

# Tarea 2

Juan Cantero Jimenez

15 de noviembre, 2021

## Ejercicio 1

¿Cómo seleccionaríamos los casos impares de un banco de datos con la función `subset()`? ¿Y los casos que corresponden a las filas numeradas con un múltiplo de cinco?

```
data <- read.csv("separaciones.csv",header = TRUE)
data <- data[,-1]
h <- subset(data, 1:nrow(data)%%2 == 1)
#Se seleccionan los casos impares haciendo uso de la función
#subset y de una expresión logica, 1:nrow(data)%%2 == 1, que
# divide las distintas posiciones entre 2, se queda con el
#resto del cociente y devuelve True si este es 1.
h2 <- subset(data, 1:nrow(data)%%5 == 0)
#Se seleccionan los casos que son multiplos de 5 haciendo u
#so de la función subset de una expresión logica similar a la
# anterior, 1:nrow(data)%%5 == 0, que divide las distintas
#posiciones entre 5, se queda con el resto del cociente y
#devuelve True si es 0.
```

## Ejercicio 2

Generar una muestra aleatoria de tamaño 100 de una normal de media 165 y desviación típica 5 (imaginar que corresponden a la altura de los hombres) y otra muestra de tamaño 40 de una normal de media 160 y desviación típica

10 (ahora de las mujeres). Crear un vector con todos los datos y un factor que identifique los individuos de uno y otro grupo.

```
n_h <- 100
#se define la variable n_h que representa el número de hombre
n_m <- 40
#se define la variable n_m que representa el número de mujeres
muestra <- c(rnorm(n_h, 165, 5), rnorm(n_m, 160, 10))
#se generan n_h datos que siguen una N(165, 5) y
# n_m datos que siguen una N(160, 10), que se concatenan y se
# asignan al vector muestra.
grupos <- as.factor(c(rep("hombre", n_h), rep("mujer", n_m)))
#se genera un factor que recoge los distintos
#grupos de la muestra, para esto se crea un vector con-
#catenando n_h veces la palabra hombre y n_m veces la palab-
#a mujer, y este se pasa a factor.
```

## Ejercicio 3

Construir un **data.frame** con los datos del ejercicio anterior y nombrarlo **datos.alturas**. Exportar dicho **data.frame** a un fichero externo de nombre **datos-alturas.txt** utilizando el comando **write.table** y comprobar como queda volviéndolo a importar con **read.table** (todo ello sin salir de R). Construir un nuevo vector con la transformación logarítmica de las mediciones e incluirlo en el banco de datos

```
datos.alturas <- data.frame(alturas=muestra,
                           grupos=grupos)
#se crea un data.frame con las variables alturas, que
#poseen los datos del vector muestra,
#y grupos que posee los datos
# del factor grupos.
write.table(datos.alturas, "datosalt.txt")
#se convierte el dataframe creado al archivo .txt, datosalt.txt,
#con la función write.table.
datos.alturabis<-read.table("datosalt.txt")
```

```
#se leen dichos datos con la función read.table
datos.alturas <- cbind(datos.alturas,
logaritmo=log(datos.alturas$alturas))
#se añade la variable logaritmos
#al dataframe datos.alturas haciendo uso de la función
# cbind(), estas variable contienen el resultado de apli
#car la función log() a datos.alturas$alturas
```

## Ejercicio 4

Utilizar la ayuda para averiguar como obtener estadísticos básicos. Obtener la mediana, la desviación típica, el máximo, el coeficiente de asimetría y curtosis de los vectores del ejercicio anterior, tanto de todos los datos como por grupos.

```
#Comentarios generales, se calcula el estadístico
#pedido en cada caso del vector datos.alturas$alturas
# y datos.alturas$logaritmo haciendo uso de la función de
#finida para cada uno de ellos. Para el calculo de
#estos estadísticos
# por grupos, se hace uso de la función tapply
#mediana
median(datos.alturas$alturas)

## [1] 163.7133

median(datos.alturas$logaritmo)

## [1] 5.098117

tapply(datos.alturas$alturas, datos.alturas$grupos, median)

## hombre mujer
## 165.4270 159.4257

tapply(datos.alturas$logaritmo, datos.alturas$grupos, median)
```

```

##      hombre      mujer
## 5.108530 5.071572

#desviación típica
sd(datos.alturas$alturas)

## [1] 7.491404

sd(datos.alturas$logaritmo)

## [1] 0.04642624

tapply(datos.alturas$alturas, datos.alturas$grupos, sd)

##      hombre      mujer
## 5.557548 9.645704

tapply(datos.alturas$logaritmo, datos.alturas$grupos, sd)

##      hombre      mujer
## 0.03361510 0.06048941

#máximo
max(datos.alturas$alturas)

## [1] 181.8618

max(datos.alturas$logaritmo)

## [1] 5.203247

tapply(datos.alturas$alturas, datos.alturas$grupos, max)

##      hombre      mujer
## 178.2781 181.8618

tapply(datos.alturas$logaritmo, datos.alturas$grupos, max)

##      hombre      mujer
## 5.183345 5.203247

