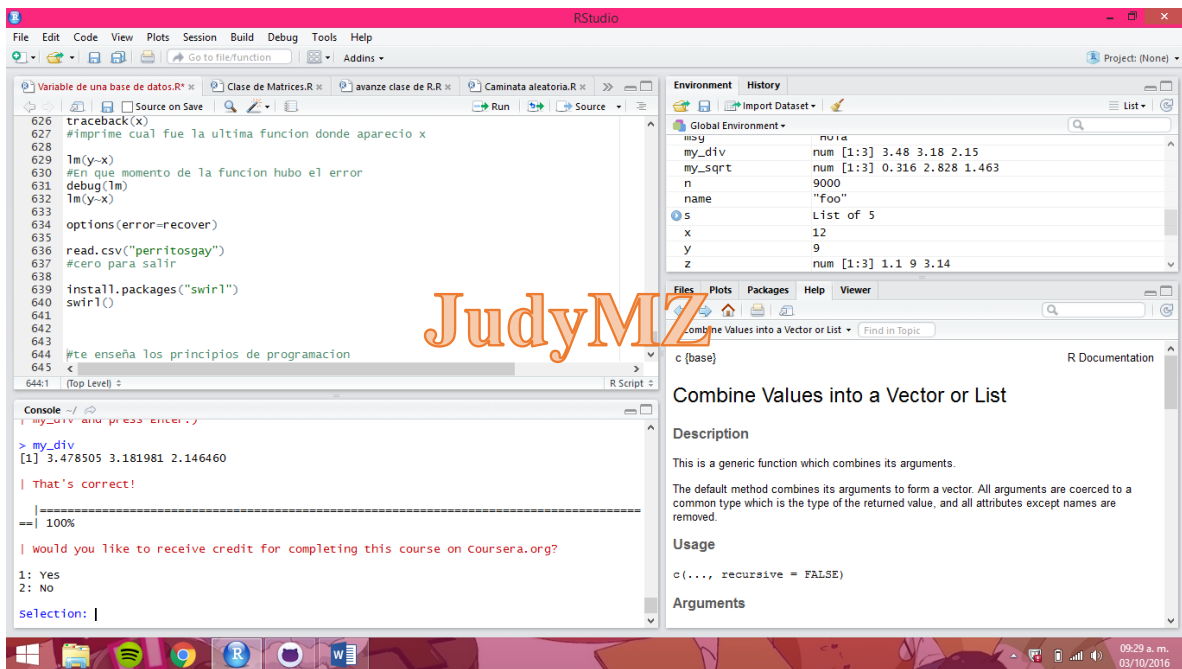


Lección 1:



Variable de una base de datos.R **Clase de Matrices.R** **avanza clase de R.R** **Caminata aleatoria.R** **Run** **Source** **Addins**

```
626 traceback(x)
627 #imprime cual fue la ultima funcion donde aparecio x
628
629 lm(y~x)
630 #En que momento de la funcion hubo el error
631 debug(lm)
632 lm(y~x)
633
634 options(error=recover)
635
636 read.csv("perritosgay")
637 #cero para salir
638
639 install.packages("swirl")
640 swirl()
641
642
643
644 #te enseña los principios de programación
645
```

Environment **History**

Global Environment

Variable	Value
my_div	num [1:3] 3.48 3.18 2.15
my_sqrt	num [1:3] 0.316 2.828 1.463
n	9000
name	"foo"
s	List of 5
x	12
y	9
z	num [1:3] 1.1 9 3.14

Combine Values into a Vector or List

Description

This is a generic function which combines its arguments.

