

PRUEBA DE EVALUACIÓN R



UNIVERSIDAD
COMPLUTENSE
DE MADRID



MÁSTER BIG DATA Y BUSINESS ANALYTICS

AUTORA: ESTHER FERNÁNDEZ BARCHÍN

FECHA DE ENTREGA: 26 DE ABRIL 2022

Pregunta 1 (20 puntos): Un boleto del sorteo de la ONCE consta de dos partes, la primera es un número de 4 dígitos y la segunda es un número de tres dígitos que forman la serie del boleto.

Aquí consideramos sólo el número, por ejemplo,

0 2 0 9

Se pide:

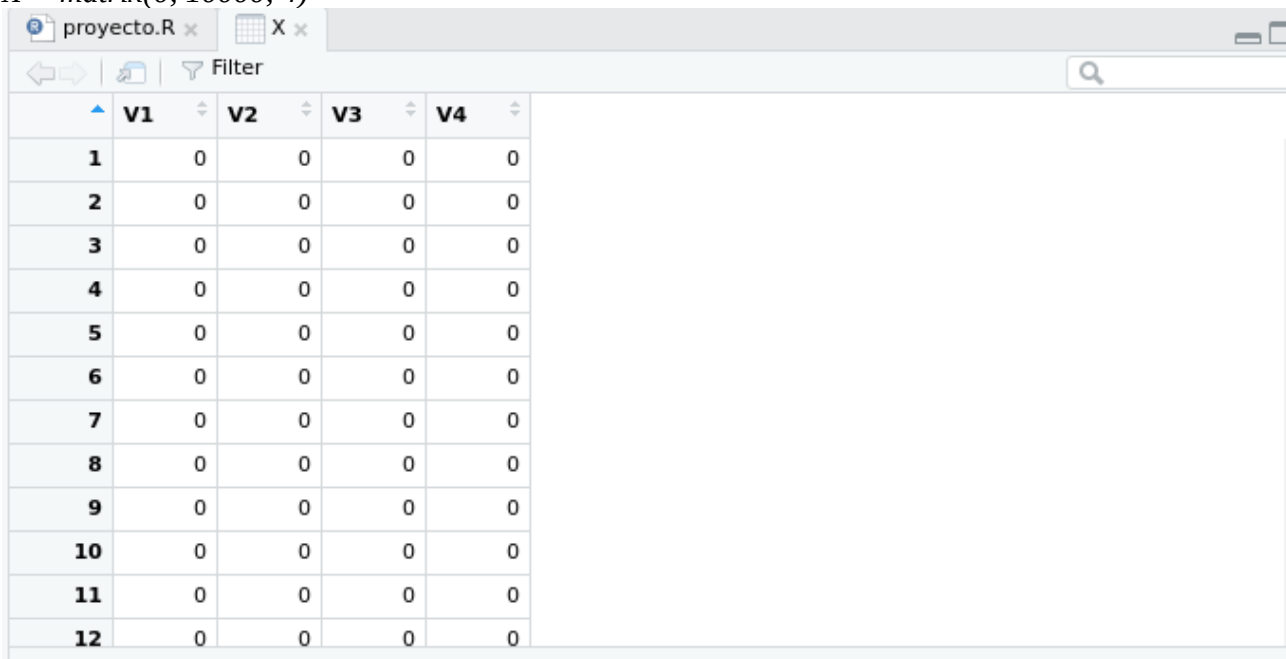
a) Genera todos los números que entran en el sorteo de la ONCE y mostrarlos con los cuatro dígitos. $b = \text{sum}(\text{table}(\text{rowSums}(X)))$

b) ¿Cuál es la suma de los números de un boleto que más se repite?

a)

Primero de todo creamos la matriz y sus dimensión 10000x4

$X = \text{matrix}(0, 10000, 4)$



	V1	V2	V3	V4
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0

Pero nos sale una matriz de 10.000x4 completamente llena de ceros. Para que salgan todos las posibles combinaciones de números vamos a crear un bucle.