```

```

#coeficiente de asimetria

e1071::skewness(datos.alturas$alturas)
## [1] -0.4994281

e1071::skewness(datos.alturas$logaritmo)
## [1] -0.6393779

tapply(datos.alturas$alturas,
datos.alturas$grupos,
e1071::skewness)

##      hombre      mujer
## 0.0116007 0.1554944

tapply(datos.alturas$logaritmo,
datos.alturas$grupos,
e1071::skewness)

##      hombre      mujer
## -0.05732976 0.04064294

#curtosis

e1071::kurtosis(datos.alturas$alturas)
## [1] 0.2191883

e1071::kurtosis(datos.alturas$logaritmo)
## [1] 0.4307139

tapply(datos.alturas$alturas,
datos.alturas$grupos,
e1071::kurtosis)

##      hombre      mujer
## -0.6544841 -0.7319618

tapply(datos.alturas$logaritmo,
datos.alturas$grupos,
e1071::kurtosis)

##      hombre      mujer
## -0.6470613 -0.7980252

```

## Ejercicio 5

Importa a **R** el fichero ambiente.sav, que tiene un formato SPSS. Observa el tipo de las variables que lo componen. Investiga el comando by. Utilízalo para calcular la media de la variable PH en cada provincia de la Comunitat Valenciana.

```
dataset <- haven::read_sav("ambiente.sav")
#Se importa el dataset ambiente.sav haciendo uso de la función
# read_sav alojada en el paquete haven
medias_ph <- by(dataset$PH, dataset$PROVIN, mean)
#Se calculan las medias de la variable PH por provi
#ncias hacienod uso de la función by()
medias_ph

## dataset$PROVIN: 1
## [1] 5.90558
## -----
## dataset$PROVIN: 2
## [1] 5.968839
## -----
## dataset$PROVIN: 3
## [1] 5.894974
```

## Ejercicio 6

Transforma el banco de datos de la tarea anterior en un array donde cada dimensión simbolice una variable categórica. Con dicho array:

1. Calcula la media por año y provincia de la variable estudiada.
2. Calcula la varianza por provincia.
3. Calcula los 3 cuartiles principales por año.
4. Representa gráficamente la evolución temporal media por provincia de la variable estudiada.

```

data_array <- as.array(rep(0,nrow(data)))
#Se crea un array vacio con un tamaño igual al número de o
#bservaciones del dataframe data.
dim(data_array) <- unlist(lapply(data, function(x){
  length(unique(x))
}))[1:ncol(data)-1]
#Se definen las dimensiones de dicho array, para esto se hace
#uso de la función dim() que permite definir
#las dimensiones del
#array mediante un vector. Este vector
#se creará mediante un lapply que
#recorre las variables del dataset, y
#devuelve la longitud del vector
# de datos únicos de cada columna.
dimnames(data_array) <- list(Provincias=unique(as.character(data$Provincias)),
Tipo.de.disolución.matrimonial=unique(data$Tipo.de.disolución.matrimonial)
,Periodo=unique(data$Periodo))
#Se definen los nombres
#de las dimensiones del array data_array, mediante la funció
#n dimnames, que toma una lista de vectores
#como valor de entrada para
#fijar el nombre de las dimensiones.

for (x in 1:nrow(data)){
  data_array[as.character(data[x,1]),
  as.character(data[x,2]),
  as.character(data[x,3])] <- data[x,4]
}
#Se rellena el array mediante un
#bucle for que recorre las distintas
# observaciones de data, extrayendo el
#valor de cada variable. El valor de
#las tres primeras variables,
#Provincias, Tipo de disolución
#y Periodo serán usados para definir la posición
#de la variable Total en el array,
#correspondiente a cada observación.

