Maestría en Computo Estadístico
Inferencia Estadística
Tarea 2
7 de septiembre de 2020
Enrique Santibáñez Cortés
Repositorio de Git: Tarea 2, IE.

1. Cuando una máquina no se ajusta adecuadamente tiene una probabilidad 0.15 de producir un artículo defectuoso. Diariamente, la máquina trabaja hasta que se producen 3 artículos defectuosos. Se detiene la máquina y se revisa para ajustarla. ¿Cuál es la probabilidad de que una máquina mal ajustada produzca 5 o más artículos antes de que sea detenida? ¿Cuál es el número promedio de artículos que la máquina producirá antes de ser detenida?

RESPUESTA
Sea X el número de artículos producidos antes de que se produzcan 3 artículos defectuosos, entonces podemos decir que $X \sim BN(3, 0.15)$. Por lo tanto, **la probabilidad de que una máquina mal ajustada produzca 5 o más artículos antes de que sea detenida** es (ocupamos la función en R pnbinom(1, 3, 0.15)):

$$\mathbb{P}(X \geq 5) = 1 - \mathbb{P}(X \leq 4) = 1 - \sum_{x=3}^{4} \binom{x-1}{3-1}(1-0.15)^{x-3}(0.15)^3 = 1 - 0.01198125 = 0.9880187.$$

Por como se distribuye X podemos decir que **el número promedio de artículos que la máquina producirá antes de ser detenida** es

$$\mathbb{E}(X) = \frac{r}{p} = \frac{3}{0.15} = 20.\blacksquare.$$

2. Los empleados de una compañía de aislantes son sometidos a pruebas para detectar residuos de asbesto en sus pulmones. Se le ha pedido a la compañía que envíe a tres empleados, cuyas pruebas resulten positivas, a un centro médico para realizarles más análisis. Si se sospecha que el 40 % de los empleados tienen residuos de asbesto en sus pulmones, encuentre la probabilidad de que deban ser analizados 10 trabajadores para poder encontrar a 3 con resultado positivo.

RESPUESTA
Sea Y el número de trabajadores que se realizan las pruebas hasta encontrar 3 empleados con resultados positivos. Y como la probabilidad de que algún empleado tenga residuos de asbesto en sus pulmones (dar positivo en la pruebas) es de 0.40. Entonces podemos concluir que $Y \sim BN(3, 0.4)$. Por lo que la **probabilidad de que deban analizar 10 trabajadores para encontrar a 3 con resultado positivo es** (ocupamos la función en dnbinom(10, 3, 0.40) en R):

$$\mathbb{P}(Y = 10) = \binom{10-1}{3-1}(1-0.40)^{10-3}(0.40)^3 = 0.06449725 \quad \blacksquare.$$

3. Para el siguiente ejercicio es necesario usar R.

a) Considere una moneda desequilibrada que tiene probabilidad p de obtener águila. Usando el comando sample, escriba una función que simule N veces lanzamientos de esta moneda hasta obtener un águila. La función deberá recibir como parámetros a la probabilidad p de obtener águila y al número N de veces que se repite el experimento; y tendrá que regresar un vector de longitud N que contenga el número de lanzamientos hasta obtener un águila en cada uno de los N experimentos.

RESPUESTA
Si X es el número de lanzamientos de la modena hasta obtener un águila, con probabilidad p de obtener

águila en un lanzamiento. Entonces, $X \sim Geo(p)$. Por lo que la función que solicitan sería la simulación de X N veces. Ocupando la siguiente notación de 1:águila y 0:sol:

```
moneda_geometrica <- function(p, N){ # p: probabilidad de aguila, N # repeticiones.
  resultados <- c() # Inicializamos un vector.
  for (i in 1:N) { # Repetimos el experimentos N veces.
    contador <- 0 # Inicializamos el número de lanzamientos.
    while(sample(x=c(1,0), size=1, prob=c(p,1-p))!=1){ # si ya se obtuvo águila deterner.
      contador <- contador + 1
    }
    resultados[i] <- contador
  }
  resultados # regresamos los resultados.
}
```

Observamos que en los incisos siguientes se ocupa esta funcipon para N un poco grandes, por lo que vectorizo la función anterior para tener lo mismo en un tiempo más corto. La diferencia entre estas dos funciones radica basicamente en el *sample*, ya que nosotros simularemos por bloques, es decir, como si estuvieramos muchas modenas lanzandose al mismo tiempo.

```
moneda_geometrica_optimizada <- function(p, N, potencia){ # potencia: tamaño del bloques.
  resultados <- c() # Inicializamos un vector.
  while(length(resultados)<N) { # Repetimos el hasta tener N resultados
    contador <- 0 # Inicializamos el número de lanzamientos.
    resultados_preliminar <- c() # Inicializamos los resultados por bloques.
    while(length(resultados_preliminar)<potencia){
      contador_s<- sum(sample(x=c(1,0), size=potencia-length(resultados_preliminar),
                              prob=c(p,1-p), replace=TRUE))
      contador <- contador + 1
     resultados_preliminar<- c(resultados_preliminar, rep(contador, contador_s)) # Concatenamos l
    }
    resultados <- c(resultados, resultados_preliminar) # Concatenamos los resultados por bloques
  }
  resultados # regresamos los resultados.
}
```

