

# LECCIÓN 1

Sebastian Delgado

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function Addins

Avance de clase.R\*

```
618 s <- split(airquality,airquality$Month)
619 lapply(s, function(x) colMeans(x[,1:3]))
620
621 sapply(s, function(x) colMeans(x[,1:3]))
622
623
624 # interpretar lo siguiente
638.8 Clase 29/09/2016 split
```

Console

```
| Finally, let's pretend you'd like to view the contents of a variable that you created earlier, but
| you can't seem to remember if you named it my_div or myDiv. You could try both and see what works,
| or...
...
| ===== 97%
| You can type the first two letters of the variable name, then hit the Tab key (possibly more than
| once). Most programming environments will provide a list of variables that you've created that begin
| with 'my'. This is called auto-completion and can be quite handy when you have many variables in your
| workspace. Give it a try. (If auto-completion doesn't work for you, just type my_div and press
| Enter.)
> my_div
[1] 3.478505 3.181981 2.146460
| Perseverance, that's the answer.
| ===== 100%
| would you like to receive credit for completing this course on coursera.org?
1: No
2: Yes
Selection:
```

Environment

Global Environment

Variable	Class	Value
my_div	num [1:3]	3.48 3.18 2.15
my_div2	num [1:3]	0.316 2.828 1.463

Files Plots Packages Help Viewer

R: Combine Values into a Vector or List

Combine Values into a Vector or List

Description

This is a generic function which combines its arguments.

The default method combines its arguments to form a vector. All arguments are coerced to a common type which is the type of the returned value, and all attributes except names are removed.

Usage

c(..., recursive = FALSE)

Arguments

... objects to be concatenated.

recursive logical. If recursive = TRUE, the function recursively descends through lists (and pairlists) combining all their elements into a vector.

Pregúntame cualquier cosa

09:28 p. m. 03/10/2016

# LECCIÓN 2

Sebastian Delgado

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function Addins

Avance de clase.R\*

```
618 s <- split(airquality,airquality$Month)
619 lapply(s, function(x) colMeans(x[,1:3]))
620
621 sapply(s, function(x) colMeans(x[,1:3]))
622
623
624 # interpretar lo siguiente
638.8 Clase 29/09/2016 split
```

Console

```
| Take nothing but results. Leave nothing but assumptions. That sounds like 'Take nothing but
| pictures. Leave nothing but footprints.' But it makes no sense! Surely our readers can come up
| with a better motto . . .
...
| ===== 95%
| In this lesson, you learned how to examine your R workspace and work with the file system of your
| machine from within R. Thanks for playing!
...
| ===== 100%
| would you like to receive credit for completing this course on coursera.org?
1: Yes
2: No
Selection:
```

Environment

Global Environment

Variable	Class	Value
my_div	num [1:3]	3.48 3.18 2.15
my_div2	num [1:3]	0.316 2.828 1.463

Files Plots Packages Help Viewer

R: Manipulation of Directories and File Permissions

dir.exists(paths)

dir.create(path, showWarnings = TRUE, recursive = FALSE, mode = "0777", use\_umask = TRUE)

Sys.chmod(paths, mode = "0777", use\_umask = TRUE)

Sys.umask(mode = NA)

Arguments

path a character vector containing a single path name. Tilde expansion (see [path.expand](#)) is done.

paths character vectors containing file or directory paths. Tilde expansion (see [path.expand](#)) is done.

showWarnings logical: should the warnings on failure be shown?

recursive logical: Should elements of the path other than the last be created? If true, like the Unix command mkdir -p.

mode the mode to be used on Unix-alikes: it will be coerced by [as.octmode](#). For Sys.chmod it is recycled along paths.

use\_umask logical: should the mode be restricted by the umask setting?

Details

Pregúntame cualquier cosa

10:17 p. m. 03/10/2016

## LECCION 3

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function Addins

Avance de clase.R

```
636 install.packages("swirl")
637 library("swirl")
638 swirl()
639
640
641
642
```

Sebastian Delgado

Environment History

Import Dataset

Global Environment

1L  
moneda  
my\_seq  
num [1:30] 5 5.17 5.34 5.52 5.69 ...