Para sacar cada dígito hay que calcular el resto.

```
for (i in 0:10000) {  
  # digito 1  
  X[i,4] = i%%10  
  # digito 2  
  X[i,3] = trunc(i/10)%%10 # 3542 / 10 = 354.2 -> trunc(354.2) = 354 -> 354 %% 10 = 4  
  # digito 3  
  X[i,2] = trunc(i/100)%%10 # 3542 / 100 = 35.42 -> trunc(35.42) = 35 -> 35 %% 10 = 5  
  # digito 4  
  X[i,1] = trunc(i/1000)%%10  
}
```

	▲ V1 ▼	▲ V2 ▼	▲ V3 ▼	▲ V4 ▼
1	0	0	0	1
2	0	0	0	2
3	0	0	0	3
4	0	0	0	4
5	0	0	0	5
6	0	0	0	6
7	0	0	0	7
8	0	0	0	8
9	0	0	0	9
10	0	0	1	0
11	0	0	1	1
12	0	0	1	2
13	0	0	1	3
14	0	0	1	4
15	0	0	1	5
16	0	0	1	6
17	0	0	1	7
18	0	0	1	8
19	0	0	1	9
20	0	0	2	0
21	0	0	2	1
22	0	0	2	2

b)

Para calcular la suma de cada boleto de lotería utilizamos la función `table()`

`b = table(rowSums(X))`

```

  0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22
1   4  10  20  35  56  84 120 165 220 282 348 415 480 540 592 633 660 670 660 633 592 540
23  24  25  26  27  28  29  30  31  32  33  34  35  36
480 415 348 282 220 165 120 84 56 35 20 10 4 1

```

Y para encontrar la suma de boletos que más se repite utilizamos la siguiente función:

`max(table(rowSums(X)))`

ó

`max(b)` (ya que hemos llamado a la función anterior como `b`)

[1] 670

Por lo tanto la suma que más se repite es 18.

Pregunta 2 (20 puntos): En la carpeta covid_19 hay una serie de archivos sobre el covid-19 en España (Fuente:

<https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov-China/situacionActual.htm>).

Se pide:

- a) Leer los archivos “datos_provincias.csv”, “CodProv.txt” y “CodCCAA.dat”. Añade el código de la comunidad autónoma al fichero “datos_provincias.csv” (no manualmente).
b) Selecciona los datos de la comunidad autónoma que te corresponda. Para saber cuál es tu comunidad autónoma realiza la siguiente operación

DNI o Pasaporte mod 17 por ejemplo (12345678 % 17 = 6 → Castilla y León)

hay que seleccionar las provincias de Castilla y León

Gráficos

- c) Realizar un gráfico que muestre adecuadamente la evolución de los casos nuevos.

Justifica el gráfico elegido.

- d) Presenta en único gráfico la evolución de las distintas variables (columnas) por medio de un gráfico de líneas múltiples. Utiliza diferentes colores y añade una leyenda muestre el origen de cada línea.

a)

```
library(readr)
```

```
datosbrutos_provincias <-
```

```
read_csv("~/Escritorio/MASTER/4.R/Datos/Cod_19/datos_provincias.csv")
```

```
codigo_CCAA <- read_delim("~/Escritorio/MASTER/4.R/Datos/Cod_19/CodCCAA.csv",  
                          delim = "\t", escape_double = FALSE,  
                          trim_ws = TRUE)
```

```
codigo_provincia <- read_csv("~/Escritorio/MASTER/4.R/Datos/Cod_19/CodProv.txt")
```









#Hemos comprobado que en datos_provincias hay varios NA por lo que hay que eliminarlos.

Los datos que salen como NA corresponden con la comunidad de Navarra pero R lo lee como si fueran una fila sin datos, por ello he preferido eliminar dichas filas, además en el siguiente apartado me tocó Asturias.

