Índice

1. f.d.p. Gaussiana

Para generar una f.d.p gaussiana X con media μ_x y varianza σ_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 Θ, la función de distribución de R:

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

Igualalndo a U_1 se puede despejar:

$$F_R(r) = U_1 \Longrightarrow R = \sqrt{-2 \cdot \ln(U_1)}$$

La variable Θ tiene distribución uniforme:

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

Igualando a U_2 se despeja:

$$F_\Theta(\theta) = U_2 \Longrightarrow \Theta = 2\pi U_2$$

Se tiene entonces que:

$$Y = R \cdot \cos(\Theta) \Longrightarrow Y = \sqrt{-2 \cdot \ln(U_1)} \cdot \cos(2\pi U_2)$$

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:

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

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 $\sigma_x^2 = 1$.

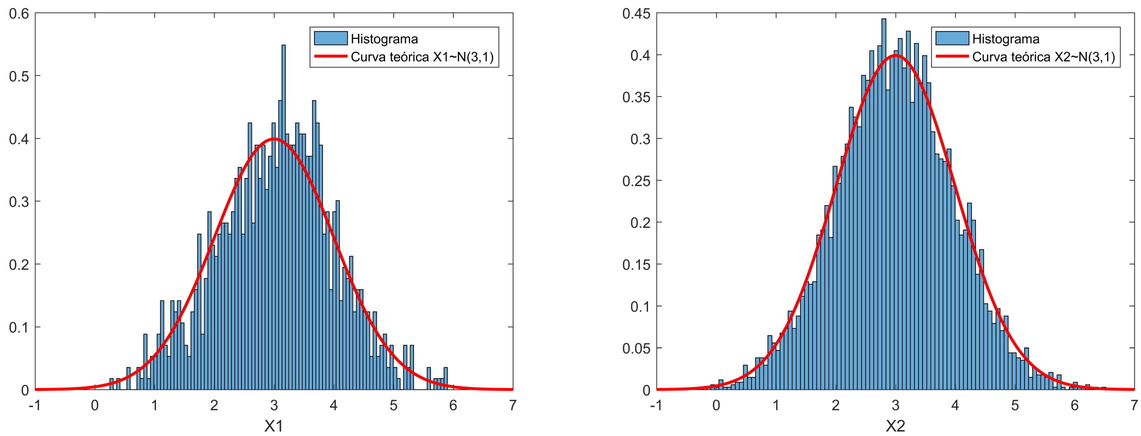

Figura 1: Comparación para vectores X_1 de 1000 elementos e X_2 de 5000 elementos