Files Plots Packages Help Viewer

R Colon Operator

Find in Topic

Colon (base)

R Documentation

Colon Operator

Description

Generate regular sequences.

Usage

from:to  
a:b

Arguments

from starting value of sequence.  
to (maximal) end value of the sequence.  
a, b **factors** of the same length.

Details

The binary operator : has two meanings: for factors a:b is equivalent to

Console

~/

```
| If instead we want our vector to contain 10 repetitions of the vector (0, 1, 2), we can do
| rep(c(0, 1, 2), times = 10). Go ahead.
> rep(c(0,1,2),times=10)
[1] 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2
| You are really on a roll!
|=====| 96%
| Finally, let's say that rather than repeating the vector (0, 1, 2) over and over again, we want
| our vector to contain 10 zeros, then 10 ones, then 10 twos. We can do this with the 'each'
| argument. Try rep(c(0, 1, 2), each = 10).
> rep(c(0,1,2),each=10)
[1] 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
| All that practice is paying off!
|=====| 100%
| would you like to receive credit for completing this course on coursera.org?
1: Yes
2: No
Selection: |
```

Pregúntame cualquier cosa

11:10 a.m.  
04/10/2016

## LECCION 4

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function Addins

Avance de clase.R

```
636 install.packages("swirl")
637 library("swirl")
638 swirl()
639
640
641
642
```

Sebastian Delgado

Environment History

Import Dataset

Global Environment

1L  
chr [1:3] "my" "name" "is"  
chr [1:4] "my" "name" "is" "Sebastian"

Files Plots Packages Help Viewer

R Colon Operator

Find in Topic

Colon (base)

R Documentation

Colon Operator

Description

Generate regular sequences.

Usage

from:to  
a:b

Arguments

from starting value of sequence.  
to (maximal) end value of the sequence.  
a, b **factors** of the same length.

Details

The binary operator : has two meanings: for factors a:b is equivalent to

Console

~/

```
| Since the character vector LETTERS is longer than the numeric vector 1:4, R simply recycles, or
| repeats, 1:4 until it matches the length of LETTERS.
...
|=====| 95%
| Also worth noting is that the numeric vector 1:4 gets 'coerced' into a character vector by the
| paste() function.
...
|=====| 97%
| we'll discuss coercion in another lesson, but all it really means is that the numbers 1, 2, 3,
| and 4 in the output above are no longer numbers to R, but rather characters "1", "2", "3", and
| "4".
...
|=====| 100%
| would you like to receive credit for completing this course on coursera.org?
1: No
2: Yes
Selection:
```

Pregúntame cualquier cosa

12:03 p.m.  
04/10/2016

## LECCIÓN 5

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function Addins

Avance de clase.R

```
1 #Clase del 25/08/2016
2
3 #Crear vectores con la función c
4 x <- vector(mode="numeric", length=5)
5 x
6
7 #Crear vectores con la función c
```

Run Source

Environment History

Global Environment

my_data	num [1:100]	NA 1.416 -0.512 NA...
my_na	logi [1:100]	TRUE FALSE FALSE ...
name		"foo"

Files Plots Packages Help Viewer

R: Colon Operator

Find in Topic

Colon (base) R Documentation

### Colon Operator

Description

Generate regular sequences.

Usage

from:to  
a:b

Arguments

from starting value of sequence.  
to (maximal) end value of the sequence.  
a, b **factors** of the same length.