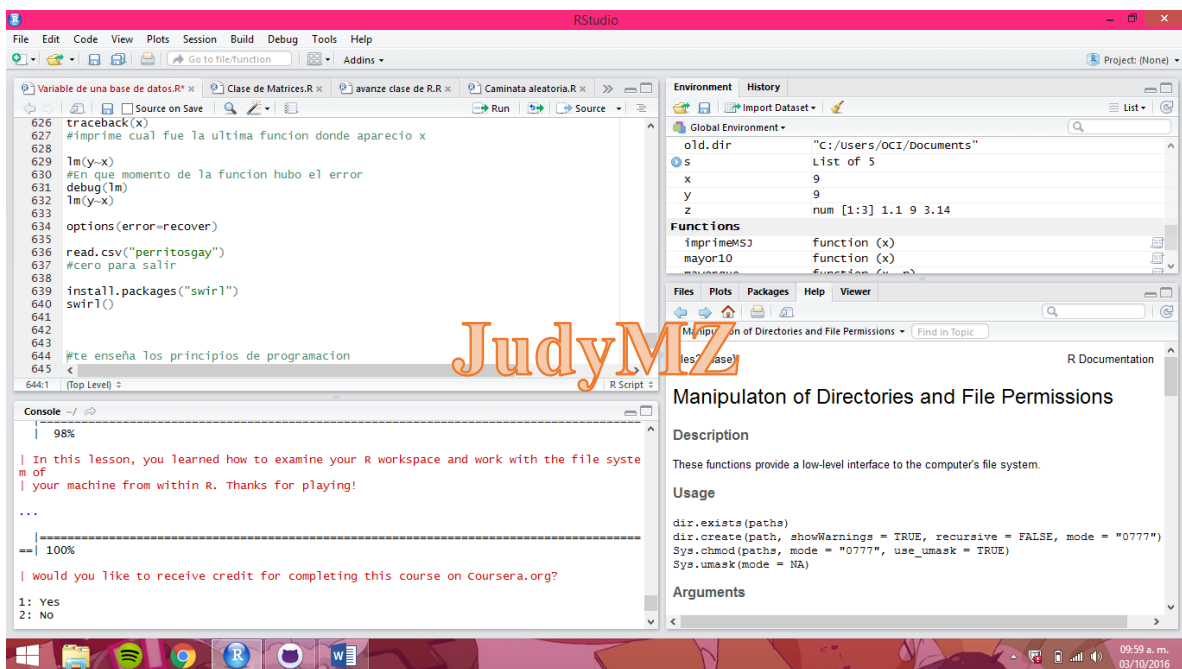
The default method combines its arguments to form a vector. All arguments are coerced to a common type which is the type of the returned value, and all attributes except names are removed.

Usage

```
c(..., recursive = FALSE)
```

Arguments

Lección 2:



Variable de una base de datos.R **Clase de Matrices.R** **avanza clase de R.R** **Caminata aleatoria.R** **Run** **Source** **Addins**

```
626 traceback(x)
627 #imprime cual fue la ultima funcion donde aparecio x
628
629 lm(y~x)
630 #En que momento de la funcion hubo el error
631 debug(lm)
632 lm(y~x)
633
634 options(error=recover)
635
636 read.csv("perritosgay")
637 #cero para salir
638
639 install.packages("swirl")
640 swirl()
641
642
643
644 #te enseña los principios de programación
645
```

Environment **History**

Global Environment

Variable	Value
old.dir	"C:/Users/OCI/Documents"
s	List of 5
x	9
y	9
z	num [1:3] 1.1 9 3.14

Functions

Function	Value
imprimeM3	function (x)
mayor10	function (x)
mayor10	function (x)

Manipulation of Directories and File Permissions

Description

These functions provide a low-level interface to the computer's file system.

Usage

```
dir.exists(paths)
dir.create(path, showWarnings = TRUE, recursive = FALSE, mode = "0777")
Sys.chmod(paths, mode = "0777", use_umask = TRUE)
Sys.umask(mode = NA)
```

Arguments

Lección 3:

[illegible]

Lección 4:

The screenshot shows the RStudio IDE with the following content:

Source Editor:

```

=====
| 92%
|
| 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: 1

```

Console:

```

=====
| 92%
|
| 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: 1

```

Environment Pane:

Variable	Class	Value
my_char	chr	[1:3] "my" "name" "is"
my_name	chr	[1:4] "my" "name" "is" " JUDY_MONTERO"
my_seq	num	[1:30] 5 5.17 5.34 5.52 5.69 ...
num_vect	num	[1:4] 0.5 55 -10 6
tf	logi	[1:4] TRUE FALSE TRUE FALSE

Global Environment:

Global Environment

Files, Plots, Packages, Help, Viewer:

R Console Strings

Find in Topic

Details:

paste converts its arguments (via [as.character\(\)](#) to character strings, and concatenates them (separating them by the string given by sep). If the arguments are vectors, they are concatenated term-by-term to give a character vector result. Vector arguments are recycled as needed, with zero-length arguments being recycled to "".

