



# Introduction to the R Language

Loop Functions - apply

Roger Peng, Associate Professor  
Johns Hopkins Bloomberg School of Public Health

# apply

`apply` is used to evaluate a function (often an anonymous one) over the margins of an array.

- It is most often used to apply a function to the rows or columns of a matrix
- It can be used with general arrays, e.g. taking the average of an array of matrices
- It is not really faster than writing a loop, but it works in one line!

# apply

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

- **X** is an array
- **MARGIN** is an integer vector indicating which margins should be “retained”.
- **FUN** is a function to be applied
- ... is for other arguments to be passed to **FUN**

# apply

```
> x <- matrix(rnorm(200), 20, 10)
> apply(x, 2, mean)
[1]  0.04868268  0.35743615 -0.09104379
[4] -0.05381370 -0.16552070 -0.18192493
[7]  0.10285727  0.36519270  0.14898850
[10]  0.26767260

> apply(x, 1, sum)
[1] -1.94843314  2.60601195  1.51772391
[4] -2.80386816  3.73728682 -1.69371360
[7]  0.02359932  3.91874808 -2.39902859
[10]  0.48685925 -1.77576824 -3.34016277
[13]  4.04101009  0.46515429  1.83687755
[16]  4.36744690  2.21993789  2.60983764
[19] -1.48607630  3.58709251
```

# col/row sums and means

For sums and means of matrix dimensions, we have some shortcuts.

- `rowSums = apply(x, 1, sum)`
- `rowMeans = apply(x, 1, mean)`
- `colSums = apply(x, 2, sum)`
- `colMeans = apply(x, 2, mean)`

The shortcut functions are *much* faster, but you won't notice unless you're using a large matrix.

# Other Ways to Apply

Quantiles of the rows of a matrix.

```
> x <- matrix(rnorm(200), 20, 10)
> apply(x, 1, quantile, probs = c(0.25, 0.75))
```

|     | [,1]        | [,2]        | [,3]       | [,4]        |
|-----|-------------|-------------|------------|-------------|
| 25% | -0.3304284  | -0.99812467 | -0.9186279 | -0.49711686 |
| 75% | 0.9258157   | 0.07065724  | 0.3050407  | -0.06585436 |
|     | [,5]        | [,6]        | [,7]       | [,8]        |
| 25% | -0.05999553 | -0.6588380  | -0.653250  | 0.01749997  |
| 75% | 0.52928743  | 0.3727449   | 1.255089   | 0.72318419  |
|     | [,9]        | [,10]       | [,11]      | [,12]       |
| 25% | -1.2467955  | -0.8378429  | -1.0488430 | -0.7054902  |
| 75% | 0.3352377   | 0.7297176   | 0.3113434  | 0.4581150   |
|     | [,13]       | [,14]       | [,15]      | [,16]       |
| 25% | -0.1895108  | -0.5729407  | -0.5968578 | -0.9517069  |
| 75% | 0.5326299   | 0.5064267   | 0.4933852  | 0.8868922   |
|     | [,17]       | [,18]       | [,19]      | [,20]       |

# apply

Average matrix in an array

```
> a <- array(rnorm(2 * 2 * 10), c(2, 2, 10))
> apply(a, c(1, 2), mean)
      [,1]      [,2]
[1,] -0.2353245 -0.03980211
[2,] -0.3339748  0.04364908

> rowMeans(a, dims = 2)
      [,1]      [,2]
[1,] -0.2353245 -0.03980211
[2,] -0.3339748  0.04364908
```