PRÁCTICA 5

La práctica 5 se encuentra dividida en 2 ejercicios. En el primero de ellos haremos uso de dos de los métodos utilizados para la integración numérica: el método de Simpson (método determinista) y el método de Montecarlo (método estocástico).
El ejercicio 2 tratará de resolver un ejercicio con funciones, entre las que se encuentra una ecuación logística.

EJERCICIO 1: FÓRMULA DE SIMPSON

La primera parte del ejercicio 1 se basa en hacer uso de la fórmula de Simpson para aproximar lo máximo posible el valor de una integral.

ENUNCIADO

Se considera una sustancia cuya densidad viene dada por d(x), siendo x la coordenada espacial. La masa total en cierto intervalo [A,B] está dada por

$$M_{AB} = \int_A^B d(x)dx$$

Se desea realizar un programa en R para resolver dicha integral, mediante una fórmula de Simpson compuesta, cuya expresión viene dada por:

$$\int_A^B d(x)dx \approx \frac{h}{6}\left(d(A) + 2\sum_{i=2}^{n-1} d(T_i) + 4\sum_{i=1}^{n-1} d\left(\frac{T_i + T_{i+1}}{2}\right) + d(B)\right)$$

Donde se ha realizado una subdivisión del intervalo [A, B] en (n-1) subintervalos iguales, es decir, considerando n puntos T_i, y siendo h la longitud de cada subintervalo.

DATOS

El programa se ejecutará con los siguientes datos: A=0, B=3.05, densidad dada por d(x)=exp(x)*sin(x). Como número de puntos para el desarrollo de la fórmula se tomará n=35.

PROCEDIMIENTO (INCLUYE LAS FUNCIONES)

1. Se programará la expresión de la densidad en una función llamada d.
2. La fórmula de Simpson compuesta se programará en otra función llamada Simpson.
3. Los sumatorios se realizarán con bucles condicionales o secuenciales (como quiera cada uno. Si son secuenciales hay que tener la precaución de usar paréntesis: (n-1)).
4. El resultado exacto de la masa es: M= 11.97907159…

```
rm(list=ls()) #para borrar el contenido de la memoria.

d=function(x){
      exp(x)*sin(x)
}

Simpson=function(A,B,n,T,h){
      suma1=0
      for (i in 2:(n-1)){
            suma1=suma1+d(T[i])
      }
      suma2=0
      for (i in 1:(n-1)){
            pmed=(T[i]+T[i+1])/2
            suma2=suma2+d(pmed)
      }
```

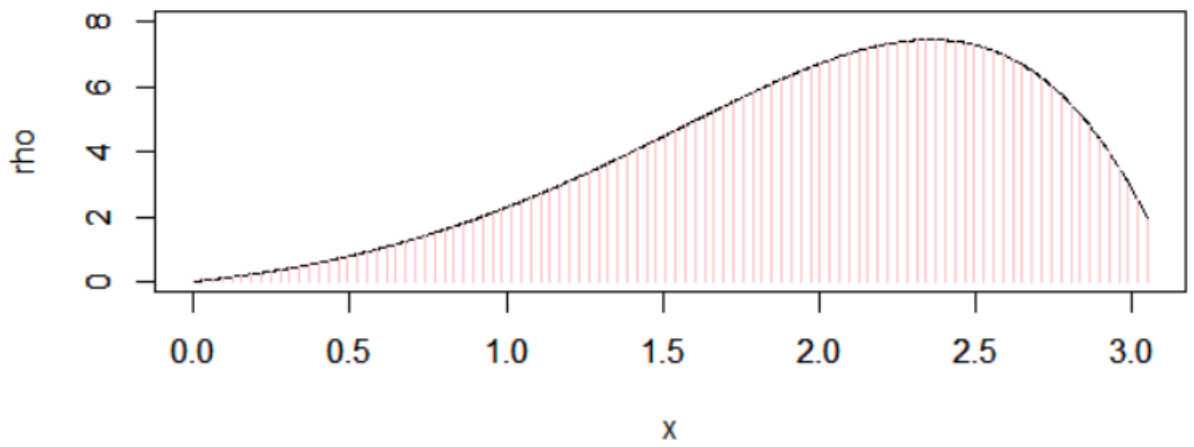
```
      Masa=h/6*(d(A)+2*suma1+4*suma2+d(B))
      return(Masa)
}

#DATOS
A=0; B=3.05; n=100; h=(B-A)/(n-1)
T=seq(A,B,length=n) #T=seq(A,B,h)
options(digits=18) #nos permite tener una mayor precisión.
MS=Simpson(A,B,n,T,h)
MS
M=integrate(d,A,B) #Con esta función se calcula el valor exacto de la integral
M
```

