APPLY FUNCTIONS IN R

IST 719

SPRING 17

Pan Chen

APPLY: WHAT ARE THESE FUNCTIONS IN R?

Help pages:

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

- They apply functions to manipulate slices of data. Their results can be use useful for the visualization
- With these apply functions we can avoid looping, and write code that is meaningful in a statistical setting.

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

- Applies to every column and/or every row in a matrix or an array
- X: an array or a matrix, MARGIN value could be 1, 2 or c(1,2). 1 means applies to every row, 2 means apply to every column, and c(1,2) means apply to every row and column.
- FUN: the function we want to apply (mean, max, median, sum, etc..or your own function)
- Returns a vector

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

```
         col1 col2 col3 col4 col5
Neo         1    2    3    4    5
Trinity     6    7    8    9   10
```

```
> apply(A,1,mean)
    Neo Trinity
      3       8
```

```
> apply(A,2,mean)
col1 col2 col3 col4 col5
 3.5  4.5  5.5  6.5  7.5
```

```
> apply(A, c(1:2), function(x) x/2)
         col1 col2 col3 col4 col5
Neo       0.5  1.0  1.5  2.0  2.5
Trinity   3.0  3.5  4.0  4.5  5.0
```

lapply (X,FUN…) sapply (X,FUN…)

- Apply function to list element(s) in a list or column(s) in a dataframe
- X: the list or vector(s) you want to apply a function on
- The difference: lapply returns a list, sapply returns a vector.

```
> lapply(mtcars,max)
$mpg
[1] 33.9

$cyl
[1] 8

$disp
[1] 472

$hp
[1] 335

$drat
[1] 4.93
```

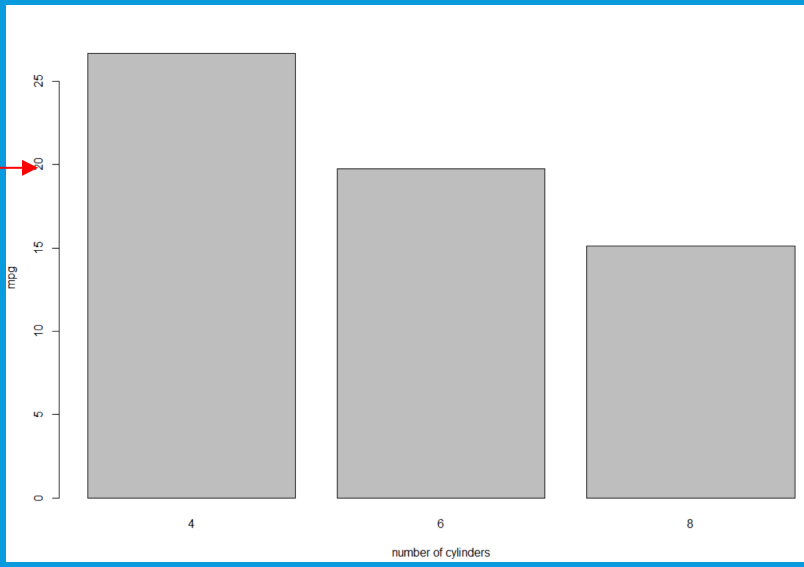
```
> sapply(mtcars,max)
   mpg    cyl   disp     hp   drat     wt   qsec     vs     am   gear   carb
33.900  8.000 472.000 335.000  4.930  5.424 22.900  1.000  1.000  5.000  8.000
```

tapply(X,INDEX,FUN…)

- Apply a function on subset of the vector broken down by a given factor variable.
- X: usually a vector, INDEX: a list of grouping factors, each of same length as X
- Useful for plotting comparisons by groups

```
> tapply(mtcars$mpg,mtcars$cyl,mean)
       4        6        8
26.66364 19.74286 15.10000
```

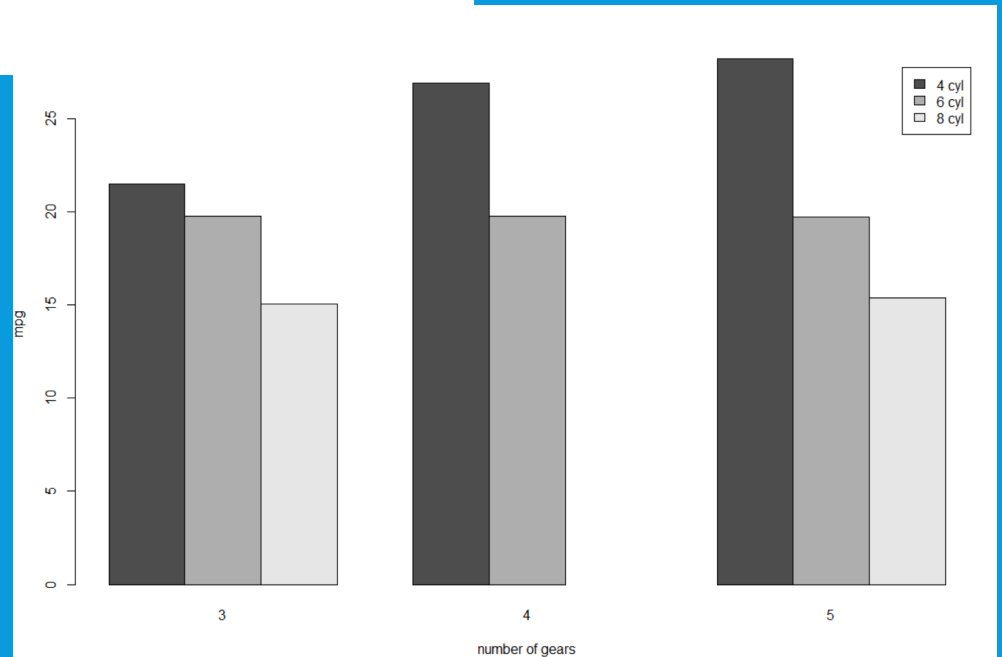
- Apply MEAN to mpg GROUP BY cyl and return an array



tapply(X,INDEX,FUN…)

```
> tapply(mtcars$mpg,list(cyl=mtcars$cyl,gear=mtcars$gear),mean)
   gear
cyl     3      4    5
  4 21.50 26.925 28.2
  6 19.75 19.750 19.7
  8 15.05     NA 15.4
```

Apply mean to mpg column group by cyl and gear and return a matrix
With cyl as rows and gear as columns



tapply() vs aggregate()

- Both are able to apply function to subsets of data

- tapply:

Return a matrix or array

- Aggregate:

Return a data frame with one row for each category

```
> tapply(mtcars$mpg,list(cyl=mtcars$cyl,gear=mtcars$gear),mean)
   gear
cyl     3      4    5
  4 21.50 26.925 28.2
  6 19.75 19.750 19.7
  8 15.05     NA 15.4
```

```
> aggregate(mpg~gear+cyl,data=mtcars,FUN=mean)
  gear cyl    mpg
1    3   4 21.500
2    4   4 26.925
3    5   4 28.200
4    3   6 19.750
5    4   6 19.750
6    5   6 19.700
7    3   8 15.050
8    5   8 15.400
```

THANK YOU!!!

Pan Chen