

ULTIMA CLASE - PRACTICA R

VECTORES Y MATRICES

#Genero vector

```
set.seed(99)
v = sample(1:100, 12)
v
```

#Genero matriz

```
M = matrix(v, 3, byrow = TRUE)
M
```

#Ultima columna

```
M[,ncol(M)]
```

#Maximo elemento impar

```
max(M[M%%2 == 1])
```

#Promedio x fila

```
apply(M, 1, mean)
```

#Maximo entre sumatorias x columna

```
max(apply(M, 2, sum))
```

#Multiplico x 2 (menos segunda columna)

```
M[,-2] = M[,c(1, 3, 4)]*2
M
```

#Armo nuevo vector c/ primera y segunda fila

```
V2 = as.vector(M[1:2,])
V2
```

#Elementos multiples de 3

```
V2[V2%%3==0]
```

#Mayor numero par

```
max(V2[V2%%2 == 0])
```

#Reemplazo por 0 elementos no multiples de 4

```
V2[V2%%4!=0] = 0
V2
```

#Contar multiples de 2 y mayores que 40

```
sum(V2%%2 == 0 & V2>40)
```

DATA FRAMES

#Ejecuto acciones

```
library(ggplot2)
data(msleep)
View(msleep)
```

```
suenio = subset(data.frame(msleep), select = c(name, vore, sleep_total, bodywt))
suenio = na.omit(suenio)
eliminadas = na.action(suenio)
suenio$vore = as.factor(suenio$vore)
dim(suenio)
```

#Ejecuto head

```
head(suenio)
```

#Subconjunto con carnivoros y sin columna vore

```
CARNIS = subset(suenio, vore=="carni", select = -vore)
CARNIS
head(CARNIS)
```

#Total carnivoros > 10kg y 5-10hs de sueño

```
sum(CARNIS$bodywt>10 & CARNIS$sleep_total<10 & CARNIS$sleep_total>5)
```

#Subconjunto > 100kg y -10hs de sueño

```
GRANDES = subset(suenio, suenio$bodywt>100 & suenio$sleep_total<10)
GRANDES
```

#Total carnivoros en GRANDES

```
sum(GRANDES$vore == "carni")
```

#Total por tipo de alimentacion en "suenio"

```
summary(suenio$vore)
```

#Nombre del animal que menos duerme en "suenio"

```
min(suenio$sleep_total)
subset(suenio, suenio$sleep_total == (min(suenio$sleep_total)), select = "name")
```

#Nombre y forma de alimentación animal mas pesado en "GRANDES"

```
max(GRANDES$bodywt)
subset(GRANDES, GRANDES$bodywt == max(GRANDES$bodywt), select = c("name",
"vore"))
```

#Promedio de peso s/tipo alimentacion

```
tapply(suenio$bodywt, suenio$vore, mean)
```

#Grafico de torta de de animales s/tipo de alimentacion

```
summary(suenio$vore)
```

```
pie(summary(suenio$vore), main = "Alimentación: proporciones")
```

#Subconjunto sin "African elephant" y "Asian elephant"

```
suenio2 = subset(suenio, suenio$name!="African elephant" & suenio$name!="Asian elephant")
```

```
suenio2
```

```
sum(suenio$name=="African elephant")
```

```
sum(suenio$name=="Asian elephant")
```

#Grafico de dispersion que relacione el peso y la cantidad de horas de sueño en "suenio2"

```
plot(suenio2$sleep_total, suenio2$bodywt, type = "p", main = "Peso vs Horas de sueño")
```

#Agregar lineas al grafico de dispersión

```
abline(h = mean(suenio2$bodywt), col = "red")
```

```
abline(v = mean(suenio2$sleep_total), col = "green")
```

#Grafico de cajas de cuantas hs duermen segun tipo de alimentacion

```
boxplot(suenio$sleep_total ~ suenio$vore, main = "¿Duermen segun lo que comen?")
```

```
# Los mas dormilones son los insectos
```

```

-----
> set.seed(99)
> v = sample(1:100, 12)
> v
[1] 48 33 44 22 62 32 13 20 31 68 9 82

> M = matrix(v, 3, byrow = TRUE)
> M
      [,1] [,2] [,3] [,4]
[1,]   48   33   44   22
[2,]   62   32   13   20
[3,]   31   68    9   82

> M[,ncol(M)]
[1] 22 20 82

> max(M[M%%2 == 1])
[1] 33
> M%%2 != 0
      [,1] [,2] [,3] [,4]
[1,] FALSE  TRUE FALSE FALSE
[2,] FALSE FALSE  TRUE FALSE
[3,]  TRUE FALSE  TRUE FALSE

> apply(M, 1, mean)
[1] 36.75 31.75 47.50

> apply(M, 2, sum)
[1] 141 133 66 124
> max(apply(M, 2, sum))
[1] 141

> M[, -2] = M[, c(1, 3, 4)] * 2
> M
      [,1] [,2] [,3] [,4]
[1,]   96   33   88   44
[2,]  124   32   26   40
[3,]   62   68   18  164

> V2 = as.vector(M[1:2,])
> V2
[1] 96 124 33 32 88 26 44 40

> V2%%3==0
[1]  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE

```

```
> V2[V2%%3==0]
[1] 96 33
```

```
> max(V2[V2%%2 == 0])
[1] 124
```

```
> V2[V2%%4!=0] = 0
> V2
[1] 96 124 0 32 88 0 44 40
```

```
> sum(V2%%2 == 0 & V2>40)
[1] 4
```

