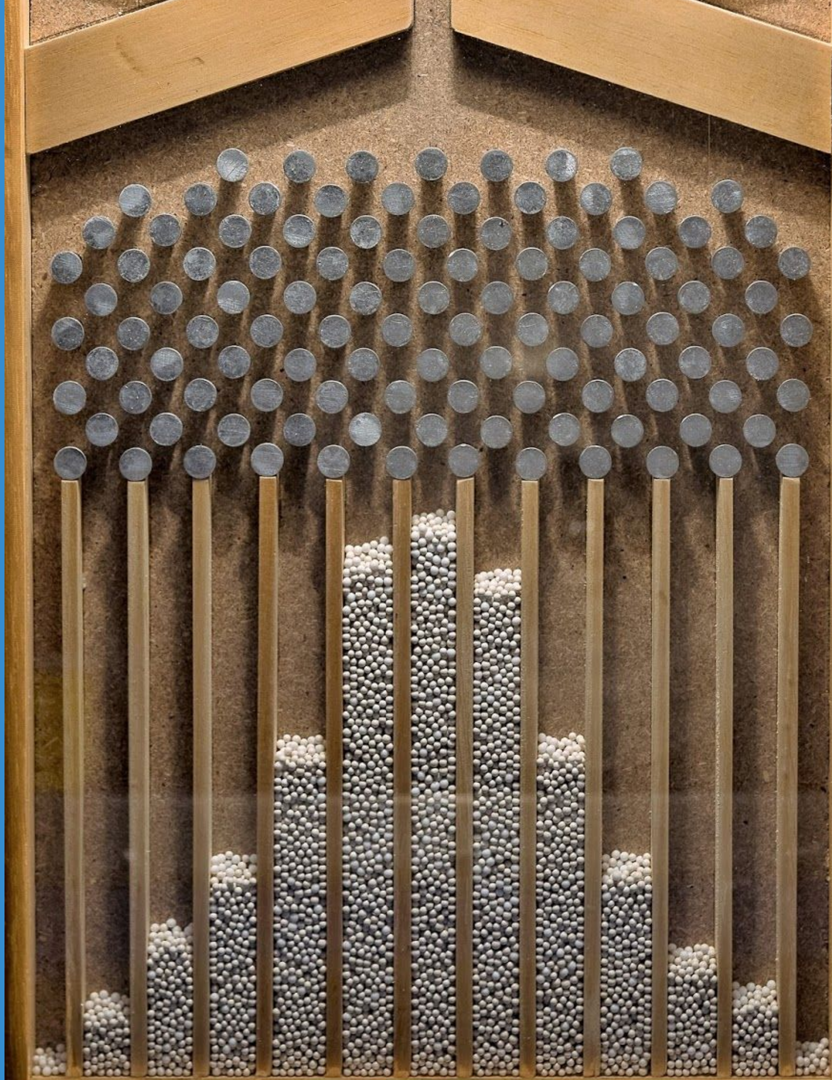
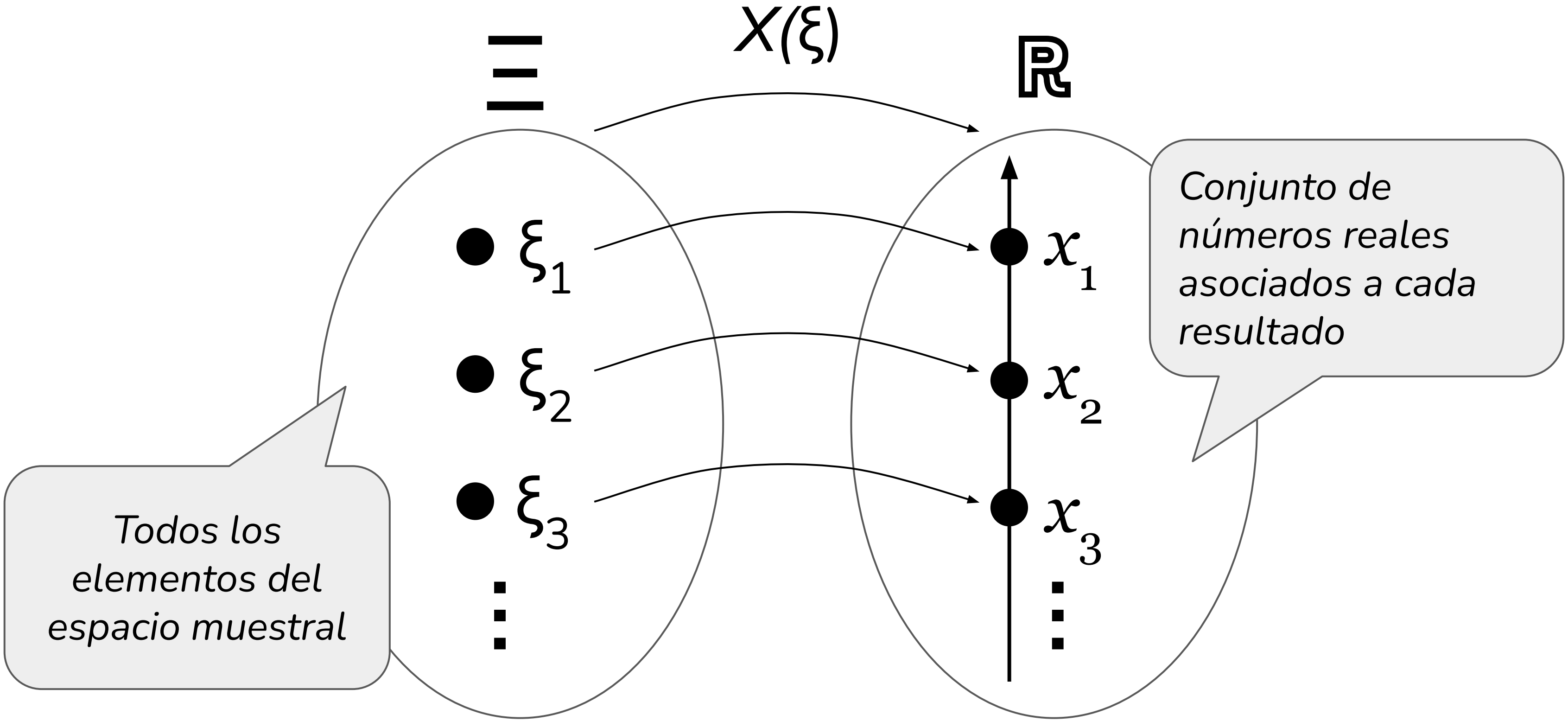
Procesos estocásticos (86.09)

