

# 11GestionDatos\_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: Data wrangling
```

```
midirectorio<-setwd("~/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/11GestionDatos")
midirectorio
```

```
## [1] "/Users/moka/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/11GestionDatos"
```

```
# I. Valores faltantes: se denotan en R por NA, hay que tener cuidado porque en otros softwares se denota con 0
# Esto puede crear problemas.
```

```
NA
```

```
## [1] NA
```

```
# Crear ejemplo
```

```
vectorna<-c(1,2,NA,"2")
```

```
vectorna
```

```
## [1] "1" "2" NA  "2"
```

```
# Como determinar si hay valores faltantes?
```

```
# Incorrecto:
```

```
# Usar expresiones del tipo
```

```
NA==NA
```

```
## [1] NA
```

```
NA+1
```

```
## [1] NA
```

```
NA^2
```

```
## [1] NA
```

```
NA/0
```

```
## [1] NA
```

```
NA^0
```

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

```
NA*0
```

```
## [1] NA
```

```
1/NA
```

```
## [1] NA
```

```
# Correcto:
```

```
# Para determinar si hay valores faltantes, podemos utilizar la funcion is.na
```

```
?is.na
```

```
is.na(vectorna)
```

```
## [1] FALSE FALSE TRUE FALSE
```

```
# Obtener el promedio de este vector
```

```
mean(vectorna)
```

```
## Warning in mean.default(vectorna): argument is not numeric or logical: returning NA
```

```
## [1] NA
```

```
# En este caso, la funcion mean nos da NA porque: 1. tenemos un dato faltante NA, 2. un valor que esta
```

```
# Otro ejemplo
```

```
vectorna2<-c(2,2,NA,2)
```

```
vectorna2
```

```
## [1] 2 2 NA 2
```

```
# Soluciones a la presencia de valores faltantes
```

```
# (Importante: esto solo se puede hacer si los valores faltantes son valores faltantes aleatoriamente, esto esta fuera del alcance de este curso)
```

```
# 1. No tomar en cuenta los valores faltantes.
```

```
mean(vectorna2,na.rm=T)
```

```
## [1] 2
```

```
?mean
```

```
# 2. Reemplazar los valores faltantes
```

```
# Opcion 1: Utilizar un vector de variables booleanas, de manera que se reemplza por cierto valor cuando la variable booleana es igual a 1.
```

```
vectorna2corrected=vectorna2
```

```
vectorna2corrected[is.na(vectorna2)]<-0
```

```
vectorna2corrected
```

```
## [1] 2 2 0 2
```

```
# II. Valores inf: inf significa que es indefinido o infinito
```

```
1/0
```

```
## [1] Inf
```

```
1/0+1
```

```
## [1] Inf
```

```
1/0^1
```

```
## [1] Inf
```

```
mean(c(1/0,4))
```

```
## [1] Inf
```

```
1/0-1/0 # NaN not a number
```

```
## [1] NaN
```

```
# III. Imputar datos
```

```
# Podemos utilizar el paquete mice
```

```
install.packages("mice")
```

```
## Error in install.packages : Updating loaded packages
```

```
library(mice)
```

```
data("mammalsleep")
```

```
head(mammalsleep)
```

```
##           species      bw   brw sws  ps   ts  mls  gt pi sei odi
## 1 African elephant 6654.000 5712.0  NA  NA  3.3 38.6 645  3  5  3
## 2 African giant pouched rat  1.000   6.6 6.3 2.0  8.3  4.5  42  3  1  3
## 3 Arctic Fox      3.385   44.5  NA  NA 12.5 14.0  60  1  1  1
## 4 Arctic ground squirrel  0.920   5.7  NA  NA 16.5   NA  25  5  2  3
## 5 Asian elephant 2547.000 4603.0 2.1 1.8  3.9 69.0 624  3  5  4
## 6 Baboon      10.550  179.5 9.1 0.7  9.8 27.0 180  4  4  4
```

```
dim(mammalsleep)
```

```
## [1] 62 11
```

```
# Determinar numero de casos faltantes
```

```
?nic
```

```
nic(mammalsleep)
```

```
## [1] 20
```

```
# Determinar el patron de datos faltantes
```

```
md.pattern(mammalsleep)
```

	species	bw	brw	pi	sei	odi	ts	mls	gt	ps	sws	
42												0
9												2
3												1
2												1
1												3
1												2
2												2
2												3
	0	0	0	0	0	0	4	4	4	12	14	38

