Loop Functions

3rd number of July

Contents

1 Lapply()

1.1 Algorythm

Lapply() does the following series of operations:

1. it loops over a list, iterating over each element in that list
2. it applies a *specified function* to each element of the list
3. it always returns a **list (l-apply)**

1.2 Arguments

1. a list *X*
2. a function *FUN*
3. other argument

1.3 Code example

```
x <- list(1:5) #this list contains just ONE element

lapply <- lapply(x, mean)
print(lapply)
```

```
## [[1]]
## [1] 3
```

```r
class(lapply) # output of LAPPLY is always a list
```

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

2 Sapply()

Sapply() behaves similarly to lapply(), but it will try to **simlify (s-apply)** the result

Result of sapply	Returned class
List with each element of length 1	`vector`
List with each element of the same length > 1	`matrix`
Something else	`list`

3 Tapply()

Tapply() is used to apply a function *over subsets of a vector*

3.1 Arguments

1. `x` is a vector
2. `INDEX` is a factor / list of factors
3. `Function`
4. `...`
5. `simplify` - should it simplify the result like sapply

4 Apply()

Apply() is used to evaluate a function **over the margins of an array** * often used to apply a function to *the rows / columns of a matrix*

4.1 Arguments

1. `x` is an array
2. `MARGIN` is an unteger vector indicating which margins should be 'retained'
   - `1` for rows
   - `2` for columns and so on
3. `Function`
4. `...`

4.2 Code example

```r
x <- matrix(1:10, 2, 5)
print(x)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    3    5    7    9
## [2,]    2    4    6    8   10
```

```r
apply1 <- apply(x, 1, mean) #take the mean of each row
print(apply1)
```

```
## [1] 5 6
```

```
class(apply1)
```

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

```
apply2 <- apply(x, c(1,2), function(y) {y-1})
# apply the funtion to each dimension (2 out of 2)
print(apply2)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    2    4    6    8
## [2,]    1    3    5    7    9
```

```
class(apply2)
```

```
## [1] "matrix"
```

5 Mapply()

Mapply() is used to apply a function **in parallel**

5.1 Arguments

1. `Function`
2. `...` contains R objects to apply over
3. `MoreArgs` is a list of other arguments to `function`
4. `Sumplify` should the result be simplify

5.2 Code example

```
mapply <- mapply(rep, 1:4, 4:1) # instead of typing
     # list(rep(1, 4), rep(2, 3), rep(3, 2), rep(4, 1))
print(mapply)
```

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