

## Índice

<b>1. f.d.p. Gaussiana</b>	<b>2</b>
1.1. Estimación con <i>hist</i> , <i>mean</i> y <i>std</i> . . . . .	2
1.2. Valores de $\mu$ y $\sigma^2$ teóricas con estimadas . . . . .	4
1.3. Estimaciones de $\mu_x$ - Histograma . . . . .	4
1.4. Error en la estimación de $\mu_x$ - Ejemplo . . . . .	5
<b>2. Variables aleatorias conjuntamente gaussianas</b>	<b>7</b>
<b>3. Proceso de Poisson</b>	<b>9</b>
3.1. Cálculo de parámetros específicos . . . . .	10
3.2. Estimación de parámetros por promedio temporal . . . . .	11

## 1. f.d.p. Gaussiana

Para generar una f.d.p gaussiana  $X$  con media  $\mu_x$  y varianza  $\sigma_x^2$ , se utilizarán dos variables aleatorias  $U_1$  y  $U_2$  con distribución uniforme en  $(0,1)$ .

Partiendo de dos variables aleatorias  $R$  y  $\Theta$ , la función de distribución de  $R$ :

$$F_R(r) = 1 - e^{-\frac{r^2}{2}} \quad 0 < r < \infty$$

Igualando a  $u_1$  se puede despejar:

$$F_R(r) = u_1 \implies R = \sqrt{-2 \cdot \ln(u_1)}$$

La variable  $\Theta$  tiene distribución uniforme:

$$F_\Theta(\theta) = \frac{\theta}{2\pi} \quad 0 < \theta < 2\pi$$

Igualando a  $u_2$  se despeja:

$$F_\Theta(\theta) = u_2 \implies \Theta = 2\pi u_2$$

Se tiene entonces que:

$$Y = R \cdot \cos(\Theta) \implies Y = \sqrt{-2 \cdot \ln(u_1)} \cdot \cos(2\pi u_2)$$

Por lo que se consigue generar una variable aleatoria  $Y \sim N(0, 1)$ . Para obtener la variable aleatoria  $X$  con media y varianza personalizadas, se utiliza la transformación lineal:

$$X = \sigma_x \cdot Y + \mu_x$$

Donde efectivamente:

$$E(X) = \sigma_x \cdot \underbrace{E(Y)}_{=0} + \mu_x = \mu_x \quad \text{y} \quad \text{Var}(X) = \sigma_x^2 \cdot \underbrace{\text{Var}(Y)}_{=1} = \sigma_x^2$$

### 1.1. Estimación con *hist*, *mean* y *std*

Se realizó la simulación utilizando vectores de 1000, 5000 y 50000 elementos, que permiten visualizar en forma clara la diferencia entre los histogramas y las curvas teóricas para cada caso, mostrando una corrida a continuación. En los histogramas se utilizaron 100 barras.

Los parámetros utilizados son  $\mu_x = 3$  y  $\sigma_x^2 = 1$ .