```
## species bw brw pi sei odi ts mls gt ps sws
## 42      1  1  1  1  1  1  1  1  1  1  1  0
## 9       1  1  1  1  1  1  1  1  1  0  0  2
## 3       1  1  1  1  1  1  1  1  0  1  1  1
## 2       1  1  1  1  1  1  1  0  1  1  1  1
## 1       1  1  1  1  1  1  1  0  1  0  0  3
## 1       1  1  1  1  1  1  1  0  0  1  1  2
## 2       1  1  1  1  1  1  0  1  1  1  0  2
## 2       1  1  1  1  1  1  0  1  1  0  0  3
##        0  0  0  0  0  0  4  4  4 12 14 38
```

```
?md.pattern
```

```
# Se puede ver que 42 casos estan completos. WSe puede ver que no hay bloques, lo que podria decir si
# los valores faltantes son al azar.
```

```
# Que hacer?
```

```
# 1. Eliminar todos los casos faltantes, pero podriamos terminar con una muestra bastante pequena
```

```
# En este caso 30% de los datos.
```

```
# 2. Imputar datos
```

```
?mice
```

```
inputed<-mice(mammalsleep) # proceso iterativo que obtiene promedios locales
```

```
##
## iter imp variable
## 1 1 sws* ps* ts* mls* gt*
## 1 2 sws* ps* ts* mls* gt*
## 1 3 sws* ps* ts* mls* gt*
## 1 4 sws* ps* ts* mls* gt*
## 1 5 sws* ps* ts* mls* gt*
## 2 1 sws* ps* ts* mls* gt*
## 2 2 sws* ps* ts* mls* gt*
## 2 3 sws* ps* ts* mls* gt*
```

```
## 2 4 sws* ps* ts* mls* gt*
## 2 5 sws* ps* ts* mls* gt*
## 3 1 sws* ps* ts* mls* gt*
## 3 2 sws* ps* ts* mls* gt*
## 3 3 sws* ps* ts* mls* gt*
## 3 4 sws* ps* ts* mls* gt*
## 3 5 sws* ps* ts* mls* gt*
## 4 1 sws* ps* ts* mls* gt*
## 4 2 sws* ps* ts* mls* gt*
## 4 3 sws* ps* ts* mls* gt*
## 4 4 sws* ps* ts* mls* gt*
## 4 5 sws* ps* ts* mls* gt*
## 5 1 sws* ps* ts* mls* gt*
## 5 2 sws* ps* ts* mls* gt*
## 5 3 sws* ps* ts* mls* gt*
## 5 4 sws* ps* ts* mls* gt*
## 5 5 sws* ps* ts* mls* gt*
```

```
## Warning: Number of logged events: 275
```

```
head(inputed) # Tenemos la base de datos original, y los valores imputados.
```

```
## $data
##           species      bw      brw  sws  ps   ts   mls    gt pi sei odi
## 1      African elephant 6654.000 5712.00   NA  NA   3.3  38.6 645.0  3  5  3
## 2 African giant pouched rat   1.000   6.60  6.3 2.0   8.3   4.5  42.0  3  1  3
## 3           Arctic Fox    3.385  44.50   NA  NA  12.5  14.0  60.0  1  1  1
## 4 Arctic ground squirrel   0.920   5.70   NA  NA  16.5   NA  25.0  5  2  3
## 5           Asian elephant 2547.000 4603.00  2.1 1.8   3.9  69.0 624.0  3  5  4
## 6           Baboon    10.550  179.50  9.1 0.7   9.8  27.0 180.0  4  4  4
## 7           Big brown bat    0.023   0.30 15.8 3.9  19.7  19.0  35.0  1  1  1
## 8 Brazilian tapir  160.000  169.00  5.2 1.0   6.2  30.4 392.0  4  5  4
## 9           Cat     3.300   25.60 10.9 3.6  14.5  28.0  63.0  1  2  1
## 10          Chimpanzee   52.160  440.00  8.3 1.4   9.7  50.0 230.0  1  1  1
## 11          Chinchilla    0.425   6.40 11.0 1.5  12.5   7.0 112.0  5  4  4
## 12           Cow    465.000  423.00  3.2 0.7   3.9  30.0 281.0  5  5  5
## 13 Desert hedgehog    0.550   2.40  7.6 2.7  10.3   NA   NA  2  1  2
## 14           Donkey  187.100  419.00   NA  NA   3.1  40.0 365.0  5  5  5
## 15 Eastern American mole    0.075   1.20  6.3 2.1   8.4   3.5  42.0  1  1  1
## 16           Echidna    3.000   25.00  8.6 0.0   8.6  50.0  28.0  2  2  2
## 17 European hedgehog    0.785   3.50  6.6 4.1  10.7   6.0  42.0  2  2  2
## 18           Galago    0.200   5.00  9.5 1.2  10.7  10.4 120.0  2  2  2
## 19           Genet    1.410  17.50  4.8 1.3   6.1  34.0   NA  1  2  1
## 20 Giant armadillo   60.000   81.00 12.0 6.1  18.1   7.0   NA  1  1  1
## 21           Giraffe  529.000  680.00   NA 0.3   NA  28.0 400.0  5  5  5
## 22           Goat    27.660  115.00  3.3 0.5   3.8  20.0 148.0  5  5  5
## 23 Golden hamster    0.120   1.00 11.0 3.4  14.4   3.9  16.0  3  1  2
## 24           Gorilla  207.000  406.00   NA  NA  12.0  39.3 252.0  1  4  1
## 25           Gray seal   85.000  325.00  4.7 1.5   6.2  41.0 310.0  1  3  1
## 26           Gray wolf   36.330  119.50   NA  NA  13.0  16.2  63.0  1  1  1
## 27 Ground squirrel    0.101   4.00 10.4 3.4  13.8   9.0  28.0  5  1  3
## 28           Guinea pig    1.040   5.50  7.4 0.8   8.2   7.6  68.0  5  3  4
## 29           Horse  521.000  655.00  2.1 0.8   2.9  46.0 336.0  5  5  5
## 30           Jaguar  100.000  157.00   NA  NA  10.8  22.4 100.0  1  1  1
## 31          Kangaroo   35.000   56.00   NA  NA   NA  16.3  33.0  3  5  4
```

