STOR 320 Programming III

Lecture 14

Yao Li

Department of Statistics and Operations Research

UNC Chapel Hill

Loop Functions (Apply Functions)

- Chapter 18 in *R Programming for Data Science*
- lapply(): Loop over a list and evaluate a function on each element
- sapply(): Same as lapply but try to simplify the result
- apply(): Apply a function over the margins of an array

lapply()

- Operates on list, data frame and vectors

```
lapply(X, FUN, …)
```

- Arguments:
  - X: A vector or an object
  - FUN: Function applied to each element of x
  - …: Other arguments not in loop

lapply()

```{r}
head(cars,5)
```

	speed <dbl>	dist <dbl>
1	4	2
2	4	10
3	7	4
4	7	22
5	8	16

5 rows

```{r}
lapply(cars, min)
```

```
$speed
[1] 4

$dist
[1] 2
```

```{r}
typeof(lapply(cars, min))
```

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

```{r}
lapply(cars, mean)
```

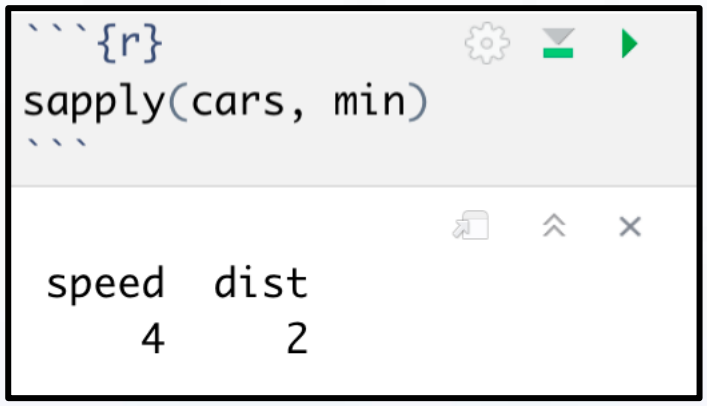
```
$speed
[1] 15.4

$dist
[1] 42.98
```

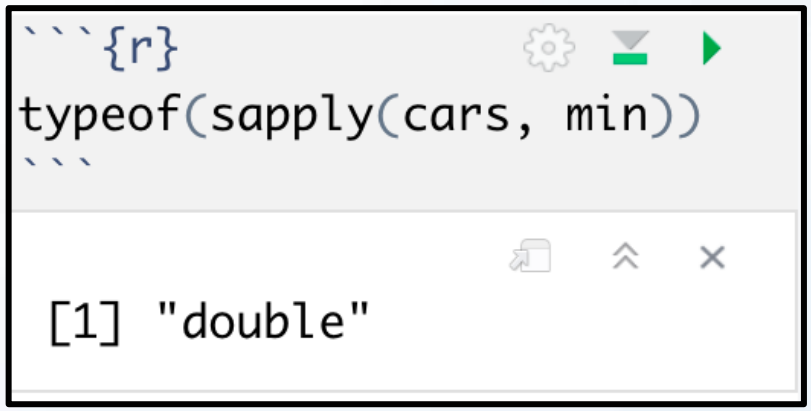
sapply()

- Like lapply() but simplifies the output

```
sapply(X, FUN, OTHER)
```

- Arguments:
  - X: A vector or an object
  - FUN: Function applied to each element of x
  - OTHER: Other arguments not in loop





Example

- Use sapply() or lapply() to generate 100 random samples from normal distributions with means of 0 to 99 and save the random samples to A.
  - The k-th component of a is generated from N(k-1,1).

```r
set.seed(200)
a = rep(NA,100)
for (k in 1:100) {
  a[k]=rnorm(1,k-1,1)
}
set.seed(200)
A = sapply(0:99,rnorm,n=1,sd=1)
identical(a,A)
```

```
[1] TRUE
```

apply()

- Takes data frame or matrix as an input
- Gives output in vector, list or array

```
apply(X, MARGIN, FUN, …)
```

  - X: an array or matrix
  - MARGIN: take a value or range between 1 and 2 to define where to apply the function:
    - MARGIN=1: the manipulation is performed on rows
    - MARGIN=2: the manipulation is performed on columns
    - MARGIN=c(1,2): the manipulation is performed on rows and columns
  - FUN: tells which function to apply.

apply()

```{r}
M = matrix(1:1000,100,10)
sum_row = apply(M, 1, sum)
sum_col = apply(M, 2, sum)
identical(sum_row,as.integer(rowSums(M)))
identical(sum_col,as.integer(colSums(M)))
```

```
[1] TRUE
[1] TRUE
```

```{r}
for(i in 1:5){
  MIN=apply(Cigar, 2, min)
  Q1=apply(Cigar, 2, quantile,probs=0.25)
  Q2=apply(Cigar, 2, quantile,probs=0.5)
  Q3=apply(Cigar, 2, quantile,probs=0.75)
  MAX=apply(Cigar, 2, max)
  FiveSum.Cigar=rbind(MIN,Q1,Q2,Q3,MAX)
}
FiveSum.Cigar
```

	state	year	price	pop	pop16	cpi	ndi	sales	pimin
MIN	1.0	63.0	23.400	319.00	215.200	30.6	1322.573	53.4	23.400
Q1	15.0	70.0	34.775	1053.00	781.175	38.8	3327.869	107.9	31.975
Q2	26.5	77.5	52.300	3174.00	2315.300	62.9	6281.201	121.2	46.400
Q3	40.0	85.0	98.100	5280.25	3914.325	107.6	11024.110	133.2	90.500
MAX	51.0	92.0	201.900	30703.30	22920.000	140.3	23074.000	297.9	178.500

Statistical Programming Assignment

- Instructions
  - Download Analysis 3 Zip Folder
  - Unzip Folder
  - Open Analysis 3 Rmd File
  - Knit to HTML
  - Read Introduction

- Three Part Assignment
  - Each Part Self Contained
  - Most Answers Require Copy-and- Paste
  - Where You See COMPLETE You Should Write/Place Code
  - Leave Code as is When You See #DO NOT CHANGE

Part 1: Discussion

- Process of Programming
    - Create Practice Example
    - Check Code Works
    - Apply Code to Real Data
    - Check Code Works
    - Create a Function of the Process
- Goals
    - Create a Function that Creates a Factor Variable of Abbreviated Weekdays (Easy)
    - Create a Function that Creates a Plot (Difficult)
- Start Working (15 min)

Part 2: Discussion

- Focus on Traffic Volume (DATA2) and Specific Location (“L103”)
- Look at Table
- Goal: Reconstruct this Table

```
head(OUTPUT) #DO NOT CHANGE
```

```
## # A tibble: 6 x 3
##     DAY median   IQR
##   <int>  <dbl> <dbl>
## 1     3     85  34
## 2     4     84  37
## 3     5     76  34.2
## 4     6     83  33
## 5     7     79  38.0
## 6    10     87  21
```

Part 2: Discussion

- Steps:
  - Given the Day in April, Create Function that Outputs the Associated Row
  - Use the Function in a Loop to Construct the Table
- Two Loops
  - Initiate with NULL
  - Initiate with Empty Tibble

- Look at Lecture on Loops

- Q3 is Tricky

- Start Working (15 min)

Part 3: Discussion

- Functions That Apply Functions Across Dimensions Of R Object
- Doesn't Require a Loop
- Tibbles are Matrices
  - Apply Functions to Rows
  - Apply Functions to Columns (Think Summarize)
- Apply() Function to Matrix
  - To Rows `apply(Matrix, 1, Function)`
  - To Columns `apply(Matrix, 2, Function)`
- Start Working (Rest of Class)