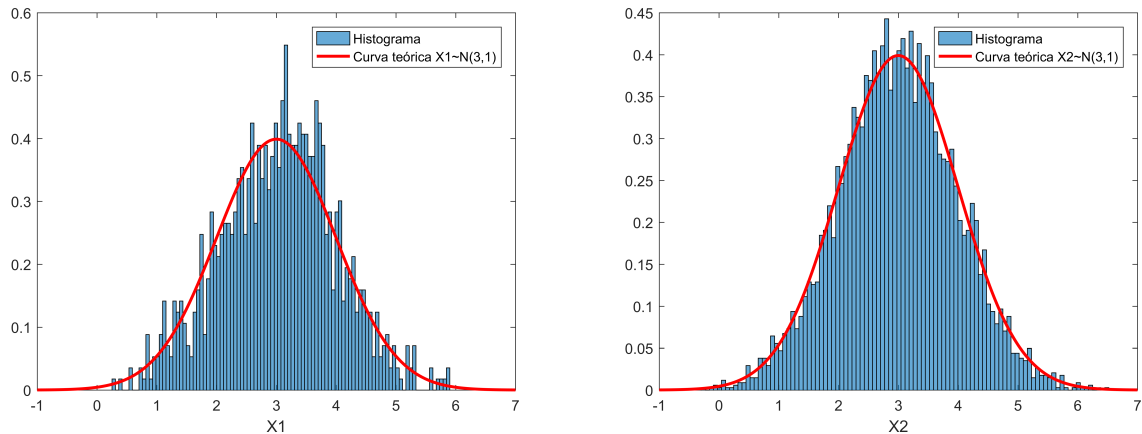


Figura 1: Comparación para vectores  $X_1$  de 1000 elementos e  $X_2$  de 5000 elementos

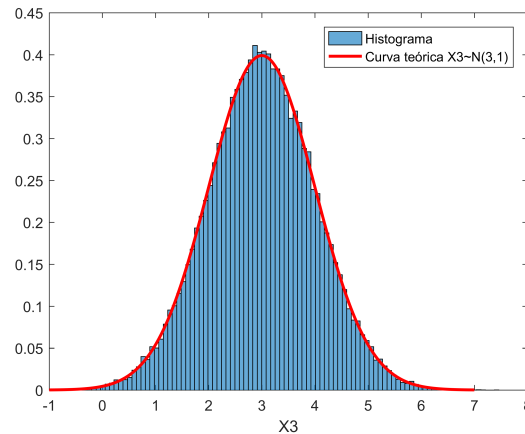


Figura 2: Comparación para vector  $X_3$  de 50000 elementos

Utilizando un número fijo de barras adecuado, se consigue el ajuste mostrado en las gráficas anteriores. Para mayor cantidad de elementos, el histograma tiende a parecerse más a la curva de la gaussiana teórica.

El código de MatLab utilizado para la simulación se muestra a continuación.

```

1 %% Valores de media y varianza personalizados
2 mu_x = 3;
3 var_x = 1;
4
5 b = mu_x;
6 a = sqrt(var_x);
7
8 %% Item A
9 % Generacion de variables aleatorias uniformes en [0,1]
10 x1 = rand(1000,1);
11 x2 = rand(1000,1);
12 x3 = rand(5000,1);
13 x4 = rand(5000,1);
14 x5 = rand(50000,1);
15 x6 = rand(50000,1);
16
17 % Generacion de variables aleatorias gaussianas a partir de dos uniformes
18 y1 = a.*(sqrt(-2.*log(x1)).*cos(2.*pi.*x2)) + b;
19 y2 = a.*(sqrt(-2.*log(x3)).*cos(2.*pi.*x4)) + b;
20 y3 = a.*(sqrt(-2.*log(x5)).*cos(2.*pi.*x6)) + b;
21
22 % Curva teorica de comparacion
23 xi = (-4*a+b):0.001:(4*a+b);
24 z = normpdf(xi,mu_x,sqrt(var_x));
25
26 bars = 100; %Numero de barras
27
28 % Histogramas normalizados
29 figure;
30 histogram(y1,bars,'Normalization','pdf');
31 hold on;
32 plot(xi,z,'red','linewidth',2);
33 xlabel('X1');
34 legend('Histograma','Curva teorica X1~N(3,1)');
35 %print('X1_1000','-dpng','-r600');
36
37 figure;
38 histogram(y2,bars,'Normalization','pdf');
39 hold on;
40 plot(xi,z,'red','linewidth',2);
41 xlabel('X2');
42 legend('Histograma','Curva teorica X2~N(3,1)');
43 %print('X2_5000','-dpng','-r600');
44
45 figure;
46 histogram(y3,bars,'Normalization','pdf');
47 hold on;
48 plot(xi,z,'red','linewidth',2);
49 xlabel('X3');
50 legend('Histograma','Curva teorica X3~N(3,1)');

```

```

51 %print('X3_50000','-dpng','-r600');
52
53 mu_y = [mean(y1) mean(y2) mean(y3)];
54 var_y = [var(y1) var(y2) var(y3)];
55
56 %% Item B
57 % Calculo la diferencia en valor absoluto para comparar
58 comp_mu_y = [abs(mu_y(1) - mu_x) abs(mu_y(2) - mu_x) abs(mu_y(3) - mu_x)];
59 comp_var_y = [abs(var_y(1) - var_x) abs(var_y(2) - var_x) abs(var_y(3) - var_x)];

```

Listing 1: Código de implementación

## 1.2. Valores de $\mu$ y $\sigma^2$ teóricas con estimadas

De los vectores anteriores se estima la media y la varianza para cada caso, comparando con los teóricos en el siguiente cuadro.