- Variables y vectores aleatorios



Variables Aleatorias

Variables Aleatorias (VA)



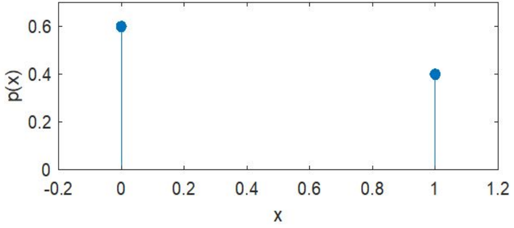
Distribuciones de probabilidad

Distribuciones de probabilidad – Funciones de masa

Bernoulli

$X \sim \text{Ber}(p)$

$$p_X(x) = \begin{cases} 1-p & \text{si } x = 0, \\ p & \text{si } x = 1. \end{cases} \qquad p \in [0,1]$$

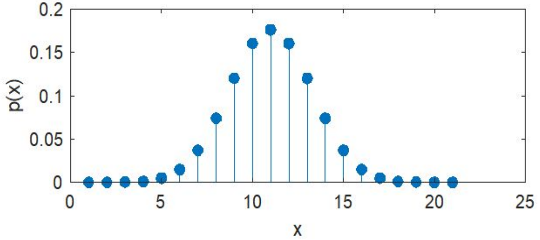


Suma de Bernoullis i.i.d.

Binomial

$X \sim Bin(p)$

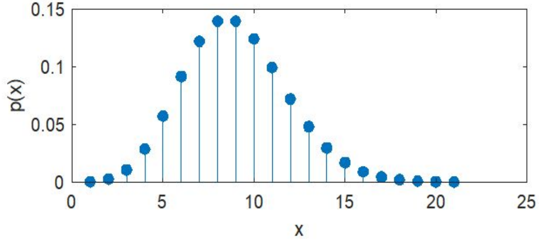
$$p_X(x) = \binom{n}{x} p^x (1-p)^{n-x} \qquad p \in [0,1] \qquad x = 0, 1, \ldots, n$$



Poisson

$X \sim Poisson(\lambda)$

$$p_X(x) = \frac{\lambda^x e^{-\lambda}}{x!} \qquad \lambda > 0 \qquad x = 1, 2 \ldots$$

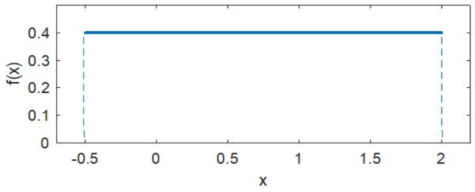


Distribuciones de probabilidad – Funciones de densidad

Uniforme

$X \sim U(a, b)$

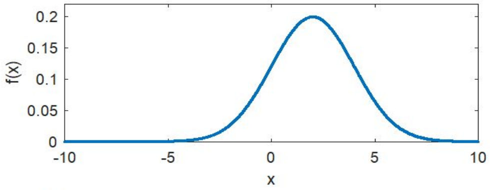
$$f_X(x) = \frac{1}{b-a}, \qquad a < x < b.$$



Normal

$X \sim N(\mu, \sigma)$

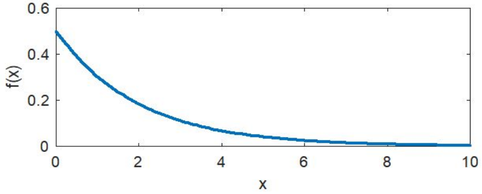
$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \qquad x \in \mathbb{R}$$



Exponencial

$X \sim Exp(\lambda)$

$$f_X(x) = \lambda\, e^{-\lambda x}, \qquad x \geq 0$$

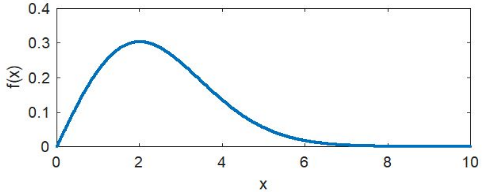


Rayleigh

$X \sim Rayl(\sigma)$

$$f_X(x) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, \qquad x \geq 0 \quad \sigma > 0$$

derivada de la gaussiana?



