**Estos ejercicios son muy básicos pero se deben controlar antes de hacer otros más complejos, ya que debemos conocer estos comandos para poder hacer estos.**

* **1ºCambio de variables**

Este es un ejercicio muy importante aunque sea muy sencillo, en nuestro año el concepto en el que se basa cayó en el primer parcial.

Intercambiar el valor de la variable a y b.

Datos: a=5 b=8

Solución: La clave del ejercicio está en ayudarse de una tercera variable que almacene temporalmente los valores durante el cambio. Es como si para intercambiar el contenido de dos vasos utilizamos un tercer vaso.

a=5

b=8

c=a

a=b

b=c

Si te cuesta entenderlo te animamos a comprobarlo con una tabla de valores, un truco muy sencillo que puedes aplicar en todos estos ejercicios.

Ejemplo:

a b c

1º 5 8 -

2º5 8 5

3º8 8 5

4º8 5 5

* **2º CALCULAR LA MEDIA DE VALORES ALEATORIOS**

Calcular la media de 100 valores aleatorios comprendidos entre 0 y 100.

Solución:

C=runif(0,100)

X=mean (C)

X

**3º LISTAS DE OBJETOS**

Listar todos los objetos siguientes: a=”no” b= “puedes”, c=”con\_esta\_carrera”, d=”8” obtener sus características, eliminar el objeto a y volver a listar todos los objetos.

Solución:

a=”no”

b= “puedes”

c=”con\_esta\_carrera”

d=”8”

ls()

ls.str()

rm(a)

ls()

**4º MATRICES Y VECTORES (MUY IMPORTANTE)**

Las matrices A y B están constituidas respectivamente por los vectores z1=(8,99,2), z2=(7,0,8) y z3=(5, 8, 9), z4=(4,8,9). Hacer sus productos escalares y componente por componente.

NOTA: se recomienda hacer el ejercicio formando las matrices mediante los comandos cbind primero y nrow ncol después.

IMPORTANTE: para el producto de dos matrices por componentes estas deben tener las mismas dimensiones, como en el enunciado. Sin embargo, para multiplicar matricialmente 2 matrices la primera debe tener debe tener el mismo número de columnas que la segunda de filas. Amn%\*%Anm=Amm. Para conseguir esto en nuestro ejercicio haremos la matriz traspuesta de B.

Opción 1: cbind

A=cbind(z1,z2)

B=cbind(z3,z4)

Opción 2: nrow, ncol

A=matrix (c(8,99,2,7,0,8), ncol=2 nrow=3)

B=matrix(c(5,8,9,4,8,9), ncol=2, nrow=3)

Producto componente por componente:

A\*B

Producto matricial

t(B)

A%\*%B

**5º TABLA O DATA.FRAME**

Construye una tabla con los nombres de 3 amigos, su carrera o trabajo y su edad.

amigos=c(“Jesús”, “María”, “Elvira”)

carrera=c(“Ingeniería”, “Historia”, “Psicología”)

AA=data.frame(amigos,carrera)

AA

**6º REPRESENTACIÓN FUNCIONES**

1. Representar una función con los vectores tiempo=(0,15,30,45,60) y millones\_de\_bacterias siendo este último un vector formado por 5 valores aleatorios tomados entre 1 y 10. Llamar a los ejes apropiadamente, utilizar como título de la gráfica "cultivo" y hacer que los puntos de la representación aparezcan y que además estén unidos por una línea.

millones\_de\_bacterias=seq (1,10, length=5)

tiempo=c(0,15,30,45,60)

plot(tiempo,millones\_de\_bacterias, main=”cultivo”, xlab=”tiempo”, ylab=”millones de bacterias”, type=”b”)

1. Añade al ejercicio anterior otra gráfica con otros valores aleatorios para practicar la representación simultánea de 2 gráficas.

millones\_de\_bacterias=seq (1,10, length=5)

tiempo=c(0,15,30,45,60)

millones2=seq(1,10,length=5)

plot(tiempo,millones\_de\_bacterias, main=”cultivo”, xlab=”tiempo”, ylab=”millones de bacterias”, type=”b”)

par(new=TRUE)

plot(tiempo,millones2,, type=”b”)

Una buena forma de practicar R es programando un algoritmo que haya salido en clase puesto que es similar a lo que se pedirá en el examen. Te animo a que lo hagas, mientras tanto, aquí tienes resuelto un ejercicio del pasado examen de algoritmia en R.

**7º BUCLES Y CONDICIONALES**

Obtener el valor aproximado del polinomio de Lagrange en el punto 3 con los siguientes datos: a=1;b=10;n=3;f=c(10,25,60);x=c(2,5,8)

a=2

b=8

n=3

f=c(10,25,60)

x=3

h=(b-a)/(n-1)

S=0

for (i in 1:n) {S[i]=a+(i-1)\*h}

#Observa que en R los subíndices que indican la componente del vector se indican con las llaves[] y no con paréntesis, parece una tontería pero puede ser un fallo

P=0

for (i in 1:n) **{**

L[i]=1

for (j in 1:n)**{**

if (i!=j)**{**

L[i]=L[i]\*(x-S[j])/(S[i]-S[j])

**}**

**}**

#Si el que no se cumpliera la condición hubiera implicado otra acción, esto se habría indicado con el comando "else {}"

#En R puede ser difícil visualizar los bucles anidados y condiciones, puesto que los signos son siempre iguales. Por ello, debemos recordar que el primero que se abre es el último que se cierra y viceversa. Los hemos coloreado para que sea más claro.

P=P+f[i]\*L[i]

**}**

P

L