Elementos	$\mu_x$			$\sigma_x^2$		
	Teórico	Estimado	$\Delta\mu_x$	Teórico	Estimado	$\Delta\sigma_x^2$
1000	3	3.0402	0.0402	1	0.9348	0.0652
5000	3	2.9882	0.0118	1	0.9577	0.0423
50000	3	3.0006	0.0006	1	0.9984	0.0016

Cuadro 1: Valores estimados y teóricos

Dado que el número de elementos es bastante grande en los casos analizados, los valores estimados caen siempre muy cerca de los teóricos. A mayor cantidad de elementos la diferencia entre el valor teórico y el estimado tiende a ser menor.

## 1.3. Estimaciones de $\mu_x$ - Histograma

Para este caso, se realizan 10000 estimaciones en distribuciones con 10000 elementos cada una. El código de MatLab utilizado para la simulación es el siguiente.

```

1 %% Valores de media y varianza personalizados
2 mu_x = 3;
3 var_x = 1;
4
5 b = mu_x;
6 a = sqrt(var_x);
7
8 %% Item D
9 % Se hace 10000 veces para ver la distribucion de las medias en conjunto
10 mu_x_dist = zeros(10000,1);
11 for i=1:10000
12     x1d = rand(10000,1);
13     x2d = rand(10000,1);
14     xd = a.*(sqrt(-2.*log(x1d)).*cos(2.*pi.*x2d)) + b;
15     mu_x_dist(i) = mean(xd);
16 end
17 figure;
18 mean(mu_x_dist);
19 std(mu_x_dist);
20 histogram(mu_x_dist,bars,'Normalization','pdf');
21 xlabel('Medias de X');
22 %print('MediasX','-dpng','-r600');

```

Listing 2: Código de implementación

Realizando un histograma de los valores obtenidos, se tiene:

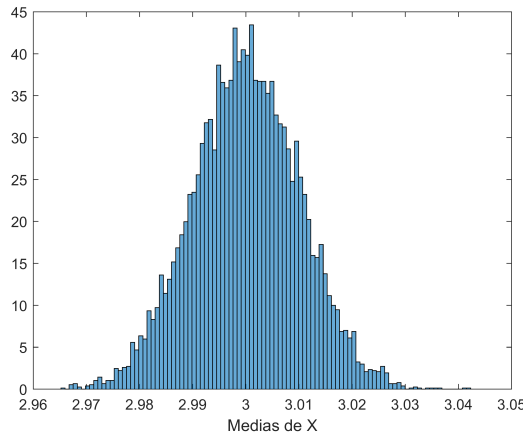


Figura 3: Histograma de las medias estimadas, con resultante  $\mu = 3,0001 - \sigma = 0,01$

Si llamamos a la media estimada  $m_X$ , se la define en este caso como:

$$m_X = \frac{1}{N} \cdot \sum_{i=1}^N \mu_{X_i} \quad \text{Con } N = 10000$$

Como la distribución normal es reproductiva, la suma de  $N$  variables aleatorias con dicha distribución resulta también normal. Esto último también puede verse desde el Teorema Central del Límite, dado que el valor de  $N$  utilizado es suficientemente grande. Para el caso en cuestión, los valores teóricos de  $\mu$  y  $\sigma$  son:

$$E(m_X) = \mu = \frac{1}{N} \cdot \underbrace{\sum_{i=1}^N E(\mu_{X_i})}_{\text{Por linealidad}} = \frac{1}{N} \cdot \sum_{i=1}^N \mu_x = \mu_x \Rightarrow \mu = 3$$

$$\sigma^2(m_X) = \sigma^2 = \frac{1}{N^2} \cdot \underbrace{\sum_{i=1}^N \sigma^2(\mu_{X_i})}_{\text{Por independencia}} = \frac{1}{N^2} \cdot \sum_{i=1}^N \sigma_x^2 = \frac{\sigma_x^2}{N} \Rightarrow \sigma = 0,0001$$

De donde se observa que los valores estimados sobre la media del vector graficado en el histograma resultan muy cercanos a los teóricos.