```

## 32 Lesser short-tailed shrew    0.005    0.14  7.7 1.4  9.1    2.6  21.5  5  2  4
## 33          Little brown bat    0.010    0.25 17.9 2.0 19.9   24.0  50.0  1  1  1
## 34          Man        62.000 1320.00  6.1 1.9  8.0 100.0 267.0  1  1  1
## 35          Mole rat    0.122    3.00  8.2 2.4 10.6    NA  30.0  2  1  1
## 36          Mountain beaver    1.350    8.10  8.4 2.8 11.2    NA  45.0  3  1  3
## 37          Mouse    0.023    0.40 11.9 1.3 13.2    3.2  19.0  4  1  3
## 38          Musk shrew    0.048    0.33 10.8 2.0 12.8    2.0  30.0  4  1  3
## 39          N. American opossum    1.700    6.30 13.8 5.6 19.4    5.0  12.0  2  1  1
## 40          Nine-banded armadillo    3.500   10.80 14.3 3.1 17.4    6.5 120.0  2  1  1
## 41          Okapi    250.000  490.00    NA 1.0    NA   23.6 440.0  5  5  5
## 42          Owl monkey    0.480   15.50 15.2 1.8 17.0   12.0 140.0  2  2  2
## 43          Patas monkey   10.000  115.00 10.0 0.9 10.9   20.2 170.0  4  4  4
## 44          Phanlanger    1.620   11.40 11.9 1.8 13.7   13.0  17.0  2  1  2
## 45          Pig    192.000  180.00  6.5 1.9  8.4   27.0 115.0  4  4  4
## 46          Rabbit    2.500   12.10  7.5 0.9  8.4   18.0  31.0  5  5  5
## 47          Raccoon    4.288   39.20    NA  NA 12.5   13.7  63.0  2  2  2
## 48          Rat    0.280    1.90 10.6 2.6 13.2    4.7  21.0  3  1  3
## 49          Red fox    4.235   50.40  7.4 2.4  9.8    9.8  52.0  1  1  1
## 50          Rhesus monkey    6.800  179.00  8.4 1.2  9.6   29.0 164.0  2  3  2
## 51          Rock hyrax (Hetero. b)    0.750   12.30  5.7 0.9  6.6    7.0 225.0  2  2  2
## 52          Rock hyrax (Procyon hab)    3.600   21.00  4.9 0.5  5.4    6.0 225.0  3  2  3
## 53          Roe deer   14.830   98.20    NA  NA  2.6   17.0 150.0  5  5  5
## 54          Sheep   55.500  175.00  3.2 0.6  3.8   20.0 151.0  5  5  5
## 55          Slow loris    1.400   12.50    NA  NA 11.0   12.7  90.0  2  2  2
## 56          Star nosed mole    0.060    1.00  8.1 2.2 10.3    3.5    NA  3  1  2
## 57          Tenrec    0.900    2.60 11.0 2.3 13.3    4.5  60.0  2  1  2
## 58          Tree hyrax    2.000   12.30  4.9 0.5  5.4    7.5 200.0  3  1  3
## 59          Tree shrew    0.104    2.50 13.2 2.6 15.8    2.3  46.0  3  2  2
## 60          Vervet    4.190   58.00  9.7 0.6 10.3   24.0 210.0  4  3  4
## 61          Water opossum    3.500    3.90 12.8 6.6 19.4    3.0  14.0  2  1  1
## 62          Yellow-bellied marmot    4.050   17.00    NA  NA  NA   13.0  38.0  3  1  1
##
## $imp
## $imp$species
## [1] 1 2 3 4 5
## <0 rows> (or 0-length row.names)
##
## $imp$bw
## [1] 1 2 3 4 5
## <0 rows> (or 0-length row.names)
##
## $imp$brw
## [1] 1 2 3 4 5
## <0 rows> (or 0-length row.names)
##
## $imp$sws
##      1      2      3      4      5
## 1  2.1 11.9  3.2  2.1  2.1
## 3  2.1 13.2  3.2  9.7  2.1
## 4  3.2 11.9  2.1  8.6  2.1
## 14 3.3 11.9  3.2  8.2  2.1
## 21 2.1 11.9  3.3  7.4  2.1
## 24 2.1 13.2  3.3  8.4  2.1
## 26 3.2 11.9  3.2  8.6  2.1

```