REPRESENTACIÓN GRÁFICA

```
par(mfrow=c(2,1)) #para dibujar dos gráficas en dos filas y una columna.
x=seq(A,B,length=1000) #secuencia de valores en el intervalo A,B
plot(x,d(x),type='l',xlim=c(A,B),ylim=c(0,8),xlab='x',ylab='rho')#Ponemos Type='l' para
poner que sea una línea. xlim es un vector que delimita el intervalo de la función
entre A y B. Esa es la función de densidad.
par(new='true') #para poner dos funciones en la misma gráfica.
plot(T,d(T),type='h',xlim=c(A,B),ylim=c(0,8),xlab='',ylab='',col='pink')#Ponemos tipo
histograma para que nos salgan las líneas. Dibujo del recubrimiento con rectángulos.
```

La gráfica obtenida tras este proceso será:



EJECICIO 1: MÉTODO DE MONTECARLO

Los métodos de Montecarlo abarcan una colección de técnicas que permiten obtener soluciones de problemas matemáticos o físicos por medio de pruebas aleatorias repetidas. Una aplicación de este método consiste en encontrar un valor aproximado de una integral definida. El proceso consiste en calcular el área comprendida entre una curva y el eje de abscisas (es decir, su integral) y definir un rectángulo que contenga esa área. Al generar puntos aleatorios se calcula la fracción entre la cantidad de puntos que hay dentro del área entre la curva y el eje de abscisas y la cantidad total de puntos (es decir, puntos en el rectángulo).

DATOS

Introducir una variable denominada num_puntos que contiene el número de valores a considerar (p.ej. 1000 puntos y luego incrementar). Definir extremos del dominio de cálculo: AA=0; BB=7.4 ; A=0; B=3.05 (en nuestro caso).
Generar los vectores ss, ff que contengan num_puntos aleatorios en el intervalo [A,B] y [AA,BB] respectivamente.

```
num_puntos=1000
AA=00; BB=7.4; A=0; B=3.05
ss=runif(num_puntos,A,B); ff=runif(num_puntos,AA,BB) #el comando runif se utiliza para
obtener numéros aleatorios en un intervalo determinado.
```

PROCEDIMIENTO

Inicializar las variables puntos_dentro=0; puntos_fuera=0; jacinto=0.
Para i desde 1 hasta num_puntos
Si (ff[i]<=d(ss[i]) entonces if( ){
puntos_dentro = puntos_dentro+1
jacinto[i]='blue'
si no }else{
puntos_fuera = puntos_fuera+1
jacinto[i]='orange '
Fin condición }
Fin bucle

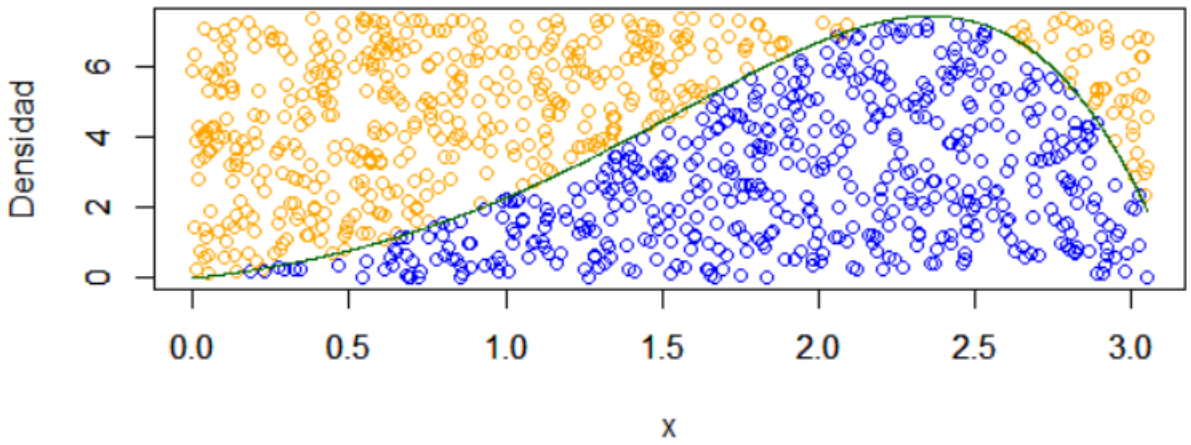
```
puntos_dentro=0; puntos_fuera=0; jacinto=0

for (i in 1:num_puntos){
      if(ff[i]<=d(ss[i])){
            puntos_dentro=puntos_dentro+1
            jacinto[i]='blue'
      } else{
            puntos_fuera=puntos_fuera+1
            jacinto[i]='orange'
      }
}
```

