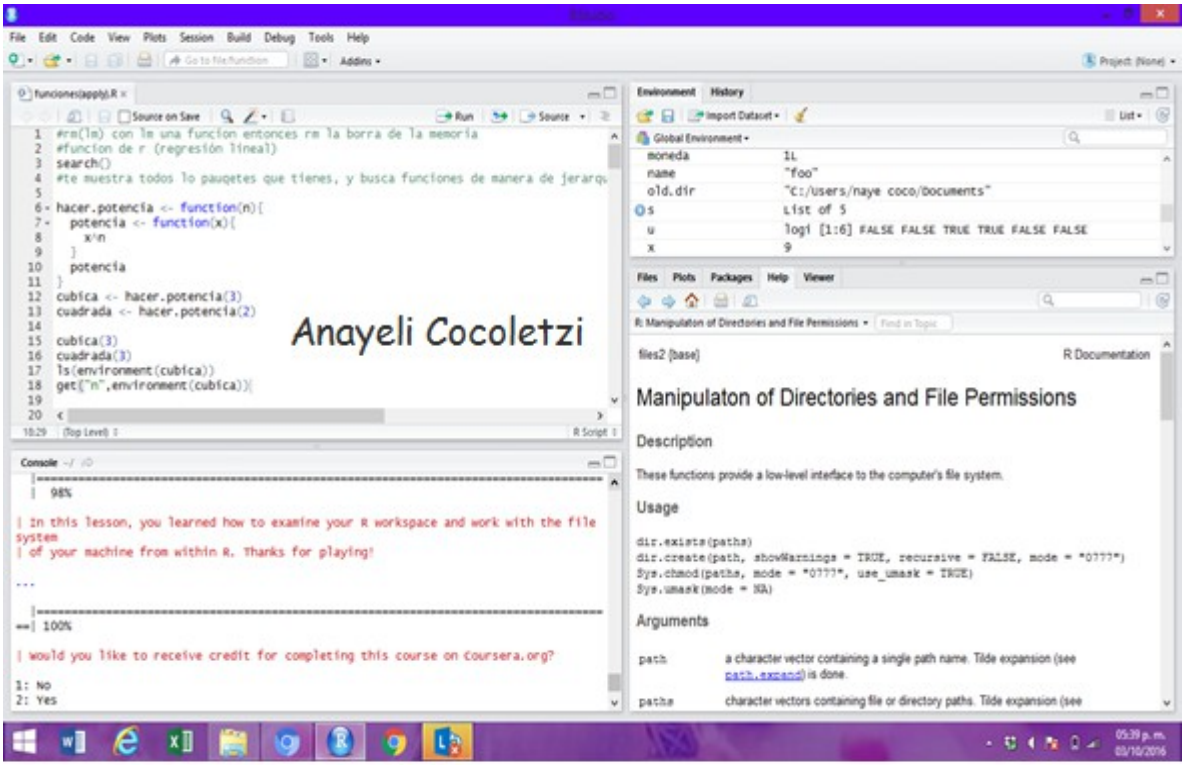
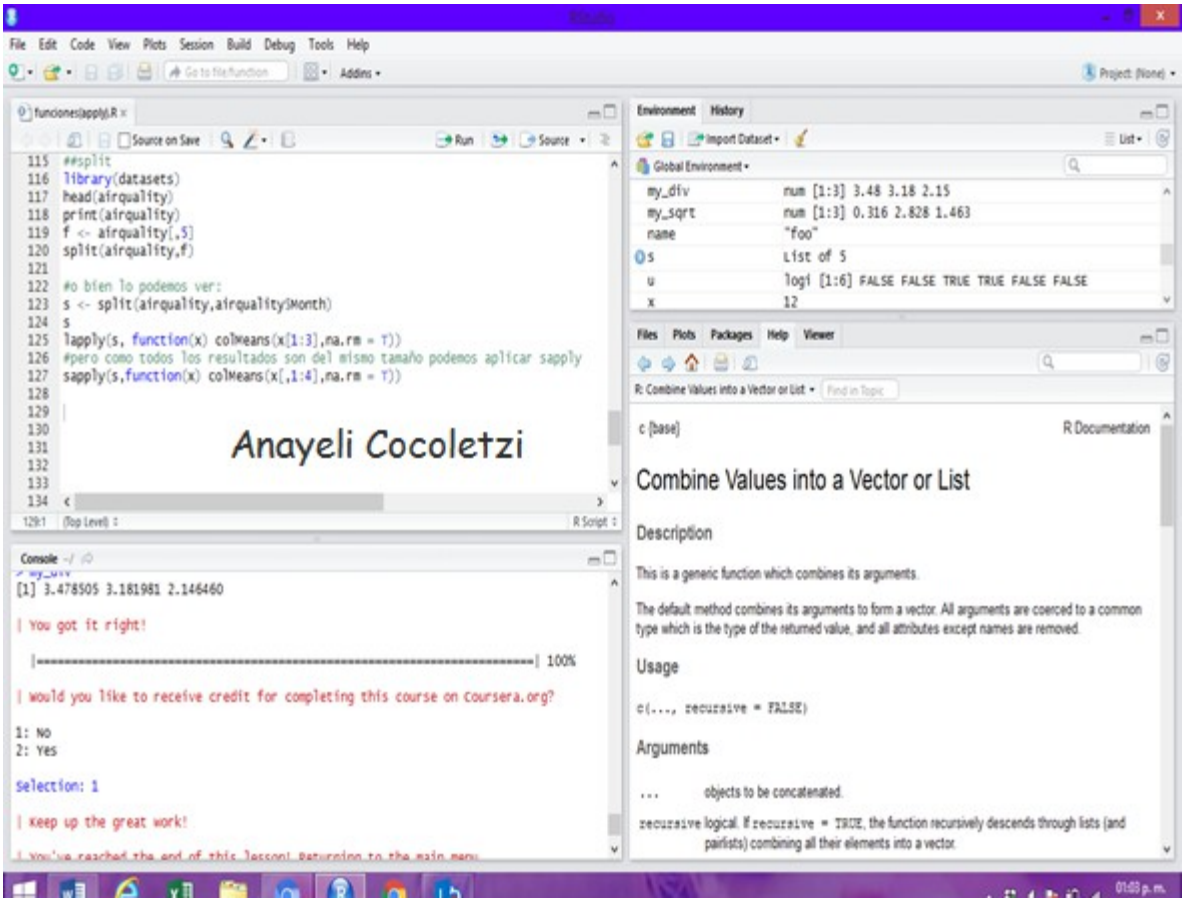
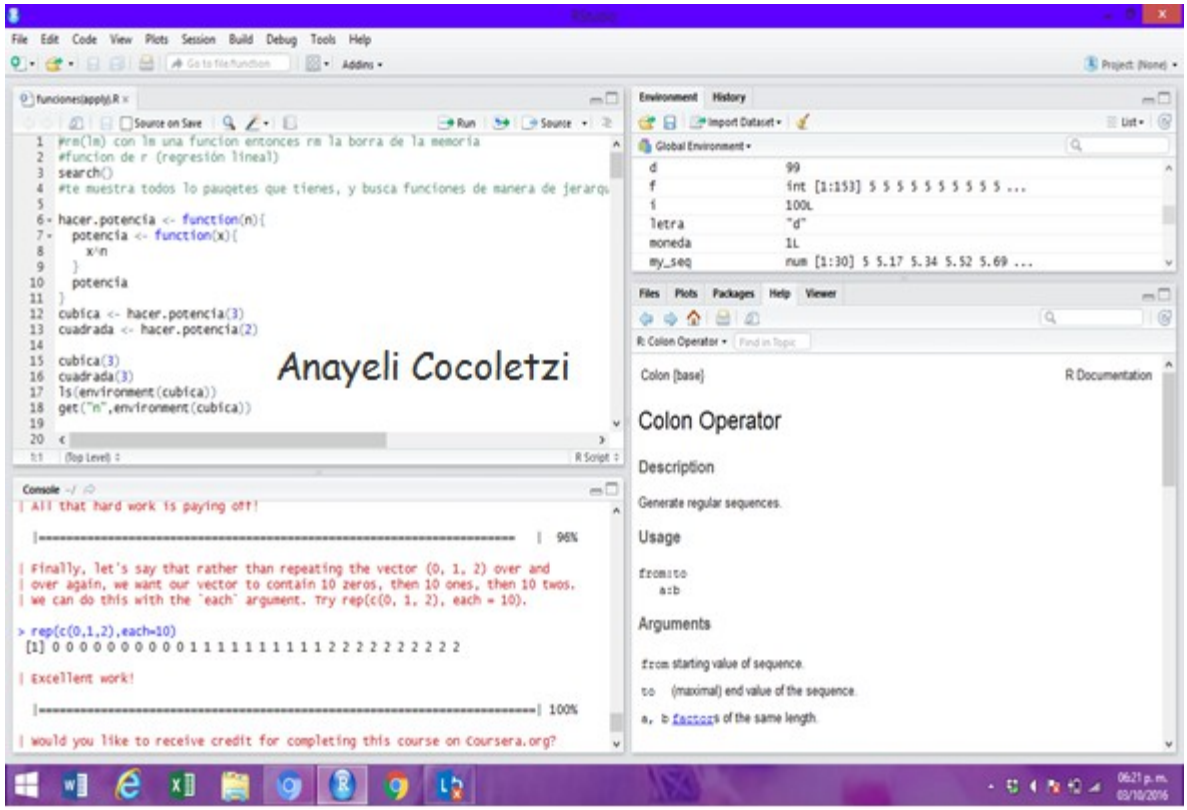