```

## 30 3.2 11.9 3.2 9.1 3.2
## 31 3.3 11.9 3.3 8.4 2.1
## 41 3.3 13.2 2.1 8.2 3.3
## 47 2.1 12.8 2.1 9.5 2.1
## 53 2.1 11.9 3.3 8.4 2.1
## 55 3.2 12.0 3.2 8.6 3.3
## 62 3.2 11.9 2.1 9.5 3.3
##
## $imp$ps
##      1      2      3      4      5
## 1   3.4 0.8 1.9 1.8 1.9
## 3   0.7 2.3 1.8 1.8 1.9
## 4   0.6 2.4 1.9 1.9 1.8
## 14  0.7 2.2 1.8 1.8 1.8
## 24  0.7 2.2 1.8 1.8 1.9
## 26  0.8 2.3 1.9 1.9 1.8
## 30  0.7 2.3 1.8 1.8 1.9
## 31  0.7 2.4 1.9 1.8 1.9
## 47  0.6 2.6 1.9 1.9 1.9
## 53  0.8 2.2 1.8 1.9 1.8
## 55  0.7 2.6 1.8 1.8 1.9
## 62  0.7 2.4 1.8 1.9 1.8
##
## $imp$ts
##      1      2      3      4      5
## 21  3.1 3.8 3.8 3.8 3.8
## 31  3.8 3.3 3.8 2.9 3.3
## 41  3.8 3.8 3.8 2.9 3.1
## 62  3.8 2.9 3.1 3.1 2.9
##
## $imp$m1s
##      1      2      3      4      5
## 4   38.6 34.0 34.0 39.3 7.0
## 13  34.0 39.3 38.6 40.0 6.0
## 35  41.0 38.6 39.3 40.0 6.5
## 36  41.0 39.3 39.3 34.0 7.0
##
## $imp$gt
##      1      2      3      4      5
## 13  392 392 645 400 392
## 19  440 392 624 400 392
## 20  645 645 400 440 392
## 56  400 645 400 624 400
##
## $imp$pi
## [1] 1 2 3 4 5
## <0 rows> (or 0-length row.names)
##
## $imp$sei
## [1] 1 2 3 4 5
## <0 rows> (or 0-length row.names)
##
## $imp$odi
## [1] 1 2 3 4 5

```