Para hacer esta parte se ha utilizado la librería TidyR ya que nos permite ordenar los datos “sucios” para obtener objetos de datos en R .

```
library(tidyR)
```

```
datos_provincias<-datosbrutos_provincias[!is.na(datosbrutos_provincias$provincia_iso),]
```

Data		
 codigo_CCAA	17 obs. of 3 variables	
 codigo_provincia	50 obs. of 3 variables	
 datos_provincias	12087 obs. of 7 variables	
 datosbrutos_provinci...	12324 obs. of 7 variables	

#Antes de unir las tablas datos_provincias y codigo_CCAA, hay que hacer una serie de pasos

#Separamos el codigo ES y las iniciales del código de provincia

Para hacer este paso, lo primero de todo creamos un subset con el codigo de comunidad autonomo para despues separarlo en dos y quedarnos con la columna

```
pro1 <-subset(codigo_provincia, select= "Código")
```

	Código
1	ES-C
2	ES-VI
3	ES-AB
4	ES-A

```
pro2<-pro1%>%separate("Código",c('ES','provincia_iso'),"-",remove = TRUE)
```

	ES	provincia_iso
1	ES	C
2	ES	VI
3	ES	AB
4	ES	A

```
pro3 <- subset(pro2,select = "provincia_iso")
```

	provincia_iso
1	C
2	VI
3	AB
4	A

#Ahora crearemos un dataset aparte para poder adjuntarlo con datosbrutos_provincia

```
codigo_provincia <- cbind(codigo_provincia, provincia_iso = c(pro3))
```

	Código	Nombre de la subdivisión en la ISO1	Comunidad autónoma	provincia_iso
1	ES-C	A Coruña	GA	C
2	ES-VI	Araba	PV	VI
3	ES-AB	Albacete	CM	AB
4	ES-A	Alicante/Alacant	VC	A

```
codigo_provincia2 <- subset(codigo_provincia,select = c("provincia_iso","Comunidad autónoma"))
```

	provincia_iso	Comunidad autónoma
1	C	GA
2	VI	PV
3	AB	CM
4	A	VC

#Creamos un nuevo dataframe a partir del merge

```
datos <- merge(datosbrutos_provincias,codigo_provincia2,by = "provincia_iso")
```

	provincia_iso	fecha	num_casos	num_casos_prueba_pcr	num_casos_prueba_test_ac
1	A	2020-04-16	44	21	
2	A	2020-04-08	39	30	
3	A	2020-09-02	213	213	
4	A	2020-02-01	0	0	

#Reordenamos las columnas dataset para que el código provincia este al lado del código de la Comunidad Autónoma

```
datos1 <- datos [ , c(1,8,2,3,4,5,6,7)]
```

	provincia_iso	Comunidad autónoma	fecha	num_casos	num_casos_prueba_pcr
1	A	VC	2020-04-16	44	21
2	A	VC	2020-04-08	39	30
3	A	VC	2020-09-02	213	213
4	A	VC	2020-02-01	0	0

b)

Para saber la comunidad autonoma tendremos que usar el data set codigo_CCAA

#Primero de todo hacemos la operacion con el DNI

```
DNI <- 48231944
```

```
indice_ca <- DNI %% 17
```

```
indice_ca
```

```
[1] 3
```

```
codigo_ca <- codigo_CCAA[indice_ca,1]
```

```
ca <- codigo_CCAA[indice_ca,2]
```

```
cat("Mi comunidad autonoma es",ca[[c(1)]],"con codigo",codigo_ca[[c(1)]])
```

```

Mi comunidad autonoma es Asturias, Principado de con codigo ES-AS

```

Ahora que ya sabemos la Comunidad Autónoma a estudiar creamos la nueva base de datos con la que vamos a trabajar.

```
datos_asturias <- datos1[datos1$`Comunidad autónoma` == "AS",]
```

	provincia_iso	Comunidad autónoma	fecha	num_casos	num_casos_prueba_pcr
6874	O	AS	2020-07-17	1	1
6875	O	AS	2020-07-23	4	4
6876	O	AS	2020-03-16	106	106
6877	O	AS	2020-03-26	77	77
6878	O	AS	2020-03-21	68	68
6879	O	AS	2020-07-21	2	2
6880	O	AS	2020-03-14	81	81
6881	O	AS	2020-07-13	0	0
6882	O	AS	2020-03-25	57	57
6883	O	AS	2020-07-18	1	1