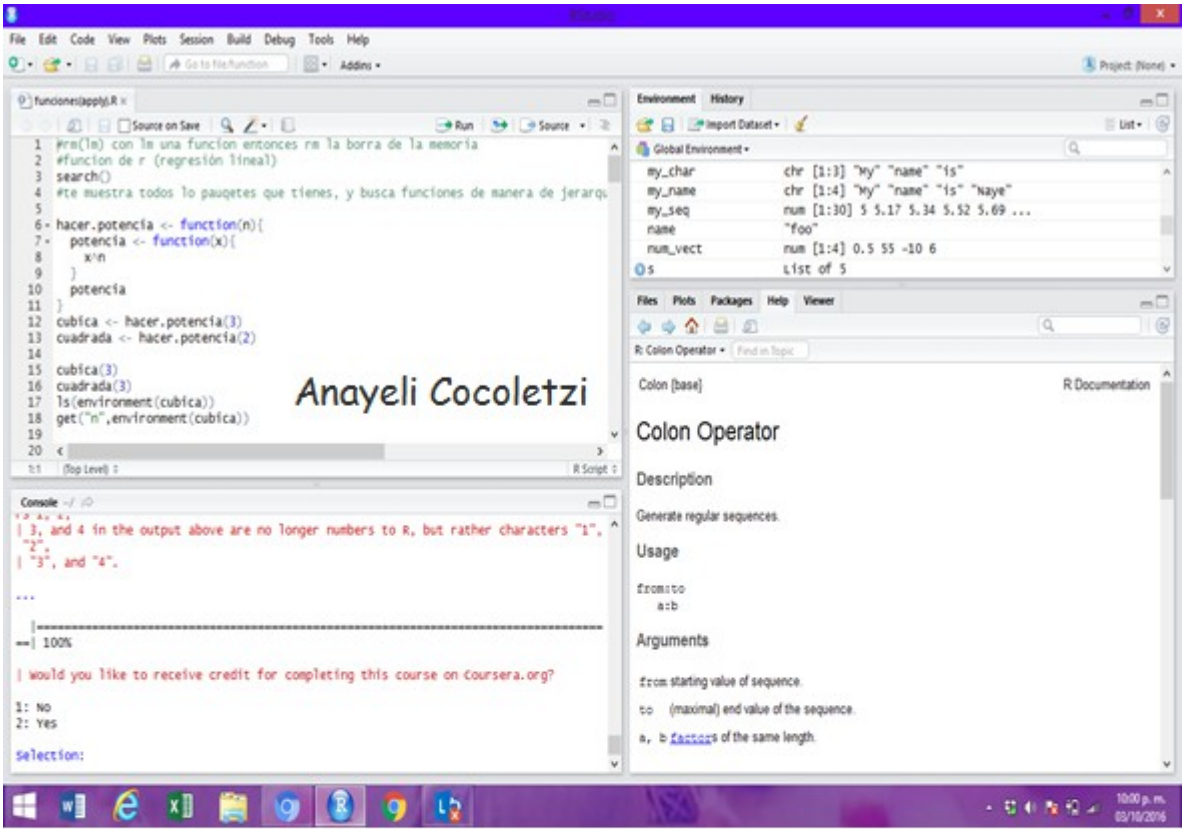
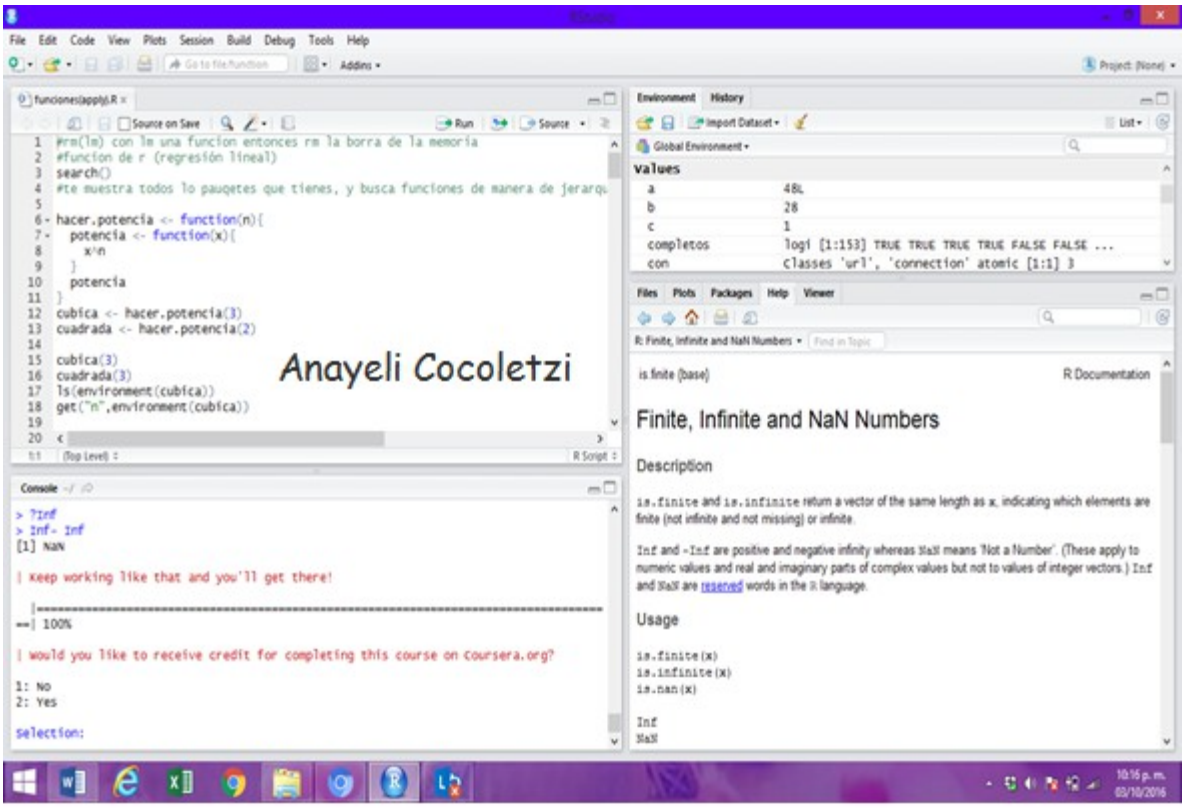
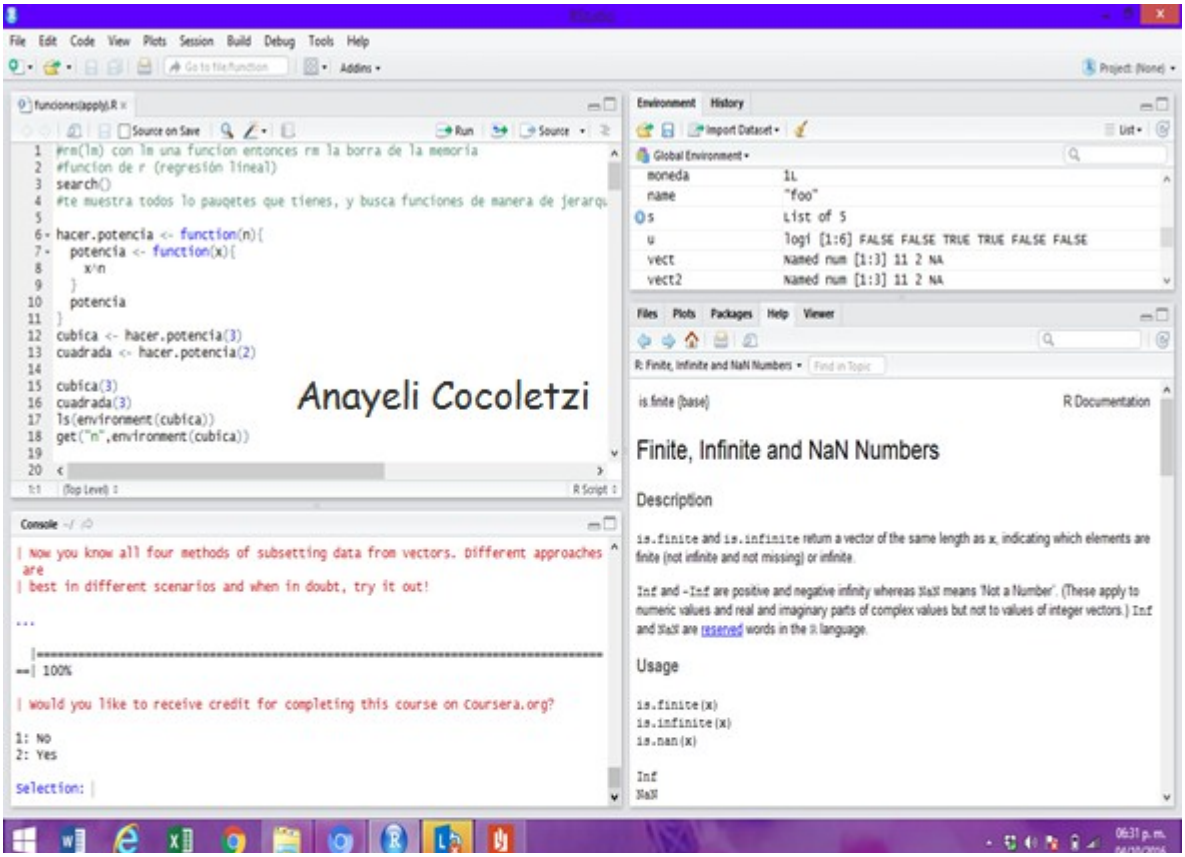