Donde el parámetro potencia representa el tamaño del bloque, es decir, cuantas monedas se lanzarán al mismo tiempo. Algo curioso de este parámetro por intución entre más grande sea más rápido será, pero no es así aunque no estoy muy seguro por que sucede.

b) Usando la función anterior simule $N = 10^4$ veces una variable aleatoria Geom(p) para $p = 0.5, 0.1, 0.01$. Grafique las frecuencias normalizadas en color azul. Sobre está última figura empalme en rojo la gráfica de la función de masa correspondiente. ¿Qué observa?

RESPUESTA

Creemos otra función que utilice la función del inciso a) y que grafique las frecuencias normalizadas en azul y en rojo las frecuencias obtenidas de función de distribución de un variable Geometrica.

```
library(tidyverse) # ggplot and dplyr
geometric_graph_simula_and_teoric <- function(p, N, potencia, titulo, estadisticos=0){
  # Utilizamos la opción del inciso a).
  simular_geometrica <- data.frame(resultado=moneda_geometrica_optimizada(p, N, potencia))
  if(estadisticos==1){
```

```
    print("La media de las simulaciones es:")
    print(mean(simular_geometrica$resultado))
        print("La desviación estandar de las simulaciones es:")
        print(sqrt(var(simular_geometrica$resultado)))
  }
  # Generamos las frecuenciass normalizadas.
  simular_geometrica <- data.frame(table(simular_geometrica)/N)
  names(simular_geometrica) <- c("x", "y")
  simular_geometrica$x <- as.numeric(simular_geometrica$x)
  # Variable auxiliar.
  simular_geometrica$origen <- "simulacion"
  max_resul <- max(simular_geometrica$x)
  # Función de distribución utilizando la formula.
  teoric_geometrica <- data.frame(x=seq(1,max_resul,1),
                                  y=dgeom(x=seq(0,(max_resul-1),1),1),

  # Concatenamos las frecuencias obtenidas.
  geometrica <- rbind(teoric_geometrica, simular_geometrica)
  # Graficamos
  g <- ggplot(geometrica, mapping=aes(x,y,fill=origen))+
    geom_histogram(position="dodge", stat="identity", bins = max_resul)+
    labs(title=titulo)
  return(g)
}
```

Por lo que las gráficas variando el parámetro p son

```
set.seed(08081997)
#geometric_graph_simula_and_teoric(0.5, 10^4, 10^4,
#                                  "Simulación de una variable Geometrica(0.5)")
#geometric_graph_simula_and_teoric(0.1, 10^4, 10^4,
#                                  "Simulación de una variable Geometrica(0.1)")
#geometric_graph_simula_and_teoric(0.01, 10^4, 10^4,
#                                  "Simulación de una variable Geometrica(0.01)")
```

Observemos que si comparamos las frecuencias de las simulaciones y las frecuencias obtenidas de la función de probabilidad de una geometrica se ven muy cercanas. Pero conforme p se acerca a 0 la comparaciones entre estas frecuencias son más notorias. Esto se puede explicar debido a que cuando p es más chico la $\mathbb{P}(X = x)$ se va hacieno más pequeña, por lo que x toma un rango más amplio de valores posibles. No hay que confundirse por el hecho de que como la función de distribución de una variable aleatoria geometrica esta defina en todos los naturales. Ya que si p es cercano a 1, las probabilidades convergen más rapido a 0, y viceversa, si p es cercano a 0 las probabilidad convergen más lento a 0.

c) Repita el inciso anterior para $N = 10^6$. Además calcule el promedio y la desviación estándar de las simulaciones que realizó ¿Qué observa?

```
set.seed(08081997)
#geometric_graph_simula_and_teoric(0.5, 10^6, 10^5,
#"Simulación de una variable Geometrica(0.5)",1)
#geometric_graph_simula_and_teoric(0.1, 10^6, 10^5,
#"Simulación de una variable Geometrica(0.1)",1)
```

```
#geometric_graph_simula_and_teoric(0.01, 10^6, 10^5,
#"Simulación de una variable Geometrica(0.01)",1)
```

Cómo el número de simulaciones son mayores que el inciso anterior, observamos que las diferencias se entre las frecuencias simuladas y frecuencias calculados son muy cercanas "casi nulas". Y esto incita a concluir que la distribución Geometrica modela bien este esperimento de lanzamiento de monedas. Ahora analizando los promedios y desviaciones de las contra la experanza de X para cada P:

■.