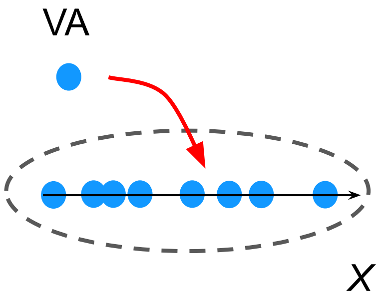
Simulación de Variables Aleatorias

Simulación de Variables Aleatorias

Funciones de Matlab/Octave para generar muestras de distribuciones comunes:

```
x = rand(1,N);                  % Uniforme estándar
x = unifrnd(a, b, 1, N);=       % Uniforme
x = randn(1,N);                 % Normal estándar
x = normrnd(mu, sig, 1, N);     % Normal
x = binornd(n, p, 1, N);        % Binomial
x = poissrnd(mu, 1, N);         % Poisson (mu = lambda)
x = exprnd(mu, 1, N);           % Exponencial(mu = 1/lambda)
x = raylrnd(b, 1, N);           % Rayleigh
```

= rand(1,N)
si a=0 y b=1

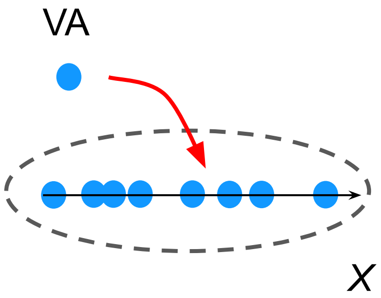


N muestras con cierta distribución (realizaciones)

Simulación de Variables Aleatorias

Funciones de Python para generar muestras de distribuciones comunes:

```
import numpy as np

x = np.random.uniform(a, b, N)               # Uniforme

x = np.random.normal(mu, sig, N)             # Normal

x = np.random.binomial(n, p, N)              # Binomial

x = np.random.poisson(mu, N)                 # Poisson

x = np.random.exponential(mu, N)             # Exponencial

x = np.random.rayleigh(scale=b, size=N)      # Rayleigh
```



N muestras con cierta distribución (realizaciones)

Simulación de Variables Aleatorias (Matlab)

```
>> x = rand(1,5)

x =

    0.0975    0.2785    0.5469    0.9575    0.9649
```

Realizaciones independientes

```
>> x = randn(1,6)

x =

   -1.3499    3.0349    0.7254   -0.0631    0.7147   -0.2050
```

Simulación de Variables Aleatorias (Python)

```
import numpy as np
x = np.random.uniform(0, 1, 5) # x = np.random.rand(5)
print(x)
```

```
[0.24230626 0.56564437 0.14763606 0.97714927 0.31140779]
```

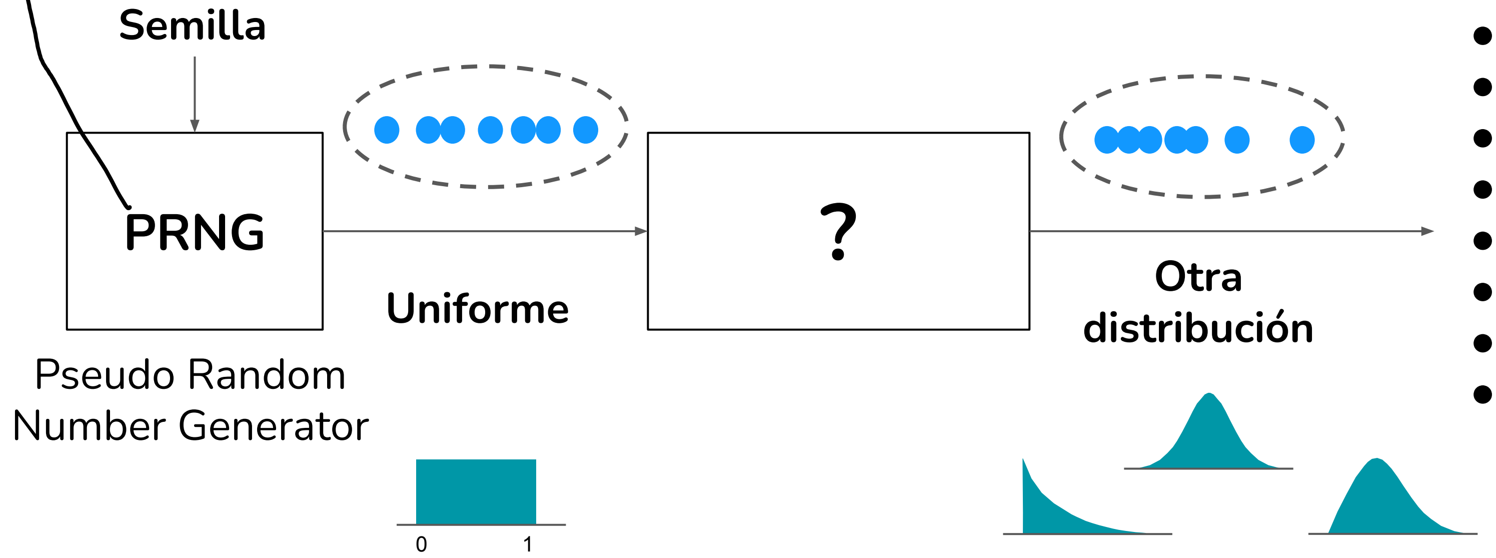
```
x = np.random.normal(0, 1, 6) # x = np.random.randn(6)
print(x)
```

```
[ 1.03918166  0.50593943 -0.35094316  1.12961749 -0.73663994 -0.6805176 ]
```

Simulación de Variables Aleatorias

¿Cómo generar muestras de diferentes distribuciones?