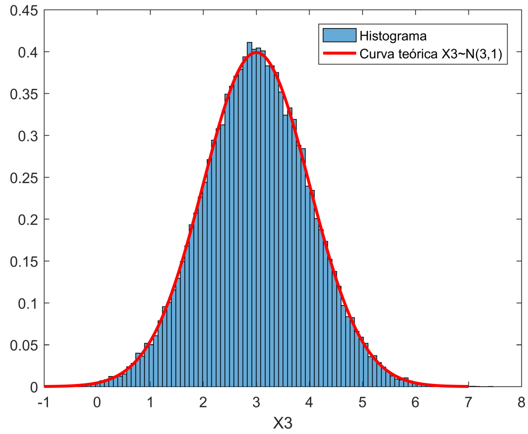

Figura 2: Comparación para vector X_3 de 50000 elementos

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.

```
%% Valores de media y varianza personalizados
mu_x = 3;
var_x = 1;

b = mu_x;
a = sqrt(var_x);

%% Item A
% Generacion de variables aleatorias uniformes en [0,1]
x1 = rand(1000,1);
x2 = rand(1000,1);
x3 = rand(5000,1);
x4 = rand(5000,1);
x5 = rand(50000,1);
x6 = rand(50000,1);

% Generacion de variables aleatorias gaussianas a partir de dos uniformes
y1 = a.*(sqrt(-2.*log(x1)).*cos(2.*pi.*x2)) + b;
y2 = a.*(sqrt(-2.*log(x3)).*cos(2.*pi.*x4)) + b;
y3 = a.*(sqrt(-2.*log(x5)).*cos(2.*pi.*x6)) + b;

% Curva teorica de comparacion
xi = (-4*a+b):0.001:(4*a+b);
z = normpdf(xi,mu_x,sqrt(var_x));

bars = 100; %Numero de barras

% Histogramas normalizados
figure;
histogram(y1,bars,'Normalization','pdf');
hold on;
plot(xi,z,'red','linewidth',2);
xlabel('X1');
legend('Histograma','Curva teorica X1~N(3,1)');
%print('X1_1000','-dpng','-r600');

figure;
histogram(y2,bars,'Normalization','pdf');
hold on;
plot(xi,z,'red','linewidth',2);
xlabel('X2');
legend('Histograma','Curva teorica X2~N(3,1)');
%print('X2_5000','-dpng','-r600');

figure;
histogram(y3,bars,'Normalization','pdf');
hold on;
plot(xi,z,'red','linewidth',2);
xlabel('X3');
legend('Histograma','Curva teorica X3~N(3,1)');
```

```
%print('X3_50000','-dpng','-r600');

mu_y = [mean(y1) mean(y2) mean(y3)];
var_y = [var(y1) var(y2) var(y3)];

%% Item B
% Calculo la diferencia en valor absoluto para comparar
comp_mu_y = [abs(mu_y(1) - mu_x) abs(mu_y(2) - mu_x) abs(mu_y(3) - mu_x)];
comp_var_y = [abs(var_y(1) - var_x) abs(var_y(2) - var_x) abs(var_y(3) - var_x)];
```

Listing 1: Codigo de implementación

1.2. Valores de μ y σ^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	μ_x			σ_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.

1.3. Estimaciones de μ_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.

```
%% Valores de media y varianza personalizados
mu_x = 3;
var_x = 1;

b = mu_x;
a = sqrt(var_x);

%% Item D
% Se hace 10000 veces para ver la distribucion de las medias en conjunto
mu_x_dist = zeros(10000,1);
for i=1:10000
    x1d = rand(10000,1);
    x2d = rand(10000,1);
    xd = a.*(sqrt(-2.*log(x1d)).*cos(2.*pi.*x2d)) + b;
    mu_x_dist(i) = mean(xd);
end
figure;
mean(mu_x_dist);
std(mu_x_dist);
histogram(mu_x_dist,bars,'Normalization','pdf');
xlabel('Medias de X');
%print('MediasX','-dpng','-r600');
```

Listing 2: Código de implementación

Realizando un histograma de los valores obtenidos, se tiene:

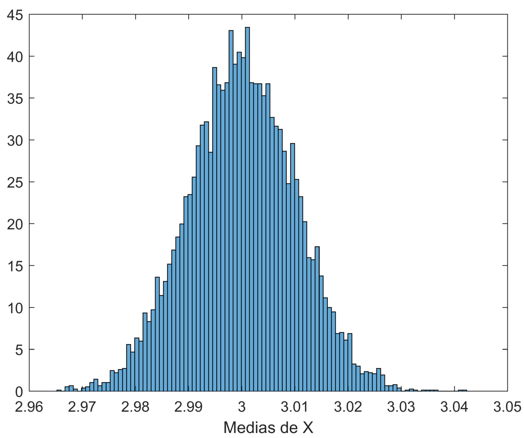

Figura 3: Histograma de las medias estimadas, con resultante $\mu = 3{,}0001$ - $\sigma = 0{,}01$

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

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

Como la distribución normal es reproductiva, la suma de N variables aleatorias con dicha distribución resulta tambi'en 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 μ y σ son:

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

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

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 μ_x - Ejemplo

Se toma ahora un caso particular, donde la cantidad de elementos 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:

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

Si se trabaja con la expresión:

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

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

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

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

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

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

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

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:

```
%% Valores de media y varianza personalizados
mu_x = 3;
var_x = 1;

%% Item E
%mx representa el promedio de la suma de N variables aleatorias
% Media: mu_x - Desviacion: sigma_x/sqrt(N)

N = 3382.3.*var_x./(mu_x.^2); %Analiticamente resulta eso y verifica

mx = makedist('Normal','mu',mu_x,'sigma',sqrt(var_x ./ N));

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	μ_x	μ_y	σ_x	σ_y	ρ
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 montecarlo (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.