4. Usando las ideas del inciso anterior escriba una función en R que simule N veces los lanzamientos de moneda hasta obtener r águilas. La función deberá recibir como parámetros a la probabilidad p de obtener águila, al número r de águilas a observar antes de detener el experimento y al número N de veces que se repite el experimento; y tendrá que regresar un vector de longitud N que contenga el número de lanzamientos hasta obtener las r águilas en cada uno de los N experimentos. Grafique las frecuencias normalizadas de los experimentos para $N = 10^6$, $p = 0.2, 0.1$ y $r = 2, 7$ y compárelos contra la función de masa de la distribución más adecuada para modelar este tipo de experimentos.

RESPUESTA

Sea X el número de lanzamientos hasta obtener r aguilas. Esto implica que $X \sim BN(r, p)$, donde p es la probabilidad de obtener águila en un lanzamiento. Entonces la función que simula este experimento sería:

```
moneda_nbinom <- function(r, p, N){
  resultados <- c()
  for(i in 1:N){
    contador <- 0
    lanzamiento <- ""
    num_aguilas <- 0
    while(num_aguilas<r){
      lanzamiento <- sample(x=c("aguila", "sol"), size=1, prob=c(p,1-p))
      contador <- contador + 1
      if(lanzamiento=="aguila"){
        num_aguilas<-num_aguilas+1
      }
    }
    resultados[i] <- contador
  }
  resultados
}
```

La función anterior tiene un problema, ya que es muy lenta. Por lo que se vectorizo para tener un mejor rendimiento.

```
moneda_nbinom_optimizada <- function(r, p, N, potencia){
  resultados <- c()
  while(length(resultados)<N) { # Repetimos el experimentos N veces.
    contador <- 0 # Inicializamos el número de lanzamientos.
    resultados_preliminar <- c()
    inicial <- rep(0, potencia)
    while(length(resultados_preliminar)<potencia){ # si ya se obtuvo águila deterner.
      inicial <- inicial + sample(x=c(1,0), size=potencia-length(resultados_preliminar),
                                  prob=c(p,1-p), replace=TRUE)
```

```
      contador_s <- sum(inicial==r)
      contador <- contador + 1
     resultados_preliminar<- c(resultados_preliminar, rep(contador, contador_s))
     inicial <- inicial[inicial<r]
    }
    resultados <- c(resultados, resultados_preliminar)
  }
  resultados # regresamos los resultados.
}
```

Ahora modificamos la función del problema 3 para adaptarla a este problema,

```
bimneg_graph_simula_and_teoric <- function(r, p, N, potencia, titulo, estadisticos=0){
  # Utilizamos la opción del inciso a).
  simular_geometrica <- data.frame(resultado=moneda_nbinom_optimizada(r, p, N, potencia))
  if(estadisticos==1){
    print("La media de las simulaciones es:")
    print(mean(simular_geometrica$resultado))
        print("La desviación estandar de las simulaciones es:")
        print(sqrt(var(simular_geometrica$resultado)))
  }
  # Generamos las frecuenciass normalizadas.
  simular_geometrica <- data.frame(table(simular_geometrica)/N)
  names(simular_geometrica) <- c("x", "y")
  simular_geometrica$x <- as.numeric(simular_geometrica$x)+r-1
  # Variable auxiliar.
  simular_geometrica$origen <- "simulacion"
  max_resul <- max(simular_geometrica$x)
  # Función de distribución utilizando la formula.
  teoric_geometrica <- data.frame(x=seq(r,max_resul,1),
                                  y=dnbinom(x=seq(0,(max_resul-r),1), size=r,prob = p), origen=re

  # Concatenamos las frecuencias obtenidas.
  geometrica <- rbind(teoric_geometrica, simular_geometrica)
  # Graficamos
  g <- ggplot(geometrica, mapping=aes(x,y,fill=origen))+
    geom_histogram(position="dodge", stat="identity", bins = max_resul)+
    labs(title=titulo)
  return(g)
}
```

Por lo que las gráficas variando el parámetro p y r son:

```
set.seed(08081997)
#bimneg_graph_simula_and_teoric(2, 0.1, 10^6, 10^5,
#                               "Simulación de una variable NBinom(2, 0.1)", 1)
#bimneg_graph_simula_and_teoric(2, 0.2, 10^6, 10^5,
#                               "Simulación de una variable NBinom(2, 0.2)", 1)
#bimneg_graph_simula_and_teoric(7, 0.1, 10^6, 10^5,
#                               "Simulación de una variable NBinom(7, 0.1)", 1)
#bimneg_graph_simula_and_teoric(7, 0.2, 10^6, 10^5,
```

```
#                                   "Simulación de una variable NBinom(7, 0.2)", 1)
```