Details

The binary operator : has two meanings: for factors a:b is

Console

```
|=====| 90%
| Now that we've got NAs down pat, let's look at a second type of missing value -- NaN, which stands for 'not a
| number'. To generate NaN, try dividing (using a forward slash) 0 by 0 now.
> 0/0
[1] NaN
| You're the best!
|=====| 95%
| Let's do one more, just for fun. In R, Inf stands for infinity. What happens if you subtract Inf from Inf?
> Inf-Inf
[1] NaN
| All that practice is paying off!
|=====| 100%
| would you like to receive credit for completing this course on coursera.org?
1: Yes
2: No
Selection: |
```

Pregúntame cualquier cosa

07:07 p.m.  
04/10/2016

## LECCIÓN 6

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function Addins

Avance de clase.R

```
1 #Clase del 25/08/2016
2
3 #Crear vectores con la función c
4 x <- vector(mode="numeric", length=5)
5 x
6
7 #Crear vectores con la función c
```

Run Source

Environment History

Global Environment

logi	[1:5]	FALSE FALSE TRUE FA...
vect	Named num [1:3]	11 2 NA
vect2	Named num [1:3]	11 2 NA

Files Plots Packages Help Viewer

R: Colon Operator

Find in Topic

Colon (base) R Documentation

### Colon Operator

Description

Generate regular sequences.

Usage

from:to  
a:b

Arguments

from starting value of sequence.  
to (maximal) end value of the sequence.  
a, b **factors** of the same length.

Details

The binary operator : has two meanings: for factors a:b is

Console

```
|=====| 95%
| Likewise, we can specify a vector of names with vect[c("foo", "bar")]. Try it out.
> vect[c("foo", "bar")]
foo bar
11 2
| You are doing so well!
|=====| 97%
| Now you know all four methods of subsetting data from vectors. Different approaches are best in different
| scenarios and when in doubt, try it out!
...
|=====| 100%
| would you like to receive credit for completing this course on coursera.org?
1: No
2: Yes
Selection: |
```

Pregúntame cualquier cosa

07:22 p.m.  
04/10/2016

## LECCIÓN 7

Sebastian Delgado

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function Addins

Project: (None)

Environment History

Global Environment

Data

my\_data 4 obs. of 6 variables

my\_matrix int [1:4, 1:5] 1 2 3 4 5 6 7...

my\_matrix2 int [1:4, 1:5] 1 2 3 4 5 6 7...

my\_vector int [1:4, 1:5] 1 2 3 4 5 6 7...

Files Plots Packages Help Viewer

R: Matrices

Find in Topic

matrix (base) R Documentation

Matrices

Description

matrix creates a matrix from the given set of values.

as.matrix attempts to turn its argument into a matrix.

is.matrix tests if its argument is a (strict) matrix.

Usage

matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)

as.matrix(x, ...)

## S3 method for class 'data.frame'

as.matrix(x, rownames.force = NA, ...)

Console

Let's see if that got the job done. Print the contents of my\_data.

> my\_data

patient	age	weight	bp	rating	test
1	Billy	1	5	9	13
2	Gina	2	6	10	14
3	Kelly	3	7	11	15
4	Sean	4	8	12	16

You are really on a roll!

In this lesson, you learned the basics of working with two very important and common data structures -- matrices and data frames. There's much more to learn and we'll be covering more advanced topics, particularly with respect to data frames, in future lessons.

would you like to receive credit for completing this course on coursera.org?

1: No

2: Yes

Selection: 1

## LECCIÓN 8

Sebastian Delgado

RStudio

File Edit Code View Plots Session Build Debug Tools Help

Go to file/function Addins

Project: (None)

Environment History

Global Environment

ints int [1:10] 6 4 10 1 8 2 3 5 9 7

Files Plots Packages Help Viewer

R: Which indices are TRUE?

Find in Topic

which (base) R Documentation

Which indices are TRUE?

Description

Give the TRUE indices of a logical object, allowing for array indices.

Usage

which(x, arr.ind = FALSE, useNames = TRUE)

arrayInd(ind, .dim, .dimnames = NULL, useNames = FALSE)

Arguments

x a logical vector or array. NAs are allowed and omitted (treated as if FALSE).

arr.ind logical; should array indices be returned when x is an array?

ind integer-valued index vector, as resulting from which(x).

.dim dim(.) integer vector

.dimnames optional list of character dimnames(.). If useNames is true, to be used for constructing dimnames for arrayInd(.) (and hence, which(\*, arr.ind=TRUE)). If names(.dimnames) is not empty

Console

That's the answer I was looking for.

which of the following evaluates to TRUE?