Como Asturias es uniprovincial solamente sale Oviedo.

c)

```
library(ggplot2)
```

```
library(lubridate)
```

```
#Ordenamos los datos
```

```
datos_asturias <- datos_asturias[ order ( as.Date (datos_asturias$ fecha, format = " % m /% d /% Y ")),]
```

	provincia_iso	Comunidad autónoma	fecha	num_casos
6945	O	AS	2020-01-31	0
6929	O	AS	2020-02-01	0
6928	O	AS	2020-02-02	0
6899	O	AS	2020-02-03	0
6887	O	AS	2020-02-04	0
6922	O	AS	2020-02-05	0
6894	O	AS	2020-02-06	0
6904	O	AS	2020-02-07	0
6903	O	AS	2020-02-08	0
6942	O	AS	2020-02-09	0

```
#Hacemos un attach para acceder mejor a los datos
```

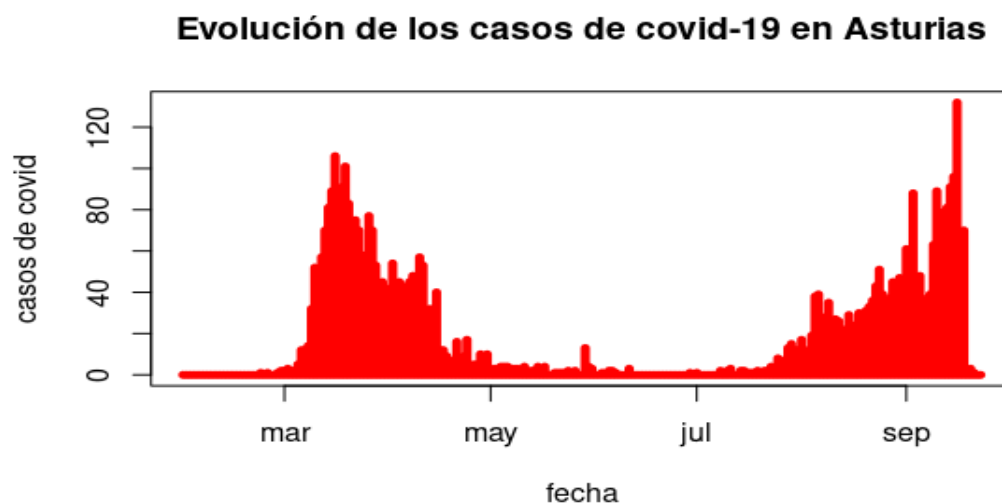
```
attach(datos_asturias)
```

```
x <- fecha
```

```
y <- num_casos
```

```
plot(x,y, type = "h",col = "red", lwd = 5,main = "Evolución de los casos de covid-19 en Asturias",  
xlab = "fecha",  
ylab = "casos de covid")
```

He utilizado un gráfico de barras porque recoge mejor la evolución de los casos.



d)

#Podemos reutilizar el gráfico del apartado anterior ya que tenemos que añadir más variables.

#Las cuales las definiremos a continuación

```
y1 <- num_casos_prueba_pcr
```

```
y2 <- num_casos_prueba_test_ac
```

```
y3 <- num_casos_prueba_otras
```

```
y4 <- num_casos_prueba_desconocida
```

```
plot(x,y1, type = "h",col = "red", lwd = 2,main = "Evolución de los casos de Covid-19 en Asturias",
```

```
      xlab = "fecha",ylab = "casos de covid")
```

```
#Añadimos la segunda variable
```

```
lines(x,y2, type = "h", col = "blue", lwd = 2)
```

```
#La tercera variable
```

```
lines(x,y3, type = "h", col = "green", lwd = 4)
```

```
#La cuarta variable
```

```
lines(x,y4, type = "h", col = "yellow", lwd = 2)
```

```
#Dibujamos la leyenda
```

```
windows(width = 3, height = 3)
```

```
# Cambiamos los márgenes del gráfico (el primero es el margen inferior)
```

```
par(mar = c(1, 1.5, 1.5, 1.5))
```

```
legend(x = "bottom",
```

```
      inset = c(0, -0.5), # Tendrás que ajustar el segundo valor
```

```
      # dependiendo del tamaño de tu gráfico
```

```
      legend = c("num_casos_prueba_pcr","num_casos_prueba_test_ac",
```

```
                "num_casos_prueba_otras","num_casos_prueba_desconocida"),
```

```
      lty = c(1, 1,1,1,1),
```