Podemos concluir que la distribución Binomial Negativa ajusta muy bien este experimento. Los parámetros p y r influyen de cierta manera, para ver como in

5. Considera X una v.a. con función de distribución F y función de densidad f, y sea A un intervalo de la línea real $\mathbb{R}$. Definamos la función indicadora $1_A(x)$:

$$1_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{en otro caso} \end{cases}$$

Sea $Y = 1_A(x)$. Encuentre una expresión para la distribución acumulada y el valor esperado de Y.
RESPUESTA
cuando X es discreta tenemos que la función de densidad es

$$f(Y = y) = \mathbb{P}(1_A(x) = y) = \mathbb{P}(\{x : 1_A(x) = y\}).$$

$$f(y) = \begin{cases} \mathbb{P}(x \in A) & \text{para y=1} \\ \mathbb{P}(x \notin A) & \text{para y=0} \\ 0 & \text{en otro caso} \end{cases}$$

El valor esperado es

$$\mathbb{E}[Y] = \sum yf(y) = 1 \cdot f(1) + 0 \cdot f(0) = 1 \cdot f(1) = \mathbb{P}(x \in A) \quad \blacksquare.$$

6. Las calificaciones de un estudiante de primer semestre en un examen de química se describen por la densidad de probabilidad

$$f_y(y) = 6y(1 - y) \qquad 0 \leq y \leq 1,$$

donde y representa la proporción de preguntas que el estudiante contesta correctamente. Cualquier calificación menor a 0.4 es reprobatoria. Responda lo siguiente:

a) ¿Cuál es la probabilidad de que un estudiante repruebe?

b) Si 6 estudiantes toman el examen, ¿cuál es la probabilidad de exactamente 2 reprueben?

RESPUESTA

Por como esta definida la función de probabilidad podemos decir que Y es es una variable continua. Ahora, solo para comprobación veamos que realmente sea una función de probabilidad, para ello observemos que

$$\int_{-\infty}^{\infty} f_y(y) = \int_0^1 6y(1 - y) = 3y^2 - 2y^3|_0^1 = 1.$$

Por lo tanto observamos que si es una función de probabilidad.

Entonces **la probabilidad de que un estudiante repruebe es**

$$f_y(Y < 0.4) = \int_0^{0.4} f_y(y) = \int_0^{0.4} 6y(1 - y) = 3y^2 - 2y^3|_0^{0.4} = 0.352$$

Ahora, sea X el número de estudiantes de reprueban el examen de un conjunto de 6 estudiantes que realizaron el examen. Por definición podemos decir que $X \sim Bin(6, p)$ donde p es la probabilidad de reprobar, pero si consideramos que las calificaiones de los estudiantes se distribuye como la variable Y, entonces podemos concluir que $X \sim Bin(6, 0.352)$. Por lo tanto, **la probabilidad de que exactamente 2 estudiantes reprueben es** (usamos la función dbinom(x=4, size = 6, prob = 0.325):

$$\mathbb{P}(X = 2) = \binom{6}{2} 0.325^2(1 - 0.325)^4 = 0.328907 \quad \blacksquare.$$

7. Escriba una función en R que simule una aproximación al proceso Poisson a partir de las 5 hipótesis que usamos en clase para construir tal proceso. Usando esta función, simule tres trayectorias de un proceso Poisson $\lambda = 2$ sobre el intervalo $[0, 10]$ y grafíquelas. Además simule 10^4 veces un proceso de Poisson N con $\lambda 1/2$ y hasta el tiempo $t = 1$. Haga un histograma de $N(1)$ en su simulación anterior y compare contra la distribución de Poisson correspondiente.