#### 1.4. Error en la estimación de $\mu_x$ - Ejemplo

Se toma ahora un caso particular, donde la cantidad de elementos (vectores con  $K$  números aleatorios con distribución normal) es desconocida, y se busca que la probabilidad de que el valor estimado  $m_X$  se aparte más del 4 % del valor teórico sea menor al 2 %. Es decir:

$$P[|m_X - \mu_x| > 0,04\mu_x] \leq 0,02$$

Se utiliza un  $K$  lo suficientemente grande para considerar que los vectores  $X$  tienen media  $\mu_x$  y desvío estándar  $\sigma_x$  iguales a los teóricos con los que se generaron los números aleatorios que éstos contienen. Si se trabaja con la expresión:

$$2 \cdot P[m_X - \mu_x < -0,04\mu_x] \leq 0,02$$

$$P\left[\left(\frac{m_X - \mu_x}{\sigma_x}\right) \cdot \sqrt{N} < -0,04 \cdot \frac{\mu_x}{\sigma_x} \cdot \sqrt{N}\right] \leq 0,01$$

La variable aleatoria  $\left(\frac{m_X - \mu_x}{\sigma_x}\right) \cdot \sqrt{N}$  tiene distribución normal estándar. La llamamos  $\Phi$ :

$$\Phi\left(-0,04 \cdot \frac{\mu_x}{\sigma_x} \cdot \sqrt{N}\right) \leq 0,01$$

$$\Phi\left(0,04 \cdot \frac{\mu_x}{\sigma_x} \cdot \sqrt{N}\right) \geq 0,99$$

$$0,04 \cdot \frac{\mu_x}{\sigma_x} \cdot \sqrt{N} \geq \Phi^{-1}(0,99) \approx 2,3263$$

$$N \geq 3382,3 \cdot \frac{\sigma_x^2}{\mu_x^2}$$

Para el caso analizado,  $\mu_x = 3$  y  $\sigma_x^2 = 1$ :

$$N \geq 375,9$$

El código utilizado para verificar por simulación es el siguiente:

```

1 %% Valores de media y varianza personalizados
2 mu_x = 3;
3 var_x = 1;
4
5 %% Item E
6 %mx representa el promedio de la suma de N variables aleatorias
7 %Media: mu_x - Desviacion: sigma_x/sqrt(N)
8
9 N = 3382.3.*var_x./(mu_x.^2); %Analiticamente resulta eso y verifica
10
11 mx = makedist('Normal','mu',mu_x,'sigma',sqrt(var_x./N));
12
13 P = 2.*cdf(mx,0.96.*mu_x) %La probabilidad pedida

```

Listing 3: Código de implementación

Donde la probabilidad P calculada efectivamente resulta  $\leq 0,02$ .

## 2. Variables aleatorias conjuntamente gaussianas

En este caso se analizan dos variables aleatorias conjuntamente gaussianas, cuyos parámetros son:

Caso	$\mu_x$	$\mu_y$	$\sigma_x$	$\sigma_y$	$\rho$
A	2	3	1	1	-0.5
B	2	3	1	0.5	-0.8

Cuadro 2: Parámetros para los casos analizados

Las variables  $X_n$  con distribución normal estándar son generadas utilizando el método de simulación de monte-carlo (mediando dos variables aleatorias con distribución uniforme al igual que antes), y mediante la transformación lineal siguiente se halla la variable  $X$ :

$$X = \sigma_x X_n + \mu_x$$

Análogamente se halla la variable  $Y$ :

$$Y = \sigma_y Y_n + \mu_y$$

Para dibujar la nube de puntos con ambas variables se tiene la función de densidad de probabilidad gaussiana bivariable:

$$f_{XY}(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 + \left(\frac{y-\mu_y}{\sigma_y}\right)^2 - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}\right]}$$

