

Parcial

Santiago Lozano-Fulanito González

21 de marzo de 2020

Este examen parcial se debe realizar INDIVIDUAL parciales iguales tendrán nota de cero, y lo deben entregar antes de las 9:00 PM, deben entregar un script, es decir un .R, con las codificaciones respectivas y debidamente comentado, es obligatorio usar las funciones vistas en las diapositiva

1

- a. Cree 5 vectores. Uno de caracteres, uno de numeric, uno de integer y uno de complex, cada uno con más de 5 elementos

```
x<- c(1.4,3.8,4.7,5.9,1.6,3.6)
y<-c("Santiago","Enrique","Lozano", "González","Fulanito")
z<- as.integer(1:6)
w<- c(1+5i,2+6i,3+7i,4+8i,5+1i,6+2i,7+3i)
```

- b. Chequee si los vectores son del tipo especificado ¿Cómo lo hace en R?

```
class(x)

## [1] "numeric"

class(y)

## [1] "character"

class(z)

## [1] "integer"

class(w)

## [1] "complex"
```

- c. Cree una lista de varios tipos de estructuras de datos, mínimo 4 elementos debe tener la lista

```
Mat<-matrix(1:12,nrow=4,byrow=TRUE)
L<-list(12,c(34,67),Mat,1:15,list(10,11))
L

## [[1]]
## [1] 12
##
## [[2]]
## [1] 34 67
##
## [[3]]
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
## [4,]   10   11   12
##
## [[4]]
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
##
## [[5]]
## [[5]][[1]]
## [1] 10
##
## [[5]][[2]]
## [1] 11
```

d. Cree una matriz de 3 filas por 4 columna con datos arreglados del 1 al 12 en forma horizontal

```
M <- matrix(1:12,nrow = 3,byrow = T)
M
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    6    7    8
## [3,]    9   10   11   12
```

e. Cree una matriz de 3 filas y 4 columnas con números arbitrarios y con acomodación vertical

```
x<-runif(12)
Ma <- matrix(x,nrow = 3,byrow = F)
Ma
```

```
##      [,1]      [,2]      [,3]      [,4]
## [1,] 0.8231838 0.80720703 0.36101877 0.240627309
## [2,] 0.5406383 0.07906838 0.08840597 0.805871282
## [3,] 0.5175967 0.95005285 0.76252430 0.001787599
```

f. Cree un vector de factores con calificaciones de estudiantes (Deficiente, insuficiente,aceptable, sobresaliente, excelente), que contenga mínimos 13 entradas distribuidas de manera aleatoria

```
calificaciones <- factor(c("deficiente","excelente","insuficiente","deficiente","aceptable",
,"suficiente","excelente","insuficiente","excelente","insuficiente","deficiente","insuficiente",
,"insuficiente","excelente","excelente"))
calificaciones
```

```
## [1] deficiente    excelente    insuficiente deficiente    aceptable
## [6] suficiente    excelente    insuficiente excelente    insuficiente
## [11] deficiente    insuficiente insuficiente excelente    excelente
## Levels: aceptable deficiente excelente insuficiente suficiente
```

g. Cree un vector de tipo ordered con niveles de riesgo en un hospital (bajo, medio, alto) donde bajo es el menor, medio el que le sigue y alto es mayor, debe tener mínimos 8 entradas

```
z <- ordered(c("alto","bajo","bajo","medio","alto","bajo","medio","bajo","medio","bajo",
,"alto","bajo","medio","alto","bajo","medio","alto"),
levels=c("bajo","medio","alto"))
z
```

```
## [1] alto bajo bajo medio alto bajo medio bajo medio bajo alto
## [12] bajo medio alto bajo medio alto
## Levels: bajo < medio < alto
```

2

Cree la siguiente base de datos

```
base11 <- read.table("base11.txt",header = T,sep = " ")
base11
```

##	id	sexo	año.de.nacimiento	fecha.de.confirmación
## 539	539	female	1975	2020-02-23
## 1190	1190	female	1960	2020-02-26
## 457	457	female	1963	2020-02-23
## 230	230	female	1961	2020-02-22
## 117	117	female	1980	2020-02-21
## 487	487	female	1967	2020-02-23
## 217	217	female	1962	2020-02-22
## 532	532	male	1956	2020-02-23

```
attach(base11)
id <- c(539,1190,457,230,117,487,217,532)
sexo <-c("female","female","female","female","female","female","female","male")
ano.de.nacimiento <- c(1975,1960,1963,1961,1980,1967,1962,1956)
fecha.de.confirmacion <- c("2020-02-23","2020-02-26","2020-02-23","2020-02-22","2020-02-21",
"2020-02-23","2020-02-22","2020-02-23")
data <- data.frame("id"=id,"sexo"=sexo,"año.de.nacimiento"=ano.de.nacimiento,
"fecha.de.confirmación"=fecha.de.confirmacion)
data
```

