Práctica 4: Acceso a ficheros externos

A continuación proponemos distintas tareas relacionadas con la lectura de datos desde fichero externos de tipo texto y su depuración dentro de R. Crea un fichero script con el código que permita resolverlas, incluyendo en el mismo los comentarios que estimes oportunos. Este script deberás enviarlo a través de PRADO siguiendo las instrucciones proporcionadas en la tarea allí creada.

1. En el fichero *census.csv* disponible en PRADO se recogen datos relativos a una muestra de $n = 500$ individuos para distintas variables. Descarga el fichero desde PRADO en el directorio de trabajo y resuelve las siguientes tareas:

   a) Importa los datos en R dentro de un data frame con nombre `censo` usando la función `read.table` o `read.csv` (la que resulte más adecuada a este tipo de datos). Hazlo de modo que las columnas `cellsource`, `travel`, `getlunch` y `gender` se almacenen como tipo factor.

   b) Comprueba con una sola sentencia el tipo de datos de las columnas del data frame.

   c) Observa que en el data frame hay varios valores perdidos (`NA`). Cuenta cuántos valores perdidos hay en cada columna. [Sugerencia: Evalúa `lapply(censo,is.na)` y observa que te devuelve una lista con un vector lógico para cada columna, indicando con `TRUE` en qué fila hay un dato perdido. Con esta resultado ya tienes casi la solución al problema.]

   d) Cuenta cuántas filas del data frame están completas, esto es, no tienen ningún datos perdido (`NA`), utilizando la función `complete.cases`.[sugerencia: Comienza evaluando `complete.cases(censo)` y observa que como resultado te devuelve un vector lógico indicando con `TRUE` las filas completas (sin ningún `NA`). Con este resultado ya casi lo tienes.]

   e) Crea un nuevo data frame con nombre `censo2` copiando en él tan solo las filas de `censo` que estén completas. Resuelve esta tarea de dos formas, primero con la misma función `complete.cases` que usaste antes, y después usando la función `na.omit` (consulta la ayuda de esta última función para ver qué hace y cómo se usa).

   f) Escribe el contenido del data frame `censo2` en un fichero de texto con nombre *censo2.txt* con la función `write.table`. En el fichero los nombres de las columnas deben aparecer en la primera fila, los valores deben estar separados por tabulaciones (`sep='\t'`) y no debe contener nombres para las filas.

   g) Importa los datos del fichero *censo2.txt* que has creado antes en un data frame con nombre `censo3`. Este data frame debe coincidir en estructura y composición con `censo2`, compruébalo.

```
> censo<-read.csv('census.csv',header=TRUE,as.is=NA)
> lapply(censo,class)

$cellsource
[1] "factor"

$rightfoot
[1] "integer"

$travel
[1] "factor"

$getlunch
[1] "factor"

$height
[1] "integer"

$gender
[1] "factor"

$age
[1] "integer"

$year
[1] "integer"

$armspan
[1] "integer"

$cellcost
[1] "integer"

> # lapply(censo,is.na)
> lapply(lapply(censo,is.na),sum)

$cellsource
[1] 167

$rightfoot
[1] 23
```

```
$travel
[1] 0

$getlunch
[1] 3

$height
[1] 21

$gender
[1] 0

$age
[1] 1

$year
[1] 0

$armspan
[1] 36

$cellcost
[1] 172
```

```
> # complete.cases(censo)
> censo2<-censo[complete.cases(censo),]
> censo2<-na.omit(censo)
> write.table(censo2,file='censo2.txt',row.names=FALSE,sep='\t')
> censo3<-read.table('censo2.txt',header=TRUE,as.is=NA)
> str(censo3)
```

```
'data.frame': 282 obs. of  10 variables:
 $ cellsource: Factor w/ 4 levels "job","other",..: 4 3 3 4 4 3 3 4 3 2 ...
 $ rightfoot : int  20 25 21 20 23 19 23 35 22 30 ...
 $ travel    : Factor w/ 6 levels "bike","bus","motor",..: 6 4 3 6 4 3 3 3 3 6 ...
 $ getlunch  : Factor w/ 6 levels "dairy","friend",..: 3 2 3 3 3 3 3 6 3 6 ...
 $ height    : int  152 153 137 115 165 137 164 150 150 123 ...
 $ gender    : Factor w/ 2 levels "female","male": 2 1 2 2 1 1 1 1 1 2 ...
 $ age       : int  12 11 10 9 14 11 12 15 12 14 ...
 $ year      : int  7 6 6 5 10 7 8 11 8 9 ...
 $ armspan   : int  150 152 132 130 160 50 164 100 152 23 ...
 $ cellcost  : int  30 50 55 60 20 50 10 20 10 0 ...
```

```
> str(censo2)

'data.frame': 282 obs. of  10 variables:
 $ cellsource: Factor w/ 4 levels "job","other",..: 4 3 3 4 4 3 3 4 3 2 ...
 $ rightfoot : int  20 25 21 20 23 19 23 35 22 30 ...
 $ travel    : Factor w/ 6 levels "bike","bus","motor",..: 6 4 3 6 4 3 3 3 3 6 ...
 $ getlunch  : Factor w/ 6 levels "dairy","friend",..: 3 2 3 3 3 3 3 6 3 6 ...
 $ height    : int  152 153 137 115 165 137 164 150 150 123 ...
 $ gender    : Factor w/ 2 levels "female","male": 2 1 2 2 1 1 1 1 1 2 ...
 $ age       : int  12 11 10 9 14 11 12 15 12 14 ...
 $ year      : int  7 6 6 5 10 7 8 11 8 9 ...
 $ armspan   : int  150 152 132 130 160 50 164 100 152 23 ...
 $ cellcost  : int  30 50 55 60 20 50 10 20 10 0 ...
 - attr(*, "na.action")= 'omit' Named int [1:218] 40 42 46 47 49 52 53 54 55 56 ..
  ..- attr(*, "names")= chr [1:218] "40" "42" "46" "47" ...
```