REPRESENTACIÓN GRÁFICA

Representamos conjuntamente ambas zonas (dentro y fuera del área).

plot(ss,ff,xlim=c(A,B),ylim=c(AA,BB),col=jacinto,ylab='Densidad',xlab='x')
par(new='TRUE')
ss=seq(A,B,0.001)
plot(ss,d(ss),type='l',col='dark green',xlim=c(A,B), ylim=c(AA,BB),xlab='',ylab='')



APROXIMACIÓN DEL VALOR DE LA INTEGRAL

Aproximar el valor de la integral: Area=puntos_dentro/num_puntos*B*BB.

Introducir el valor exacto de la integral: Vexact=11.97907159

```
Area=puntos_dentro/num_puntos*B*BB
Vexact=11.9790715873488693
```

OBTENCIÓN DE ERRORES

Obtener los errores cometidos con la fórmula de Simpson Compuesta y el Método de Montecarlo.

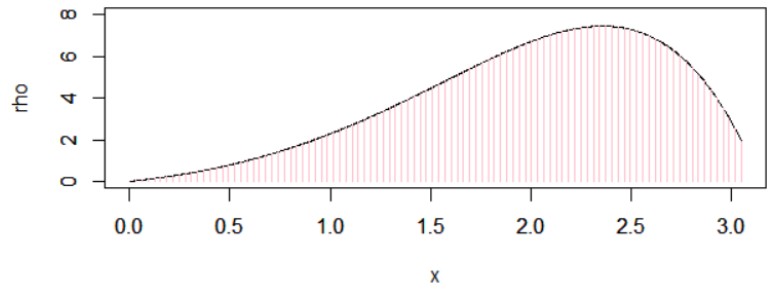
```
ErrorM=(Area-Vexact)/Vexact
ErrorS=(MS-Vexact)/Vexact
```

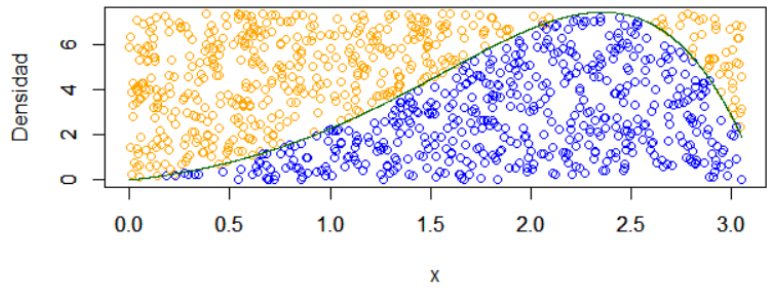
Escribir los resultados en forma de tabla (estructura data.frame).

```
Metodo=c("Valor exacto","Simpson","Montecarlo")
Valores=c(Vexact,MS,Area)
Error_Relativo=c(0,round(ErrorS,7),round(ErrorM,7))
data.frame(Metodo,Valores,Error_Relativo)
```

	Metodo	Valores	Error_Relativo
1	Valor exacto	11.9790715873488693	0.00000000000000000000
2	Simpson	11.9790715723599650	0.00000000000000000000
3	Montecarlo	11.9846700000000013	0.00046729999999999975

REPRESENTACIÓN DE AMBAS GRÁFICAS





EJERCICIO 2: ENUNCIADO

Consideramos el problema:

$$\begin{cases} y'(t) = r y(t).\left(1 - \dfrac{y(t)}{K}\right), & t \in (0,3] \\ y(0) = 2 \end{cases}$$

donde la ecuación diferencial se denomina ecuación logística. La variable t representa el tiempo, mientras que la variable y(t) representa la densidad de la población. En la ecuación aparecen también las constantes positivas r, K. El problema dado tiene una solución analítica (exacta) dada por la expresión:

$$y(t) = \frac{2K}{2 + e^{(-rt)}K - 2e^{(-rt)}}$$

PROCEDIMIENTO

Realiza un programa en R en el que: a) Se genere un vector t de 20 componentes que son puntos equidistantes en [0,3]. b) Se asigne y1=2 y w1=2 (tanto y como w serán vectores de 20 componentes). c) Se resuelva el problema (1) generando la sucesión (método de Euler).