PORONGA



- Bernoulli
- Binomial
- Poisson
- Exponencial
- Gaussiana
- Gamma
- Beta
- Etc.

Generadores de Números Pseudoaleatorios (PRNG)

- Los números aleatorios son esenciales en simulaciones de Monte Carlo, criptografía, aprendizaje automático, etc.
- Los PRNG generan secuencias que parecen aleatorias pero son deterministas. ¿?
- Se basan en una semilla inicial.

Generadores de Números Pseudoaleatorios (PRNG)

Algoritmo Congruencial Lineal (LCG)

$$X_{n+1} = (a \cdot X_n + c) \bmod m$$

X_0 es la semilla, algún numero dado

- LCG genera todos los enteros posibles en {0,1,...,m-1} antes de repetirse.
- Algunos valores comunes (MINSTD) son:

  $a = 16807$, $c=0$, $m = 2^{31}-1$.

- Otros generadores: Mersenne Twister, Xorshift, Permuted Congruential Generator, etc.

¿Qué tiene de aleatorio esto? → No mucho, es pseudoaleatorio → "La tautología al palo" J.L.

Simulación de Variables Aleatorias

1. **Método de la transformación inversa.**
2. Método de aceptación rechazo.
3. Método de la transformación de Box-Muller
4. Otros

Simulación de VA – Método de la Transformación Inversa

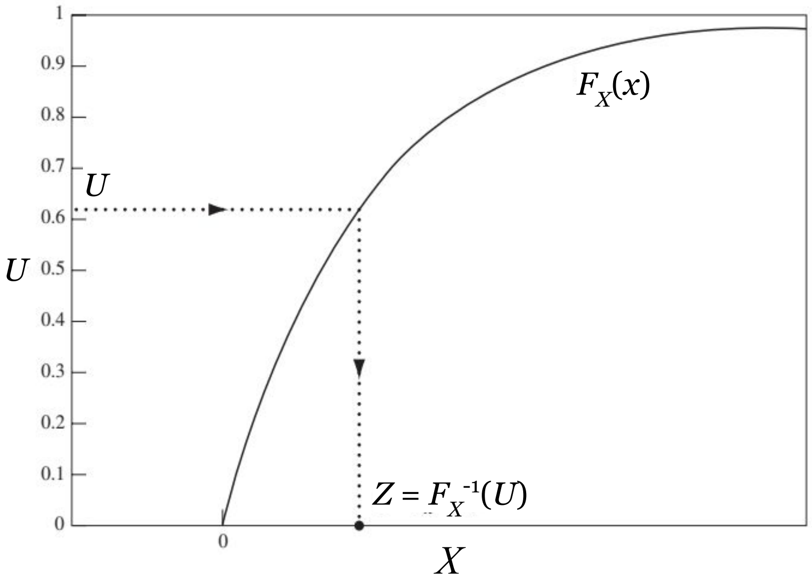
Queremos una transformación g: $\mathbb{R} \to \mathbb{R}$ para obtener realizaciones de una VA X (de cierta distribución) a partir de una VA uniforme $U \sim U(0,1)$.

Requerimientos

- $F_X(x)$ debe ser una función continua
- $F_X(x)$ debe ser monótona creciente
- $F_X(x)$ debe ser invertible

Procedimiento del método

1. Generar un número random $U \sim U(0,1)$
2. Obtener una realización de $Z = F_X^{-1}(U)$



Simulación de VA – Método de la Transformación Inversa

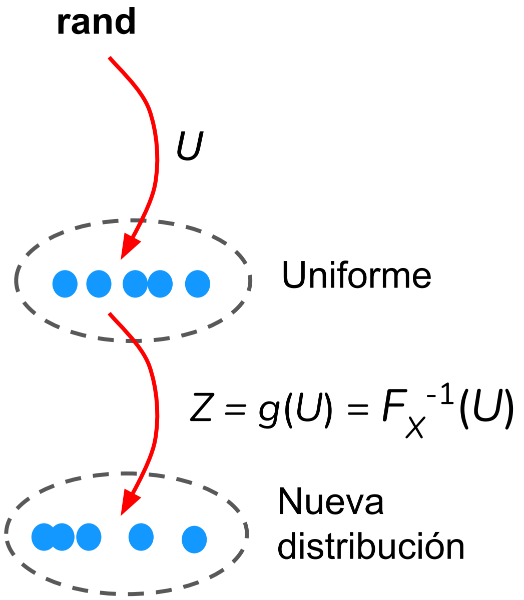
Queremos una transformación g: $\mathbb{R} \rightarrow \mathbb{R}$ para obtener realizaciones de una VA X (de cierta distribución) a partir de una VA uniforme $U \sim U(0,1)$.