File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R
#rm(lm) con lm una funcion entonces rm la borra de la memoria
#funcion de r (regresión lineal)
search()
#te muestra todos lo paugetes que tienes, y busca funciones de manera de jerarq
hacer.potencia <- function(n){
potencia <- function(x){
x^n
}
potencia
}
cubica <- hacer.potencia(3)
cuadrada <- hacer.potencia(2)
cubica(3)
cuadrada(3)
ls(environment(cubica))
get("n",environment(cubica))
Anayeli Cocoletzi
Console
| 98%
| 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: No
2: Yes
Environment History
Global Environment
moneda 1L
name "foo"
old.dir "C:/Users/naye coco/Documents"
s List of 5
u logi [1:6] FALSE FALSE TRUE TRUE FALSE FALSE
x 9
Files Plots Packages Help Viewer
R: Manipulaton of Directories and File Permissions
files2 {base} R Documentation
Manipulaton 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
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
05:39 p. m.
03/10/2016

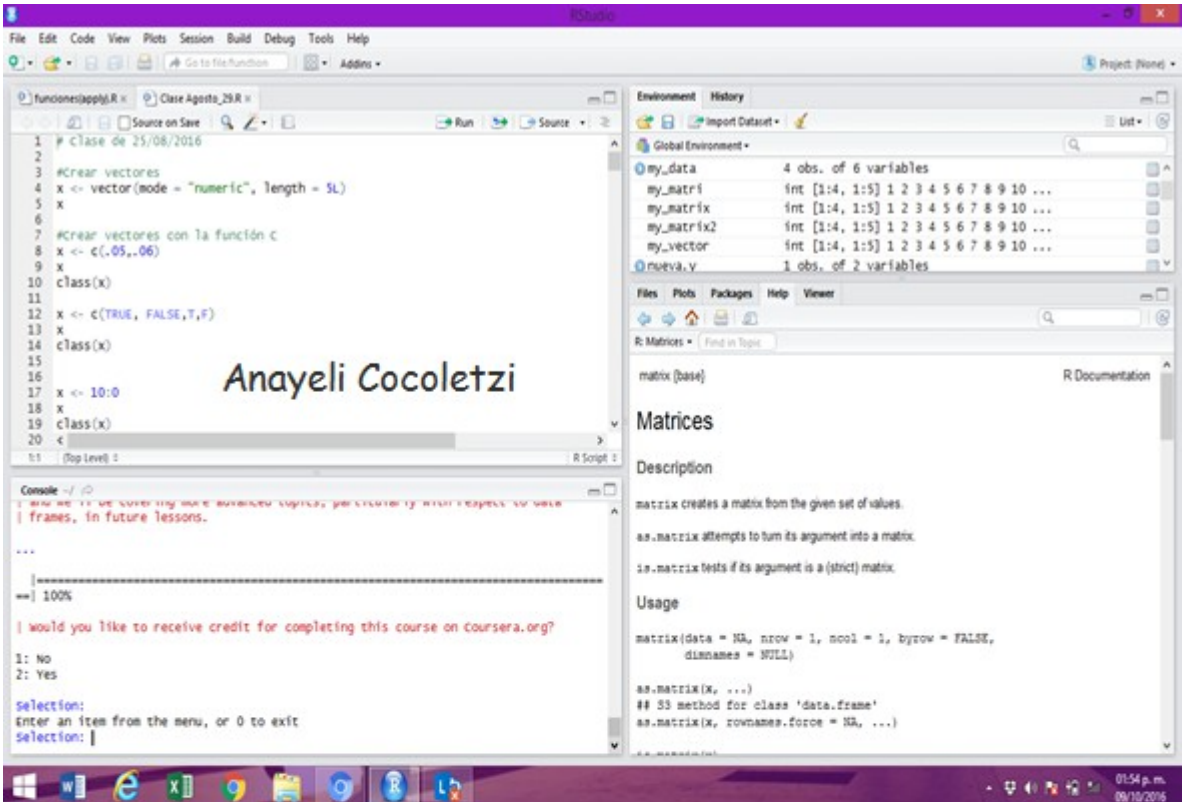

File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R
Source on Save
Run
Source
115 ##split
116 library(datasets)
117 head(airquality)
118 print(airquality)
119 f <- airquality[,5]
120 split(airquality,f)
121
122 #o bien lo podemos ver:
123 s <- split(airquality,airquality$Month)
124 s
125 lapply(s, function(x) colMeans(x[1:3],na.rm = T))
126 #pero como todos los resultados son del mismo tamaño podemos aplicar sapply
127 sapply(s,function(x) colMeans(x[,1:4],na.rm = T))
128
129
130
131
132
133
134
Anayeli Cocoletzi
129:1 (Top Level) R Script
Console
[1] 3.478505 3.181981 2.146460
| You got it right!
|==| 100%
| Would you like to receive credit for completing this course on Coursera.org?
1: NO
2: Yes
Selection: 1
| Keep up the great work!
Environment History
Import Dataset
Global Environment
my_div num [1:3] 3.48 3.18 2.15
my_sqrt num [1:3] 0.316 2.828 1.463
name "foo"
s List of 5
u logi [1:6] FALSE FALSE TRUE TRUE FALSE FALSE
x 12
Files Plots Packages Help Viewer
R: Combine Values into a Vector or List
c {base} R Documentation
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.