```
%% [2] V.A Conjuntamente Gaussianas
clear Workspace;
clc;

% Generacion de uniformes [0 , 1]
N = 5000;
X1 = rand(N,1);
X2 = rand(N,1);
X3 = rand(N,1);
X4 = rand(N,1);

% Generacion de gaussianas estandard
Y1 = sqrt(-2.*log(X1)).*cos(2.*pi.*X2);
Y2 = sqrt(-2.*log(X3)).*cos(2.*pi.*X4);

%% CASO A)
u1a = 2;
u2a = 3;
sigma1a = 1;
sigma2a = 1;
roa = -0.5;

% Gaussianas con media y varianza personalizadas
Z1a = sigma1a.*Y1 + u1a;
Z2a = sigma1a.*Y2 + u2a;

% Gaussiana bivariable
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))) ));

figure;
scatter3(Z1a,Z2a,Z12a,2,'filled');
title('Caso A')
xlabel('Z1');
ylabel('Z2');
zlabel('f(Z1,Z2)');
set(gca,'XLim',[-4 8],'YLim',[-4 8])
view(-45,40)
print('Za','-dpng','-r600');

%% CASO B)
u1b = 2;
```

```
u2b = 3;
sigma1b = 1;
sigma2b = 0.5;
rob = −0.8;

% Gaussianas con media y varianza personalizadas
Z1b = sigma1b.∗Y1 + u1b;
Z2b = sigma1b.∗Y2 + u2b;

% Gaussiana bivariable
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))) ));

figure;
scatter3(Z1b,Z2b,Z12b,2,'filled');
title('Caso B')
xlabel('Z1');
ylabel('Z2');
zlabel('f(Z1,Z2)');
set(gca,'XLim',[−4 8],'YLim',[−4 8])
view(−45,40)
print('Zb','−dpng','−r600');
```

Listing 4: Código de implementación

Los resultados obtenidos para ambos casos se muestran a continuación.

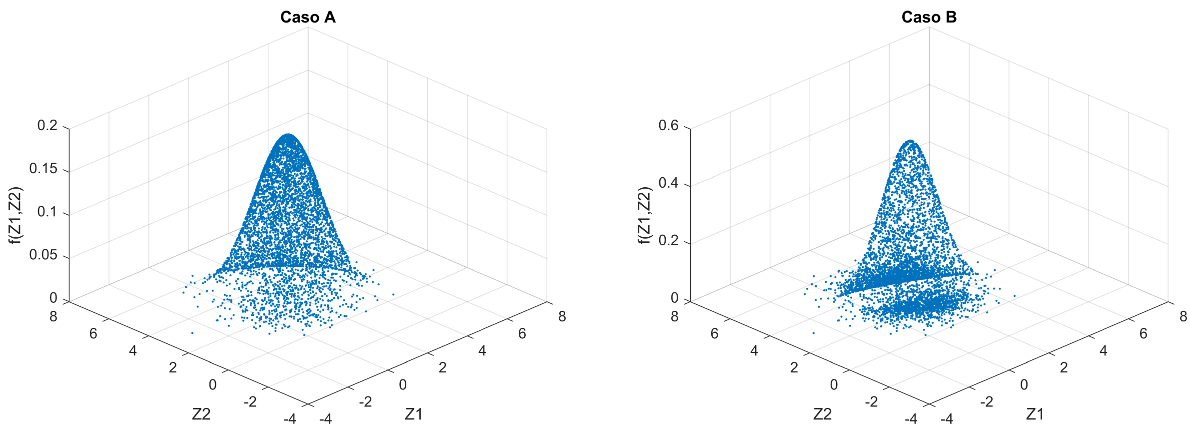

Figura 4: Nube de puntos resultante para ambos casos

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 ρ. 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 $|\rho| \longrightarrow 1$, 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