Requerimientos

- $F_X(x)$ debe ser una función continua
- $F_X(x)$ debe ser monótona creciente
- $F_X(x)$ debe ser invertible

Procedimiento del método

1. Generar un número random $U \sim U(0,1)$
2. Obtener una realización de $Z = F_X^{-1}(U)$



Simulación de VA – Método de la Transformación Inversa

Ejemplo: Generar una distribución exponencial

Función de densidad de X ~ Exp(λ):

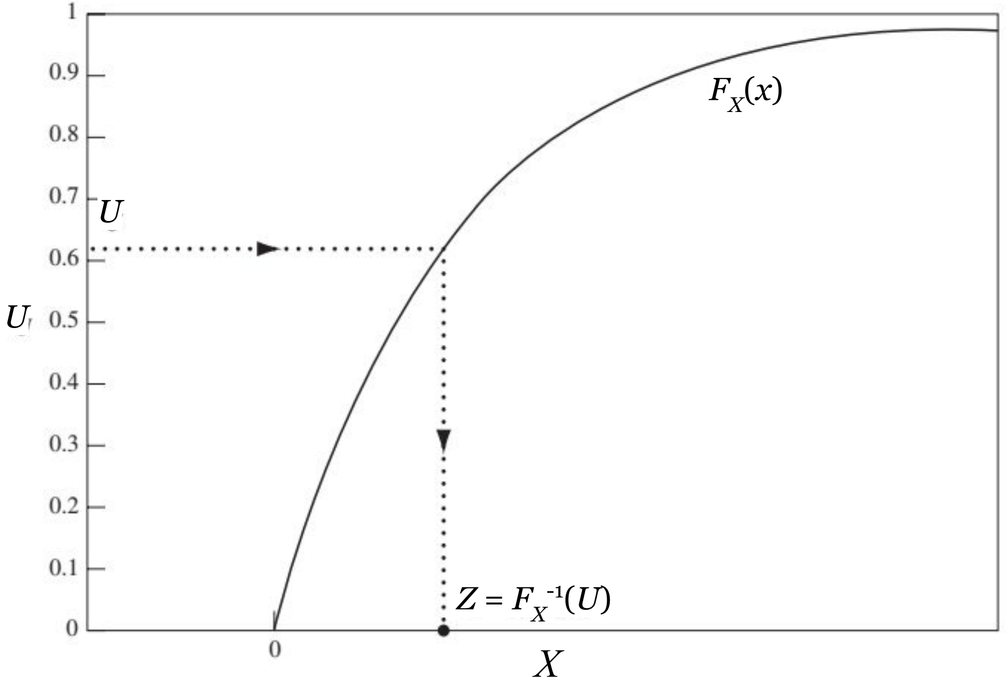
$$f_X(x) = \lambda e^{-\lambda x} \quad ; x \geq 0$$

Función de distribución de X:

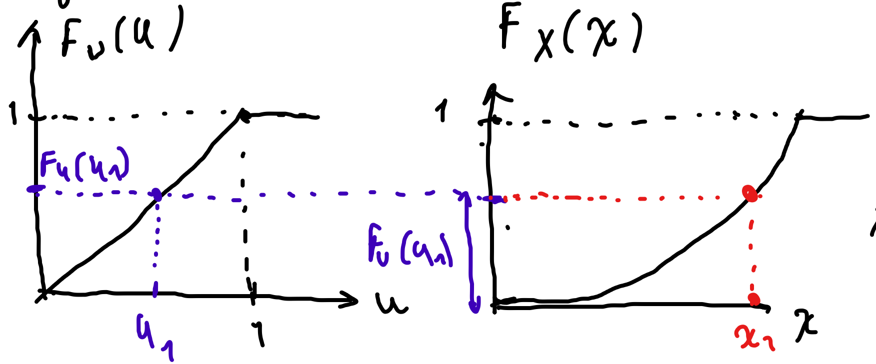
$$F_X(x) = \int_0^x f_X(x)dt = 1 - e^{-\lambda x}$$

Obtener Z con inversa de F_X(x):

$$Z = F_X^{-1}(U) = -\frac{\ln(1-U)}{\lambda}$$



Algunas anotaciones:



Realizo un experimento aleatorio uniforme estándar
La probabilidad de observar u1 o menos es FU(u1)
Hago de cuenta que eso es la realización de x1, que tienen probabilidad FX(x1)
¿A qué x corresponde la probabilidad de observar ese evento o menor?
Al pedo todo este texto, el dibujito es más claro

Aguanten los dibujitos

Histogramas

Histogramas

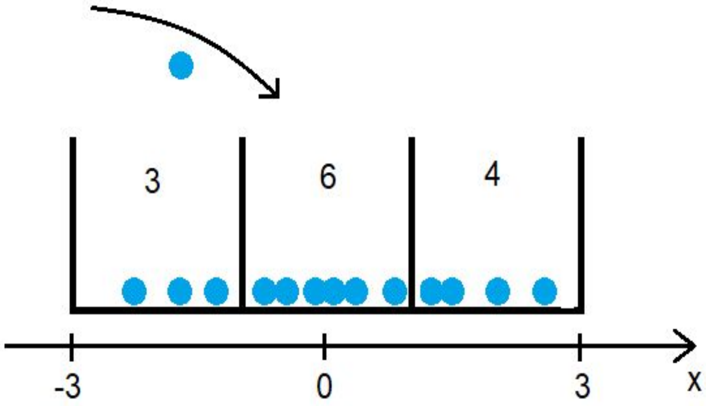
- Permite representar una aproximación de la función de **densidad / masa de probabilidad**

- Representa la **frecuencia de ocurrencia** de una dada VA dentro de cada bin

- Divide el eje de la VA en **intervalos (bins)**

- Puede representarse **normalizada** con área unitaria.

¿Tanto quilombo para contar bochas?