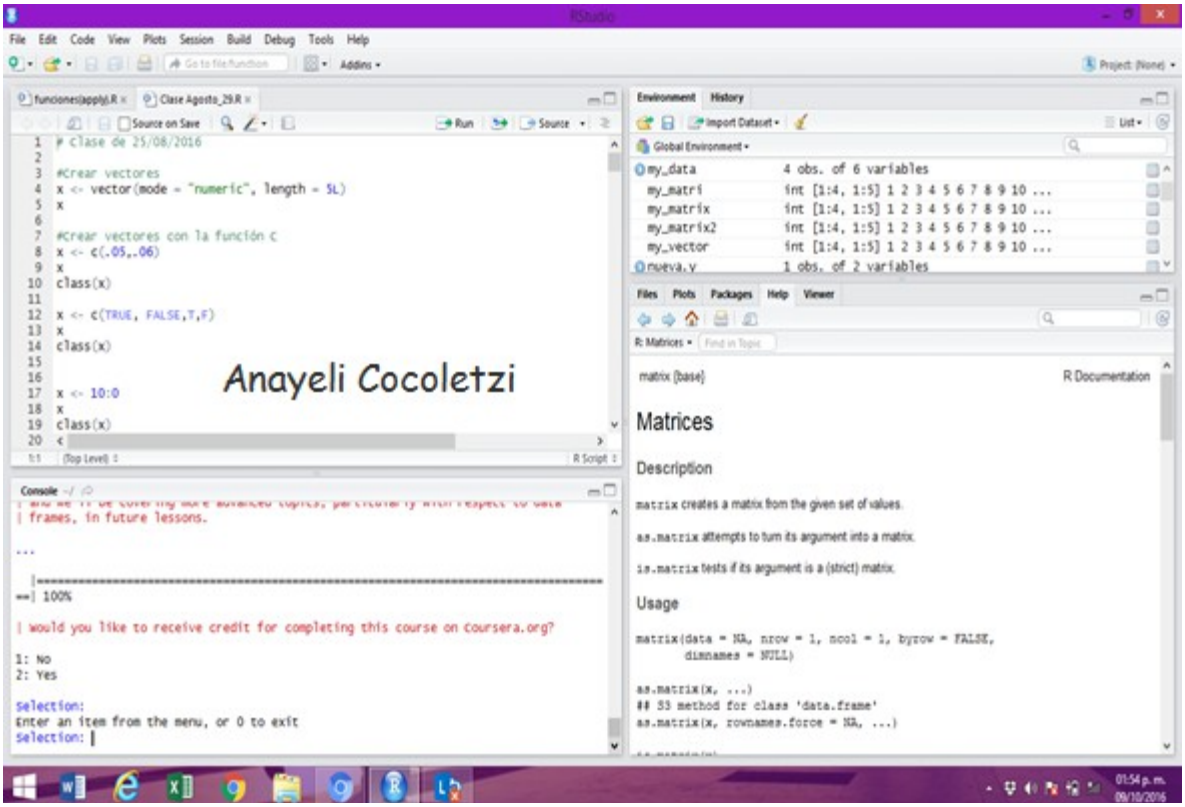

File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R
#rm(lm) con lm una funcion entonces rm la borra de la memoria
#funcion de r (regresión lineal)
search()
#te muestra todos lo paugetes que tienes, y busca funciones de manera de jerarqu
hacer.potencia <- function(n){
potencia <- function(x){
x^n
}
potencia
}
cubica <- hacer.potencia(3)
cuadrada <- hacer.potencia(2)
cubica(3)
cuadrada(3)
ls(environment(cubica))
get("n",environment(cubica))
Anayeli Cocoletzi
Console
| All that hard work is paying off!
| 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 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
| Excellent work!
| Would you like to receive credit for completing this course on Coursera.org?
Environment History
Import Dataset
Global Environment
d 99
f int [1:153] 5 5 5 5 5 5 5 5 5 5 ...
i 100L
letra "d"
moneda 1L
my_seq num [1:30] 5 5.17 5.34 5.52 5.69 ...
Files Plots Packages Help Viewer
R: Colon Operator
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.

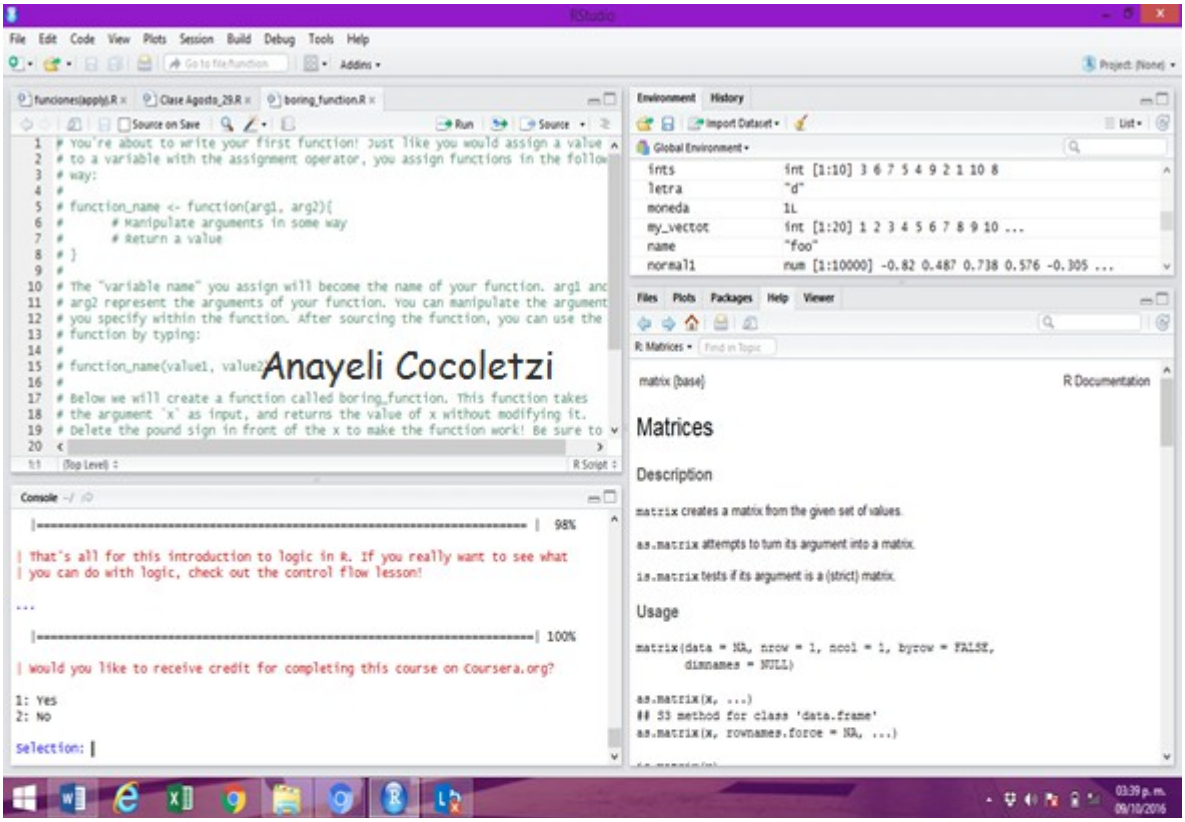

File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R
Source on Save
Run
Source
#rm(ls) con ls una funcion entonces rm la borra de la memoria
#funcion de r (regresión lineal)
search()
#te muestra todos lo paugetes que tienes, y busca funciones de manera de jerarqu
hacer.potencia <- function(n){
potencia <- function(x){
x^n
}
potencia
}
cubica <- hacer.potencia(3)
cuadrada <- hacer.potencia(2)
cubica(3)
cuadrada(3)
ls(environment(cubica))
get("n",environment(cubica))
Anayeli Cocoletzi
Environment History
Import Dataset
Global Environment
my_char chr [1:3] "My" "name" "is"
my_name chr [1:4] "My" "name" "is" "Naye"
my_seq num [1:30] 5 5.17 5.34 5.52 5.69 ...
name "foo"
num_vect num [1:4] 0.5 55 -10 6
s List of 5
Files Plots Packages Help Viewer
R: Colon Operator
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.
Console
| 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:
10:00 p. m.
03/10/2016