##	id	sexo	año.de.nacimiento	fecha.de.confirmación
## 1	539	female	1975	2020-02-23
## 2	1190	female	1960	2020-02-26
## 3	457	female	1963	2020-02-23
## 4	230	female	1961	2020-02-22
## 5	117	female	1980	2020-02-21
## 6	487	female	1967	2020-02-23
## 7	217	female	1962	2020-02-22
## 8	532	male	1956	2020-02-23

3

Importe la base de datos base22

```
base22 <- read.table("base22.txt",header = T,sep = " ")
base22
```

##	país	sexo	Estado	id
## 531	Korea	female	isolated	531
## 457	Korea	female	isolated	457
## 481	Korea	female	isolated	481
## 117	Korea	female	isolated	117
## 1184	Korea	male	isolated	1184
## 539	Korea	female	isolated	539
## 224	Korea	female	isolated	224
## 217	Korea	female	isolated	217

a. Extraiga los id de las mujeres (sexo=="female")

```
attach(base22)
```

```
## The following objects are masked _by_ .GlobalEnv:
```

```
##
##      id, sexo
## The following objects are masked from base11:
##
##      id, sexo
id[sexo=="female"]
```

```
## [1] 539 1190 457 230 117 487 217
```

b. Extraiga las filas de la 4 a 7 y las columnas de 2 a 4

```
data[4:7,2:4]

##      sexo año.de.nacimiento fecha.de.confirmación
## 4 female           1961           2020-02-22
## 5 female           1980           2020-02-21
## 6 female           1967           2020-02-23
## 7 female           1962           2020-02-22
```

4

De la matriz

```
ymatrix <- matrix(data = c(6,34,923,5,0, 112:116, 5,9,34,76,2, 545:549),nrow=5)
ymatrix
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    6  112    5  545
## [2,]   34  113    9  546
## [3,]  923  114   34  547
## [4,]    5  115   76  548
## [5,]    0  116    2  549
```

a. Saque el promedio de cada fila

```
apply(ymatrix,MARGIN = 1,FUN = mean)
```

```
## [1] 167.00 175.50 404.50 186.00 166.75
```

b. ordene cada fila de mayor a menor

```
apply(ymatrix,MARGIN = 1,FUN =function(x){rev(sort(x))})
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  545  546  923  548  549
## [2,]  112  113  547  115  116
## [3,]    6   34  114   76    2
## [4,]    5    9   34    5    0
```

5.

Importe la base de datos worms

```
worms <- read.table("worms.txt",header = T,dec = ".")
attach(worms)
worms
```

##	Field.Name	Area	Slope	Vegetation	Soil.pH	Damp	Worm.density
## 1	Nashs.Field	3.6	11	Grassland	4.1	FALSE	4
## 2	Silwood.Bottom	5.1	2	Arable	5.2	FALSE	7
## 3	Nursery.Field	2.8	3	Grassland	4.3	FALSE	2
## 4	Rush.Meadow	2.4	5	Meadow	4.9	TRUE	5
## 5	Gunness.Thicket	3.8	0	Scrub	4.2	FALSE	6
## 6	Oak.Mead	3.1	2	Grassland	3.9	FALSE	2
## 7	Church.Field	3.5	3	Grassland	4.2	FALSE	3
## 8	Ashurst	2.1	0	Arable	4.8	FALSE	4
## 9	The.Orchard	1.9	0	Orchard	5.7	FALSE	9
## 10	Rookery.Slope	1.5	4	Grassland	5.0	TRUE	7
## 11	Garden.Wood	2.9	10	Scrub	5.2	FALSE	8
## 12	North.Gravel	3.3	1	Grassland	4.1	FALSE	1
## 13	South.Gravel	3.7	2	Grassland	4.0	FALSE	2
## 14	Observatory.Ridge	1.8	6	Grassland	3.8	FALSE	0
## 15	Pond.Field	4.1	0	Meadow	5.0	TRUE	6
## 16	Water.Meadow	3.9	0	Meadow	4.9	TRUE	8
## 17	Cheapside	2.2	8	Scrub	4.7	TRUE	4
## 18	Pound.Hill	4.4	2	Arable	4.5	FALSE	5
## 19	Gravel.Pit	2.9	1	Grassland	3.5	FALSE	1
## 20	Farm.Wood	0.8	10	Scrub	5.1	TRUE	3

- a. saque el promedio de los elementos de Area con respecto a Vegetation, es decir cuál es el promedio de las Areas de Grassland y así sucesivamente

```
tapply(Area,Vegetation,mean)
```

##	Arable	Grassland	Meadow	Orchard	Scrub
##	3.866667	2.911111	3.466667	1.900000	2.425000