9DataFrames_cheatsheet.R

moka

2023-04-20

```
# Autor: Monika Avila Marquez, Ph.D.
# Fecha: 12.04.2023
# Objetivo: dataframes, explorar datos
# Definicion: data.frame es una lista pero que tiene propiedades de listas y matrices.  Son importantes
# almacenar datos categoricos y numericos. En esta lista, tenemos que tener mismo numero
# de observaciones por compnente o variable.
# Referencia: Basado en R Programming Fundamentals, StanfordOnline XDFS112
```

```
# Limpiar el espacio de trabajo
rm(list=ls())
```

```
# Configurar el directorio
```

```
midirectorio<-setwd("~/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/9Dataframes")
midirectorio
```

```
## [1] "/Users/moka/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/9Dataframes"
```

```
# Instalar paquetes y cargar paquetes
```

```
# install.packages("MASS") (Nota: despues de instalar por primera vez, podemos
# comentar esta inea.  No hay necesidad de instalar el paquete todas las veces, a menos que
# queramos realizar una actualizacion.
library(MASS)
```

```
# Cargar datos
```

```
# Ver las bases de datos que estan cargadas en nuestro espacio de trabajo de R
data()
# Vamos a utilizar UScereal
?UScereal # Obtener una descripcion de la base de datos
head(UScereal) # Ver las primeras observaciones.
```

```
##                            mfr calories   protein      fat   sodium     fibre    carbo   sugars shelf
## 100% Bran                    N 212.1212 12.121212 3.030303 393.9394 30.303030 15.15152 18.18182     3
## All-Bran                     K 212.1212 12.121212 3.030303 787.8788 27.272727 21.21212 15.15151     3
## All-Bran with Extra Fiber    K 100.0000  8.000000 0.000000 280.0000 28.000000 16.00000  0.00000     3
## Apple Cinnamon Cheerios      G 146.6667  2.666667 2.666667 240.0000  2.000000 14.00000 13.33333     1
## Apple Jacks                  K 110.0000  2.000000 0.000000 125.0000  1.000000 11.00000 14.00000     2
## Basic 4                      G 173.3333  4.000000 2.666667 280.0000  2.666667 24.00000 10.66667     3
##                            potassium vitamins
## 100% Bran                  848.48485 enriched
## All-Bran                   969.69697 enriched
## All-Bran with Extra Fiber 660.00000 enriched
## Apple Cinnamon Cheerios    93.33333 enriched
## Apple Jacks                30.00000 enriched
```

```
## Basic 4                    133.33333 enriched
```

```
UScereal[10:14,7:11] # Ver las observaciones 10 a 14, y las variables 7 a 11.
```

```
##                          carbo sugars shelf potassium vitamins
## Cheerios               13.60000    0.8     1        84 enriched
## Cinnamon Toast Crunch 17.33333   12.0     2        60 enriched
## Clusters               26.00000   14.0     3       210 enriched
## Cocoa Puffs            12.00000   13.0     2        55 enriched
## Corn Chex              22.00000    3.0     1        25 enriched
```

```
UScereal[1:10,1:6] # Ver las observaciones 1 a 10, y las variables 1 a 6.
```

```
##                            mfr calories   protein      fat   sodium     fibre
## 100% Bran                    N 212.1212 12.121212 3.030303 393.9394 30.303030
## All-Bran                     K 212.1212 12.121212 3.030303 787.8788 27.272727
## All-Bran with Extra Fiber    K 100.0000  8.000000 0.000000 280.0000 28.000000
## Apple Cinnamon Cheerios      G 146.6667  2.666667 2.666667 240.0000  2.000000
## Apple Jacks                  K 110.0000  2.000000 0.000000 125.0000  1.000000
## Basic 4                      G 173.3333  4.000000 2.666667 280.0000  2.666667
## Bran Chex                    R 134.3284  2.985075 1.492537 298.5075  5.970149
## Bran Flakes                  P 134.3284  4.477612 0.000000 313.4328  7.462687
## Cap'n'Crunch                 Q 160.0000  1.333333 2.666667 293.3333  0.000000
## Cheerios                     G  88.0000  4.800000 1.600000 232.0000  1.600000
```

```
class(UScereal) # UScereal es de tipo dataframe.  Nos permite combinar diferentes tipos de datos
```

```
## [1] "data.frame"
```

```
# Podemos ver que esta base de datos contiene variables categoricas y variables numericas
summary(UScereal)
```

```
##  mfr       calories        protein             fat            sodium          fibre
##  G:22   Min.   : 50.0   Min.   : 0.7519   Min.   :0.000   Min.   :  0.0   Min.   : 0.000
##  K:21   1st Qu.:110.0   1st Qu.: 2.0000   1st Qu.:0.000   1st Qu.:180.0   1st Qu.: 0.000
##  N: 3   Median :134.3   Median : 3.0000   Median :1.000   Median :232.0   Median : 2.000
##  P: 9   Mean   :149.4   Mean   : 3.6837   Mean   :1.423   Mean   :237.8   Mean   : 3.871
##  Q: 5   3rd Qu.:179.1   3rd Qu.: 4.4776   3rd Qu.:2.000   3rd Qu.:290.0   3rd Qu.: 4.478
##  R: 5   Max.   :440.0   Max.   :12.1212   Max.   :9.091   Max.   :787.9   Max.   :30.303
##      carbo           sugars          shelf         potassium          vitamins
##  Min.   :10.53   Min.   : 0.00   Min.   :1.000   Min.   : 15.00   100%    : 5
##  1st Qu.:15.00   1st Qu.: 4.00   1st Qu.:1.000   1st Qu.: 45.00   enriched:57
##  Median :18.67   Median :12.00   Median :2.000   Median : 96.59   none    : 3
##  Mean   :19.97   Mean   :10.05   Mean   :2.169   Mean   :159.12
##  3rd Qu.:22.39   3rd Qu.:14.00   3rd Qu.:3.000   3rd Qu.:220.00
##  Max.   :68.00   Max.   :20.90   Max.   :3.000   Max.   :969.70
```

```
str(UScereal$mfr)
```