Teniendo los elementos anteriores, se implementa el siguiente código en MatLab para obtener los gráficos resultantes.

```

1 %% [2] V.A Conjuntamente Gaussianas
2 clear Workspace;
3 clc;
4
5 % Generacion de uniformes [0 , 1]
6 N = 5000;
7 X1 = rand(N,1);
8 X2 = rand(N,1);
9 X3 = rand(N,1);
10 X4 = rand(N,1);
11
12 % Generacion de gaussianas estandard
13 Y1 = sqrt(-2.*log(X1)).*cos(2.*pi.*X2);
14 Y2 = sqrt(-2.*log(X3)).*cos(2.*pi.*X4);
15
16 %% CASO A)
17 u1a = 2;
18 u2a = 3;
19 sigma1a = 1;
20 sigma2a = 1;
21 roa = -0.5;
22
23 % Gaussianas con media y varianza personalizadas
24 Z1a = sigma1a.*Y1 + u1a;
25 Z2a = sigma1a.*Y2 + u2a;
26
27 % Gaussiana bivariable
28 Z12a = (1/(2.*pi.*sigma1a.*sigma2a.*sqrt(1-roa.^2))).*exp(-(1/(2.*(1-roa.^2))).*(((Z1a-u1a)/sigma1a).^2 + ((Z2a-u2a)/sigma2a).^2 - (2.*roa.*(Z1a-u1a).*(Z2a-u2a)/(sigma1a.*sigma2a))));
29
30 figure;
31 scatter3(Z1a,Z2a,Z12a,2,'filled');
32 title('Caso A')
33 xlabel('Z1');
34 ylabel('Z2');
35 zlabel('f(Z1,Z2)');
36 set(gca,'XLim',[-4 8],'YLim',[-4 8])
37 view(-45,40)
38 print('Za','-dpng','-r600');
39
40 %% CASO B)
41 u1b = 2;

```

```

42 u2b = 3;
43 sigma1b = 1;
44 sigma2b = 0.5;
45 rob = -0.8;
46
47 % Gaussianas con media y varianza personalizadas
48 Z1b = sigma1b.*Y1 + u1b;
49 Z2b = sigma1b.*Y2 + u2b;
50
51 % Gaussiana bivariable
52 Z12b = (1/( 2.*pi.*sigma1b.*sigma2b.*sqrt(1-rob.^2) )).*exp(-( 1/(2.*(1-rob.^2)) ).*(((Z1b-u1b)/sigma1b).^2 + ((
    Z2b-u2b)/sigma2b).^2 - (2.*rob.*(Z1b-u1b).*(Z2b-u2b)/(sigma1b.*sigma2b)) ) ));
53
54 figure;
55 scatter3(Z1b,Z2b,Z12b,2,'filled');
56 title('Caso B');
57 xlabel('Z1');
58 ylabel('Z2');
59 zlabel('f(Z1,Z2)');
60 set(gca,'XLim',[-4 8],'YLim',[-4 8]);
61 view(-45,40)
62 print('Zb','-dpng','-r600');

```

Listing 4: Código de implementación

Los resultados obtenidos para ambos casos se muestran a continuación.