##DATA FRAMES##

```
> library(ggplot2)
```

Aviso:

package 'ggplot2' was built under R version 4.4.2

```
> data(msleep)
```

```
> View(msleep)
```

```
> suenio = subset(data.frame(msleep), select = c(name, vore, sleep_total,
bodywt))
```

```
> suenio = na.omit(suenio)
```

```
> eliminadas = na.action(suenio)
```

```
> suenio$vore = as.factor(suenio$vore)
```

```
> dim(suenio)
```

```
[1] 76 4
```

```
> head(suenio)
```

	name	vore	sleep_total	bodywt
1	Cheetah	carni	12.1	50.000
2	Owl monkey	omni	17.0	0.480
3	Mountain beaver	herbi	14.4	1.350
4	Greater short-tailed shrew	omni	14.9	0.019
5	Cow	herbi	4.0	600.000
6	Three-toed sloth	herbi	14.4	3.850

#Subconjunto con carnivoros y sin columna vore

```
> CARNIS = subset(suenio, vore=="carni", select = -vore)
```

```
> head(CARNIS)
```

	name	sleep_total	bodywt
1	Cheetah	12.1	50.00
7	Northern fur seal	8.7	20.49
9	Dog	10.1	14.00
18	Long-nosed armadillo	17.4	3.50
28	Domestic cat	12.5	3.30
31	Pilot whale	2.7	800.00

```
#Total carnivoros > 10kg y 5-10hs de sueño
```

```
> sum(CARNIS$bodywt>10 & CARNIS$sleep_total<10 & CARNIS$sleep_total>5)
[1] 4
```

```
#Subconjunto > 100kg y -10hs de sueño
```

```
> GRANDES = subset(suenio, suenio$bodywt>100 & suenio$sleep_total<10)
> GRANDES
```

	name	vore	sleep_total	bodywt
5	Cow	herbi	4.0	600.000
21	Asian elephant	herbi	3.9	2547.000
23	Horse	herbi	2.9	521.000
24	Donkey	herbi	3.1	187.000
30	Giraffe	herbi	1.9	899.995
31	Pilot whale	carni	2.7	800.000
36	African elephant	herbi	3.3	6654.000
77	Brazilian tapir	herbi	4.4	207.501
80	Bottle-nosed dolphin	carni	5.2	173.330

```
#Total carnivoros en GRANDES
```

```
> sum(GRANDES$vore == "carni")
[1] 2
```

```
#Total por tipo de alimentacion en suenio
```

```
> summary(suenio$vore)
carni  herbi insecti  omni
  19    32      5    20
```

```
#Nombre del animal que menos duerme en "suenio"
```

```
> subset(suenio, suenio$sleep_total == (min(suenio$sleep_total)), select =
"name")
      name
30 Giraffe
```

```
#Nombre y forma de alimentacion animal mas pesado en "GRANDES"
```

```
> subset(GRANDES, GRANDES$bodywt == max(GRANDES$bodywt), select = c("name",
"vore"))
      name  vore
36 African elephant herbi
```

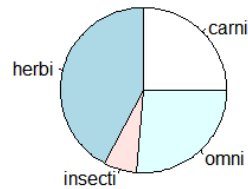
```
#Promedio de peso s/tipo alimentacion
```

```
> tapply(suenio$bodywt, suenio$vore, mean)
carni  herbi  insecti  omni
90.75111 366.87725  12.92160  12.71800
```

```
#Grafico de torta de de animales s/tipo de alimentacion
```

```
> summary(suenio$vore)
  carni   herbi insecti   omni
    19     32      5     20
> pie(summary(suenio$vore), main = "Alimentación: proporciones")
```

Alimentación: proporciones



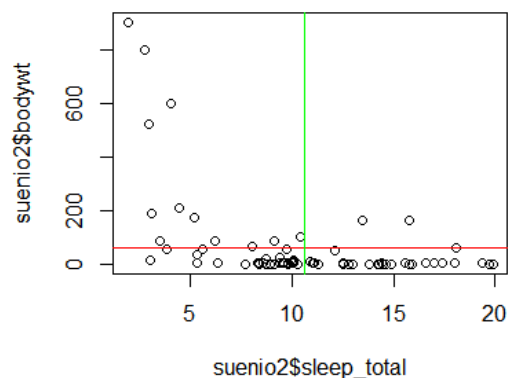
#Subconjunto sin "African elephant" y "Asian elephant"

```
> suenio2 = subset(suenio, suenio$name!="African elephant" &
suenio$name!="Asian elephant")
> sum(suenio$name=="African elephant")
[1] 1
> sum(suenio$name=="Asian elephant")
[1] 1
```

#Grafico de dispersion que relacione el peso y la cantidad de horas de sueño en "suenio2"

```
> plot(suenio2$sleep_total, suenio2$bodywt, type = "p", main = "Peso vs Horas
de sueño")
> abline(h = mean(suenio2$bodywt), col = "red")
> abline(v = mean(suenio2$sleep_total), col = "green")
```

Peso vs Horas de sueño



#Grafico de cajas de cuantas hs duermen segun tipo de alimentacion

```
> boxplot(suenio$sleep_total ~ suenio$vore, main = "¿Duermen segun lo que
comen?")
```

¿Duermen segun lo que comen?

