

# 8Listas\_cheatsheet.R

moka

2023-04-20

```
# Autor: Monika Avila Marquez, Ph.D.
# Fecha: 11.04.2023
# Referencia: Basado en R Programming Fundamentals, StanfordOnline XDFS112
# Objetivo: manipular datos que son heterogeneos (numericos, caracteres, booleano)
# Definiciones:
  #Listas son un metodo de combinar diferentes tipos de datos

rm(list=ls())

# Configurar el directorio

midirectorio<-setwd("~/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/8Listas")
midirectorio

## [1] "/Users/moka/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/8Listas"

# Ejemplo de lista

load("~/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/6Matrices/math2.Rdata")

# Si no guardo esta matriz, puede crearla usando el codigo siguiente (No olvide quitar los simbolos de
# Datos como matriz
# Crear conjunto de datos
observNames<-c("I1", "I2")
vbleNames<-c("v1", "v2")
class(vbleNames)

## [1] "character"

vecH<-rnorm(4)
math<-matrix(vecH,nrow=2)
rownames(math)<-observNames
colnames(math)<-vbleNames
dim(math)

## [1] 2 2

length(colnames)

## [1] 1

length(vbleNames)

## [1] 2

vecH2<-runif(4)
math2<-matrix(vecH2,nrow=2)
```

```
dimnames(matH2)<-list(observNames,vbleNames)
str(matH2)
```

```
## num [1:2, 1:2] 0.824 0.682 0.783 0.904
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:2] "I1" "I2"
## ..$ : chr [1:2] "v1" "v2"
```

```
class(matH2)
```

```
## [1] "matrix" "array"
```

```
summary(matH2) # Sabe qué tipo de unción utilizar a partir del tipo de objeto.
```

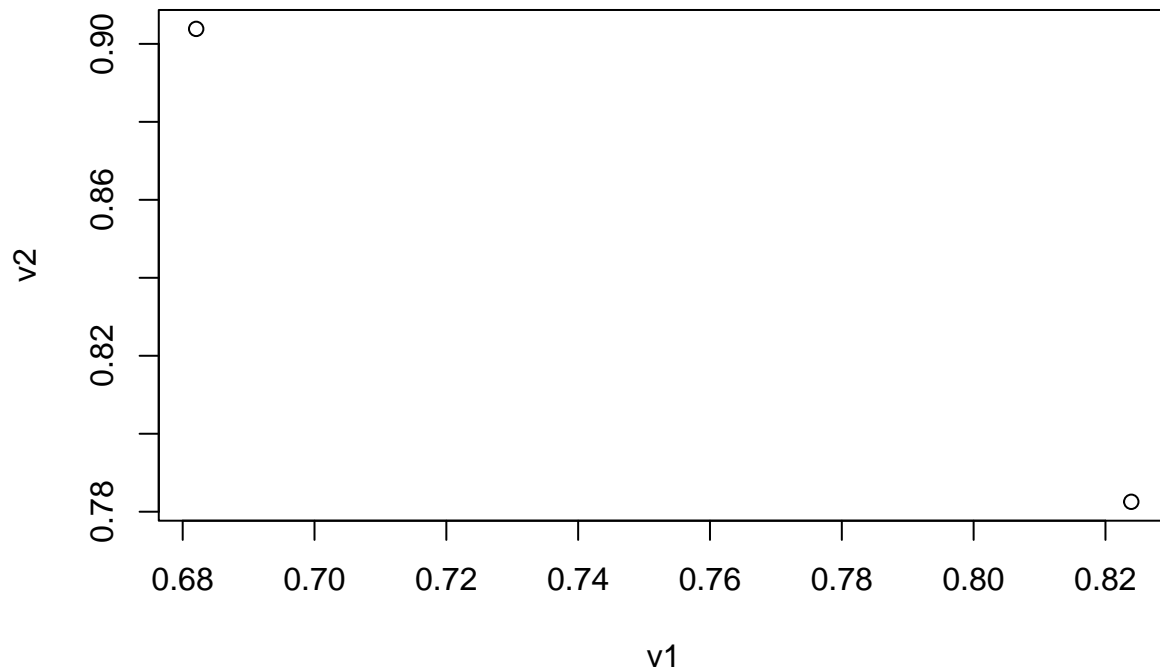
```
##          v1          v2
## Min.   :0.6820   Min.   :0.7825
## 1st Qu.:0.7175   1st Qu.:0.8129
## Median :0.7530   Median :0.8432
## Mean   :0.7530   Mean    :0.8432
## 3rd Qu.:0.7885   3rd Qu.:0.8735
## Max.   :0.8239   Max.    :0.9038
```

```
# Así, R puede ser considerado como un lenguaje OO.
```

```
?summary
```

```
?summary.matrix
```

```
plot(matH2) # Tomará las dos primeras columnas de la matriz
```



```
?plot
```

```
# Nota: list es para combinar objetos que no son del mismo
```

```
# tipo o longitud. Podemos combinar objetos que no son homogéneos.
```

```
list.dirs()
```

```
## [1] "."
```

```
# Guardar matriz
```

```
save(matH2,file="matH2.Rdata") # Guardar en formato binario que solo R puede leer
```

```

# Los nombres de las columnas y filas de nuestra matriz math2 son una lista

# Obtener nombres de columnas y filas
class(dimnames(math2))

## [1] "list"

dimnames(math2)

## [[1]]
## [1] "I1" "I2"
##
## [[2]]
## [1] "v1" "v2"

# Obtener nombres de filas
dimnames(math2)[[1]]

## [1] "I1" "I2"

# Cambiar nombre de una fila
dimnames(math2)[[1]][1]="H10"
dimnames(math2)[[1]] # Verificar el cambio

## [1] "H10" "I2"

# Crear una lista complicada con diferentes componentes

A<-matrix(c(4,2,0,3,1,7,2,8,4,5), # the data elements
          nrow=2, # the number of rows
          ncol=5) # the number of columns
lista1=list(name="mokametrics",bydate=1125,Amatrix=A,size=5.5,urban=FALSE)
lista1 # Esta lista tiene componentes con diferentes nombres que nosotros hemos asignado.

## $name
## [1] "mokametrics"
##
## $bydate
## [1] 1125
##
## $Amatrix
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    4    0    1    2    4
## [2,]    2    3    7    8    5
##
## $size
## [1] 5.5
##
## $urban
## [1] FALSE

# 2 metodos para acceder los componentes de la lista
# Metodo 1
lista1$name

## [1] "mokametrics"

# Metodo 2
lista1[[1]]