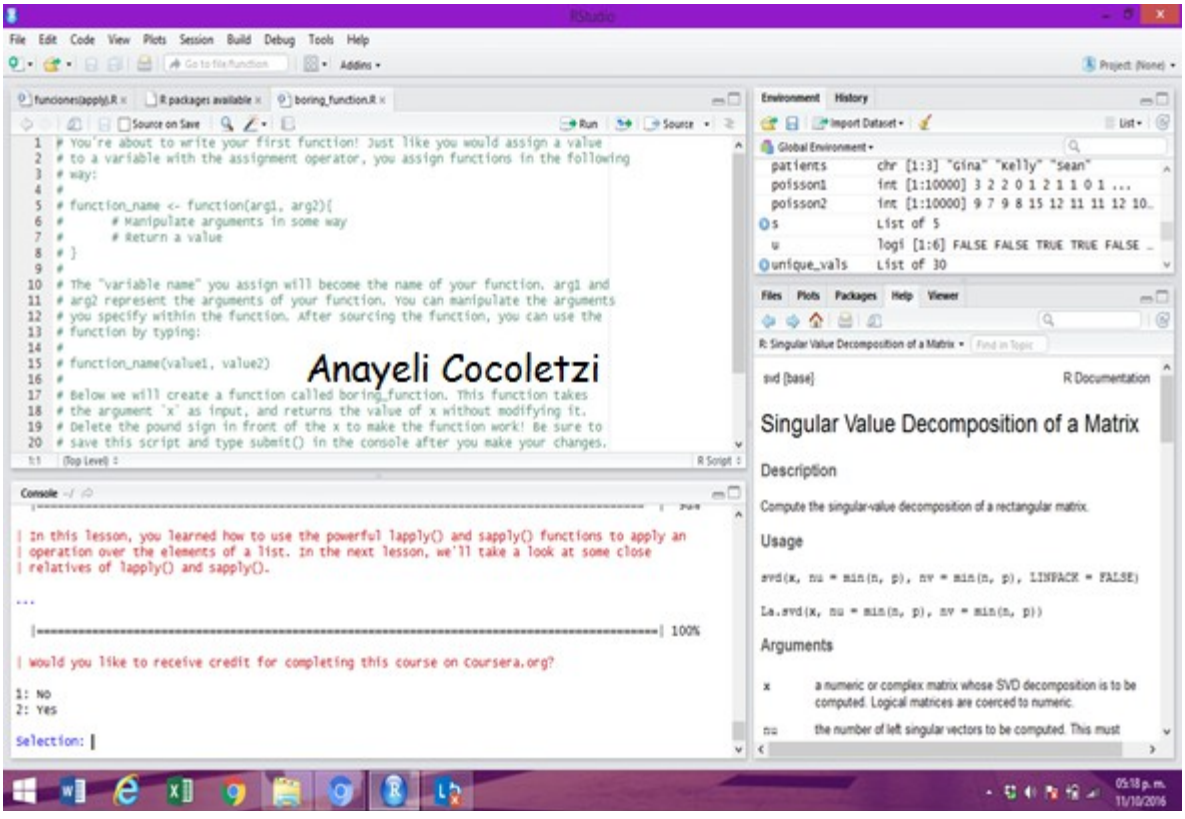

File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R
Source on Save
Run
Source
#rm(ls) con ls una funcion entonces rm la borra de la memoria
#funcion de r (regresión lineal)
search()
#te muestra todos lo paugetes que tienes, y busca funciones de manera de jerarqu
hacer.potencia <- function(n){
potencia <- function(x){
x^n
}
potencia
}
cubica <- hacer.potencia(3)
cuadrada <- hacer.potencia(2)
cubica(3)
cuadrada(3)
ls(environment(cubica))
get("n",environment(cubica))
Anayeli Cocoletzi
Environment History
Import Dataset
Global Environment
values
a 48L
b 28
c 1
completos logi [1:153] TRUE TRUE TRUE TRUE FALSE FALSE ...
con Classes 'url', 'connection' atomic [1:1] 3
Files Plots Packages Help Viewer
R: Finite, Infinite and NaN Numbers
is.finite {base} R Documentation
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)
Inf
NaN
Console
> ?Inf
> Inf- Inf
[1] NaN
| Keep working like that and you'll get there!
==| 100%
| Would you like to receive credit for completing this course on Coursera.org?
1: No
2: Yes
Selection:
10:16 p. m.
03/10/2016