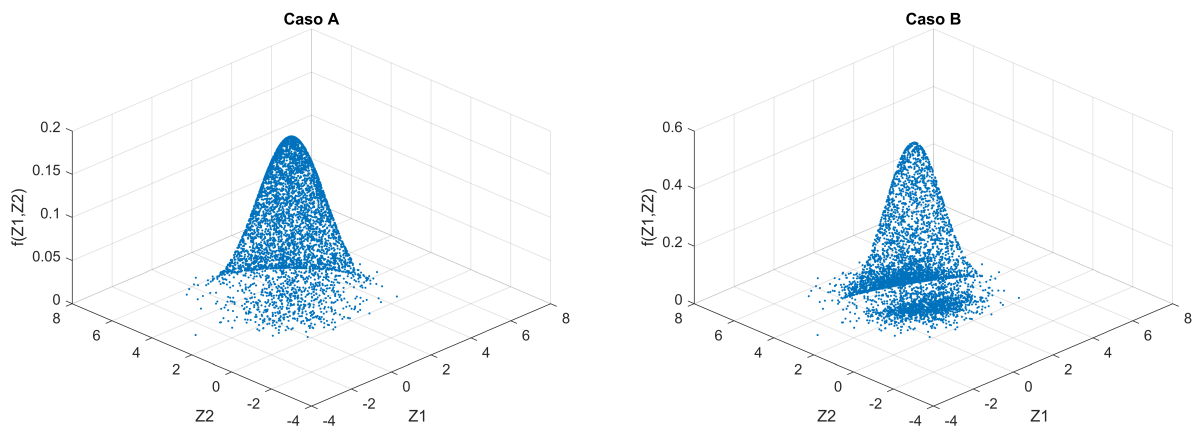


Figura 4: Nube de puntos resultante para ambos casos

Para el caso A, se tiene que  $\sigma_x = \sigma_y$ , por lo que si se realizan las curvas de nivel se observarían círculos, donde la orientación de sus ejes está dada por el coeficiente de correlación  $\rho$ . En el caso B, se tiene  $\sigma_x \neq \sigma_y$ , por lo que las curvas de nivel resultantes son elipses. El  $|\rho|$  es mayor que en el caso anterior, por lo que la orientación de sus ejes resulta distinta. Se observa que a medida que el  $|\rho| \rightarrow 1$ , las elipses se van 'estirando', hasta que en el límite tiende a una recta.

En dicho límite, se tiene una relación directamente lineal entre  $X$  e  $Y$ , por lo que conociendo cuánto vale uno ya se conoce el otro, mediante:

$$y = \frac{\sigma_y}{\sigma_x}x + \mu_y - \frac{\sigma_y}{\sigma_x}\mu_x$$



### 3. Proceso de Poisson

El proceso aleatorio asignado en este caso es el proceso de Poisson. Se utiliza un contador  $Q(t)$ , que indica el número de eventos ocurridos hasta el tiempo  $t$ .

El código utilizado para generar las funciones muestra se detalla a continuación. Se ajustó de manera tal que la media  $\lambda t$  tenga  $\lambda \approx 1$ .

```

1 %% [3] Proceso aleatorio de Poisson
2 N = 10000; %Cantidad de funciones muestra
3 T = 1000; %Longitud de tiempo
4
5 %% Item A
6 Qa = zeros(N,1);
7 for i=2:N
8     xa = rand;
9     ya = log(1/(1-xa));
10    if (ya > 7.5)
11        Qa(i) = Qa(i-1) + 1;
12    else
13        Qa(i) = Qa(i-1);
14    end
15 end
16 z = 1:1:N;
17 figure;
18 plot(z,Qa);
19 xlabel('t');
20 ylabel('Q(t)');
21 print('Q', '-dpng', '-r600');

```

Listing 5: Código de implementación

En lo que sigue se plasman algunas funciones muestra generadas.

