

Ejercicios Tema 1

Juan Andrés Peraira Pérez

6 de abril de 2018

Ejercicio 1: Resuelva las siguientes operaciones

- $(3^5 + 2^9) - \log(2)$
- $\frac{1}{\sqrt{\log(2) * 2^5}}$

Solución:

```
x<-(3^5+2^9)-log(2)
x
```

```
## [1] 754.3069
```

```
y<-1/sqrt(log(2)*2^5)
y
```

```
## [1] 0.2123305
```

Ejercicio 2: Defina los siguientes vectores $x=\{4,2,1\}$ e $y=\{1,0,-1\}$:

calcule la suma de los elementos de “x”.

calcule la suma de los vectores “x” + “y”

Solución:

```
##-- Suma de los elementos del vector X
x<-c(4,2,1)
sum(x)
```

```
## [1] 7
```

```
##-- Suma de los dos vectores
x<-c(4,2,1)
y<-c(1,0,-1)
x+y
```

```
## [1] 5 2 0
```

Ejercicio 3: Construya las siguientes matrices de datos:

$$A = \begin{pmatrix} 5 & 1 \\ 3 & 2 \end{pmatrix}$$

$$B = \begin{pmatrix} 4 & 1 \\ 3 & 2 \end{pmatrix}$$

Multiplique por dos la matriz A y sume ambas matrices

Solución:

```
#-- Construcción de matrices  
A<-matrix(c(5,3,1,2),nrow = 2,ncol = 2)  
A
```

```
##      [,1] [,2]  
## [1,]    5    1  
## [2,]    3    2
```

```
B<-matrix(c(1,4,7,5),nrow = 2,ncol = 2)  
B
```

```
##      [,1] [,2]  
## [1,]    1    7  
## [2,]    4    5
```

```
#--Multiplicación por dos de la matriz A  
A * 2
```

```
##      [,1] [,2]  
## [1,]   10    2  
## [2,]    6    4
```

```
#--Suma de las matrices  
A+B
```

```
##      [,1] [,2]  
## [1,]    6    8  
## [2,]    7    7
```

Ejercicio 4: Construya un data.frame con la siguiente información:

Sexo	Edad	Peso
M	30	80
F	35	61
M	31	85

Solución:

```
Data_frame<-data.frame(Sexo=c("M","F","M"),Edad=c(30,35,31),Peso=c(80,61,85))
Data_frame
```

```
##   Sexo Edad Peso
## 1    M   30   80
## 2    F   35   61
## 3    M   31   85
```

Ejercicio 5: Crea una función que tome un vector numérico como argumento y calcule la media y tamaño (o largo) del vector. En el resultado de la función se debe obtener ambos valores (media y tamaño).

Solución:

```
Funcion_Vector<-function(x){
  as.vector(x)
  media<-mean(x)
  longitud<-length(x)
  return(list(media=media,longitud=longitud))
}
```

```
x<-(1:65)
Funcion_Vector(x)
```

```
## $media
## [1] 33
##
## $longitud
## [1] 65
```

Ejercicio 6: Crea una función que calcule el IRPF e IVA de una factura. Recuerden que en una factura al monto inicial se le resta el IRPF (porque lo paga la empresa) y se suma el IVA (porque lo paga el comprador). En el resultado se debe observar: importe inicial, costo del IRPF (pueden tomar el 19% o dejarlo como un argumento), costo del IVA (21% o dejarlo como un argumento a ingresar) e importe final de la factura.

