Grado en Ingeniería de Sistemas de Información
Examen parcial conv. ordinaria: Estadística (Duración: 3 horas)

Normas:

- **ANTES DE EMPEZAR**: Renombra `EXAMEN_NOMBRE_APELLIDOS` y muévelo a tu USB. Haz `source` de ütils.Rz asegúrate de que funciona.
- **De acuerdo con la Ley Orgánica de Protección de Datos, le informamos que la sesión de su ordenador podrá ser grabada y monitorizada por motivos de integridad académica.**
- El examen se realizará únicamente con bolígrafo, los formularios proporcionados y los programas R y Wolfram Alpha. Cualquier otro software/página web están prohibidos.
- No escribas en los formularios proporcionados.
- No puedes usar simulaciones a no ser que lo diga el enunciado.
- Las respuestas deberán estar razonadas para considerarse correctas.
- Un código que genere errores puntuará, como máximo, la mitad de la pregunta del apartado.
- Los ejercicios que deben hacerse obligatoriamente en R están marcados en negrita. En el resto de ejercicios puedes entregar los resultados en R o copiarlos a mano. ¡La única condición es que se entienda! Por ejemplo, sería válido:

  Opción 1:
  $\int_0^5 xdx$ = *La integral se resuelve con integrales.R, linea 5.*
  El resultado es 12.5.

  Opción 2:
  $\int_0^5 xdx$ = *La integral se resuelve con Wolfram/R, usando el código* `[código empleado, aquí]`.
  El resultado es 12.5.
- **NO HAY PREGUNTAS DURANTE EL EXAMEN**. La interpretación del enunciado forma parte de la evaluación. Si crees que algún enunciado es ambiguo o que faltan datos haz las asunciones que consideres oportunas para resolverlo, indicándolo claramente.
- Al final del examen, si tienes que entregar ficheros .R, **AVISA AL PROFESOR** y entrégalos en las actividades del campus virtual.

1. Problemas

1.1. (3 puntos) *(40 minutos)* La probabilidad de que desde un huevo nazca un insecto es $p = 0.35$. En una flor, el número de huevos puestos por estos insectos sigue una distribución Poisson de media $\lambda = 100$.

(a) (1 punto) Calcular la distribución del número de insectos condicionada a 10 huevos en la flor y representar gráficamente dicha distribución. ¿Cuántos insectos podemos esperar? ¿Cuál es el número más probable de insectos que tendrá la flor?

(b) (1 punto) [**Entregar huevos_nombre_apellido.R**] ¿Cuál es la probabilidad de que nazcan más de 50 insectos de una sola flor? Haz los cálculos mediante simulaciones en R.

(c) (1 punto) Se ha observado una flor y se ha constatado que han nacido 50 insectos. Calcular la distribución del número de huevos que había en la flor y representar gráficamente dicha distribución. ¿Cuál es el número de huevos que podemos esperar? ¿Cuál es el número más probable de huevos que tendrá la flor?

Solution:

```r
# Datos:
lambda = 100
p = 0.35

print("Apartado (a)")
```

```r
# Función de distribución condicionada:
py_x = function(y,x){
  dbinom(y,x,p)
}
py_x = Vectorize(py_x)

# Obtenemos la gráfica:
x=10
y = 0:x
plot(y,py_x(y,x),type = "h")

# Esperanza:
esperanza = sum(y * py_x(y,x))
print(paste("Se esperan",esperanza,"insectos en la flor si hay 10 huevos."))

# Moda:
moda = which.max(py_x(y,x)) - 1
print(paste("Lo más probable es obtener",moda,"insectos si hay 10 huevos."))

print("Apartado (b)")

N=10000
sims = replicate(N,{
  nhuevos = rpois(1,lambda)
  ninsectos = rbinom(1,nhuevos,p)
  if (ninsectos > 50){
    TRUE
  }else{
    FALSE
  }
})
prob = mean(sims)

print(paste0("La probabilidad de que haya más de 50 insectos es ",round(prob,4)))

print("Apartado (c)")

M = 200 # aproximación del infinito

# Distribución:
py = function(y){
  x = 0:M
  px = dpois(x,lambda)
  pyx = py_x(y,x)
  sum(pyx * px)
}
py = Vectorize(py)
y = 50
# Teorema de Bayes
px_y = function(x,y){
  py_x(y,x) * dpois(x,lambda) / py(y)
```

```
}
px_y = Vectorize(px_y)

# Gráfica
x = 0:M
plot(x,px_y(x,y),type = "h")

# Esperanza:
esperanza = sum(x*px_y(x,y))
print(paste("Se espera que hubiera",esperanza,"huevos en la flor."))

# Moda:
px_ymax = max(px_y(x,y))
moda = which.max(px_y(x,y)) - 1
print(paste("Lo más probable es que hubiera",moda,"huevos."))
```

1.2. (2 puntos) *(40 minutos)*

En una empresa cárnica han comprado una maquina de embase al vacío. En el manual de instrucciones se indica una probabilidad del 3 % de que entre aire en el paquete. En la venta al por mayor, venden lotes de 100 filetes de ternera cada uno envasado al vacío. Sabiendo que el cliente rechaza el lote entero si hay cuatro o más de los productos en mal estado:

(a) (1 punto) ¿Cuál es la probabilidad de que el cliente rechace un lote?