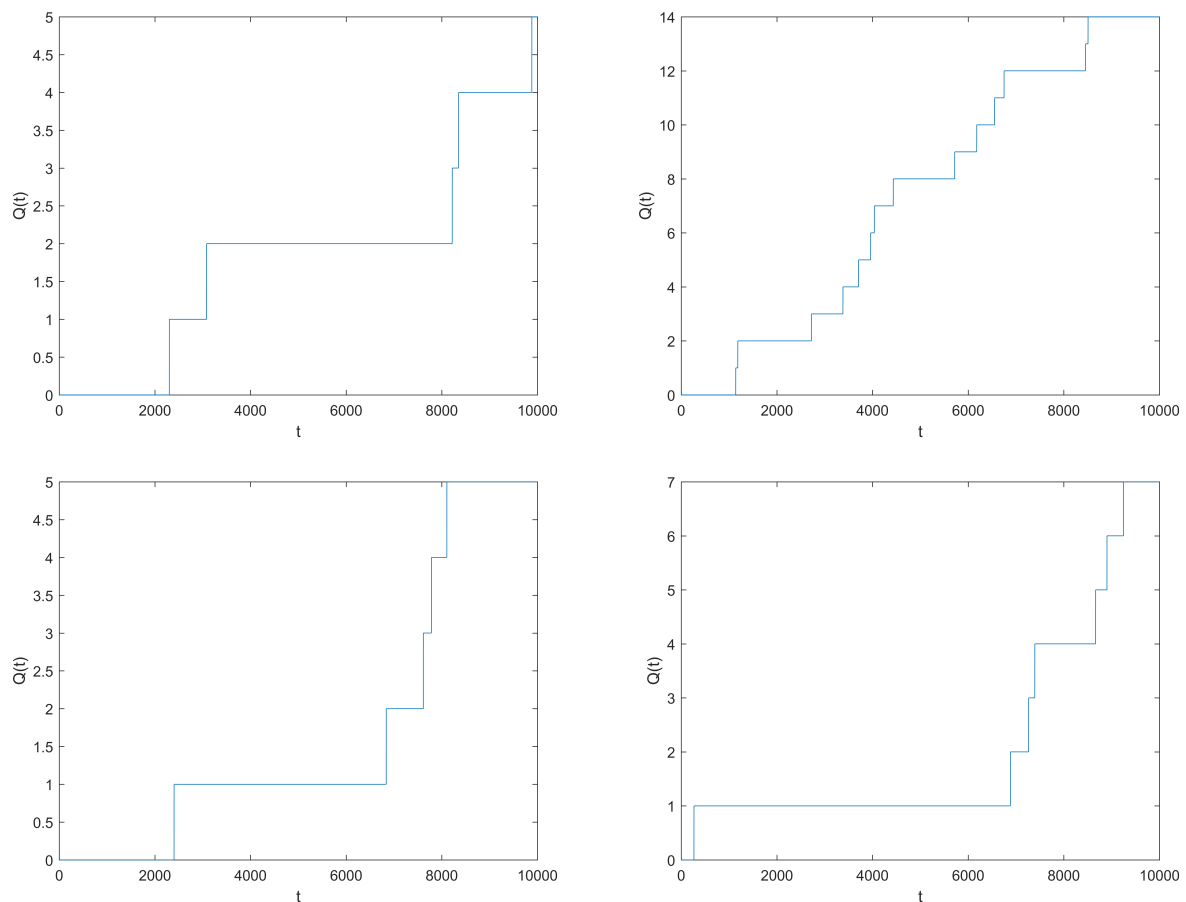


Figura 5: Algunas funciones muestra generadas

### 3.1. Cálculo de parámetros específicos

Teniendo en cuenta el valor teórico de  $\lambda = 1$  (y que  $t_1 < t_2$ ), se calculan algunos parámetros en particular:

$$E[Q(t)] = \lambda t \Rightarrow E[Q(4)] = 4$$

$$\sigma^2[Q(t)] = \lambda t \Rightarrow \sigma^2[Q(7)] = 7$$

$$R_{QQ}(t_1, t_2) = \lambda^2 t_1 t_2 + \lambda t_1 \Rightarrow R_{QQ}(5, 8) = 45$$

$$C_{QQ}(t_1, t_2) = R_{QQ}(t_1, t_2) - E[Q(t_1)] \cdot E[Q(t_2)] \Rightarrow r_{QQ}(t_1, t_2) = \frac{C_{QQ}(t_1, t_2)}{\sqrt{C_{QQ}(t_1, t_1) \cdot C_{QQ}(t_2, t_2)}} \Rightarrow r_{QQ}(2, 3) = 0,8165$$

Se comparan los valores teóricos con los simulados durante una corrida en el siguiente cuadro.