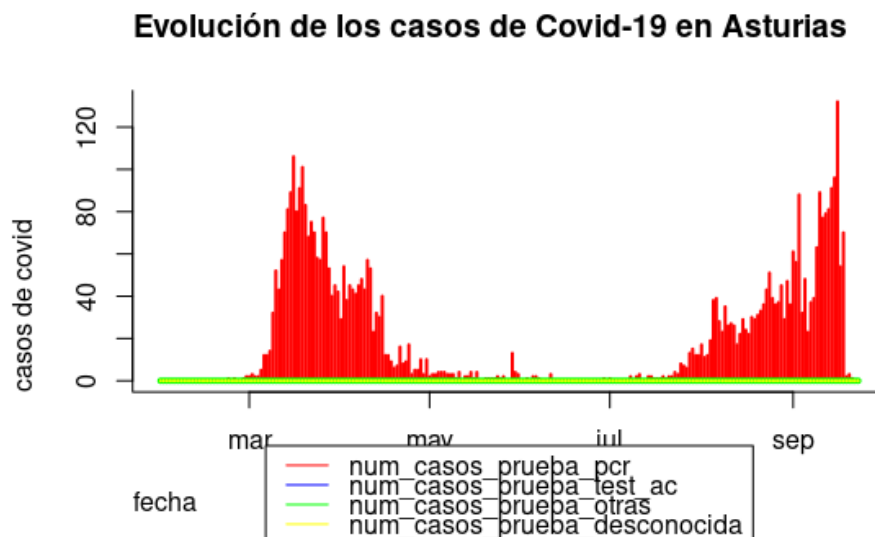
```
      col = c("red","blue","green","yellow"),
```

```
      xpd = TRUE)# Tienes que especificar este parámetro gráfico
```

```
      # para poner la leyenda fuera del gráfico
```

```
# Volvemos a los parámetros gráficos por defecto
```

```
on.exit(par(opar))
```



3. Pregunta 3 (30 puntos): Una tienda vende una serie de artículos cuyo precio y volumen de ventas están recogidos en el archivo “articulo.xlsx”. Al llegar el periodo de rebajas se plantea aplicar un descuento para lo cual decide clasificar los artículos en tres tipos A, B y C. Los artículos se clasifican en función del ingreso bruto que proporcionan ($\text{ingresobruto} = \text{PVP} \times \text{cantidad}$), de forma que el 20% de los que más aportan se clasifican de tipo A, el 30% siguiente de Clase B y el resto de clase C.

El porcentaje de rebaja que aplicará en función de esta clasificación se encuentra en el archivo “descuento_aplicar.txt”.

Se pide:

a) Crear dos data frame de nombres, uno de nombre articulo leyendo la información de “articulo.xlsx” y el otro de nombre descuento que guarde la información de “descuento_aplicar.txt”. (5 pts)

b) Crear una variable llamada tipo que clasifique los artículos en los tipos A, B y C en la forma indicada y añadirlo al data frame articulo y calcula, en una única sentencia, el total de ingresos brutos por cada tipo de artículo.

c) Unir los dos data frame, articulo y descuento, de forma adecuada, para crear el data frame clientes. Calcular la variable nuevo_pvp = pvp – cantidad a descontar. Suponiendo que el volumen de ventas se mantiene constante dar una estimación del porcentaje de decremento en los ingresos brutos al aplicar los descuentos.



a)

Usamos la librería readr y readxl para facilitar la lectura de los ficheros uno en csv y el otro en xlxs.
library(readr)

```
descuento <- read_delim("~/Escritorio/MASTER/4.R/Datos/descuento_aplicar.txt",
                        delim = "\t", escape_double = FALSE,
                        trim_ws = TRUE, show_col_types = FALSE)
```

```
library(readxl)
```

```
articulo <- read_excel("~/Escritorio/MASTER/4.R/Datos/articulo.xlsx")
```