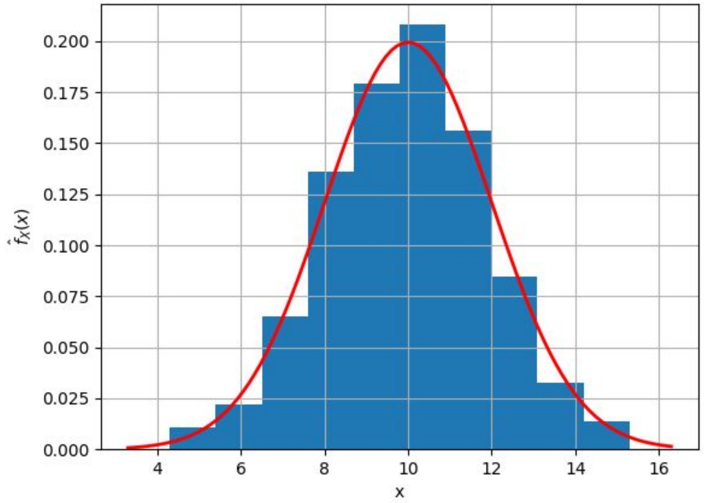
$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n}\sum_{j}\mathbf{1}\{x_i \in B_j\}\cdot\mathbf{1}\{x \in B_j\},$$

$$B_j = [m_j - \frac{h}{2}, m_j + \frac{h}{2}]$$

h y m_j son la longitud y centro del intervalo.

Histogramas

- Permite representar una aproximación de la función de **densidad**
- Representa la **frecuencia de ocurrencia** de una dada VA dentro de cada bin
- Divide el eje de la VA en **intervalos (bins)**
- **Debe** representarse **normalizada** con área unitaria.



X

Histogramas

Matlab

abre q' no tengo matlab

```
histogram(x)              % Por defecto gráfica frecuencia de ocurrencia con
                          % bins en automático(se ajusta)
```

No normaliza

```
histogram(x, bins)        % Se puede especificar la cantidad de bins

histogram(x,bins,'Normalization','pdf') % Normalización (para comparar
                                        % con la función de densidad)
```

Así hay q' ponerlo en el TP

Matlab/Octave

```
h = hist(x, bins);          % Guardar en una variable los valores de hist()

[h, xc] = hist(x, bins); % Guardar histograma normalizado y graficar
bar(xc, h /(sum(h)*(xc(2)-xc(1))));
```

Histogramas

Python

```
import matplotlib as plt

plt.hist(x) # Por defecto gráfica frecuencia de ocurrencia con 20 bins

plt.hist(x, bins) # Se puede especificar la cantidad de bins

plt.hist(x, bins, density = True)  # Histograma normalizado

                                   # PDF aprox. (el área cierra a 1)


# Histograma para distribuciones discretas

plt.hist(x, bins = np.arange(xmin, xmax + 1) - 0.5, density = True)
```

Bromomento

Momentos de una variable aleatoria

Momentos de una variable aleatoria

Esperanza

$$\mathbb{E}[X] = \mu_X$$

Varianza (medida de dispersión)

$$\mathbb{V}[X] = \sigma_X^2 = \mathbb{E}\left[(X - \mu_X)^2\right]$$

Covarianza entre dos VA

$$\mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$$

Los cuatro jinetes del apocalipsis

Coeficiente de correlación

$$\rho(X, Y) = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \, \sigma_Y}$$

$-1 \leq \rho \leq 1$

Estimación de algunas medidas estadísticas (Matlab)

Funciones de Matlab para **estimar** algunas de las medidas estadísticas

vector de realizaciones.

```
mean(x);            % Media de x

var(x);             % Varianza de x

std(x); = √var(x)   % Desvío de x

corrcoef(x, y); % Coeficiente de correlación entre x e y (columnas)
```

Estimación de algunas medidas estadísticas (Matlab)

Funciones de Matlab/Octave para estimar algunas de las medidas estadísticas

```
>> mean(x)

ans =

    0.6462
```

```
>> var(x)

ans =

    0.1242
```

```
>> std(x)

ans =

    0.3524
```

```
>> corrcoef(x,y)

ans =

    1.0000    0.2038
    0.2038    1.0000
```

Estimación de algunas medidas estadísticas (Python)

Funciones de Python para **estimar** algunas de las medidas estadísticas

```
import numpy as np

mean_x = np.mean(x)                    # Media

var_x = np.var(x)                      # Varianza

std_x = np.std(x)                      # Desvío

corr_coef = np.corrcoef(x, y) # Coeficiente de correlación
```

Estimación de algunas medidas estadísticas (Python)

```
import numpy as np
x = np.random.normal(0, 1, 1000)
y = np.random.normal(0, 1, 1000)
```

```
mean_x = np.mean(x)
print(mean_x)
```

```
-0.014969036499220583
```

```
std_x = np.std(x)
print(std_x)
```

```
1.000321491280756
```

```
var_x = np.var(x)
print(var_x)
```

```
1.0006430859181559
```

```
corr_coef = np.corrcoef(x, y)
print(corr_coef)
```

```
[[ 1.         -0.04528167]
 [-0.04528167  1.        ]]
```

Actividad 1

Actividad 1
Variables aleatorias e Histogramas

Genere N experimentos de una variable aleatoria Rayleigh con parámetro $b = 0.5$. Grafique su histograma para los siguientes parámetros:

1. $N = 100$, bins = 10
2. $N = 100$, bins = 30
3. $N = 10000$, bins = 30

Actividad 2

Actividad 2
Variables aleatorias e Histogramas

Sea x una variable aleatoria exponencial, $X \sim \mathrm{Exp}(\lambda)$, de parámetro $\lambda = 0.5$

