, array, or list
sapply	Vector	Elements	Vector, matrix, or list
	Data frame	Variables	Vector, matrix, or list
	List	Elements	Vector, matrix, or list
lapply	Vector	Elements	List
	Data frame	Variables	List
	List	Elements	List

apply, sapply, lapply table

References

1. <https://www.rdocumentation.org/packages/base/versions/3.4.1/topics/apply>
2. <https://datascienceplus.com/using-the-apply-family-of-functions-in-r/>
3. <https://nsaunders.wordpress.com/2010/08/20/a-brief-introduction-to-apply-in-r/>
4. <https://www.r-bloggers.com/r-tutorial-on-the-apply-family-of-functions/>
5. <https://www.r-bloggers.com/using-apply-sapply-lapply-in-r/>
6. <https://www.datacamp.com/community/tutorials/r-tutorial-apply-family#as>
7. <http://www.dummies.com/programming/r/how-to-use-the-apply-family-of-functions-in-r/>