Note that paste() coerces [NA character](#), the character missing value, to "NA" which may seem undesirable, e.g. when pasting two character vectors, or very desirable, e.g. in paste("the value of p is ", p).

paste0(..., collapse) is equivalent to paste(..., sep = "", collapse), slightly more efficiently.

If a value is specified for collapse, the values in the result are then concatenated into a single string, with the elements being separated by the value of collapse.

Value

<

Lección 5:

Source

```
now.  
> 0/0  
[1] NaN  
  
| You are quite good my friend!  
|=====|  
| 95%  
|=====|  
  
| Let's do one more, just for fun. In R, Inf stands for infinity. what happens if you subtract Inf from Inf?  
  
> ?Inf  
> n  
Error: object 'n' not found  
> Inf  
[1] Inf  
  
| Not quite, but you're learning! Try again. or, type info() for more options.  
| Type Inf - Inf. Can you guess the result?  
  
> Inf - Inf  
[1] NaN  
  
| You are amazing!  
|=====|  
| 100%  
|=====|  
  
| would you like to receive credit for completing this course on Coursera.org?  
1: Yes  
2: No  
Selection:
```

Environment

Variable	Class	Value
my_char	chr	[1:3] "My" "name" "is"
my_data	num	[1:100] NA -0.331 0.252 NA NA ...
my_na	logi	[1:100] TRUE FALSE FALSE TRUE TRUE FALSE ...
my_name	chr	[1:4] "My" "name" "is" " JUDY_MONTERO"
my_seq	num	[1:30] 5 5.17 5.34 5.52 5.69 ...
num_vect	num	[1:4] 0.5 55 -10 6
tf	logi	[1:4] TRUE FALSE TRUE FALSE
x	num	[1:4] 44 NA 5 NA

Files **Plots** **Packages** **Help** **Viewer**

Finite, Infinite and NaN Numbers

Description

is.finite and is.infinite return a vector of the same length as x, indicating which elements are finite (not infinite and not missing) or infinite.

Inf and -Inf are positive and negative infinity whereas NaN means 'Not a Number'. (These apply to numeric values and real and imaginary parts of complex values but not to values of integer vectors.) Inf and NaN are [reserved](#) words in the R language.

Usage

```
is.finite(x)  
is.infinite(x)  
is.nan(x)
```

Lección 6:

Source

```
| Now, try it out.  
> vect["bar"]  
bar  
2  
  
| Great job!  
|=====|  
| 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 really on a roll!  
|=====|  
| 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: |
```

Environment

Variable	Class	Value
my_char	chr	[1:3] "My" "name" "is"
my_data	num	[1:100] NA -0.331 0.252 NA NA ...
my_na	logi	[1:100] TRUE FALSE FALSE TRUE TRUE FALSE ...
my_name	chr	[1:4] "My" "name" "is" " JUDY_MONTERO"
my_seq	num	[1:30] 5 5.17 5.34 5.52 5.69 ...
num_vect	num	[1:4] 0.5 55 -10 6
tf	logi	[1:4] TRUE FALSE TRUE FALSE
vect	Named num	[1:3] 11 2 NA
vect2	Named num	[1:3] 11 2 NA

Files **Plots** **Packages** **Help** **Viewer**

Finite, Infinite and NaN Numbers

Description

is.finite and is.infinite return a vector of the same length as x, indicating which elements are finite (not infinite and not missing) or infinite.

Inf and -Inf are positive and negative infinity whereas NaN means 'Not a Number'. (These apply to numeric values and real and imaginary parts of complex values but not to values of integer vectors.) Inf and NaN are [reserved](#) words in the R language.

Usage

```
is.finite(x)  
is.infinite(x)  
is.nan(x)
```