PACKAGES, FILES and HELP

install.packages('dplyr')	Download and install a package.
library(dplyr)	Load the package into the session.
dplyr::select	Use a function from a package.
getwd()	Find the current working directory.
setwd('C://file/path')	Change the current work. directory.
dir()	Returns data in the named directory.
?mean	Get help of a particular function.F1.
help.search('mean')	Search the help files for a phrase.
help(package = 'dplyr')	Find help for a package.

DATA

read.table() or .csv()	Download and install a package.
url() with read.*()	Search the help files for a phrase.
write.csv() or .csv()	Find help for a package.
readlines()	Read text lines from a connection.
writelines()	Write text lines from a connection.
save() and load() (extension .rda or .Rdata	Save writes an external representa-tion of objects to the specified file.

R AS A CALCULATOR

+ Addition	**^or** Exponential**
- Substraction	**x%%y Remainder**
*** Multiplication**	**x%/% Integer**
/ Divisionf	

LOGICAL OPERATORS

a > b	Is a greater than b?
a >= b	Is a greater than or equal to b?
a < b	Is a less than b?
a <= b	Is a less than or equal to b?
a == b	Is a equal to b?
a != b	Is a not equal to b?
a %in% c(a, b, c)	Is a in the group c(a, b, c)?
x | y	x OR y
x & y	x AND y
isTRUE(x)	test if X is TRUE

VARIABLES ASSIGNMENT

A is diff. to a	R is case sensitive
x <- 1	<- Assignation operator

DATA TYPES

TRUE, T, FALSE, F	Logical
1, 1.11, 111	Double
1L	Integer
1 + 1i	Complex
"1"	Character
class(x)	Find the class an object belongs to.
str(x)	Get a summary of an ob. structure.
as.logical(x)	Convert from higher level to lower.

(ATOMIC) VECTORS

v1 <- 1:3	Create a vector from sequence.	*#[1] 1 2 3*
v2 <- c(3, 4, 5)	Create vector using c() function.	*#[1] 3 4 5*
seq(from , to, by)	Generate a sequence.	*seq(1,10,2) #[1] 1 3 5 7 9*
rep(1:3, ntimes)	Repeat x n times.	*rep(1:3, 2) #[1] 1 2 3 1 2 3*
lenght(v)	Get or set the length.	*length(v1) #[1] 3*
v + 1	Vectorize operations.	*v1 * 2 #[1]1 4 6*
v[1] or v[2:3]	Getting element by index. (No 0 index.)	*v1[1] #[1] 1*
v[v2]	Getting with another vector.	*v1[T,T,F] #[1] 1 2*
names(v1) <- v3	a character vector giving each element a name.	*names(v1) <- c("a", "b", "c") or v1 <- c(a = 1, b = 2, c = 3).*
unname(v1)	Remove the names or dimnames attribute of an R object.	*v1 <- unname(v1)*
attr(v1,"name attr") <- "value attr"	All objects can have arbitrary ad-ditional attributes, used to store metadata about the object.	*attr(v1, "my_attribute") <- "This is a vector"*

SET OPERATIONS

intersect (v1, v2)	Return obs. in both v1 and v2.	*#[1] 3*
union (v1, v2)	Return unique obs. in v1 and v2.	*#[1] 1 2 3 4 5*
setdiff (v1, v2)	Return obs. in v1, but not in v2.	*#[1] 1 2*

VECTOR FUNCTIONS

sort (v1)	Sort the elements of a vector.	*sort(c(5,9,3)) #[1] 3,5,9*
table (v1)	Count the elements of a vector.	*c(5,9,3,3)) # 3 5 9 2 1 1*

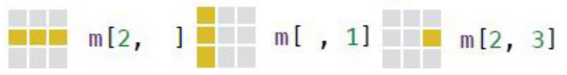
SPECIAL NUMBERS

+ OR - Inf	positive and negative infinity.	*1 / 0 #[1] Inf*
NaN	'Not a Number', undefined.	*1 / 1 #[1] NaN*
NA	'Not Available', missing value.	*c(1, NA, 2) #[1] 1 NA 2*
is.na(x) OR is.nan(x)	Check values which are na or NaN	*x <- c(1, 2, NA, 4, NA, 5) x [!is.na(x)]*

MATRICES

m1 <- matrix (1:6, nrow = 2, ncol = 3)	Create a matrix.
dim(m1)	Retrieve or set the dimension of an object.
t(m1)	Traspose.
m %*% n	Matrix multiplication. *Columns of m must be rows of n or other way round.*
rbind() or cbind()	Combine vectors by rows or columns to form a matrix.

*matrix will fill up the matrix column by column by default, but you can fill the matrix row by row if you include the argument byrow = TRUE.