```

```

#1
media_perodo <- sapply(dimnames(data_array)$Periodo,
function(x,y){
  mean(y[, "Total", x], na.rm=TRUE)
}, y=data_array)
#Se obtiene la media agrupando los
#datos por periodo, para esto
#se hace uso de la función sapply
#que recorre un vector con los distintos
#periodos presentes en el dataset, y gracias a este valor se
#extrae los valores necesarios del array y se realiza su media

media_provincia <- sapply(dimnames(data_array)$Provincia,
function(x,y){
  mean(y[x, "Total", ], na.rm=TRUE)
}, y=data_array)
#Equivalente a lo descrito en la línea anterior
#2
varianza_provincia <- sapply(dimnames(data_array)$Provincia,
function(x,y){
  var(y[x, "Total", ], na.rm=TRUE)
}, y=data_array)
#Equivalente a lo realizado en el apartado anterior
#3
cuartiles_perodo <- sapply(dimnames(data_array)$Periodo,
function(x,y){
  quantile(y[, "Total", x], na.rm=TRUE, c(0.25, 0.50, 0.75))
}, y=data_array)
#Equivalente a lo realizado en el apartado anterior
#4
for_plot <- as.data.frame.table(data_array[, "Total", ])
#Se genera un data.frame con el total de
#separaciones por periodo y provincia.

par(cex=0.3)
#Se ajusta el tamaño de la fuente, esto es conveniente

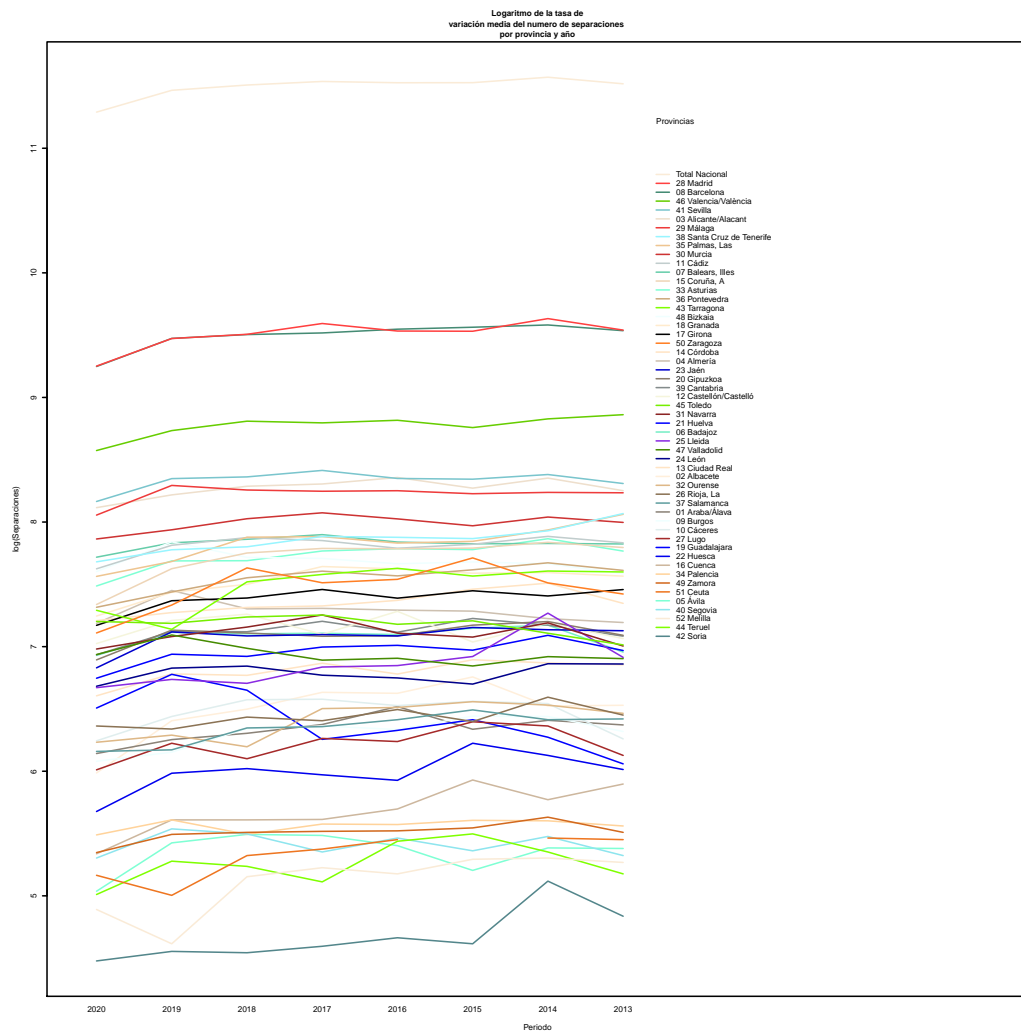
```



```

#debido a que la leyenda posee mucha información.
interaction.plot(x.factor=for_plot$Periodo,
                 trace.factor=for_plot$Provincias,
                 response=log(for_plot$Freq),
                 legend = TRUE,
                 type="l",
                 col=colors()[2:length(unique(for_plot$Provincias))+1],
                 lty=1,
                 xpd=TRUE,
                 lwd=1,
                 xlab="Periodo",
                 ylab="log(Separaciones)",
                 trace.label = "Provincias")
title(main="Logaritmo de la tasa de
variación media del numero de separaciones
por provincia y año")

```



#Se crea un plot de la evolución temporal media por pro  
#vincia del total de separaciones, nulidades y divorcios por  
# año y por comunidad autónoma, para esto se hace uso de la  
#función `interaction.plot` que toma como valores de entrada,  
#`x.factor` que indica la variable que agrupa el eje de las x,  
#`trace.factor` que indica la otra variable de agrupación, y  
#finalmente `response` en el que se indican los datos a agrupar.

#Se ha decidido realizar el logaritmo  
#antes de calcular la variación

*#media para que todas las líneas se encontrarán en una #escala  
#similar*