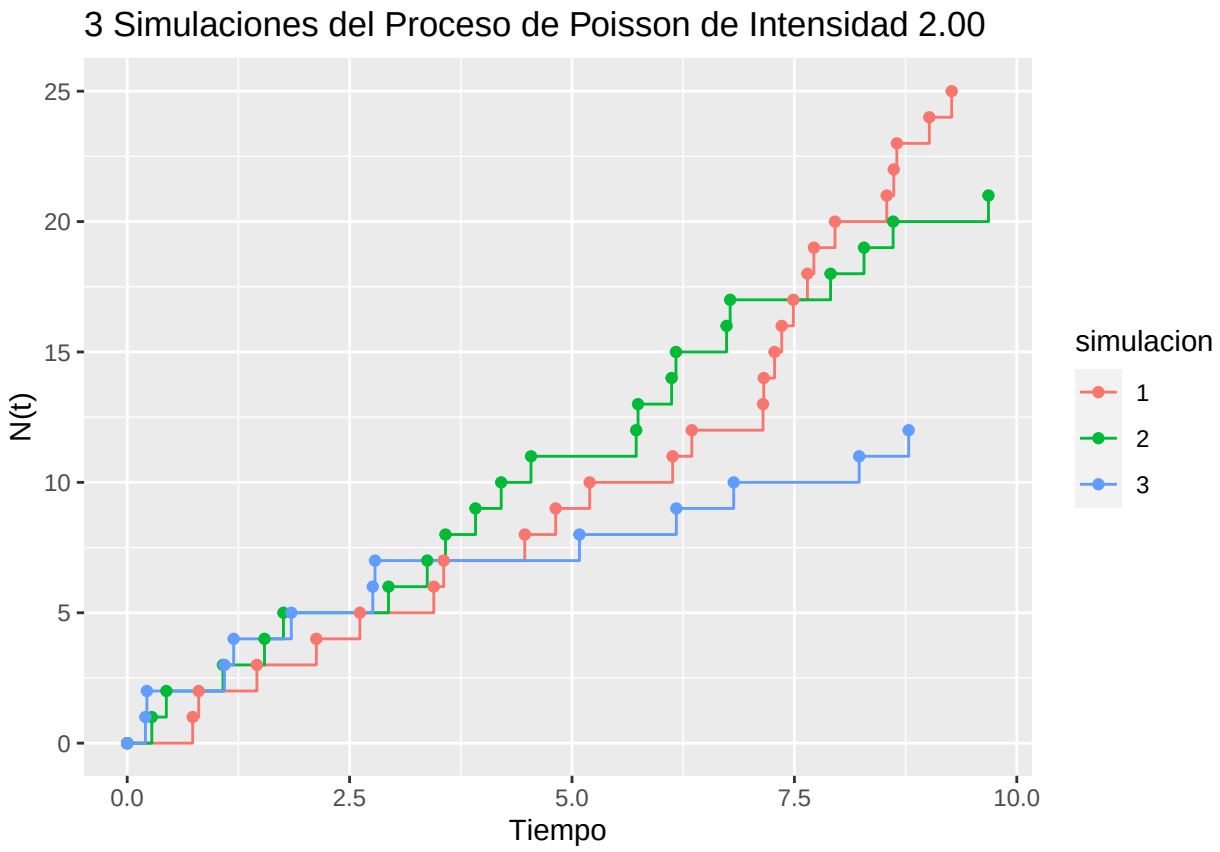
RESPUESTA

```
ProcesoPois<- function(t,lambda){
  N<- rpois(1,t*lambda) #Paso 1
  C<- sort(runif(N,0,t)) #Paso 2 y 3
  data.frame(x=c(0,0,C),y=c(0,0:N))
}
```

```
library(plyr)
NPois<-function(n,t,rate){
  C<- lapply(1:n, function(n)
    #Genera N dataframes con los procesos
    data.frame(ProcesoPois(t,rate),simulacion=n))
  C<-ldply(C, data.frame) # Une en una sola dataframe
  C$simulacion<-factor(C$simulacion) # Convierte en factores
  C
}
```