```
##  Factor w/ 6 levels "G","K","N","P",..: 3 2 2 1 2 1 6 4 5 1 ...
```

```
str(UScereal$vitamins)
```

```
##  Factor w/ 3 levels "100%","enriched",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```
# Notar que para variables categoricas tenemos informacion diferente que
# para variables continuas

# UScereal es un dataframe, pero tambien una lista.  A pesar de ello, puede
```

```
# ser indexada como una matriz
class(UScereal)
```

```
## [1] "data.frame"
```

```
?data.frame
str(UScereal)
```

```
## 'data.frame':    65 obs. of  11 variables:
##  $ mfr      : Factor w/ 6 levels "G","K","N","P",..: 3 2 2 1 2 1 6 4 5 1 ...
##  $ calories : num  212 212 100 147 110 ...
##  $ protein  : num  12.12 12.12 8 2.67 2 ...
##  $ fat      : num  3.03 3.03 0 2.67 0 ...
##  $ sodium   : num  394 788 280 240 125 ...
##  $ fibre    : num  30.3 27.3 28 2 1 ...
##  $ carbo    : num  15.2 21.2 16 14 11 ...
##  $ sugars   : num  18.2 15.2 0 13.3 14 ...
##  $ shelf    : int  3 3 3 1 2 3 1 3 2 1 ...
##  $ potassium: num  848.5 969.7 660 93.3 30 ...
##  $ vitamins : Factor w/ 3 levels "100%","enriched",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```
# data.frame tiene propiedades de listas y matrices.
# Indexar items en un data frame
UScereal$mfr
```

```