Data		
articulo	58 obs. of 3 variables	
descuento	3 obs. of 2 variables	

b)

#Primero de todo añadimos una nueva columna con el ingreso bruto por articulo

```
articulo$ingresobruto <- articulo$PVP*articulo$CANTIDAD
```

	PRODUCTO	PVP	CANTIDAD	ingresobruto
1	1	5000	6000	30000000
2	2	50	7600	380000
3	3	860	2400	2064000
4	4	1475	1500	2212500
5	5	10800	4050	43740000
6	6	4500	1400	6300000
7	7	245	2400	588000
8	8	505	0	0

```
# Ordenando por la columna ingresobruto
```

```
df_ordenado <- articulo[order(-articulo$ingresobru), ]
```

Para conseguir los cuantiles he tenido que utilizar la librería stats, ya que permite cualquier operación estadística y no hay ninguna parecida en librería base.

```
#Calculamos los cuantiles agrupandolos por ingresobru acumulado
```

```
deciles <- quantile(articulo$ingresobru, probs = seq(.1,.9,by = .1))
```

```
deciles
```

```
      10%      20%      30%      40%      50%      60%      70%      80%      90%
411360  559648  643300  1017400  1797000  2142360  3020952  6108000  17367800
```

```
#Para el 80% acumulado sería 6108000, para el 70% y 50%, 3020952 y 1797000 consecutivamente
#y el resto el 50%
```

```
check <- function(ingresobru) {
```

```
  if (ingresobru > 6108000) {
```

```
    result <- "A"
```

```
  }
```

```
  else if (ingresobru >= 1797000 & ingresobru < 6108000) {
```

```
    result <- "B"
```

```
  }
```

```
  else {
```

```
    result <- "C"
```

```
  }
```

```
  return(result)
```

```
}
```

```
new_colum = sapply(df_ordenado$ingresobru, check)
```

```
articulo <- cbind(df_ordenado, tipo = c(new_colum))
```

	PRODUCTO	PVP	CANTIDAD	ingresobru	tipo
1	5	10800	4050	43740000	A
2	29	6600	5800	38280000	A
3	1	5000	6000	30000000	A
4	37	9700	3030	29391000	A
5	28	11500	2000	23000000	A
6	15	5400	4000	21600000	A
7	24	7700	2020	15554000	A
8	13	1880	4140	7783200	A
9	44	1600	4790	7664000	A
10	55	3500	2000	7000000	A
11	23	1700	4000	6800000	A
12	6	4500	1400	6300000	A
13	49	1940	3000	5820000	B
14	30	3000	1600	4800000	B
15	16	2310	1900	4389000	B

c)

2.

#Primero de todo unimos los dos dataframes

clientes <- merge(articulo,descuento,by = "tipo")

	tipo	PRODUCTO	PVP	CANTIDAD	ingresobrut	descuento
1	A	5	10800	4050	43740000	10
2	A	29	6600	5800	38280000	10
3	A	1	5000	6000	30000000	10
4	A	37	9700	3030	29391000	10
5	A	28	11500	2000	23000000	10
6	A	15	5400	4000	21600000	10
7	A	24	7700	2020	15554000	10
8	A	13	1880	4140	7783200	10
9	A	44	1600	4790	7664000	10
10	A	55	3500	2000	7000000	10
11	A	23	1700	4000	6800000	10
12	A	6	4500	1400	6300000	10