1: any(ints == 2.5)

2: any(ints == 10)

3: all(c(TRUE, FALSE, TRUE))

4: all(ints == 10)

Selection: 2

You nailed it! Good job!

That's all for this introduction to logic in R. If you really want to see what you can do with logic, check out the control flow lesson!

would you like to receive credit for completing this course on coursera.org?

1: No

2: Yes

Selection: 1

## LECCIÓN 9

The screenshot shows the RStudio IDE interface. The main editor window displays an R script with the following content:

```

9 # Let's say I wanted to define a binary operator that multiplied two numbers and
10 # then added one to the product. An implementation of that operator is below:
11
12 # "%mult_add_one%" <- function(left, right) { Notice the quotation marks!
13 #   left * right + 1
14 # }
15
16 # I could then use this binary operator like `4 %mult_add_one% 5` which would
17 #   21
18
19 # "%mult_add_one%"

```

The console window shows the output of the script:

```

| Use %p% in between each string.
> "I" %p% "love" %p% "R!"
[1] "I love R!"
| Excellent work!
| ===== | 98%
| We've come to the end of our lesson! Go out there and write some great
| functions!
...
| ===== | 100%
| Would you like to receive credit for completing this course on Coursera.org?
1: No
2: Yes

```

The Environment pane on the right shows the function being called:

```

Functions
%mult_add_one% function (left, right)
horning_functi function (x)
evaluate function (func, dat)
f function (a, b)
hilbert function (n)
hilert function (n)

```

The Files pane on the right shows the project structure:

```

Files Plots Packages Help Viewer
R: Concatenate Strings Find in Topic
paste (base) R Documentation
Concatenate Strings
Description
Concatenate vectors after converting to character.
Usage
paste(..., sep = " ", collapse = NULL)
paste0(..., collapse = NULL)
Arguments

```

## LECCIÓN 10

The screenshot shows the RStudio IDE interface. The main editor window displays R code and console output. The code defines a function `lapply()` and uses it to process a list. The console output shows the results of the function calls. The environment pane on the right shows the 'Global Environment' with variables like 'i', 'j', 'letra', 'moneda', 'name', and 'u'. The Files pane shows 'R: Concatenate Strings'.

**Code in Editor:**

```
#Clase del 25/08/2016
1
2
3
...
# The only difference between previous examples and this one is that we are defining
# and using our own function right in the call to lapply(). Our function has no name
# and disappears as soon as lapply() is done using it. So-called 'anonymous functions'
# can be very useful when one of R's built-in functions isn't an option.
...
# In this lesson, you learned how to use the powerful lapply() and sapply() functions
# to apply an operation over the elements of a list. In the next lesson, we'll take a
# look at some close relatives of lapply() and sapply().
...
# Would you like to receive credit for completing this course on Coursera.org?
1: Yes
2: No
Selection:
```

**Environment Pane:**

Variable	Value
i	100L
j	3L
letra	"d"
moneda	1L
name	"foo"
u	logi [1:5] FALSE FALSE TRUE FALS...
unique_vals	List of 30

**Files Pane:**

R: Concatenate Strings

**Viewer Pane:**

### Concatenate Strings

**Description**

Concatenate vectors after converting to character.

**Usage**

```
paste (... , sep = " ", collapse = NULL)
paste0 (... , collapse = NULL)
```

**Arguments**

...

## LECCIÓN 11

RStudio interface showing Lesson 11. The console displays progress from 3% to 100% with a 'You are amazing!' message. The Environment pane shows functions like hilbert, hilert, mayor10, and mayorque. The Viewer pane displays the 'Apply a Function Over a Ragged Array' documentation.