1. Genere $N = 10^4$ muestras de X (usando el método de **transformación inversa**).

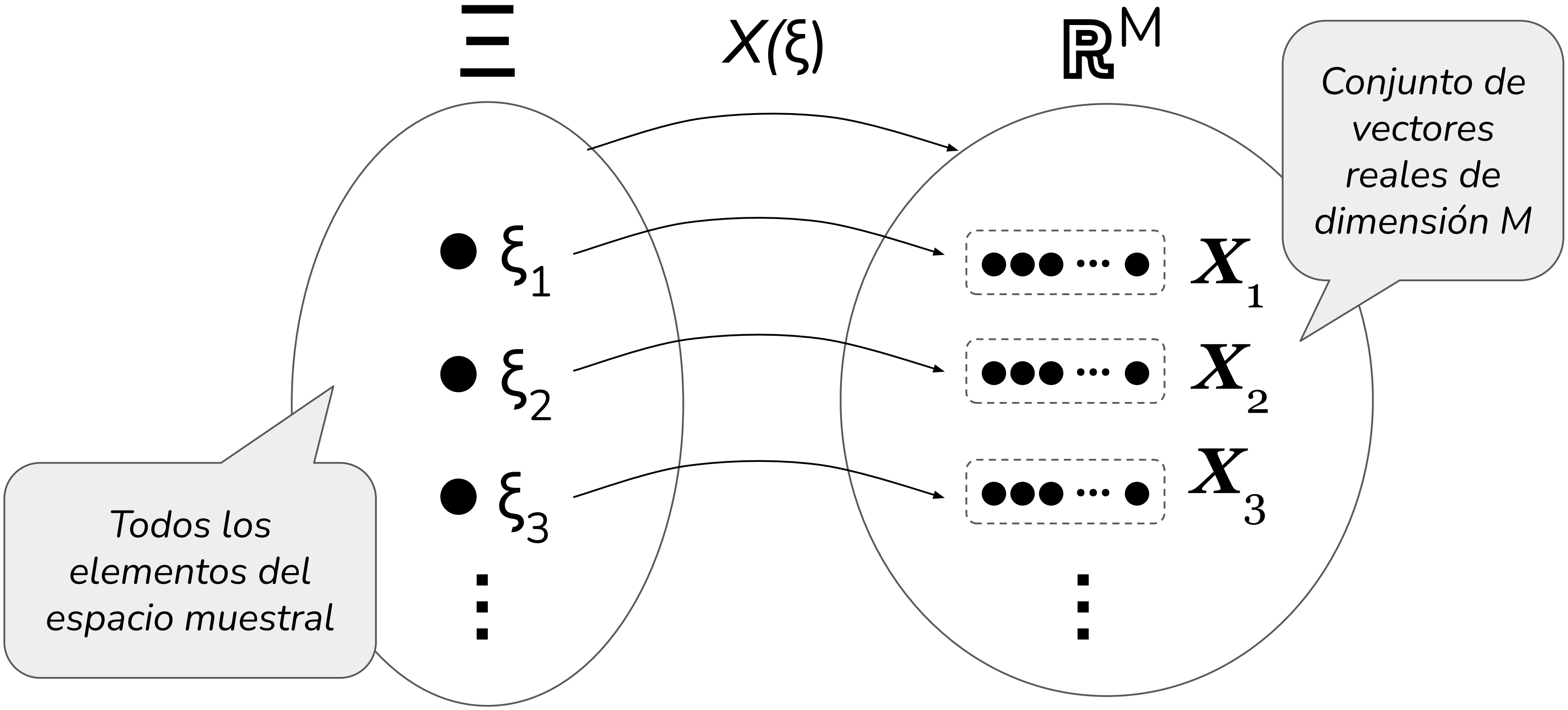
2. Estime la media y la varianza muestrales de X y comparelas con las teóricas ($\mu = 1/\lambda$, $\sigma^2 = 1/\lambda^2$).

3. Construya el **histograma** de las muestras de X. Normalice el histograma para que tenga área 1. Compare la función obtenida con la función de densidad de probabilidad teórica.

⚠ HACER EN CASA ⚠

Vectores aleatorios

Variables Aleatorias (VA)



Momento de primer orden de un Vector Aleatorio

Vector aleatorio

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}$$

VA $\in \mathbb{R}$
(escalar)

VA $\in \mathbb{R}^M$
(vectorial)

Media de un vector aleatorio $\boldsymbol{X}$

$$\mathbb{E}[\mathbf{X}] = \mu_{\mathbf{X}} = \mathbb{E}\left[\begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}\right] = \begin{bmatrix} \mathbb{E}[X_1] \\ \vdots \\ \mathbb{E}[X_n] \end{bmatrix}$$

Simulación de Vectores aleatorios

Simulación de un Vector Aleatorio (Matlab)

Realizaciones iid de un VeA uniforme de dimensión 1x5

```
>> x = rand(1,5)

x =

   0.0975    0.2785    0.5469    0.9575    0.9649
```

Esta es UNA realización del V.e.A.

