Biostatistics II: Introduction to R

The apply family

Eleni-Rosalina Andrinopoulou

Department of Biostatistics, Erasmus Medical Center

e.andrinopoulou@erasmusmc.nl

@erandrinopoulou

In this Section

- The apply family
- A lot of practice

What is the apply Family

Manipulate **vectors** or slices of data from **matrices**, **arrays**, **data frames** and **lists** in a repetitive way

- An aggregating function, like for example the mean, or the sum
- Other transforming or subsetting functions
- Other vectorized functions, which return more complex structures like lists, vectors and matrices

What is the apply Family

apply(), lapply() , sapply(), tapply(), mapply()

But how and when should we use these?

How To Use apply() in R

- Operates on **matrices**, **arrays** and **data frames**

- By column

```
mat <- matrix(1:4, nrow = 2, ncol = 2)
mat
```

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

```
apply(mat, 2, sum)
```

```
[1] 3 7
```

- By row

```
apply(mat, 1, sum)
```

```
[1] 4 6
```

How To Use apply() in R

- Operates on **matrices**, **arrays** and **data frames**

- By column

```
mat <- matrix(1:4, nrow = 2, ncol = 2)
mat
```

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

```
apply(mat, 2, mean)
```

```
[1] 1.5 3.5
```

- By row

```
apply(mat, 1, mean)
```

```
[1] 2 3
```

How To Use apply() in R

- You can also apply your own functions
- By column

```
mat <- matrix(1:4, nrow = 2, ncol = 2)
mat
```

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

```
apply(mat, 2, function(x) { sum(x)/(length(x)-1) } )
```

```
[1] 3 7
```

How To Use lapply() in R

- Apply a given function to every element of a `list` and return a `list`

- The difference with `apply()`:
  - It can be used for other objects like `vector`, `data.frame` or `list`
  - The output returned is a list

How To Use lapply() in R

```
myList <- list(x = c(1:6),
               y = c("m", "f"),
               z = c(30, 4, 23))
myList
```

```
$x
[1] 1 2 3 4 5 6

$y
[1] "m" "f"

$z
[1] 30  4 23
```

- Use pre-specified functions

```
lapply(myList, length)
```

```
$x
[1] 6

$y
[1] 2

$z
[1] 3
```

How To Use lapply() in R

```
myList <- list(x = c(1:6),
               y = c("m", "f"),
               z = c(30, 4, 23))
myList
```

```
$x
[1] 1 2 3 4 5 6

$y
[1] "m" "f"

$z
[1] 30  4 23
```

- Use pre-specified functions

```
lapply(myList, median)
```

```
$x
[1] 3.5

$y
[1] NA

$z
[1] 23
```

- You can also apply your own functions!

How To Use sapply() in R

- `sapply()` is similar to `lapply()`, but it tries to simplify the output

```
myList <- list(x = c(1:6),
               y = c("m", "f"),
               z = c(30, 4, 23))
myList
```

```
$x
[1] 1 2 3 4 5 6

$y
[1] "m" "f"

$z
[1] 30  4 23
```

- Use pre-specified functions

```
sapply(myList, length)
```

```
x y z
6 2 3
```

```
sapply(myList, median)
```

```
   x    y    z
 3.5   NA 23.0
```

- You can also apply your own functions!

How To Use tapply() in R

- Apply a function to subsets of a `vector` - the subsets are defined by some other `vector`, usually a factor

```
tapply(pbc$bili, pbc$sex, mean)
```

```
       m        f
2.865909 3.262567
```

```
tapply(pbc$age, pbc$sex, median)
```

```
       m        f
54.00137 50.19302
```

How To Use tapply() in R

- You can also apply your own functions

```
tapply(pbc$bili, pbc$sex, function(x) { sum(x)/(length(x)-1) } )
```

```
       m        f
2.932558 3.271314
```

How To Use mapply() in R

- Multivariate apply
- Its purpose is to be able to vectorize arguments to a function that is not usually accepting `vectors` as arguments
- `mapply()` applies a function to multiple `list` or multiple `vector` arguments

```
mapply(length, pbc)
```

```
      id     time   status      trt      age      sex  ascites
     418      418      418      418      418      418      418
  hepato  spiders    edema     bili     chol  albumin   copper
     418      418      418      418      418      418      418
alk.phos      ast     trig platelet  protime    stage
     418      418      418      418      418      418
```

How To Use mapply() in R

- Overlapping between functions

```
myList <- list(x = c(1:6),
               y = c("m", "f"),
               z = c(30, 4, 23))
mapply(length, myList,
       SIMPLIFY = FALSE)
```

```
$x
[1] 6

$y
[1] 2

$z
[1] 3
```

```
lapply(myList, length)
```

```
$x
[1] 6

$y
[1] 2

$z
[1] 3
```

- You can also apply your own functions!

Summary

apply(), lapply() , sapply(), tapply(), mapply()

- Useful functions for data manipulation
- Overlapping between functions

Practice

- Use the following webpage to further investigate the apply family https://emcbiostatistics.shinyapps.io/the_apply_family/