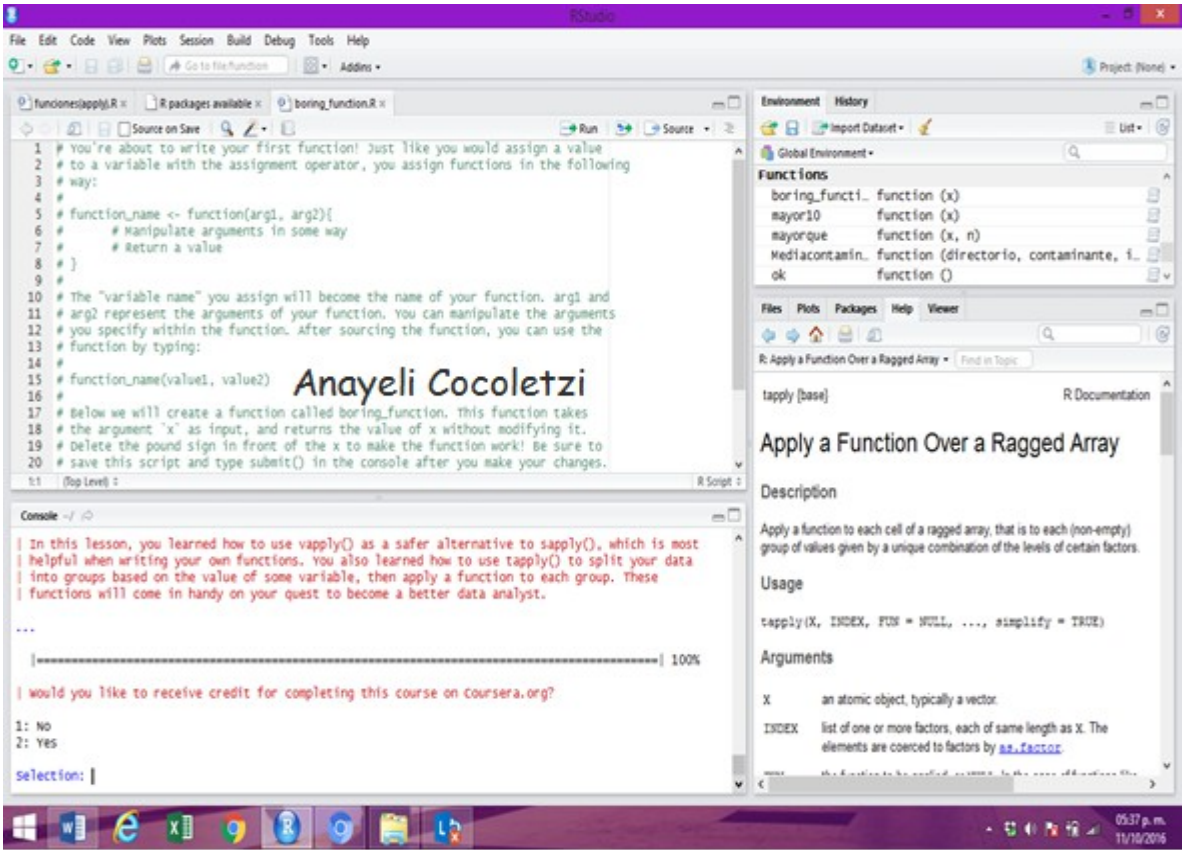

File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R
Source on Save
Run
Source
#rm(lm) con lm una funcion entonces rm la borra de la memoria
#funcion de r (regresión lineal)
search()
#te muestra todos lo paugetes que tienes, y busca funciones de manera de jerarqu
hacer.potencia <- function(n){
potencia <- function(x){
x^n
}
potencia
}
cubica <- hacer.potencia(3)
cuadrada <- hacer.potencia(2)
cubica(3)
cuadrada(3)
ls(environment(cubica))
get("n",environment(cubica))
Anayeli Cocoletzi
(Top Level)
R Script
Console
| 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 History
Import Dataset
Global Environment
moneda 1L
name "foo"
s List of 5
u logi [1:6] FALSE FALSE TRUE TRUE FALSE FALSE
vect Named num [1:3] 11 2 NA
vect2 Named num [1:3] 11 2 NA
Files Plots Packages Help Viewer
R: Finite, Infinite and NaN Numbers
is.finite (base)
R Documentation
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)
Inf
NaN
06:31 p. m.

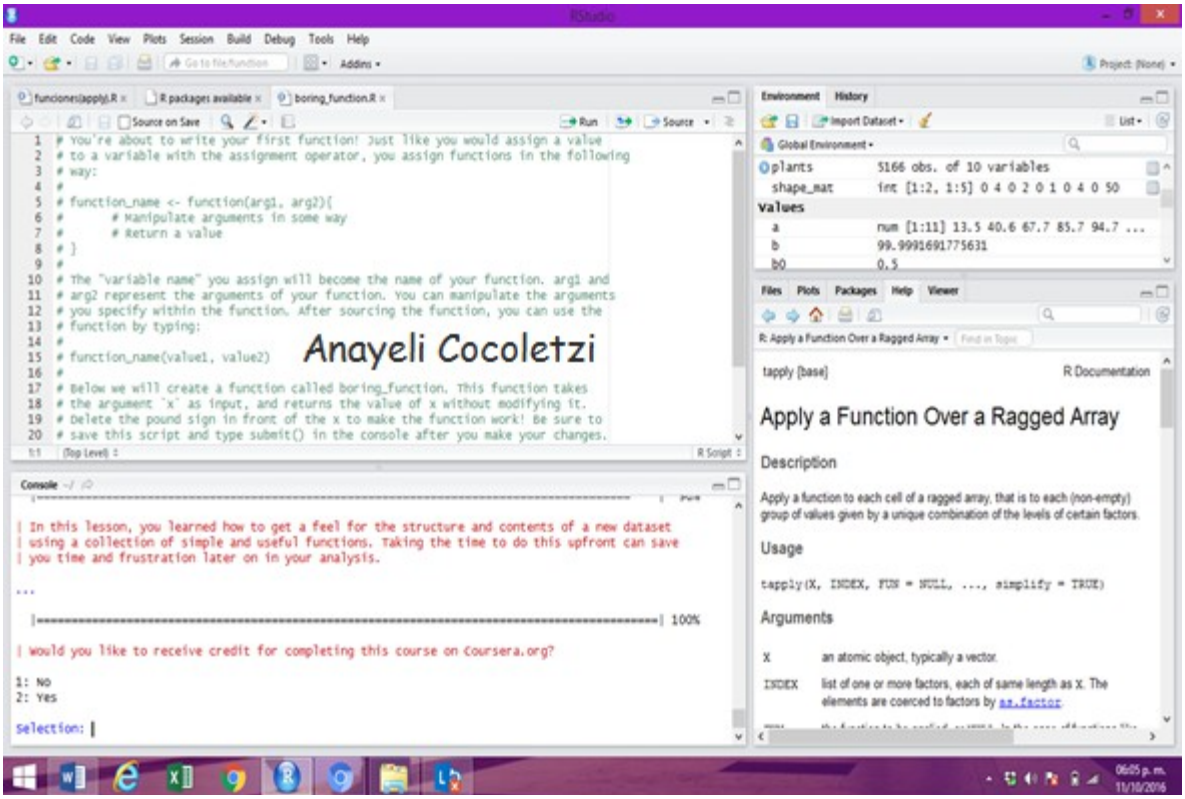

File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R
Clase Agosto_29.R
Source on Save
Run
Source
Clase de 25/08/2016
#Crear vectores
x <- vector(mode = "numeric", length = 5L)
x
#Crear vectores con la función c
x <- c(.05,.06)
x
class(x)
x <- c(TRUE, FALSE,T,F)
x
class(x)
Anayeli Cocoletzi
x <- 10:0
x
class(x)
Environment
History
Import Dataset
Global Environment
my_data 4 obs. of 6 variables
my_matri int [1:4, 1:5] 1 2 3 4 5 6 7 8 9 10 ...
my_matrix int [1:4, 1:5] 1 2 3 4 5 6 7 8 9 10 ...
my_matrix2 int [1:4, 1:5] 1 2 3 4 5 6 7 8 9 10 ...
my_vector int [1:4, 1:5] 1 2 3 4 5 6 7 8 9 10 ...
nueva.v 1 obs. of 2 variables
Files
Plots
Packages
Help
Viewer
R: Matrices
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
frames, in future lessons.
==| 100%
Would you like to receive credit for completing this course on coursera.org?
1: No
2: Yes
selection:
Enter an item from the menu, or 0 to exit
selection:
01:54 p. m.