##  [1] N K K G K G R P Q G G G G R K K G K K G R K K K P K P P G P P P Q G P K G Q G K G K K G P K Q Q
## [50] G K R K N N K K G G G G G R G G
## Levels: G K N P Q R
```

```
names(UScereal)
```

```
##  [1] "mfr"       "calories"  "protein"   "fat"       "sodium"    "fibre"     "carbo"     "sugars"
##  [9] "shelf"     "potassium" "vitamins"
```

```
UScereal[[1]]
```

```
##  [1] N K K G K G R P Q G G G G R K K G K K G R K K K P K P P G P P P Q G P K G Q G K G K K G P K Q Q
## [50] G K R K N N K K G G G G G R G G
## Levels: G K N P Q R
```

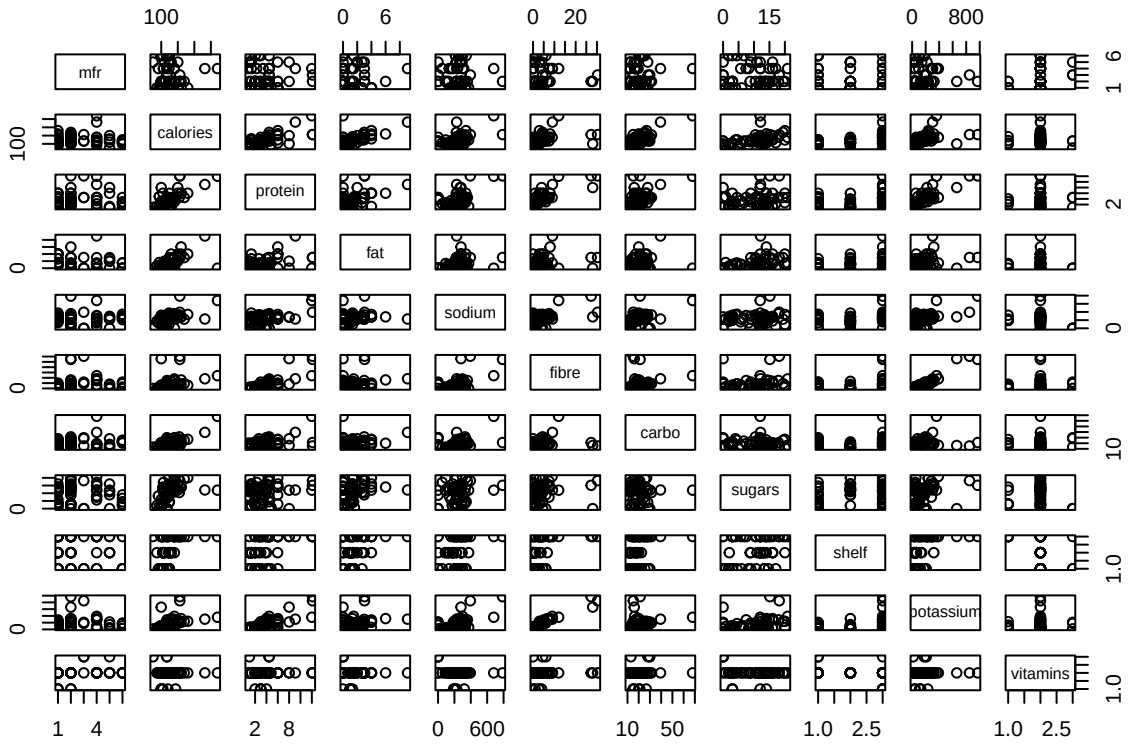
```
UScereal[,1]
```

```
##  [1] N K K G K G R P Q G G G G R K K G K K G R K K K P K P P G P P P Q G P K G Q G K G K K G P K Q Q
## [50] G K R K N N K K G G G G G R G G
## Levels: G K N P Q R
```

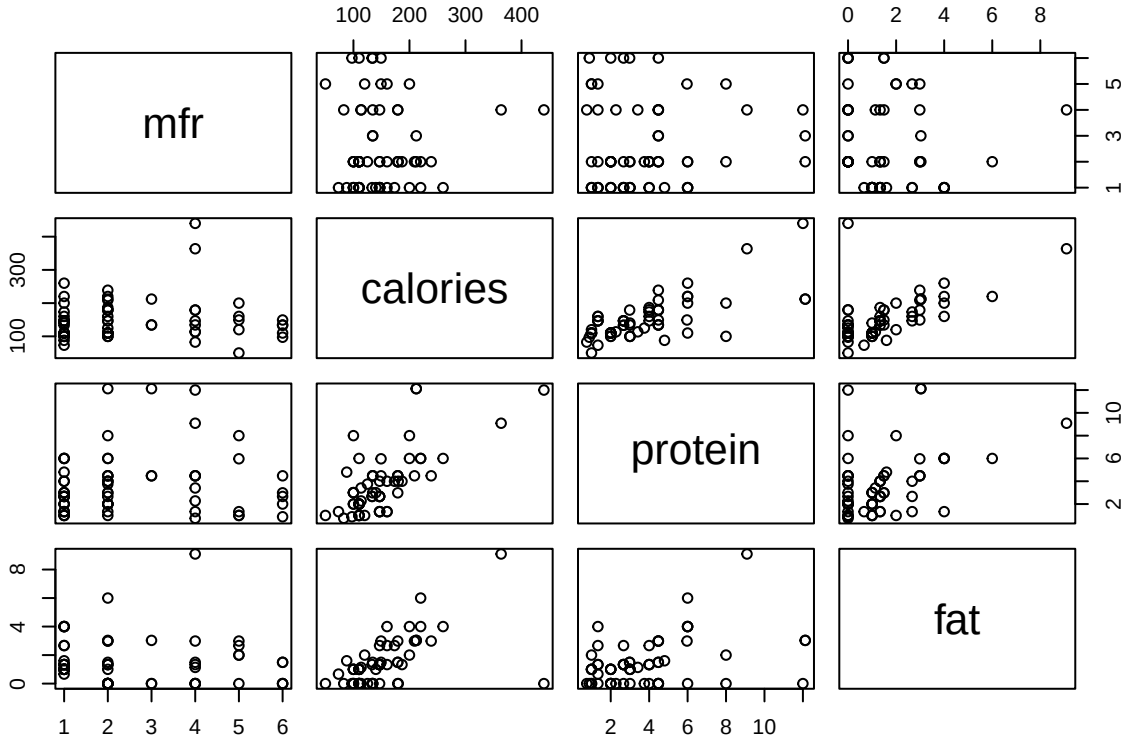
```
# Obtener dimensiones
dim(UScereal)
```

```
## [1] 65 11
```

```
# La funcion plot me da una matriz de dispersion
plot(UScereal)
```




```
plot(UScereal[,1:4])
```




```
length(UScereal)
```

```
## [1] 11
```

```
# Importante! ggplot2 solamente puede trabajar con dataframes (marco de datos)
# Otras funciones
```

```
# as.data.frame()

# No quitar el comentario de la linea inferior. Solamente copiar en la consola para que ejecute
#rmarkdown::render("9DataFrames_cheatsheet.R",c("pdf_document","html_document"))
```