

Programación con R. Práctica 4

Ejercicio 1

La librería **stats** está instalada por defecto y se invoca con la siguiente orden: **require(stats)** o se instala con **library(stats)**; dicha librería contiene algunos data.frames de ejemplo con los que poder trabajar, uno de ellos es **swiss**. Podemos ver su estructura y su contenido con **str(swiss)**; **head(swiss)**; vamos a coger los datos de uno de los campos (una sola variable) y vamos a ordenar todo de forma estándar, es decir, de menor a mayor por el método clásico:

```
x <- swiss$Education[1:25]
```

```
sort(x, method="sh", index.return = TRUE)
```

se puede sustituir "sh" por "shell" o por "s"

también podemos generar una secuencia aleatoria y realizar una ordenación rápida de los datos

```
x <- as.integer(rnorm(200, 5, 7))
```

```
sort(x, method="quick")
```

Donde se puede sustituir "quick" por "qu" o por "q". A veces es interesante realizar la ordenación solo de una parte de los datos, esto puede hacerse con: `sort(x, partial = 1:5)`, en este caso, devuelve el vector con las cinco primeras posiciones del vector ordenado pero el resto sin ordenar. En selección directa, corta el bucle cuando llega al cinco

Ejercicio 2. Calcular en **x**, 100 números aleatorios según una normal de media 42 y varianza 2.5. Dicha variable redondeada a entero representará el número de pie de un chico de 18 años. Calcular en **y**, 100 números aleatorios según una normal de media 177 y varianza 10. Dicha variable redondeada representa la estatura.

- Meter ambas variables en un data.frame denominado **datos**
- Calcular la media y la varianza de **x** e **y** usando **apply** sobre '**datos**'
- Determinar la covarianza y la correlación, representar mediante un gráfico
- Determinar la regresión lineal entre las variables y representar los resultados.

Ejercicio 3. Calcular una matriz de **k1** filas por **k2** columnas cuyas componentes sean números aleatorios según una distribución de Poisson de parámetro **Lambda**. Calcular la traspuesta de dicha matriz y comprobar los resultados usando la función **t** de R.

Observaciones. Los datos **k1**, **k2** y **Lambda** deben solicitarse por teclado al usuario del script generado. La implementación de todo el programa debe realizarse utilizando sentencias de iteración (**for**, **while**, **repeat**). La función **t** de R sirve para calcular la traspuesta de una matriz. El ejercicio debe comprobar que los resultados son los mismos que los que se obtendrían con esta función.