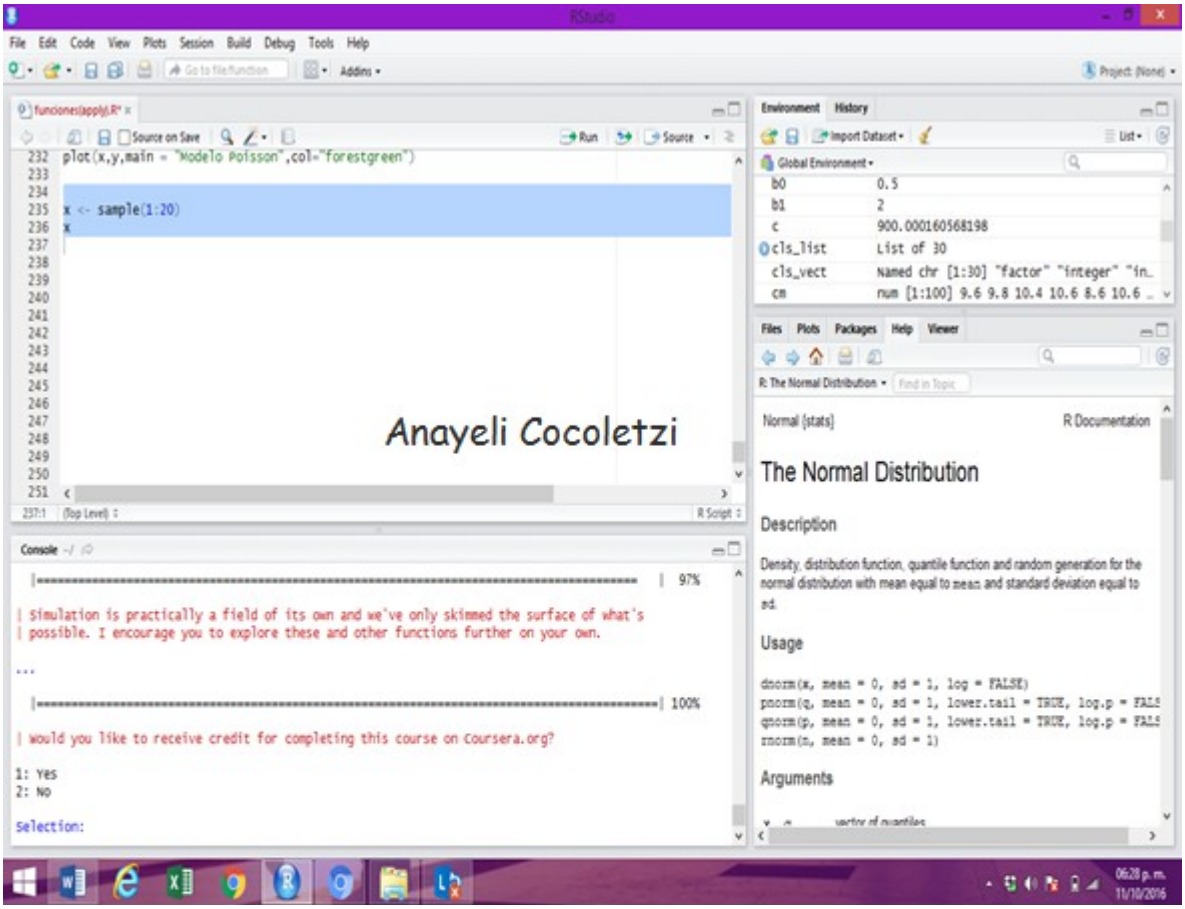

File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R
Clase Agosto_29.R
Source on Save
Run
Source
Clase de 25/08/2016
#Crear vectores
x <- vector(mode = "numeric", length = 5L)
x
#Crear vectores con la función c
x <- c(.05,.06)
x
class(x)
x <- c(TRUE, FALSE,T,F)
x
class(x)
Anayeli Cocoletzi
x <- 10:0
x
class(x)
Environment
History
Import Dataset
Global Environment
my_data 4 obs. of 6 variables
my_matri int [1:4, 1:5] 1 2 3 4 5 6 7 8 9 10 ...
my_matrix int [1:4, 1:5] 1 2 3 4 5 6 7 8 9 10 ...
my_matrix2 int [1:4, 1:5] 1 2 3 4 5 6 7 8 9 10 ...
my_vector int [1:4, 1:5] 1 2 3 4 5 6 7 8 9 10 ...
nueva.v 1 obs. of 2 variables
Files
Plots
Packages
Help
Viewer
R: Matrices
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
frames, in future lessons.
==| 100%
Would you like to receive credit for completing this course on coursera.org?
1: No
2: Yes
selection:
Enter an item from the menu, or 0 to exit
selection:
01:54 p. m.