$$y_{i+1} = y_i + hf(r,K,t_i,y_i), \quad (i = 1,\ldots,(N-1)) \qquad \text{siendo} \quad f(r,K,t,y) = r.y.\left(1-\frac{y}{K}\right)$$

Se resuelva nuevamente el problema (1) generando la sucesión (método de Heun):

$$z = w_i + hf(r,K,t_i,w_i)$$
$$w_{i+1} = w_i + \frac{h}{2}(f(r,K,t_i,w_i) + f(r,K,t_{i+1},z)), \quad (i = 1,\ldots,(N-1))$$

empleando, en ambos casos, los valores de las constantes: r=2, K=1, N=20.

```
rm(list=ls(all=TRUE)) #Comando para borrar todas las variables definidas previamente
t=seq(0,3,length=N)
```

`y=c(0); w=c(0)` # el vector y tendrá una solución en cada instante de tiempo, al igual que w, pero cada uno se calculará por un método diferente.

```
y[1]=2 ;w[1]=2

N=20
r=2
K=1
h=(3-0)/(N-1) #h es la distancia entre cada uno de los puntos del vector t, los cuales
son equidistantes.

f=function(r,K,t,y){
r*y*(1-y/K)
}

for(i in 1:(N-1)){
      y[i+1]=y[i]+h*f(r,K,t[i],y[i]) #El objetivo del bucle es calcular cada componente
de y a partir de la anterior #Método de Euler
      z=w[i]+h*f(r,K,t[i],w[i])
      w[i+1]=w[i]+h/2*(f(r,K,t[i],w[i])+f(r,K,t[i+1],z))
}
y
w
```

REPRESENTACIÓN GRÁFICA

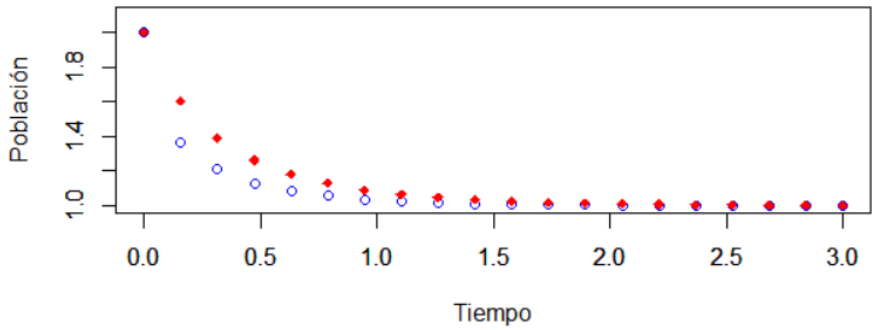
Se represente, en un mismo gráfico, la soluciones: exacta (*) en línea continua verde oscuro, la solución del apartado c) con símbolos azules y la solución del apartado d) con símbolos rojos usando pch=18.

Introducir leyenda en la esquina superior derecha con los títulos: ‘Exacta’, ‘Euler’, ‘Heun’ con relleno ‘dark green’, ‘blue’, ‘red’.

(*) para la exacta generar el vector tt de 10000 componentes en [0,3]

Una vez resuelto el ejercicio incrementar el valor de N (N=60, N=100, N=500).

```
plot(t,y,col='blue',xlab='Tiempo',ylab='Población',xlim=c(0,3),ylim=c(1,2.1))
par(new='true')
plot(t,w,col='red',pch=18,xlim=c(0,3),ylim=c(1,2.1),xlab='',ylab='')
```



PARA REPRESENTAR LA SOLUCIÓN EXACTA

```
#Queremos representar la solución exacta.

x=seq(0,3,length=1000)

Ex=c(0)  #en cada uno de estos puntos calculo la exacta.

for(i in 1:1000){

   Ex[i]=2*K/(2+exp(-r*x[i])-2*exp(-r*x[i]))

}

par(new='TRUE')

plot(x,Ex,type='l',co='dark green',xlim=c(0,3),ylim=c(1,2.1),xlab='',ylab='')
```

Este proceso hay que realizarlo a parte para que la gráfica no quede dividida en dos partes.