

# FUNCIONES R

ME!

2022-04-20

#Lo que pude rescatar de Coursera  
Lo primero y superimportante es la pregunta de investigation  
Para ayuda en R: **?command**  
Los foros de ciencia son *Stack Overflow* y *Cross Validated*  
Hay como 5 idiomas en R -> <https://r4stats.com/>  
Paquetes [^1] son descargables desde  
[^1] <https://www.rdocumentation.org/>

- **RCRAN** *Red integral de archivos R*
- **Bioconductor** *Paquetes de enfoque bioinformatico*
- **Github**
  - `install.packages(" ")`
  - `library ( )`
  - `remove.packages(" ")`
  - `detach() | detach("package:", unload=T)`
  - `version | sessionInfo()`
  - `help`

##Comandos a patir de pack readr

readr solo funciona en versiones de 4.2.0 de RDescargar esa version aqui

Luego se instala, y si hay problemas, probar con re-instalar "tzdb", eso sugieren las soluciones en foros Para  
mostrar columnas en un solo Data, se escribe algo asi `equipos <- read_csv("C:/Users/luis_/AppData/Local/R/win-library/4.2/readr/extdata/mtcars.csv.bz2", col_types = cols_only(mpg = col_number()),  
n_max = 32)`

##Almacenar datos

###Formato de texto

#### Sacar con 'dput' un archivo

`dput(y, file = "y.R")`

#### Meter con 'dput' desde un archivo

`new.y <- dget("y.R")`

Este no se que pex

`dump(c("x", "y"), file = "data.R")`

###Formato binario

Es clave **save()**, **save.image()**, and **serialize()**.

Para varios objetos la extension debe ser .RData y .rda con `save()` para otros softwares

```

Pa serializar x <- lista(1, 2, 3)
serialize(x, NULL)
##Conectarse a:
###Archivos
str(file, gzfile, url)
function (description = "", open = "", blocking = TRUE, encoding = getOption("encoding"), raw = FALSE,
method = getOption("url.method", "default"))
description es el nombre del dcto
open, es según necesitemos solo leer (r), editarlo (w), adjuntarlo (a), o modo binario ("rb", "wb", "ab")
Create a connection se simplifica con variable <- read_csv("Nombre del .txt")
Warning message:
In for (stage in c("install", "render")) if (stage %in% stages) Rd <- processRdSexprs(Rd, :
closing unused connection 5 (http://www.jhsph.edu)
###MATRICES
matrix()
Luego cuando se extrae con [] y deseamos conservar el formato de matriz y no un vector de longitud, entonces
debe ser: drop=False
###PARA LISTAS
x <- list(foo = 1:4, bar = 0.6)
Se extrae como los otros, pero ahora ya funciona $ y [[]], en esta ultima si se abrevian nombres, el elemento
primero va entre comillas y luego debe poner exact = False]]
###Valores NA
Si faltan valores y quiero eliminar entonces uso is.na(objeto) primero a un objeto y luego objeto[!objeto con
funcion] Elementos ausentes se imprimen como verdaderos con la primera funcion, pero se vuelven falsos
con complete.cases() Aplica igual en data frame, objeto(description[objeto con funcion, ])
##dplyr
Funcion buena porque se puede tener una idea de que se ejecuta por su significado directo en la palabra
select(description, object)
filter(description, object) n arrange(description, object)
arrange(description, desc(object))
rename(description, nuevo = antiguo)
mutate(description, nueva = calculo remplazo medida (variable, na.rm = TRUE)) #Si quieren eliminar las
cols no implicadas es con transmute
summarize()
group_by(description, objeto)
quantil(description$objeto, seq(0, 1, 0.2), na.rm = TRUE)
Para no crear conjunto de variables temporales se debe usar al final de cada conjunto %>%
— ##Controles
#####if() {#do something} #Continuar con el Code, else if() {# do something different} else {# do
something different si no se cumple}
#####for(i in seq_along(x)) {print(x[i])} sin seq_along, se pondria el intervalo.
#####objeto <- matrix(Contenido, n.row, n.col) for(i in seq_len(nrow(x))) {
for(j in seq_len(ncol(x))) {
print(x[i, j])
}
}
#####while(condition){ se hace y tambien por lo que se hace se hace algo mas } funciona bien porque se
repite hasta que se vuelva falso la condicion primaria
x0 <- 1 tol <- 1e-8 #####repeat {x1 <- computeEstimate() if(abs(x1 - x0) < tol) { ## Close enough?
break} else {x0 <- x1}} puede ser infinito
#####Para discontinuar una iteracion se usa next para saltar a algo si se cumple alguna condicion, y se
usa break para salir inmediatamente
##FUNCIONES
Agrupar serie de expresiones en un objeto y listo
Ejemploo

```

```

args(lm)
function (formula, data, subset, weights, na.action, method = "qr", model = TRUE, x = FALSE, y =
FALSE, qr = TRUE, singular.ok = TRUE, contrasts = NULL, offset, ...)
NULL
###make.power para construir otras funciones.  make.power <- function(n) { +pow <- function(x) {
+x^n} +pow + }

```