```
## <0 rows> (or 0-length row.names)
##
##
## $m
## [1] 5
##
## $where
##   species   bw   brw   sws   ps   ts   mls   gt   pi   sei   odi
## 1   FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 2   FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3   FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 4   FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
## 5   FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 6   FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 7   FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 8   FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 9   FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 10  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 11  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 12  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 13  FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE
## 14  FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 15  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 16  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 17  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 18  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 19  FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## 20  FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## 21  FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 22  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 23  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 24  FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 25  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 26  FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 27  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 28  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 29  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 30  FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 31  FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
## 32  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 33  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 34  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 35  FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## 36  FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## 37  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 38  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 39  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 40  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 41  FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## 42  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 43  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 44  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 45  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 46  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```



```

## 47  FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 48  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 49  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 50  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 51  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 52  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 53  FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 54  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 55  FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## 56  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## 57  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 58  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 59  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 60  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 61  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 62  FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
##
## $blocks
## $blocks$species
## [1] "species"
##
## $blocks$bw
## [1] "bw"
##
## $blocks$brw
## [1] "brw"
##
## $blocks$sws
## [1] "sws"
##
## $blocks$ps
## [1] "ps"
##
## $blocks$ts
## [1] "ts"
##
## $blocks$mls
## [1] "mls"
##
## $blocks$gt
## [1] "gt"
##
## $blocks$pi
## [1] "pi"
##
## $blocks$sei
## [1] "sei"
##
## $blocks$odi
## [1] "odi"
##
## attr(,"calltype")
## species      bw      brw      sws      ps      ts      mls      gt      pi      sei      odi
## "type" "type" "type" "type" "type" "type" "type" "type" "type" "type" "type"

```

```
##
## $call
## mice(data = mammalsleep)

?mice
names(inputed)

## [1] "data"          "imp"            "m"              "where"          "blocks"
## [6] "call"           "nmis"           "method"         "predictorMatrix" "visitSequence"
## [11] "formulas"       "post"           "blots"          "ignore"         "seed"
## [16] "iteration"      "lastSeedValue"  "chainMean"      "chainVar"       "loggedEvents"
## [21] "version"        "date"

str(inputed)

## List of 22
## $ data          :'data.frame': 62 obs. of  11 variables:
## ..$ species: Factor w/ 62 levels "African elephant",...: 1 2 3 4 5 6 7 8 9 10 ...
## ..$ bw       : num [1:62] 6654 1 3.38 0.92 2547 ...
## ..$ brw      : num [1:62] 5712 6.6 44.5 5.7 4603 ...
## ..$ sws      : num [1:62] NA 6.3 NA NA 2.1 9.1 15.8 5.2 10.9 8.3 ...
## ..$ ps       : num [1:62] NA 2 NA NA 1.8 0.7 3.9 1 3.6 1.4 ...
## ..$ ts       : num [1:62] 3.3 8.3 12.5 16.5 3.9 9.8 19.7 6.2 14.5 9.7 ...
## ..$ mls      : num [1:62] 38.6 4.5 14 NA 69 27 19 30.4 28 50 ...
## ..$ gt       : num [1:62] 645 42 60 25 624 180 35 392 63 230 ...
## ..$ pi       : int [1:62] 3 3 1 5 3 4 1 4 1 1 ...
## ..$ sei      : int [1:62] 5 1 1 2 5 4 1 5 2 1 ...
## ..$ odi      : int [1:62] 3 3 1 3 4 4 1 4 1 1 ...
## $ imp          :List of 11
## ..$ species:'data.frame': 0 obs. of  5 variables:
## .. ..$ 1: logi(0)
## .. ..$ 2: logi(0)
## .. ..$ 3: logi(0)
## .. ..$ 4: logi(0)
## .. ..$ 5: logi(0)
## ..$ bw       :'data.frame': 0 obs. of  5 variables:
## .. ..$ 1: logi(0)
## .. ..$ 2: logi(0)
## .. ..$ 3: logi(0)
## .. ..$ 4: logi(0)
## .. ..$ 5: logi(0)
## ..$ brw      :'data.frame': 0 obs. of  5 variables:
## .. ..$ 1: logi(0)
## .. ..$ 2: logi(0)
## .. ..$ 3: logi(0)
## .. ..$ 4: logi(0)
## .. ..$ 5: logi(0)
## ..$ sws      :'data.frame': 14 obs. of  5 variables:
## .. ..$ 1: num [1:14] 2.1 2.1 3.2 3.3 2.1 2.1 3.2 3.2 3.3 3.3 ...
## .. ..$ 2: num [1:14] 11.9 13.2 11.9 11.9 11.9 13.2 11.9 11.9 13.2 ...
## .. ..$ 3: num [1:14] 3.2 3.2 2.1 3.2 3.3 3.3 3.2 3.2 3.3 2.1 ...
## .. ..$ 4: num [1:14] 2.1 9.7 8.6 8.2 7.4 8.4 8.6 9.1 8.4 8.2 ...
## .. ..$ 5: num [1:14] 2.1 2.1 2.1 2.1 2.1 2.1 2.1 3.2 2.1 3.3 ...
## ..$ ps       :'data.frame': 12 obs. of  5 variables:
## .. ..$ 1: num [1:12] 3.4 0.7 0.6 0.7 0.7 0.8 0.7 0.7 0.6 0.8 ...
```

```

## ..$ 2: num [1:12] 0.8 2.3 2.4 2.2 2.2 2.3 2.3 2.4 2.6 2.2 ...
## ..$ 3: num [1:12] 1.9 1.8 1.9 1.8 1.8 1.9 1.8 1.9 1.9 1.8 ...
## ..$ 4: num [1:12] 1.8 1.8 1.9 1.8 1.8 1.9 1.8 1.8 1.9 1.9 ...
## ..$ 5: num [1:12] 1.9 1.9 1.8 1.8 1.9 1.8 1.9 1.9 1.9 1.8 ...
## ..$ ts      : 'data.frame':  4 obs. of  5 variables:
## ..$ 1: num [1:4] 3.1 3.8 3.8 3.8
## ..$ 2: num [1:4] 3.8 3.3 3.8 2.9
## ..$ 3: num [1:4] 3.8 3.8 3.8 3.1
## ..$ 4: num [1:4] 3.8 2.9 2.9 3.1
## ..$ 5: num [1:4] 3.8 3.3 3.1 2.9
## ..$ mls     : 'data.frame':  4 obs. of  5 variables:
## ..$ 1: num [1:4] 38.6 34 41 41
## ..$ 2: num [1:4] 34 39.3 38.6 39.3
## ..$ 3: num [1:4] 34 38.6 39.3 39.3
## ..$ 4: num [1:4] 39.3 40 40 34
## ..$ 5: num [1:4] 7 6 6.5 7
## ..$ gt      : 'data.frame':  4 obs. of  5 variables:
## ..$ 1: num [1:4] 392 440 645 400
## ..$ 2: num [1:4] 392 392 645 645
## ..$ 3: num [1:4] 645 624 400 400
## ..$ 4: num [1:4] 400 400 440 624
## ..$ 5: num [1:4] 392 392 392 400
## ..$ pi      : 'data.frame':  0 obs. of  5 variables:
## ..$ 1: logi(0)
## ..$ 2: logi(0)
## ..$ 3: logi(0)
## ..$ 4: logi(0)
## ..$ 5: logi(0)
## ..$ sei     : 'data.frame':  0 obs. of  5 variables:
## ..$ 1: logi(0)
## ..$ 2: logi(0)
## ..$ 3: logi(0)
## ..$ 4: logi(0)
## ..$ 5: logi(0)
## ..$ odi     : 'data.frame':  0 obs. of  5 variables:
## ..$ 1: logi(0)
## ..$ 2: logi(0)
## ..$ 3: logi(0)
## ..$ 4: logi(0)
## ..$ 5: logi(0)
## $ m          : num 5
## $ where      : logi [1:62, 1:11] FALSE FALSE FALSE FALSE FALSE ...
## ..- attr(*, "dimnames")=List of 2
## ..$ : chr [1:62] "1" "2" "3" "4" ...
## ..$ : chr [1:11] "species" "bw" "brw" "sws" ...
## $ blocks     :List of 11
## ..$ species: chr "species"
## ..$ bw      : chr "bw"
## ..$ brw     : chr "brw"
## ..$ sws     : chr "sws"
## ..$ ps      : chr "ps"
## ..$ ts      : chr "ts"
## ..$ mls     : chr "mls"
## ..$ gt      : chr "gt"