#creamos la nueva columna con el nuevo precio

clientes\$nuevoPVP <- clientes\$PVP-clientes\$descuento

#calculamos el nuevo ingreso

	tipo	PRODUCTO	PVP	CANTIDAD	ingresobrut	descuento	nuevoPVP
1	A	5	10800	4050	43740000	10	10790
2	A	29	6600	5800	38280000	10	6590
3	A	1	5000	6000	30000000	10	4990
4	A	37	9700	3030	29391000	10	9690
5	A	28	11500	2000	23000000	10	11490
6	A	15	5400	4000	21600000	10	5390
7	A	24	7700	2020	15554000	10	7690
8	A	13	1880	4140	7783200	10	1870
9	A	44	1600	4790	7664000	10	1590
10	A	55	3500	2000	7000000	10	3490
11	A	23	1700	4000	6800000	10	1690
12	A	6	4500	1400	6300000	10	1190

clientes\$ingresobrut2 <- clientes\$nuevoPVP*clientes\$CANTIDAD

ingresobrut	descuento	nuevoPVP	ingresobrut2
43740000	10	10790	43699500
38280000	10	6590	38222000
30000000	10	4990	29940000
29391000	10	9690	29360700
23000000	10	11490	22980000
21600000	10	5390	21560000
15554000	10	7690	15533800
7783200	10	1870	7741800
7664000	10	1590	7616100
7000000	10	3490	6980000
6800000	10	1690	6760000

4. Pregunta 4 (30 puntos): La siguiente tabla representa puntuaciones de sensación de ardor para 16 sujetos en un estudio para probar un nuevo hidrogel. La primera columna da el número del sujeto. Las siguientes columnas dan la puntuación de sensación de ardor (en una escala de 1 a 4) para semanas 1 (S1) a 7 (S7). (La matriz de datos se encuentra en “matriz.R”, no se puede abrir el fichero y copiar su contenido en el script)

Propondremos una forma interesante de mostrar estos datos.

1.- Para la semana S_7 , calcule el vector $(f_1, 1 - f_1, f_2, 1 - f_2, f_3, 1 - f_3, f_4, 1 - f_4)$ donde f_i es la frecuencia (relativa) de la modalidad $i \in \{1,2,3,4\}$ observada en la semana S_7 sobre los 16 sujetos. (Sugerencia: use las funciones `tabulate()`, `cbind()`, `t()` y `as.vector()`).

Ahora, use la función `apply()` para hacer el mismo cálculo para todas las demás semanas.

Almacene el resultado en una matriz.

2.- Utilice la función `barplot()` y el argumento `col = c("black", "white")` en esta matriz. El gráfico que se obtiene ofrece una descripción general de la evolución de la Sensación de ardor con el tiempo.

3.- Cambie el gráfico anterior para que las barras que representan las frecuencias estén en rojo. Los números de las semanas deben estar en azul y en la parte superior del gráfico en lugar del fondo. Los números de modalidad deben estar a la izquierda, en azul. Agrega un título al gráfico

1.

```
matriz <- source("~/Escritorio/MASTER/4.R/Datos/matriz.R")
```

Data

matriz	List of 2	
--------	-----------	--

Al cargar el archivo `matriz.R` podemos observar que es una lista de dos elementos.

	matriz	list [2]	List of length 2
value		double [16 x 8]	1 2 3 4 5 6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
visible		logical [1]	TRUE

También podemos ver la misma información mediante la función `str`

```
#Para ver que tipo de objeto es matriz
```

```
str(matriz)
```

```
list of 2
 $ value : num [1:16, 1:8] 1 2 3 4 5 6 7 8 9 10 ...
 $ visible: logi TRUE
```

Por lo que primero de todo separamos ambos elementos de la lista y cogemos el que nos interesa que es `value`:

```
matriz$value
```

```
> matriz$value
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]    1    1    1    1    1    1    1    1
[2,]    2    1    1    1    1    1    1    2
[3,]    3    1    1    1    1    1    2    3
[4,]    4    1    1    1    1    1    3    4
[5,]    5    1    1    1    1    2    3    3
[6,]    6    1    1    1    1    1    1    1
[7,]    7    1    1    1    3    4    2    2
[8,]    8    1    1    1    1    1    1    1
[9,]    9    1    1    1    1    1    1    1
[10,]   10    1    1    1    1    1    1    4
[11,]   11    1    1    1    1    1    1    1
[12,]   12    1    1    1    1    1    1    1
[13,]   13    1    2    1    3    2    3    4
[14,]   14    1    1    1    2    2    4    4
[15,]   15    1    1    1    1    1    1    1
[16,]   16    1    1    1    1    1    1    1
```