```

```
## [1] "mokametrics"
# Ver los nombres de los componentes de la lista
names(listal)

## [1] "name"      "bydate"    "Amatrix"   "size"      "urban"
# Este tipo de listas son utiles para funciones mas sofisticadas

# Ejemplo, los resultados de la funcion lm son una lista. Podemos
# acceder todos los componentes de esta lista
data(women)
attach(women)

## The following objects are masked from women (pos = 3):
##
##      height, weight
rs<-lm(weight~height)
rs

##
## Call:
## lm(formula = weight ~ height)
##
## Coefficients:
## (Intercept)      height
##      -87.52         3.45

summary(rs)

##
## Call:
## lm(formula = weight ~ height)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.7333 -1.1333 -0.3833  0.7417  3.1167
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -87.51667     5.93694  -14.74 1.71e-09 ***
## height       3.45000     0.09114   37.85 1.09e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.525 on 13 degrees of freedom
## Multiple R-squared:  0.991, Adjusted R-squared:  0.9903
## F-statistic: 1433 on 1 and 13 DF, p-value: 1.091e-14
# rs es mas que eso, es una lista que contiene varios componentes
names(rs)

## [1] "coefficients" "residuals"      "effects"        "rank"
## [7] "qr"           "df.residual"    "xlevels"        "call"
##                    "fitted.values" "assign"
##                    "terms"         "model"

# De nuevo, podemos acceder estos componentes utilizando dos metodos
rs$fitted.values
```

```
##           1           2           3           4           5           6           7           8           9           10           11
## 112.5833 116.0333 119.4833 122.9333 126.3833 129.8333 133.2833 136.7333 140.1833 143.6333 147.0833
##           12           13           14           15
## 150.5333 153.9833 157.4333 160.8833

rs[[5]]

##           1           2           3           4           5           6           7           8           9           10           11
## 112.5833 116.0333 119.4833 122.9333 126.3833 129.8333 133.2833 136.7333 140.1833 143.6333 147.0833
##           12           13           14           15
## 150.5333 153.9833 157.4333 160.8833

rs$rank

## [1] 2

rs[[4]]

## [1] 2

rs$call

## lm(formula = weight ~ height)

rs[[10]]

## lm(formula = weight ~ height)
# No quitar el comentario de la linea inferior. Solamente copiar en la consola para que ejecute
#rmarkdown::render("8Listas_cheatsheet.R",c("pdf_document","html_document"))
```