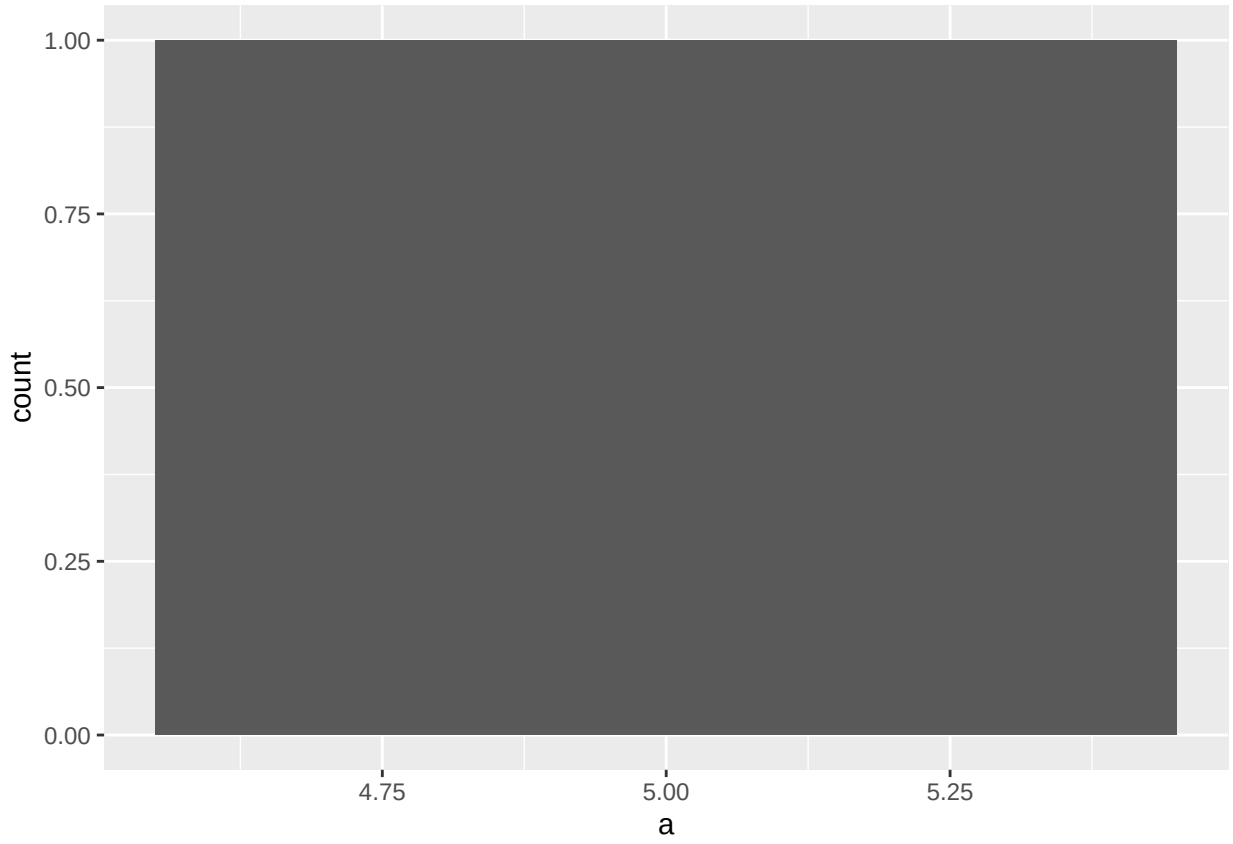
```
simulacion_process_a <- NPois(3,10,2)

# Graficamos.
qplot(x,y,data=simulacion_process_a,geom=c("step","point"),color=simulacion,
      xlab="Tiempo",ylab="N(t)",main=sprintf("%d Simulaciones del Proceso de Poisson de Intensida
```




```
set.seed(13)
prueba <- NPois(10^4, 1,0.5)

prueba %>% group_by(simulacion) %>% summarise(a=max(y)) %>%
  ggplot(aes(x=a))+geom_bar()
```




```
dpois(x = 0,lambda = 0.5)
```

```
## [1] 0.6065307
```

```
dpois(x = 1,lambda = 0.5)
```

```
## [1] 0.3032653
```

```
dpois(x = 2,lambda = 0.5)
```

```
## [1] 0.07581633
```

```
dpois(x = 3,lambda = 0.5)
```

```
## [1] 0.01263606
```

```
dpois(x = 4,lambda = 0.5)
```

```
## [1] 0.001579507
```

```
dpois(x = 5,lambda = 0.5)
```

```
## [1] 0.0001579507
```

8. En una oficina de correo los paquetes llegan según un proceso de Poisson de intensidad λ. Hay un costo de almacenamiento de c pesos por paquete y por unidad de tiempo. Los paquetes se acumulan en el local y se despachan en grupos cada T unidades de tiempo (es decir, se despachan en $T, 2T, 3T, \cdots$). Hay un costo por despacho fijo de K pesos (es decir, el costo es independiente del número de paquetes que se despachen). (a) ¿Cuál es el costo promedio por paquete por almacenamiento en el primer ciclo $[0, T]$? (b) ¿Cuál es el costo promedio por paquete por almacenamiento y despacho en el primer ciclo? (c) ¿Cuál es el valor de T que minimiza este costo promedio?