```

```

## ..$ pi      : chr "pi"
## ..$ sei     : chr "sei"
## ..$ odi     : chr "odi"
## ..- attr(*, "calltype")= Named chr [1:11] "type" "type" "type" "type" ...
## ..- attr(*, "names")= chr [1:11] "species" "bw" "brw" "sws" ...
## $ call      : language mice(data = mammalsleep)
## $ nmis      : Named int [1:11] 0 0 0 14 12 4 4 4 0 0 ...
## ..- attr(*, "names")= chr [1:11] "species" "bw" "brw" "sws" ...
## $ method    : Named chr [1:11] "" "" "" "pmm" ...
## ..- attr(*, "names")= chr [1:11] "species" "bw" "brw" "sws" ...
## $ predictorMatrix: num [1:11, 1:11] 0 1 1 1 1 1 1 1 1 1 ...
## ..- attr(*, "dimnames")=List of 2
## .. ..$ : chr [1:11] "species" "bw" "brw" "sws" ...
## .. ..$ : chr [1:11] "species" "bw" "brw" "sws" ...
## $ visitSequence : chr [1:11] "species" "bw" "brw" "sws" ...
## $ formulas     :List of 11
## ..$ species:Class 'formula' language species ~ bw + brw + sws + ps + ts + mls + gt + pi + sei + odi
## .. ..- attr(*, ".Environment")=<environment: 0x10b615fa8>
## ..$ bw      :Class 'formula' language bw ~ species + brw + sws + ps + ts + mls + gt + pi + sei + odi
## .. ..- attr(*, ".Environment")=<environment: 0x10b615fa8>
## ..$ brw     :Class 'formula' language brw ~ species + bw + sws + ps + ts + mls + gt + pi + sei + odi
## .. ..- attr(*, ".Environment")=<environment: 0x10b615fa8>
## ..$ sws     :Class 'formula' language sws ~ species + bw + brw + ps + ts + mls + gt + pi + sei + odi
## .. ..- attr(*, ".Environment")=<environment: 0x10b615fa8>
## ..$ ps      :Class 'formula' language ps ~ species + bw + brw + sws + ts + mls + gt + pi + sei + odi
## .. ..- attr(*, ".Environment")=<environment: 0x10b615fa8>
## ..$ ts      :Class 'formula' language ts ~ species + bw + brw + sws + ps + mls + gt + pi + sei + odi
## .. ..- attr(*, ".Environment")=<environment: 0x10b615fa8>
## ..$ mls     :Class 'formula' language mls ~ species + bw + brw + sws + ps + ts + gt + pi + sei + odi
## .. ..- attr(*, ".Environment")=<environment: 0x10b615fa8>
## ..$ gt      :Class 'formula' language gt ~ species + bw + brw + sws + ps + ts + mls + pi + sei + odi
## .. ..- attr(*, ".Environment")=<environment: 0x10b615fa8>
## ..$ pi      :Class 'formula' language pi ~ species + bw + brw + sws + ps + ts + mls + gt + sei + odi
## .. ..- attr(*, ".Environment")=<environment: 0x10b615fa8>
## ..$ sei     :Class 'formula' language sei ~ species + bw + brw + sws + ps + ts + mls + gt + pi + odi
## .. ..- attr(*, ".Environment")=<environment: 0x10b615fa8>
## ..$ odi     :Class 'formula' language odi ~ species + bw + brw + sws + ps + ts + mls + gt + pi + sei
## .. ..- attr(*, ".Environment")=<environment: 0x10b615fa8>
## $ post      : Named chr [1:11] "" "" "" "" ...
## ..- attr(*, "names")= chr [1:11] "species" "bw" "brw" "sws" ...
## $ blots     :List of 11
## ..$ species: list()
## ..$ bw      : list()
## ..$ brw     : list()
## ..$ sws     : list()
## ..$ ps      : list()
## ..$ ts      : list()
## ..$ mls     : list()
## ..$ gt      : list()
## ..$ pi      : list()
## ..$ sei     : list()
## ..$ odi     : list()
## $ ignore    : logi [1:62] FALSE FALSE FALSE FALSE FALSE FALSE ...
## $ seed      : logi NA

