**Script:**

require(stats)

str(swiss); head(swiss);

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")

##Ejercicio 2. Calcular en x, 100 números aleatorios según una normal de media 42 y desviación típica de 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.

x <- as.integer(rnorm(100, 42, 2.5))

sort(x, method="quick")

pie <- x

y <- (as.integer(rnorm(100,177,(10^(1/2)))))

sort(y, method="quick")

estatura <- y

# a. Meter ambas variables en un data.frame denominado datos

datos <- data.frame(x, y)

# b. Calcular la media y la varianza de x e y usando apply sobre 'datos'

apply(datos,2,mean)

apply(datos,2,var)

medias<-lapply(datos,mean)

medias

# c. Determinar la covarianza y la correlación, representar mediante un gráfico

cov(datos)

cor(datos)

plot(x,y)

plot(cov(datos))

plot(cor(datos))

plot( cov( x,y), main = 'Covariance function', xlab = 'Estatura', ylab = 'Pie' )

plot( cor( x,y), main = 'Covariance function', xlab = 'Estatura', ylab = 'Pie' )

plot(cov(datos),cor(datos))

# d. Determinar la regresión lineal entre las variables y representar los resultados.

regresion<-lm(y~x)

regresion

abline(regresion)

#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.

preguntar <- function(){

k1 <- readline("What is the value of k1?")

k2 <- readline("What is the value of k2?")

lambda <- readline("What is the value of lambda?")

k1<-as.integer(k1)

k2 <-as.integer(k2)

lambda <-as.integer(lambda)

}

preguntar()

x<- rpois(k1\*k2,lambda)

matriz<-matrix(x, nrow = k1,ncol = k2)

matriz

matriz2<-matriz

matriz3<-matriz

for(j in 1:k2){

for(i in 1:k1){

matriz3[i,j]<- x[k1\*(j-1)+i]

}

}

matriz3

mattraspuesta<-matrix(0,k2,k1)

for(i in 1:k2) {

for(j in 1:k1){

mattraspuesta[i,j]<-matriz3[j,i]

}

}

mattraspuesta

**Capturas Consola Ejercicio 4:**