Podemos observar que `matriz$value` tiene como primera columna el vector que corresponde a los individuos, pero esto se corregirá más adelante al aplicar la función de a continuación junto con `apply()`

Calculamos la función de frecuencias:

```
f <- function(x) {
  fi = tabulate(x, nbins = 4) / length(x)
  return(c(
    fi[1],
    1 - fi[1],
    fi[2],
    1 - fi[2],
    fi[3],
    1 - fi[3],
    fi[4],
    1 - fi[4]
  ))
}
```

Cálculo del vector S7

```
S7 <- as.vector(f(matriz$value[,8]))
S7
```

```
> S7
```

```
[1] 0.500 0.500 0.125 0.875 0.125 0.875 0.250 0.750
```

Cálculo completo de la matriz de frecuencias utilizando la función `apply()` con el cual cogemos la segunda columna hasta la 8

```
Z = apply(matriz$value[,2:8],2,f)
```

	V1	V2	V3	V4	V5	V6	V7
1	1	0.9375	1	0.8125	0.7500	0.6250	0.500
2	0	0.0625	0	0.1875	0.2500	0.3750	0.500
3	0	0.0625	0	0.0625	0.1875	0.1250	0.125
4	1	0.9375	1	0.9375	0.8125	0.8750	0.875
5	0	0.0000	0	0.1250	0.0000	0.1875	0.125
6	1	1.0000	1	0.8750	1.0000	0.8125	0.875
7	0	0.0000	0	0.0000	0.0625	0.0625	0.250
8	1	1.0000	1	1.0000	0.9375	0.9375	0.750

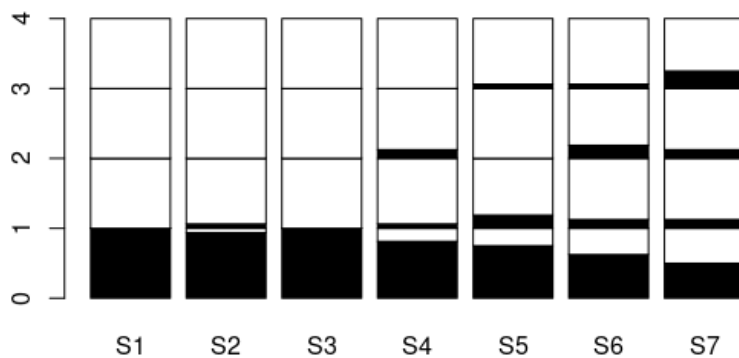
Ahora ponemos el nombre a las columnas gracias a :

```
nombres_columnas <- c("S1", "S2", "S3", "S4", "S5", "S6", "S7")
colnames(Z) <- nombres_columnas
```

	S1	S2	S3	S4	S5	S6	S7
1	1	0.9375	1	0.8125	0.7500	0.6250	0.500
2	0	0.0625	0	0.1875	0.2500	0.3750	0.500
3	0	0.0625	0	0.0625	0.1875	0.1250	0.125
4	1	0.9375	1	0.9375	0.8125	0.8750	0.875
5	0	0.0000	0	0.1250	0.0000	0.1875	0.125
6	1	1.0000	1	0.8750	1.0000	0.8125	0.875
7	0	0.0000	0	0.0000	0.0625	0.0625	0.250
8	1	1.0000	1	1.0000	0.9375	0.9375	0.750

2.

```
barplot(Z,col = c ("black", "white"))
```



3.

```
par(bty = "u",adj=0)
barplot(Z, col = c ("red", "white"),axes = FALSE,xaxt = "n")
axis(3,col.axis = "blue",at = c(0.8,1.9,3.1,4.3,5.5,6.7,7.9), labels=c("S1", "S2", "S3",
"S4", "S5", "S6", "S7"),
tck = 0,lwd = 0)
axis(2,col.axis = "blue")
title(main = "Frecuencia sensación de ardor",line = 3)
```

Frecuencia sensación de ardor

