Práctica 1: Vectores

A continuación proponemos distintas tareas, relacionadas con los conceptos estudiados en el Tema 2 relativos a vectores, y que deberás resolver escribiendo sentencias apropiadas en R en un fichero script[1].

Los ejercicios están pensados para realizarse usando objetos de tipo vector, operadores y funciones básicas del sistema base, y escribiendo sentencias lo más simples posible.

1. Crea un vector con nombre `x` que contenga una secuencia de números reales entre 1 y 10 con incrementos de 0.2. Con dicho vector realiza las siguientes tareas:

   a) Calcula su longitud y almacénala en un objeto con nombre `n`.

   b) Da nombres a cada uno de los elementos del vector del tipo `x_1`,...,`x_n`

   c) Calcula la media de `x` y almacénala en un objeto con nombre `mx`.

   d) Calcula cuántos elementos de `x` están por encima de `mx`.

   e) Calcula la posición que ocupa el elemento de `x` más próximo por encima de `mx`.

   f) Crea otro vector `y` con los primeros `n` números impares.

   g) Imprime los elementos `x` que ocupen las posiciones indicadas por los primeros 5 elementos de `y`.

```
> x<-seq(1,10,by=.2)
> n<-length(x)
> names(x)<-paste0('x_',1:n)
> mx<-mean(x);mx

[1] 5.5

> sum(x>mx) # length(x[x>mx])

[1] 23

> which(x==min(x[x>mx]))

x_24
  24

> y<-seq(1,by=2,length.out=n)
> x[y[1:5]]

x_1 x_3 x_5 x_7 x_9
1.0 1.4 1.8 2.2 2.6
```

[1] Este fichero, incluyendo en el mismo los comentarios que estimes oportunos, deberás enviarlo a través de PRADO siguiendo las instrucciones proporcionadas en la tarea allí creada.

2. Evaluar la siguiente función en una rejilla de valores equiespaciados en el intervalo $[-2, 2]$ con incremento 0.1:

$$f(x) = \begin{cases} 1 & \text{si } x < -1 \\ \log(x^2) & \text{si } -1 \leq x < 0 \\ \log(x^2+1) & \text{si } 0 \leq x < 1 \\ 2 & x \geq 1 \end{cases}$$

```
> x<-seq(-2,2,by=0.1)
> fx<-(x< -1)+(-1<=x & x<0)*log(x^2)+(0<=x & x<1)*log(x^2+1)+(x>=1)*2
> fx[is.nan(fx)]<-0 ## fx[x==0]<-0
> fx

 [1]  1.000000000  1.000000000  1.000000000  1.000000000  1.000000000  1.000000000
 [7]  1.000000000  1.000000000  1.000000000  1.000000000  0.000000000 -0.210721031
[13] -0.446287103 -0.713349888 -1.021651248 -1.386294361 -1.832581464 -2.407945609
[19] -3.218875825 -4.605170186  0.000000000  0.009950331  0.039220713  0.086177696
[25]  0.148420005  0.223143551  0.307484700  0.398776120  0.494696242  0.593326845
[31]  2.000000000  2.000000000  2.000000000  2.000000000  2.000000000  2.000000000
[37]  2.000000000  2.000000000  2.000000000  2.000000000  2.000000000
```

3. Crea un vector con nombre `x` que contenga 50 valores aleatorios de una distribución uniforme en el intervalo unidad usando la función `runif` (previamente fija la semilla de generación de números aleatorios escribiendo la sentencia `set.seed(1)`). A partir de dicho vector realiza las siguientes tareas:

   a) Calcula cuántos de sus elementos están en el intervalo $(0.25, 0.75)$.

   b) Calcula cuántos de sus elementos están por debajo de 0.1 o por encima de 0.9. Reemplaza dichos elementos por el valor `NA`. Después calcula su media.

   c) Partiendo del vector obtenido en el apartado anterior, reemplaza los valores `NA` por ceros. Después calcula su media y compara con la obtenida en el apartado anterior.

```
> set.seed(1)
> x<-runif(50)
> sum(x>0.25 & x< 0.75)

[1] 27

> sum(x<0.1 | x> 0.9)
```

```
[1] 7

> x2<-x
> x2[x<0.1 | x> 0.9]<-NA
> mean(x2,na.rm=TRUE)

[1] 0.5291079

> x2[is.na(x2)]<-0
> mean(x2)

[1] 0.4550328
```

4. Crea un vector con los 20 primeros términos de la progresión aritmética $a_n = a_1 + (n-1)d$ con $a_1 = 1$ y $d = 1.2$. A partir de él:

   a) Calcula la suma de sus elementos usando la función `sum` y comprueba que coincide con fórmula $n(a_1 + a_n)/2$, para $n = 20$.

   b) Calcula la (cuasi-)desviación típica usando la función `sd` y comprueba que coincide con $|d|\sqrt{\frac{n(n+1)}{12}}$.

   c) Calcula el producto de sus elementos usando la función `prod` y comprueba que coincide con $\prod_{k=0}^{n-1}(a_1 + kd) = d^n \frac{\Gamma(a_1/d + n)}{\Gamma(a_1/d)}$, donde Γ denota la función gamma (en R tienes esta función con el mismo nombre).

```
> a1<-1;d<-1.2;n<-20
> p20<-a1+(0:(n-1))*d
> sum(p20)

[1] 248

> 20*(p20[1]+p20[n])/2

[1] 248

> sd(p20)

[1] 7.099296
```

```
> abs(d)*sqrt(n*(n+1)/12)

[1] 7.099296

> prod(p20) # prod(a1+(0:(n-1))*d)

[1] 4.99804e+19

> d^n*gamma(a1/d+n)/gamma(a1/d)

[1] 4.99804e+19
```

5. Crea un vector `x` con elementos 2, 2, 8, 7, 6, 1 y 5. Después, escribiendo una única sentencia calcula las diferencias sucesivas entre sus elementos.

   Nota: En R existe una función que hace esto exactamente, se trata de la función `diff`. Resuelve este ejercicio sin usarla.

```
> x<-c(2,2,8,7,6,1,5)
> x[-1]-x[-length(x)]

[1]  0  6 -1 -1 -5  4
```

6. Crea un vector con nombre `ABE` con las letras del abecedario en mayúscula. Con dicho vector:

   a) Selecciona aleatoriamente 5 letras (usando la función `sample` con argumento `replace=FALSE`) y almacénalas en un vector con nombre `ABE.5`.

   b) Crea un vector (con nombre `PAL`) con 2 elementos consistentes en 2 "palabras" formadas colocando aleatoriamente las 5 letras anteriores sin repeticiones. Las palabras no tienen que estar en el diccionario.

```
> set.seed(1)
> ABE<-sort(c(LETTERS,'Ñ'))
> ABE.5<-sample(ABE,5,replace=FALSE)
> PAL<-c(paste(sample(ABE.5,replace=FALSE),collapse = ""),
+        paste(sample(ABE.5,replace=FALSE),collapse = ""))
```