```

```
## $ iteration      : num 5
## $ lastSeedValue  : int [1:626] 10403 575 1615935056 -295711774 -1465396111 -591905638 -2119852292 -
## $ chainMean      : num [1:11, 1:5, 1:5] NaN NaN NaN 2.59 1.84 ...
##   .- attr(*, "dimnames")=List of 3
##   ..$ : chr [1:11] "species" "bw" "brw" "sws" ...
##   ..$ : chr [1:5] "1" "2" "3" "4" ...
##   ..$ : chr [1:5] "Chain 1" "Chain 2" "Chain 3" "Chain 4" ...
## $ chainVar       : num [1:11, 1:5, 1:5] NA NA NA 0.33978 0.00265 ...
##   .- attr(*, "dimnames")=List of 3
##   ..$ : chr [1:11] "species" "bw" "brw" "sws" ...
##   ..$ : chr [1:5] "1" "2" "3" "4" ...
##   ..$ : chr [1:5] "Chain 1" "Chain 2" "Chain 3" "Chain 4" ...
## $ loggedEvents   : 'data.frame': 275 obs. of  5 variables:
##   ..$ it  : int [1:275] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ im  : int [1:275] 1 1 1 1 1 1 1 1 1 1 ...
##   ..$ dep : chr [1:275] "sws" "sws" "sws" "ps" ...
##   ..$ meth: chr [1:275] "pmm" "pmm" "pmm" "pmm" ...
##   ..$ out : chr [1:275] "df set to 1. # observed cases: 48 # predictors: 71" "speciesArctic Fox, sp
## $ version        :Classes 'package_version', 'numeric_version' hidden list of 1
##   ..$ : int [1:3] 3 15 0
## $ date           : Date[1:1], format: "2023-04-20"
## - attr(*, "class")= chr "mids"
```