RESPUESTA
Sea X el número de paquetes que llegan al correo en un intervalo de tiempo T, este se distribuye como un proceso Poisson con intensidad λ. Entonces el costo total promedio por almacenamiento es:

$$\mathbb{E}[C] = \mathbb{E}[X \cdot c \cdot T] = cT\mathbb{E}[X] = cT \cdot (\lambda T) = cT^2\lambda.$$

Y ahora el número esperado de paquetes en el primer ciclo es:

$$\mathbb{E}[X] = \lambda T.$$

Por lo que, **(a) el costo promedio por paquete por almacenamiento es:**

$$\frac{cT^2\lambda}{\lambda T} = cT.$$

Ahora sea G el costo total de almacenamiento y despacho para el primer ciclo $[0, T]$ definido como

$$G = cXT + K.$$

Entonces el costo promedio total por almacenamiento y despacho es

$$\mathbb{E}[G] = \mathbb{E}[cXT + K] = cT^2\lambda + K.$$

Lo anterior implica que **el costo promedio por paquete por almacenamiento y despacho en el primer ciclo** es:

$$\mathbb{E}\bar{[}G] = \frac{cT^2\lambda + K}{\lambda T}.$$

Utilizando el resultado anterior, diferenciamos e igualamos a cero para encontrar el mínimo.

$$\mathbb{E}'\bar{[}G] = c - \frac{K}{\lambda T^2}$$

Igualamos a cero:

$$\begin{aligned} c - \tfrac{K}{\lambda T^2} &= 0 \\ T^2 &= \tfrac{K}{c\lambda} \\ T &= \sqrt{\tfrac{K}{c\lambda}}. \end{aligned}$$

Usando el criterio de segunda derivada para determinar si es un máximo o minino:

$$\mathbb{E}''\bar{[}G] = 2\frac{K}{\lambda T^3}.$$

Evaluando la segunda derivada en $T = \sqrt{\frac{K}{c\lambda}}$, observamos que $\mathbb{E}''\bar{[}G] > 0$, por lo que podemos concluir que es un mínimo. En conclusión, **el valor de T para el cuál minimiza el costo promedio por paquete por almacenamiento y despacho en el primer ciclo es $\sqrt{\frac{K}{c\lambda}}$** ■.

9. Considere la siguiente función

$$F(x) = \begin{cases} 0 & \text{para } x < 0 \\ 0.1 & \text{para } x = 0 \\ 0.1 + 0.8x & \text{para } 0 < x < 3/4 \\ 1 & \text{para } 3/4 \leq x \end{cases}$$

¿Es una función de distribución? Si es una función de distribución, ¿corresponde a una variable aleatoria discreta o continua?

RESPUESTA
Observemos por como esta definida la función tenemos que $0 \leq F(x) \leq 1$. Y además $\lim_{x\to\infty^-} F(x) = 0$ y $\lim_{x\to\infty^+} F(x) = 1$. Por lo que podemos concluir que $F(x)$ si es una función de distribución. Ahora, observemos que la función esta definida para $x = 0$ y $x = 3/4$, si X fuera una X fuera una variable continua, por definición $F(x = a) = 0$, por lo que X es discreta en $x = 0$ y $x = 3/4$. Y como $F(x)$ es continua en $0 < x < 3/4$ podemos decir que X es continua en ese intervalo. Entonces como X es continua y discreta para ciertos valores, decimos que X es "mixta". Esto igual se puede mostrar observando la grafica de la función $F(X)$ ■.

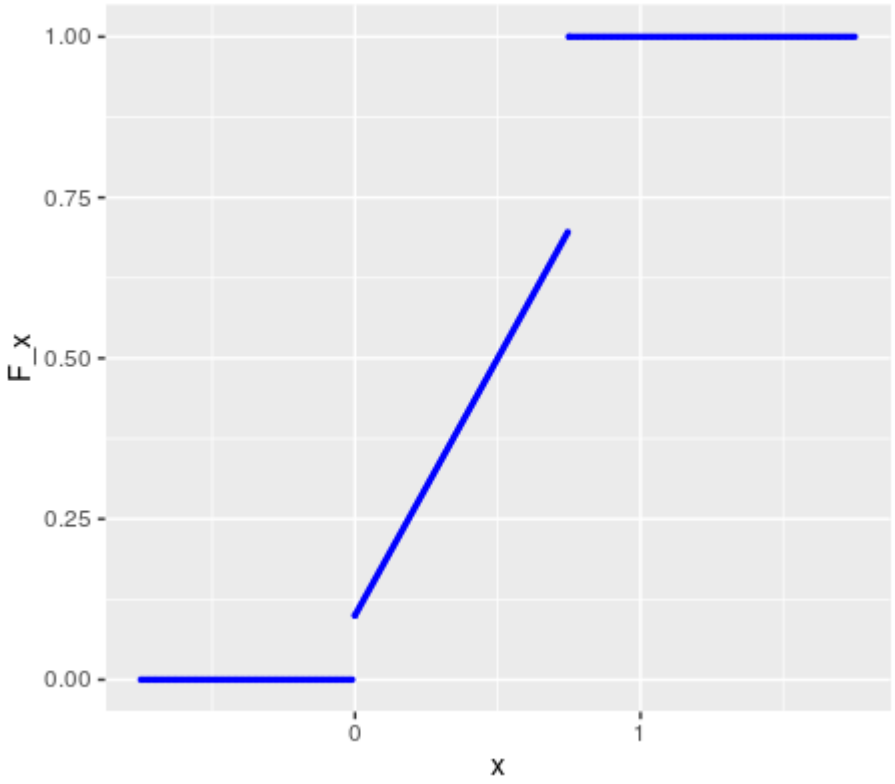


Figura 1: Función de densidad mixta.