(b) (1 punto) ¿De cuántos paquetes como máximo tiene que componerse un lote para que menos del 5 % de los lotes sean rechazados?

Solution:

```
# Datos:
p = 0.03 # Probabilidad de fracaso en 1 paquete
n = 100 # Número de paquetes en un lote
# X = número de paquetes defectuosos en un lote
# P(rechazar lote) = P(3 < X ) = 1 - P(X <= 3)
probabilidad = 1 - pbinom(3,n,p)
print(paste0("La probabilidad de rechazar un lote es del ",
             round(probabilidad*100,2),"%."))

print("Apartado (b)")

for (k in 1:n) {
  probabilidad = 1 - pbinom(3,k,p)
  if (probabilidad < 0.05){
    numero = k
  }
}
print(paste("Los lotes deben tener como máximo",
            numero,"paquetes para que se rechacen menos del 5%."))
```

1.3. (2.5 puntos) *(40 minutos)* Una máquina tiene dos componentes. Sea X la variable aleatoria: "duración del componente I (en años)" e Y la variable aleatoria: "duración del componente II (en años)". La distribución conjunta de ambas variables es

$$f(x,y) = \begin{cases} c \cdot e^{-3x} \cdot y \cdot e^{-y^2} & x \geq 0, y \geq 0 \\ 0 & \text{en otro caso} \end{cases}$$

(a) (0.5 puntos) Calcula c. Si no eres capaz de hacer este apartado, resuelve los siguientes usando $c = 1$ (no es el resultado correcto).

(b) (1 punto) Calcula la probabilidad de que ambos componentes duren entre 1 y 3 años.

(c) (1 punto) Calcula la probabilidad de que el componente II dure al menos 1 año más que el componente I.

Solution:

```r
source("utils.R")

### a)
f_xy_unnormalized = function(x, y) {
  ifelse(
    (x > 0) & (y > 0),
    exp(-3 * x) * y * exp(-y ^ 2),
    0
  )
}

result = integrate2_dydx(
  f_xy_unnormalized,
  from_x = 0,
  to_x = Inf,
  from_y = function(x) 0,
  to_y = function(x) Inf
)
# Dado que c * result = 1
c_value = 1 / result$value
print(c_value)
#-----> c_value = 6

f_xy = function(x, y) {
  c_value * f_xy_unnormalized(x, y)
}

### b) P(1 < X < 3, 1 < Y < 3)
result_b = integrate2_dydx(
  f_xy,
  from_x = 1,
  to_x = 3,
  from_y = function(x) 1,
  to_y = function(x) 3
)
print(result_b)
#-----> P(1 < X < 3, 1 < Y < 3) = 0.01826
```

```
### c) P(Y > X + 1)
result_c = integrate2_dydx(
  f_xy,
  from_x = 0,
  to_x = Inf,
  from_y = function(x) x + 1,
  to_y = function(x) Inf
)
print(result_c)
#-------> P(Y > X + 1) = 0.2062
```

1.4. (2.5 puntos) (40 minutos) [**Entregar laberinto_nombre_apellido.R**] En un misterioso laberinto, te encuentras en un punto con tres posibles caminos para explorar: A, B y C. El camino A se elige con probabilidad 0.2, el camino B con probabilidad 0.5 y el camino C con probabilidad 0.3.

Tanto el camino A como el camino B llevan de vuelta al punto de inicio. Se tardan 3 horas en atravesar el camino que parte de A, y 5 horas en atravesar el camino que parte de B. Después de aventurarte por una de estos dos caminos y llegar de nuevo al punto de inicio, experimentas una desorientación temporal, lo que te hace perder el rumbo y que olvides que camino has usado (¡puedes volver a elegir el mismo camino que acabas de usar!). El camino C, por otro lado, lleva a la salida del laberinto después de una travesía de 3 horas.

Hay una dificultad adicional. Después de aventurarse por uno de los caminos y volver al punto de inicio te sientes más cansado, lo que añade 0.5 horas extra al tiempo necesario para recorrer cualquiera de los caminos. Es decir, en el primer intento los tiempos de travesía son 3, 5 y 3 horas para los caminos A, B y C, respectivamente; en el segundo intento serán 3.5, 5.5 y 3.5; en el tercer intento serán 4, 6 y 4, etc.

¿Cuánto tiempo, en promedio, se tardar en salir del laberinto? Usa simulaciones en R.

Solution: Es análogo al ejercicio de "James Bond" de la práctica. Las únicas novedades son:

- Los caminos no son equiprobables: la forma más fácil de resolver esto es usando el argumento `prob` de `sample`: `sample(..., prob = c(0.2, 0.5, 0.3))` (ver cuaderno de ejercicios del tema 1 y `?sample`).
- Hay que tener en cuenta el cansancio. Lo más sencillo es crear una variable `cansancio` y por cada intento aumentar su valor `cansancio = cansancio + 0.5`.

(No se sube la solución detallada para no difundir la solución de la práctica.)