	Teórico	Simulado
$E[Q(4)]$	4	4.0520
$\sigma^2[Q(7)]$	7	6.8203
$R_{QQ}(5, 8)$	45	45.6080
$r_{QQ}(2, 3)$	0.8165	0.8024

Cuadro 3: Comparación de los parámetros obtenidos

El código utilizado para simular dichos valores se muestra a continuación. Se utilizaron para ello 1000 funciones muestra (N) de 10000 valores cada una (T), donde en cada caso el valor de  $t$  corresponde a  $t \cdot N$ .

```

1 N = 1000; %Cantidad de funciones muestra
2 T = 10000; %Longitud de tiempo
3
4 %% Item B
5
6 Qb = zeros(T,N);
7 for i=1:N
8     for j=1:T
9         xb = rand;
10        yb = (log(1/(1-xb)));
11        if (j == 1)
12            if (yb > 6.9)
13                Qb(j,i) = 1;
14            else
15                Qb(j,i) = 0;
16            end
17        else
18            if (yb > 6.9)
19                Qb(j,i) = Qb(j-1,i) + 1;
20            else
21                Qb(j,i) = Qb(j-1,i);
22            end
23        end
24    end
25 end
26
27 %Media en t = 4N
28 mean_Q4 = mean(Qb(4.*N,:));
29 %Varianza en t = 7N
30 var_Q7 = var(Qb(7.*N,:));
31 %Rxx(5,8)
32 Rxx5_8 = R(Qb,5.*N,8.*N);
33 %rxx(2,3)
34 rxx2_3 = C(Qb,2.*N,3.*N) / (sqrt(C(Qb,2.*N,2.*N) .* C(Qb,3.*N,3.*N)));

```

Listing 6: Código de implementación

### 3.2. Estimación de parámetros por promedio temporal

Para ver si es posible estimar los parámetros anteriores realizando un promedio temporal sobre alguna de las funciones muestra (en lugar de hacerlo sobre el ensamble), se debe verificar que se cumplan ciertos requerimientos. Se considera el promedio temporal en tiempo continuo:

$$\langle g[x(t)] \rangle_T = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} g[x(t)] dt$$

Para estimar la media con promedio temporal, debe resultar ser un proceso ergódico en la media. Para ello, debe cumplirse que:

$$\lim_{T \rightarrow \infty} E[\langle x(t) \rangle_T] = \mu_x \quad \lim_{T \rightarrow \infty} \sigma^2[\langle x(t) \rangle_T] = 0$$

Intuitivamente, se puede observar que no resultará ergódico en la media sin realizar los cálculos, dado que si se toma una función  $Q(t)$  cualquiera del ensamble, si  $T \rightarrow \infty$  en el peor caso resulta que  $Q(t) \rightarrow \infty$ , dado que es monótona creciente, por lo que el promedio tenderá a  $\infty$ . Se considera en este caso la integral desde 0 hasta T, dado que no hay tiempo negativo. Verificando la primera condición:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T E[Q(t)] dt = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \lambda t dt = \lim_{T \rightarrow \infty} \frac{1}{T} \cdot \lambda \cdot \frac{T^2}{2} = \infty$$

Por lo tanto, no puede estimarse la media mediante promedio temporal sobre una  $Q(t)$  cualquiera del ensamble.

Para estimar la autocorrelación con promedio temporal, debe resultar un proceso ergódico en la autocorrelación. Para esto debe verificarse que:

$$\lim_{T \rightarrow \infty} E[\langle R_{xx}(t) \rangle_T] = R_{xx}(t) \quad \lim_{T \rightarrow \infty} \sigma^2[\langle R_{xx}(t) \rangle_T] = \frac{1}{T} \int_{-T}^T \left(1 - \frac{|\tau|}{T}\right) C_{zz}(\tau) d\tau$$

Donde:

$$z(t) = x(t) \cdot x(t + \tau)$$

En la expresión de  $R_{QQ}(t_1, t_2)$  hallada anteriormente, se observa que depende de los tiempos en cuestión y no de la diferencia, por lo que el proceso no resulta ergódico en la autocorrelación. En consecuencia, no podrá estimarse tampoco mediante el promedio temporal de una función de muestra  $Q(t)$  cualquiera del ensamble.