Console output:

```
3: 1010.0
4: 157.00
5: 56.00

Selection:
Enter an item from the menu, or 0 to exit
Selection: 5

| You are amazing!

===== | 96%

| In this lesson, you learned how to use vapply() as a safer alternative to sapply(),
| which is most helpful when writing your own functions. You also learned how to use
| tapply() to split your data into groups based on the value of some variable, then
| apply a function to each group. These functions will come in handy on your quest to
| become a better data analyst.

...

===== | 100%

| Would you like to receive credit for completing this course on Coursera.org?
1: Yes
2: No
```

Environment pane:

Function	Class
hilbert	function (a, b)
hilert	function (n)
mayor10	function (n)
mayorque	function (x, n)

Viewer pane: Apply a Function Over a Ragged Array

Description: Apply a function to each cell of a ragged array, that is to each (non-empty) group of values given by a unique combination of the levels of certain factors.

Usage: tapply(X, INDEX, FUN = NULL, ..., simplify = TRUE)

Arguments: tapply(X, INDEX, FUN = NULL, ..., simplify = TRUE)

## LECCIÓN 12

RStudio interface showing Lesson 12. The console displays progress from 92% to 100% with a 'You are amazing!' message. The Environment pane shows data objects like plants, x, and y. The Viewer pane displays the 'Apply a Function Over a Ragged Array' documentation.

Console output:

```
...

===== | 92%

| str() is actually a very general function that you can use on most objects in R. Any time you
| want to understand the structure of something (a dataset, function, etc.), str() is a good
| place to start.

...

===== | 96%

| In this lesson, you learned how to get a feel for the structure and contents of a new dataset
| using a collection of simple and useful functions. Taking the time to do this upfront can
| save you time and frustration later on in your analysis.

...

===== | 100%

| Would you like to receive credit for completing this course on Coursera.org?
```

Environment pane:

Object	Class	Value
plants	data.frame	5166 obs. of 10 variables
x	matrix	Large matrix (1000000 elements)
y	matrix	num [1:2, 1:2] 10 10 10 10

Viewer pane: Apply a Function Over a Ragged Array

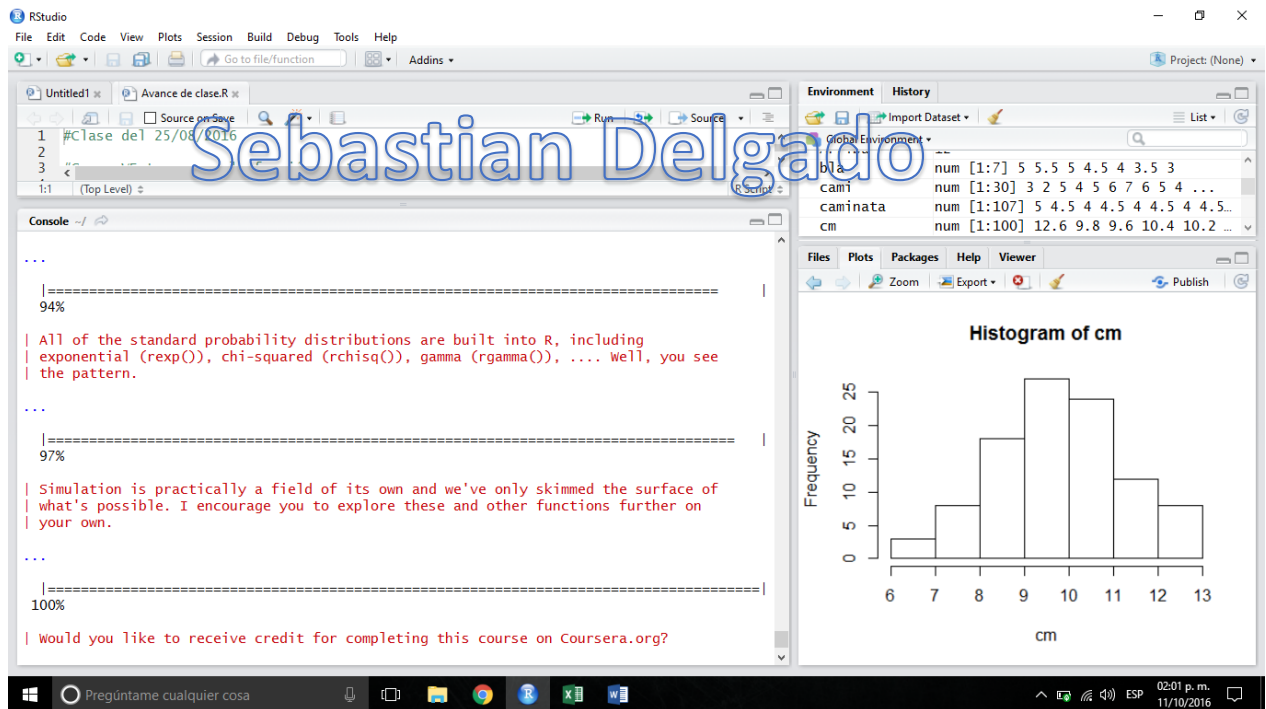
Description: Apply a function to each cell of a ragged array, that is to each (non-empty) group of values given by a unique combination of the levels of certain factors.

