Taller generación de variables aleatorias
Simulación Digital
Grupo H1

Kevin Javier Lozano Galvis
2172016

Brayan Rodolfo Barajas Ochoa
2170688

Carlos Alberto Palencia Pombo
2161342

Ejercicio 1

- Use el método de inversión para generar una variable aleatoria que tenga la siguiente función de distribución acumulativa:

$$F(x) = \frac{x^2 + x}{2},\ 0 \leq x \leq 1$$

Reemplazando $F(x)$ por $g \sim U(0, 1)$ se obtiene:

$$g = \frac{x^2 + x}{2}$$

$$2g = x^2 + x$$

$$x^2 + x - 2g = 0$$

$$x = \frac{-1 \pm \sqrt{1 - 4(-2g)}}{2}$$

$$x = \frac{-1 \pm \sqrt{1 + 8g}}{2}$$

Como $0 \leq x \leq 1$, se selecciona la raíz positiva

$$x = \frac{-1 + \sqrt{1 + 8g}}{2}$$

Además, para poder comprobar si se ha realizado correctamente es necesario conocer la función de densidad que en este caso es la derivada de la función acumulada

$$f(x) = x + \frac{1}{2},\ 0 \leq x \leq 1$$

Ejercicio 2

- Utilice una función g(x) de su elección (diferente a la distribución uniforme) para generar mediante el método del rechazo la siguiente función de densidad:

$$f(x) = xe^{-x}$$

Para poder realizar el cálculo de la constante c de forma analítica se escoge la función exponencial con $\lambda = 0.5$.

$$g(x) = 0.5e^{-0.5x}$$

Para calcular el valor de c analíticamente se tiene que

$$\frac{f(x)}{h(x)} = \frac{xe^{-x}}{0.5e^{-0.5x}} = 2xe^{-0.5x}$$

Para encontrar el valor de c se debe derivar e igualar a cero

$$(2e^{-0.5x})(1 - 0.5x) = 0$$

$$x = 2$$

De tal forma que c es igual a

$$c = 2(2)e^{-0.5(2)} \approx 1.471518$$

Ejercicio 3

- Una serie de buses llega a un evento deportivo de acuerdo a un proceso de Poisson con tasa media de 5 por hora. Cada bus tiene igual probabilidad de tener 20, 21, ... 40 aficionados. El número de aficionados en diferentes buses es independiente. Escriba un algoritmo que simule la llegada de aficionados en el tiempo t=1.

Para describir mejor la solución a este problema se coloca el código utilizado a continuación el cual contienen los comentarios que explican el mismo y hacen más entendible la solución

```
procpois.H<-function(T,lambda)
{
P<-vector() #El vector P almacena la cantidad de pasajeros que llega en cada bus
S<-vector() #El vector S almacena el tiempo de llegada de cada bus
t<-0;I<-0   #Se inicializa la variable tiempo y el contador de buses
    repeat
    {
    u1<-runif(1); t<-t-(1/lambda)*log(u1) #Se genera un tiempo t de acuerdo a una
distribucion exponencial
      if (t>T){ #Si el tiempo supera el tiempo maximo se rompe el ciclo
```

```
        break }
      else {
        I<-I+1 #Se aumenta el contador de buses
        S[I]<-t #Se guarda el tiempo en el vector correspondiente
        P[I]<-floor(runif(1, min=20, max=41)) #Se guarda la cantidad de pasajeros del
bus
        }
    }
result<- list("S" = S, "P" = P) #Se retornaron los dos vectores
return(result)
}

n_bus<-c() #Vector de cantidad de buses para cada experimento
n_pass<-c()#Vector de cantidad de pasajeros para cada experimento
for(i in 1:50){ #Se realizan 50 experimentos
  proceso <- procpois.H(1,5) #Se ejecuta un experimento
  n_bus[i]<-length(proceso$S) #Se guarda la cantidad de buses del experimento
  n_pass[i]<-sum(proceso$P) #Se guarda la cantidad de pasajeros del experimento
}

hist(n_bus, breaks=12) #Histograma para cantidad de buses
hist(n_pass, breaks= 8) #Histograma para cantidad de pasajeros
```

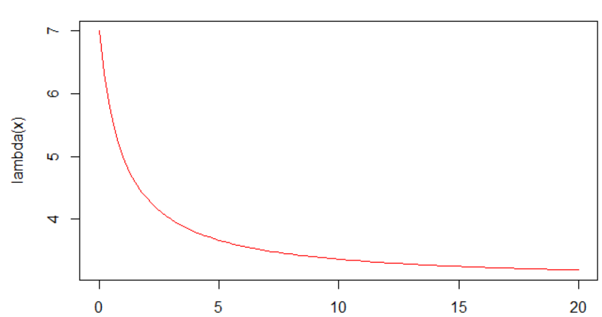
Ejercicio 4

- Escriba un programa que genere un proceso de Poisson no homogéneo con función de intensidad

$$\lambda(t) = 3 + \frac{4}{t+1}$$

Describa y justifique la solución. Verifique que el programa aproxima adecuadamente el proceso Poisson no homogéneo.

Antes de realizar el proceso no homogéneo es necesario conocer el comportamiento de la función $\lambda(t)$ para determinar un λ máximo



Luego de observar la gráfica se puede realizar la implementación del proceso no homogéneo mediante el código que se muestra a continuación el cual esta descrito por medio de comentarios

```
poisson<-function(lambda, Tmax) {
  #Se inicializan las variables
  S<-vector()
  u1<-vector()
  u2<-vector()
  t<-0;I<-0
  #Se calcula el valor de lambda maximo en la variable max
max=optimize(lambda,c(0,Tmax),maximum=TRUE)$objective
      for (i in 1:1000) {
                              #se calcula el tiempo t con una distribucion exponencial
                               u1[i]<-runif(1);    t<-t-((1/max)*log(u1[i]))
                               #Si t es mayor al tiempo maximo se rompe el ciclo
                              if (t>Tmax) {print(i);break}
                              if (t<=Tmax) u2[i]<-runif(1)
                              #Si el numero aleatorio u2 es menor a la restriccion se
guarda el tiempo t considerado como un exito
                              if(u2[i]<=(lambda(t)/max)) {I<-I+1;S[I]<-t}}
    print("S")
    print(S)
    print("tasa de aceptaci n")
    print(length(S)/i)
    print("n mero medio de realizaciones")
    print(i/length(S))
    hist(S, breaks=20)
    curve(lambda(x),xlim=c(0,20),ylab="lambda(x)",col="red",add=TRUE)
  }

poisson(lambda.p4, 20)
```