10. **Este es un problema al que se recurrirá en el futuro,** su intención es que empiecen a jugar con datos reales. El archivo Delitos.csv contiene información sobre los delitos denunciados en la ciudad de Aguascalientes, para el período comprendido entre enero de 2011 a junio del 2016. Dicho archivo contiene 5 columnas: la primera columna contiene la fecha de denuncia del delito; la columna TIPO muestra una descripción del tipo de delito; la columna CONCATENAD presenta un descripción más amplia del delito; la columna SEMANA contiene la semana del año a la que corresponde la fecha de denuncia; y la columna SEMANA_COMPLETAS indica la semana a lo largo del estudio en la cual se presentó la denuncia. A través de métodos gráficos (e.g. boxplots) traten de determinar el comportamiento semanal de los delitos y discutan alternativas de modelos para describir los delitos cometidos en forma relativamente apropiada.

```
# Cargamos las librerias a ocupar.
library(tidyverse)
library(lubridate)

# Leamos los datos.
```

```
df_delitos <- read.csv(file = "Delitos.csv")
```

Conozcamos un poco los datos.

```
names(df_delitos)
```

```
## [1] "FECHA"            "TIPO"             "CONCATENAD"       "SEMANA"
## [5] "SEMANA_COMPLETAS"
```

```
head(df_delitos,3)
```

```
##        FECHA      TIPO                           CONCATENAD SEMANA
## 1 2011-01-01 COMERCIAL COMERCIAL/EMPRESA/INDUSTRIA/FARDERO      1
## 2 2011-01-04 COMERCIAL COMERCIAL/EMPRESA/INDUSTRIA/FARDERO      1
## 3 2011-01-16 COMERCIAL COMERCIAL/EMPRESA/INDUSTRIA/FARDERO      3
##   SEMANA_COMPLETAS
## 1                1
## 2                1
## 3                3
```

```
str(df_delitos)
```

```
## 'data.frame':    44212 obs. of  5 variables:
##  $ FECHA           : Factor w/ 1988 levels "2011-01-01","2011-01-02",..: 1 4 16 21 21 23 25 25
##  $ TIPO            : Factor w/ 23 levels "BICICLETA","COMERCIAL",..: 2 2 2 2 2 2 17 3 11 2 ...
##  $ CONCATENAD      : Factor w/ 305 levels "BICICLETA/PERSONA/ASALTO",..: 44 44 44 44 44 44 223
##  $ SEMANA          : int  1 1 3 3 3 4 4 4 5 6 ...
##  $ SEMANA_COMPLETAS: int  1 1 3 3 3 4 4 4 5 6 ...
```

```
unique(df_delitos$TIPO)
```

```
##  [1] COMERCIAL                       TRANSEUNTE
##  [3] CRISTAL                         MOTOCICLETA
##  [5] VEHICULO                        TRANSEUNTE EN VEHICULO
##  [7] BICICLETA                       TRANSPORTE DE PASAJEROS CIUDAD
##  [9] DOMICILIARIO                    INSTITUCIONES PUBLICAS
## [11] INSTITUCION POLITICA            REMOLQUE/PLATAFORMA
## [13] INSTITUCION FINANCIERA          OTRO
## [15] TARJETA BANCARIA/COMERCIAL      TRANSPORTE DE CARGA CIUDAD
## [17] MAQUINARIA PESADA               TRANSPORTE DE CARGA CARRETERA
## [19] GANADO                          INSTITUCION BANCARIA
## [21] TRANSPORTE DE PASAJEROS CARRETERA No Capturado
## [23] TRACTOR AGRICOLA
## 23 Levels: BICICLETA COMERCIAL CRISTAL DOMICILIARIO ... VEHICULO
```

```
#df_delitos %>% group_by(TIPO) %>%
#  count() %>% arrange(desc(n)) %>% head()
```

Esto puede deberse a que no todos los delitos se reportan, probablemente exista un sesgo cuando las perdidas son mayores.

```
#df_delitos %>% group_by(TIPO,SEMANA) %>%
#  count() %>% group_by(TIPO,SEMANA) %>% arrange(desc(n)) %>% head(4)
```

Si observamos el calendario, probablemente se daba a las vacaciones de semana santas.

```
ggplot(data=df_delitos, aes(x=SEMANA))+
  geom_density()
```