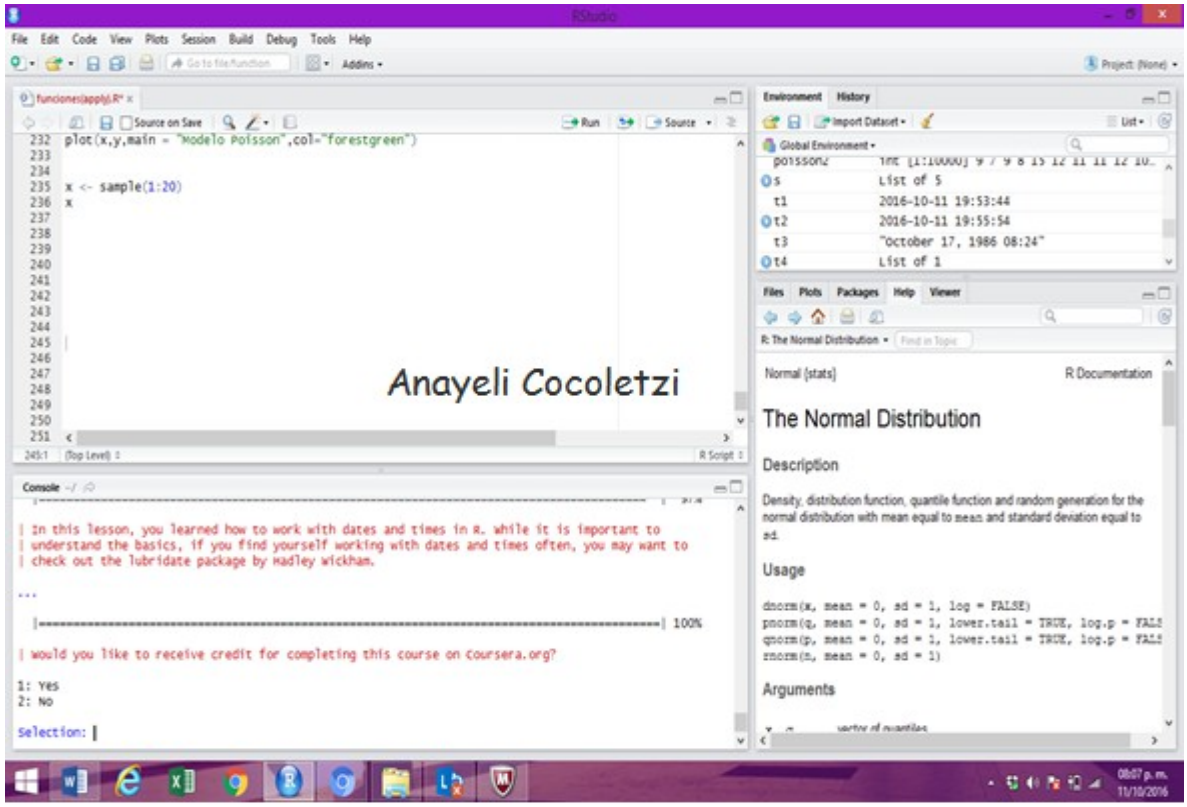

Anayeli Cocoletzi

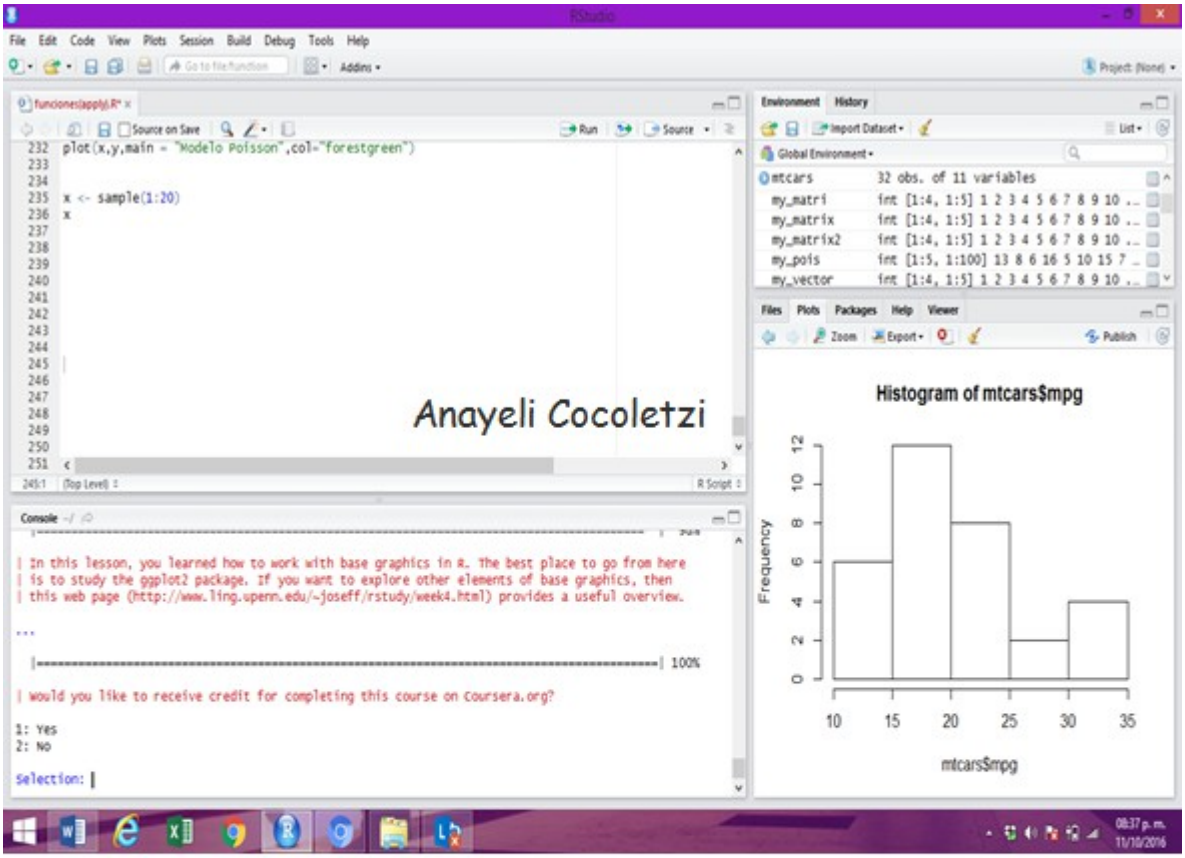

Anayeli Cocoletzi


RStudio
File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R
R packages available
boring_function.R
Source on Save
Run
Source
You're about to write your first function! Just like you would assign a value
to a variable with the assignment operator, you assign functions in the following
way:
#
function_name <- function(arg1, arg2){
Manipulate arguments in some way
Return a value
}
#
The "variable name" you assign will become the name of your function. arg1 and
arg2 represent the arguments of your function. You can manipulate the arguments
you specify within the function. After sourcing the function, you can use the
function by typing:
#
function_name(value1, value2)
#
Below we will create a function called boring_function. This function takes
the argument `x` as input, and returns the value of x without modifying it.
Delete the pound sign in front of the x to make the function work! Be sure to
save this script and type submit() in the console after you make your changes.
Anayeli Cocoletzi
(Top Level)
R Script
Console
| 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: NO
2: Yes
Selection:
Environment History
Import Dataset
Global Environment
Functions
boring_functi.. function (x)
mayor10 function (x)
mayorque function (x, n)
Mediacontamin.. function (directorio, contaminante, i..
ok function ()
Files Plots Packages Help Viewer
R: Apply a Function Over a Ragged Array
tapply {base} R Documentation
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
X an atomic object, typically a vector.
INDEX list of one or more factors, each of same length as X. The elements are coerced to factors by as.factor.
05:37 p. m.
11/10/2016


RStudio
File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R
R packages available
boring_function.R
Source on Save
Run
Source
You're about to write your first function! Just like you would assign a value
to a variable with the assignment operator, you assign functions in the following
way:
#
function_name <- function(arg1, arg2){
Manipulate arguments in some way
Return a value
}
#
The "variable name" you assign will become the name of your function. arg1 and
arg2 represent the arguments of your function. You can manipulate the arguments
you specify within the function. After sourcing the function, you can use the
function by typing:
#
function_name(value1, value2)
#
Below we will create a function called boring_function. This function takes
the argument `x` as input, and returns the value of x without modifying it.
Delete the pound sign in front of the x to make the function work! Be sure to
save this script and type submit() in the console after you make your changes.
Anayeli Cocoletzi
(Top Level)
R Script
Console
| 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?
1: NO
2: Yes
Selection:
Environment History
Import Dataset
Global Environment
plants 5166 obs. of 10 variables
shape_mat int [1:2, 1:5] 0 4 0 2 0 1 0 4 0 50
Values
a num [1:11] 13.5 40.6 67.7 85.7 94.7 ...
b 99.9991691775631
b0 0.5
Files Plots Packages Help Viewer
R: Apply a Function Over a Ragged Array
tapply {base} R Documentation
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
X an atomic object, typically a vector.
INDEX list of one or more factors, each of same length as X. The elements are coerced to factors by as.factor.
06:05 p. m.
11/10/2016


RStudio
File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R*
Source on Save
Run
Source
plot(x,y,main = "Modelo Poisson",col="forestgreen")
x <- sample(1:20)
x
Anayeli Cocoletzi
Environment History
Import Dataset
Global Environment
b0 0.5
b1 2
c 900.000160568198
cls_list List of 30
cls_vect Named chr [1:30] "factor" "integer" "in..
cm num [1:100] 9.6 9.8 10.4 10.6 8.6 10.6 ..
Files Plots Packages Help Viewer
R: The Normal Distribution
Normal {stats} R Documentation
The Normal Distribution
Description
Density, distribution function, quantile function and random generation for the normal distribution with mean equal to mean and standard deviation equal to sd.
Usage
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALS
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALS
rnorm(n, mean = 0, sd = 1)
Arguments
Console
| 97%
| simulation is practically a field of its own and we've only skimmed the surface of what's
| possible. I encourage you to explore these and other functions further on your own.
...
| 100%
| Would you like to receive credit for completing this course on Coursera.org?
1: Yes
2: No
Selection:
06:28 p. m.
11/10/2016


File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R*
Source on Save
Run
Source
plot(x,y,main = "Modelo Poisson",col="forestgreen")
x <- sample(1:20)
x
Anayeli Cocoletzi
Console
| In this lesson, you learned how to work with dates and times in R. While it is important to
| understand the basics, if you find yourself working with dates and times often, you may want to
| check out the lubridate package by Hadley Wickham.
| would you like to receive credit for completing this course on coursera.org?
1: Yes
2: No
Selection:
Environment
History
Import Dataset
Global Environment
s
List of 5
t1
2016-10-11 19:53:44
t2
2016-10-11 19:55:54
t3
"October 17, 1986 08:24"
t4
List of 1
Files
Plots
Packages
Help
Viewer
R: The Normal Distribution
Normal {stats}
R Documentation
The Normal Distribution
Description
Density, distribution function, quantile function and random generation for the normal distribution with mean equal to mean and standard deviation equal to sd.
Usage
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALS
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALS
rnorm(n, mean = 0, sd = 1)
Arguments


RStudio
File Edit Code View Plots Session Build Debug Tools Help
funciones(apply).R*
Source on Save
Run
Source
232 plot(x,y,main = "Modelo Poisson",col="forestgreen")
235 x <- sample(1:20)
236 x
Anayeli Cocoletzi
Environment History
Import Dataset
Global Environment
mtcars 32 obs. of 11 variables
my_matri int [1:4, 1:5] 1 2 3 4 5 6 7 8 9 10 ...
my_matrix int [1:4, 1:5] 1 2 3 4 5 6 7 8 9 10 ...
my_matrix2 int [1:4, 1:5] 1 2 3 4 5 6 7 8 9 10 ...
my_pois int [1:5, 1:100] 13 8 6 16 5 10 15 7 ...
my_vector int [1:4, 1:5] 1 2 3 4 5 6 7 8 9 10 ...
Files Plots Packages Help Viewer
Zoom
Export
Publish
Histogram of mtcars$mpg
Frequency
0 2 4 6 8 10 12
10 15 20 25 30 35
mtcars$mpg
Console
| In this lesson, you learned how to work with base graphics in R. The best place to go from here
| is to study the ggplot2 package. If you want to explore other elements of base graphics, then
| this web page (http://www.ling.upenn.edu/~joseff/rstudy/week4.html) provides a useful overview.
...
| 100%
| Would you like to receive credit for completing this course on Coursera.org?
1: Yes
2: No
Selection:
08:37 p. m.
11/10/2016