LISTS

l1 <- list (1,"a",TRUE)	Create a list.	*#[1] 1 "a" TRUE*
l1 [[2]]	Getting a elem.	*l1[2] #[1]"a"*
l1[2]	Getting a elem. as a list.	*l1[2] #[[1]] [1]"a"*
names(l1) <- v3	Assign a name to a element of the list.	*names(l1) <- c("a", "b", "c") or l1 <- list(a = 1, b = "a", c = T).*
unlist()	Convert list to vector.	*v1 <- unlist(l1) #[1]"1" "a"...*
$ "name"	Call element by name.	*$a #[1] 1*

* We can create a list of vector.

FACTORS

factor (v1)	Turn a vector into a factor. Can set the levels of the factor and the order.	*v1 <- c(1,1,2,3) fv1 <- factor(v1) #[1] 1 1 2 3 Levels: 1 2 3*

DATA FRAMES

data.frame(v1, v2)	Create a data frame with vectors as columns.	*df1<- data.frame (x = 1:3, y = c('a', 'b', 'c'))*
df1$x	Getting column of values.	
df1[[1]]	Getting column of values.	
df1[1] or df1["x"]	Getting column as d.f.	
df1$z <- v3	Create new column.	
df1z <- NULL	Delete column.	
rm(df1)	Delete data frame.	
rbind(df1, df2)	Combine data frames.	
names(df1)	Getting columns names.	
row.names(df1)	Getting rows names.	
df1[1:2,"x"]	Getting rows.	*#[1] 1 2*
df1[c(T,F,T),]	Getting rows by logic.	*df1[c(T,F,F),] # 1 a (as df)*
df1 [df$x>n,]	Getting rows by values of columns.	
df1 [df$x>n,1]	Getting values by r and c	

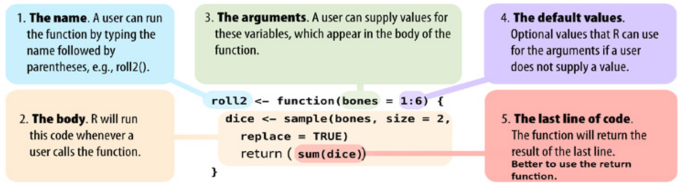
EXPLORING DATA FRAMES

?datasetname	Returns a description from de dataset.
dim()	Retrieve or set the dimension of an object.
ncol() or nrow()	Return the number of rows or columns present in x.
names()	Gets or sets the names of an object.D.F in this case.
head() or tail()	Returns the first or last 5 parts of a object. Rows in d.f.
summary()	Returns statistics summary.
str()	Returns type of values of each colum.and some samples.
table()	Returns counts at each combination of factor levels.

R STUDIO BASE 1/2

Juan Calvo. V1.0
juancalvo@glaucoestudio.com
www.glaucoestudio.com

PROGRAMMING



1. **The name**. A user can run the function by typing the name followed by parentheses, e.g., roll2().

2. **The body**. R will run this code whenever a user calls the function.

3. **The arguments**. A user can supply values for these variables, which appear in the body of the function.

4. **The default values**. Optional values that R can use for the arguments if a user does not supply a value.

5. **The last line of code**. The function will return the result of the last line. **Better to use the return function.**

```
roll2 <- function(bones = 1:6) {
  dice <- sample(bones, size = 2,
    replace = TRUE)
    return ( sum(dice))
}
```

**if (x>0){
print ("positive")}
else if(x<0){
 print("negative")}
else{
print("It's 0")}**
	Decision making if/else.

**for (x in 1:5){
print (1:x)}**
	Like a while statement, except that it tests the condition at the end of the loop body.

**while (x < 10){
print (x)
x <- x+1}**
	Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.

next	skip the current iteration of a loop without terminating it.
break	Terminates the loop statement and transfers execution to the statement immediately following the loop.

STATISTICS FUNCTIONS

min() or max()	Returns the (regular or parallel) maxima and minima of the input values: vector.
mean()	Return the mean of the input values: vector.
median()	Return the median of the input values: vector.
sd()	Return the standard deviation of the input values: vector.
var()	Return the variance of the input values: vector.
summary()	Return the data distribution. A summary.
help("Distributions")	Help for many standard probability distributions.
rnorm(n, mean, var)	Create n random numbers.
set.seed(n)	To get same values. Needed before every call to sample()
sample(x, size, replace = FALSE, prob = NULL)	x = num. from which to choose, size = number of items to choose from, replace = T for not repeated values...
data()	List the available data sets.
data("dataset")	Load specified data set.
rm("dataset")	Remove object from memory. In this case the data set.

PLOTTING

plot(x or data, y, type="type"...)	Generic function for plotting of R objects. No need of y if x is a single plotting structure (plot various plots). "type" for defining type of plot to be draw: "p" for points, "l" for lines,"b" for both...
points(x,y,col="c",...)	Draw a sequence of points in the exisiting plot.
lines(x,y,col="c",...)	Draw a line in the exisiting plot.
barplot()	Creates a bar plot with vertical or horizontal bars.
hist()	Computes a histogram of the given data values.
boxplot()	Produce box-and-whisker plot(s) of the given values.
pie()	Draw a pie chart.

R STUDIO BASE 2/2

Juan Calvo. V1.0
juancalvo@glaucoestudio.com
www.glaucoestudio.com