Solución:

```
funcion_factura<-function(factura_inicial,iva,irpf){
  factura<-factura_inicial
  porcentaje_iva<-iva*0.01
  porcentaje_irpf<-irpf*0.01
  ## calculo del iva
  iva_cantidad<-porcentaje_iva*factura
  ## calculo del irpf
  irpf_cantidad<-porcentaje_irpf*factura
  ## Resultado
```

```

resultado= factura -irpf_cantidad + iva_cantidad
return(list("factura"=factura,
           "cantidad de iva"=iva_cantidad,
           "Cantidad de IRPF" = irpf_cantidad,
           "Resultado"=resultado))
}

```

```

funcion_factura(28500,21,8)

```

```

## $factura
## [1] 28500
##
## $`cantidad de iva`
## [1] 5985
##
## $`Cantidad de IRPF`
## [1] 2280
##
## $Resultado
## [1] 32205

```

Ejercicio 7: En el siguiente vector sustituya los valores perdidos por 0:
 $x = \{0, 1, 2, NA, 3, 2, 1\}$

Solución:

```

x<-c(0,1,2,NA,3,2,1)
x

```

```

## [1] 0 1 2 NA 3 2 1

```

```

x[is.na(x)]=0
x

```

```

## [1] 0 1 2 0 3 2 1

```

Ejercicio 8: Cargue los datos “trees” del dataset (el conjunto de datos que viene instalado por defecto en R) y utilice la función “apply” para calcular la media de cada variable de la base de datos.

Solución:

```
##-- Cargamos el dataset "trees" y lo definimos como "datos"  
datos<-trees  
datos
```

```
##      Girth Height Volume  
## 1      8.3      70   10.3  
## 2      8.6      65   10.3  
## 3      8.8      63   10.2  
## 4     10.5      72   16.4  
## 5     10.7      81   18.8  
## 6     10.8      83   19.7  
## 7     11.0      66   15.6  
## 8     11.0      75   18.2  
## 9     11.1      80   22.6  
## 10    11.2      75   19.9  
## 11    11.3      79   24.2  
## 12    11.4      76   21.0  
## 13    11.4      76   21.4  
## 14    11.7      69   21.3  
## 15    12.0      75   19.1  
## 16    12.9      74   22.2  
## 17    12.9      85   33.8  
## 18    13.3      86   27.4  
## 19    13.7      71   25.7  
## 20    13.8      64   24.9  
## 21    14.0      78   34.5  
## 22    14.2      80   31.7  
## 23    14.5      74   36.3  
## 24    16.0      72   38.3  
## 25    16.3      77   42.6  
## 26    17.3      81   55.4  
## 27    17.5      82   55.7  
## 28    17.9      80   58.3  
## 29    18.0      80   51.5  
## 30    18.0      80   51.0  
## 31    20.6      87   77.0
```

```
apply(datos,2,mean)
```

```
##      Girth      Height      Volume  
## 13.24839 76.00000 30.17097
```

Ejercicio 9: Transforme los datos de formato “wide” a “long”. NOTA: los resultados pueden estar en un orden distinto.

id	Sex	V1	V2	V3
1	F	0.89	-0.45	0.21
2	M	0.30	0.21	0.37
3	F	0.22	-0.71	-0.54

id	Sex	N	V
1	F	1	0.89
1	F	2	-0.45
1	F	3	0.21
2	M	1	0.30
2	M	2	0.21
2	M	3	0.37
3	F	1	0.22
3	F	2	-0.71
3	F	3	-0.54

Solución:

```
datos<-data.frame(id=c(1,2,3),Sex=c("F","M","F"),V1=c(0.89,0.30,0.22),
                  V2=c(-0.45,0.21,-0.71),V3=c(0.21,0.37,-0.54))
```

```
datos
```

```
##   id Sex  V1  V2  V3
## 1  1  F 0.89 -0.45 0.21
## 2  2  M 0.30  0.21 0.37
## 3  3  F 0.22 -0.71 -0.54
```

```
## -- Instalamos y cargamos el paquete necesario
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 3.4.4
```

```
datos_long<- melt(datos, id=c("id","Sex"))
datos_long
```

```
##   id Sex variable value
## 1  1  F      V1  0.89
## 2  2  M      V1  0.30
## 3  3  F      V1  0.22
## 4  1  F      V2 -0.45
## 5  2  M      V2  0.21
## 6  3  F      V2 -0.71
## 7  1  F      V3  0.21
## 8  2  M      V3  0.37
## 9  3  F      V3 -0.54
```