2. Crea una matriz con nombre `matriz` con 10 filas y 5 columnas cuyos elementos sean valores aleatorios desde una distribución normal estándar.

   a) Asigna nombres a las columnas de la matriz del tipo `col1`,...`col5`.

   b) Imprime la matriz anterior en un fichero de texto *matriz.txt* usando la función `write` y separando los valores por comas. En la primera fila deben imprimirse los nombres de las columnas.

   c) Lee el fichero que has escrito y almacena su información en el espacio de trabajo en forma de data frame. Ten en cuenta que los nombres de las columnas del data frame deben tomarse de la primera fila del fichero.

```
> nomb<-paste0('col',1:5)
> matriz<-matrix(rnorm(50),10,5,dimnames = list(NULL,nomb))
> write(nomb,file='matriz.txt',ncol=5,sep=',')
> write(t(matriz),file='matriz.txt',ncol=5,sep=',',append=TRUE)
> matriz2<-read.csv('matriz.txt',header=TRUE)
> str(matriz2)

'data.frame': 10 obs. of  5 variables:
 $ col1: num  0.19639 -0.6276 -0.00752 -0.74411 -0.00076 ...
 $ col2: num  -0.904 0.638 0.54 0.356 0.435 ...
 $ col3: num  -0.501 2.625 -1.064 -1.078 -0.399 ...
 $ col4: num  -0.1866 -0.1425 -0.0634 -0.4406 -0.1649 ...
 $ col5: num  -0.9551 -0.2617 0.0892 -0.6625 -0.1988 ...
```

3. En el fichero *Olympics100m.csv* disponible en PRADO se recogen datos de una muestra de $n = 50$ atletas. Descarga el fichero desde PRADO en el directorio de trabajo y resuelve las siguientes tareas:

   a) Importa los datos en R dentro de un data frame con nombre `olimpics`. Hazlo de modo que las columnas que correspondan a factores se almacenen con ese tipo.

   b) Comprueba con una función adecuada si hay algún dato perdido, contando cuántos hay en dicho caso.

   c) Calcula un resumen descriptivo con `summary` del data frame. Almacena el valor devuelto en un objeto `resumen`, comprueba que se trata de una matriz de tipo carácter. Después imprime dicho objeto en un fichero de tipo texto (*resumen.txt*). Hazlo de forma que puedas leerlo después con `read.table` y cargarlo en formato de data frame.

   d) Calcula un resumen descriptivo ahora solo de la variable `TIME` para los distintos niveles del factor `Gender`. Hazlo de modo que el objeto resultante sea un único data frame. Después imprime dicho data frame en un fichero de tipo texto (*resumen2.txt*). Hazlo de forma que puedas leerlo después con `read.csv`, resultando un data frame conteniendo columnas con los resúmenes descriptivos (mínimo, primer cuartil, mediana, etc.) para cada grupo en formato numérico.

```
> olympics<-read.csv('Olympics100m.csv',header=TRUE,as.is=c(2))
> lapply(olympics,class)

$YEAR
[1] "integer"

$NAME
[1] "character"

$TIME
[1] "numeric"

$Country
[1] "factor"

$Gender
[1] "factor"

> lapply(lapply(olympics,is.na),sum)
```

```
$YEAR
[1] 0

$NAME
[1] 0

$TIME
[1] 0

$Country
[1] 0

$Gender
[1] 0

> resumen<-summary(olympics)
> is.matrix(resumen)

[1] TRUE

> typeof(resumen)

[1] "character"

> class(resumen) # es un matriz de una clase especial

[1] "table"

> write.table(resumen,file='resumen.txt',row.names=FALSE)
> resumen.leido<-read.table('resumen.txt')
> resumen2<-aggregate(olympics$TIME,by=list(olympics$Gender),summary)
> write.table(resumen2,file='resumen2.txt',sep=',',row.names=FALSE)
> resumen2.leido<-read.csv(file='resumen2.txt',header=T)
> str(resumen2.leido) # observa que las últimas columnas son numéricas

'data.frame': 2 obs. of  7 variables:
 $ Group.1  : chr  "female" "male"
 $ x.Min.   : num  10.54 9.63
 $ x.1st.Qu.: num  10.93 9.95
 $ x.Median : num  11.1 10.2
 $ x.Mean   : num  11.2 10.3
 $ x.3rd.Qu.: num  11.5 10.8
 $ x.Max.   : num  12.2 12
```