*# Este tema es delicado, y se necesita mayor conocimiento estadístico.*

*# IV. Detectando outliers (Valores extremos)*

```
?mammalsleep
summary(mammalsleep)
```

	species	bw	brw	sws	ps
African elephant	: 1	Min. : 0.005	Min. : 0.14	Min. : 2.100	Min. : 0.000
African giant pouched rat	: 1	1st Qu.: 0.600	1st Qu.: 4.25	1st Qu.: 6.250	1st Qu.: 0.900
Arctic Fox	: 1	Median : 3.342	Median : 17.25	Median : 8.350	Median : 1.800
Arctic ground squirrel	: 1	Mean : 198.790	Mean : 283.13	Mean : 8.673	Mean : 1.972
Asian elephant	: 1	3rd Qu.: 48.202	3rd Qu.: 166.00	3rd Qu.: 11.000	3rd Qu.: 2.550
Baboon	: 1	Max. : 6654.000	Max. : 5712.00	Max. : 17.900	Max. : 6.600
(Other)	: 56			NA's : 14	NA's : 12

  

	ts	mls	gt	pi	sei	odi
Min. :	2.60	Min. : 2.000	Min. : 12.00	Min. : 1.000	Min. : 1.000	Min. : 1.000
1st Qu.:	8.05	1st Qu.: 6.625	1st Qu.: 35.75	1st Qu.: 2.000	1st Qu.: 1.000	1st Qu.: 1.000
Median :	10.45	Median : 15.100	Median : 79.00	Median : 3.000	Median : 2.000	Median : 2.000
Mean :	10.53	Mean : 19.878	Mean : 142.35	Mean : 2.871	Mean : 2.419	Mean : 2.613
3rd Qu.:	13.20	3rd Qu.: 27.750	3rd Qu.: 207.50	3rd Qu.: 4.000	3rd Qu.: 4.000	3rd Qu.: 4.000
Max. :	19.90	Max. : 100.000	Max. : 645.00	Max. : 5.000	Max. : 5.000	Max. : 5.000
NA's :	4	NA's : 4	NA's : 4			

*# Vemos que en el caso del peso (bw), hay valores bastante altos.*

```
mammalsleep[which.max(mammalsleep$bw),]
```

```
##           species bw brw sws ps ts mls gt pi sei odi
## 1 African elephant 6654 5712 NA NA 3.3 38.6 645 3 5 3
```

*# En este caso, se cree que es un valor correcto. Preo podems encontrar un 99999 que significa que es*

*# Se necesita documentar esto explicando:*

*# valores extremos, y si son correctos o no.*

```
# V. Interactuando con datos
```

```
# Obtener una sub muestra para ver si hay algunos patrones.
```

```
s=sample(62,10)
```

```
mammalsleep[s,]
```

##	species	bw	brw	sws	ps	ts	mls	gt	pi	sei	odi
## 48	Rat	0.280	1.9	10.6	2.6	13.2	4.7	21	3	1	3
## 6	Baboon	10.550	179.5	9.1	0.7	9.8	27.0	180	4	4	4
## 18	Galago	0.200	5.0	9.5	1.2	10.7	10.4	120	2	2	2
## 49	Red fox	4.235	50.4	7.4	2.4	9.8	9.8	52	1	1	1
## 40	Nine-banded armadillo	3.500	10.8	14.3	3.1	17.4	6.5	120	2	1	1
## 20	Giant armadillo	60.000	81.0	12.0	6.1	18.1	7.0	NA	1	1	1
## 2	African giant pouched rat	1.000	6.6	6.3	2.0	8.3	4.5	42	3	1	3
## 43	Patas monkey	10.000	115.0	10.0	0.9	10.9	20.2	170	4	4	4
## 12	Cow	465.000	423.0	3.2	0.7	3.9	30.0	281	5	5	5
## 57	Tenrec	0.900	2.6	11.0	2.3	13.3	4.5	60	2	1	2

```
# No quitar el comentario de la linea inferior. Solamente copiar en la consola para que ejecute  
#rmarkdown::render("11GestionDatos_cheatsheet.R",c("pdf_document","html_document"))
```