Usage: tapply(X, INDEX, FUN = NULL, ..., simplify = TRUE)

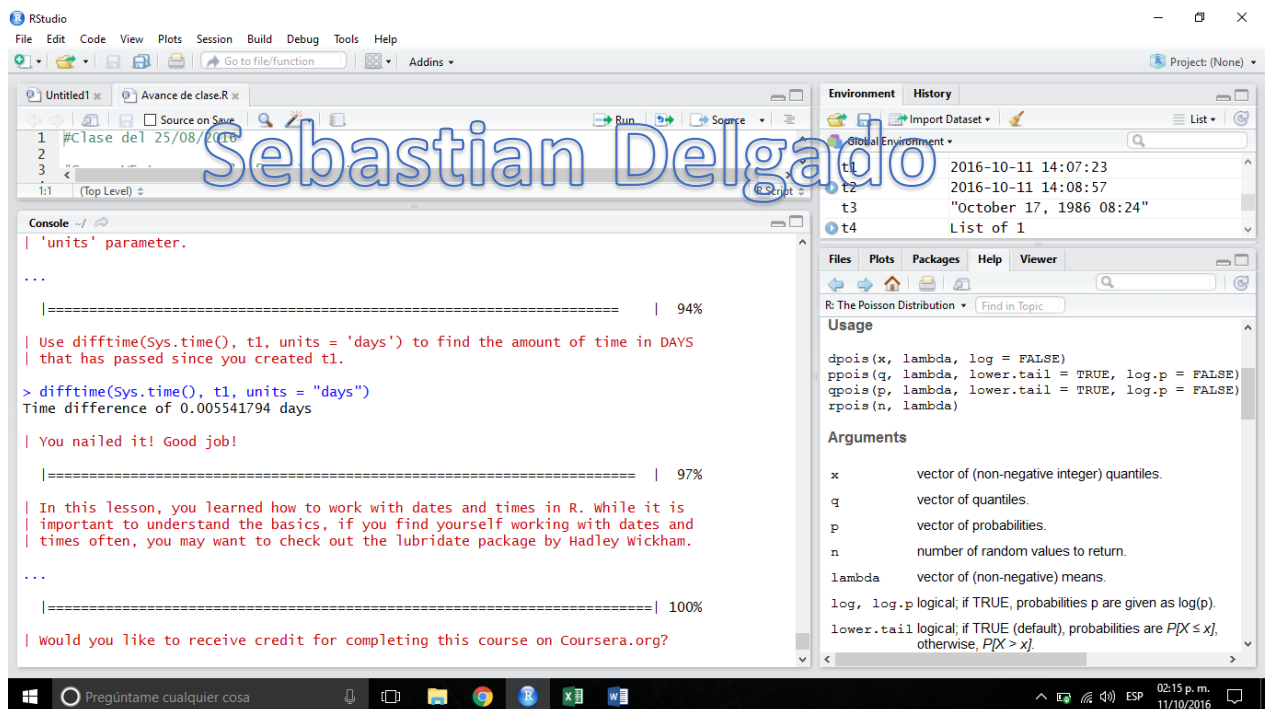
Arguments: tapply(X, INDEX, FUN = NULL, ..., simplify = TRUE)



## LECCIÓN 13



## LECCIÓN 14



## LECCIÓN 15