Realizaciones iid de un VeA normal de dimensión 1x5

```
>> x = randn(1,5)

x =

  -1.3499    3.0349    0.7254   -0.0631    0.7147
```

Simulación de un Vector Aleatorio (Matlab)

2 realizaciones iid de un VeA uniforme de dimensión 1x5

```
>> x = rand(2,5)

x =

    0.4519    0.8644    0.5398    0.6779    0.7095

    0.7685    0.7278    0.7395    0.5265    0.3678
```

También puede verse como **5 realizaciones** iid de un VeA uniforme de dimensión 2x1

```
>> x = rand(2,5)

x =

    0.4519    0.8644    0.5398    0.6779    0.7095

    0.7685    0.7278    0.7395    0.5265    0.3678
```

Simulación de un Vector Aleatorio (Matlab)

2 realizaciones iid de un VeA uniforme de dimensión 1x5

```
>> x = rand(2,5)

x =

    0.4519    0.8644    0.5398    0.6779    0.7095

    0.7685    0.7278    0.7395    0.5265    0.3678
```

Realizaciones de un VeA

También puede verse como **5 realizaciones** iid de un VeA uniforme de dimensión 2x1

```
>> x = rand(2,5)

x =

    0.4519    0.8644    0.5398    0.6779    0.7095

    0.7685    0.7278    0.7395    0.5265    0.3678
```

Simulación de un Vector Aleatorio (Matlab)

2 realizaciones iid de un VeA uniforme de dimensión 1x5

```
>> x = rand(2,5)

x =

    0.4519    0.8644    0.5398    0.6779    0.7095
    0.7685    0.7278    0.7395    0.5265    0.3678
```

También puede verse como **5 realizaciones** iid de un VeA uniforme de dimensión 2x1

```
>> x = rand(2,5)

x =

    0.4519    0.8644    0.5398    0.6779    0.7095
    0.7685    0.7278    0.7395    0.5265    0.3678
```

Realizaciones
de un VeA

Actividad 3

Actividad 3
Vectores aleatorios

Genere N = 200 muestras para definir los siguientes vectores aleatorios.:

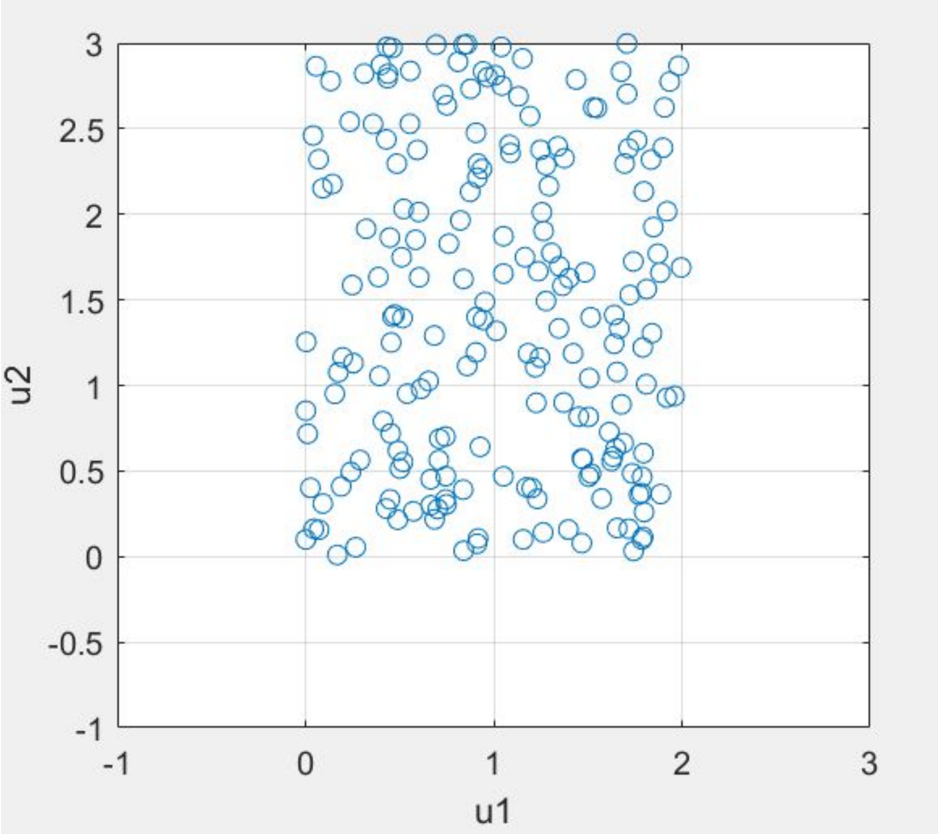
1. Para el vector $\boldsymbol{U} = [U_1\ U_2]^T$, genere dos variables uniformes, $U_1 \sim U(0;2)$ y $U_2 \sim U(0;3)$.
2. Para el vector $\boldsymbol{X} = [X_1\ X_2]^T$ genere muestras de las variables X_1 y X_2 a partir de U_1 y U_2, tal que $X_1 = 0.5\,U_1 - 0.3\,U_2$ y $X_2 = 0.7\,U_1 + 0.2\,U_2$.
3. Para el vector $\boldsymbol{Y} = [Y_1\ Y_2]^T$, genere muestras de las variables Y_1 y Y_2 a partir de U_1 y U_2, tal que $Y_1 = 1.2\,U_1 - 0.1\,U_2$ y $Y_2 = U_1 + 0.1\,U_2$.

Haga el gráfico de dispersión (ej: `scatter(u1, u2)`) y calcule el coeficiente de correlación para cada uno de casos. Terminar en Tu parte casa

Nota: defina el límite de los ejes del gráfico con axis([-1 3 -1 3]).

Actividad 4
Vectores aleatorios

$\varrho(U_1, U_2) =$
-0.0112



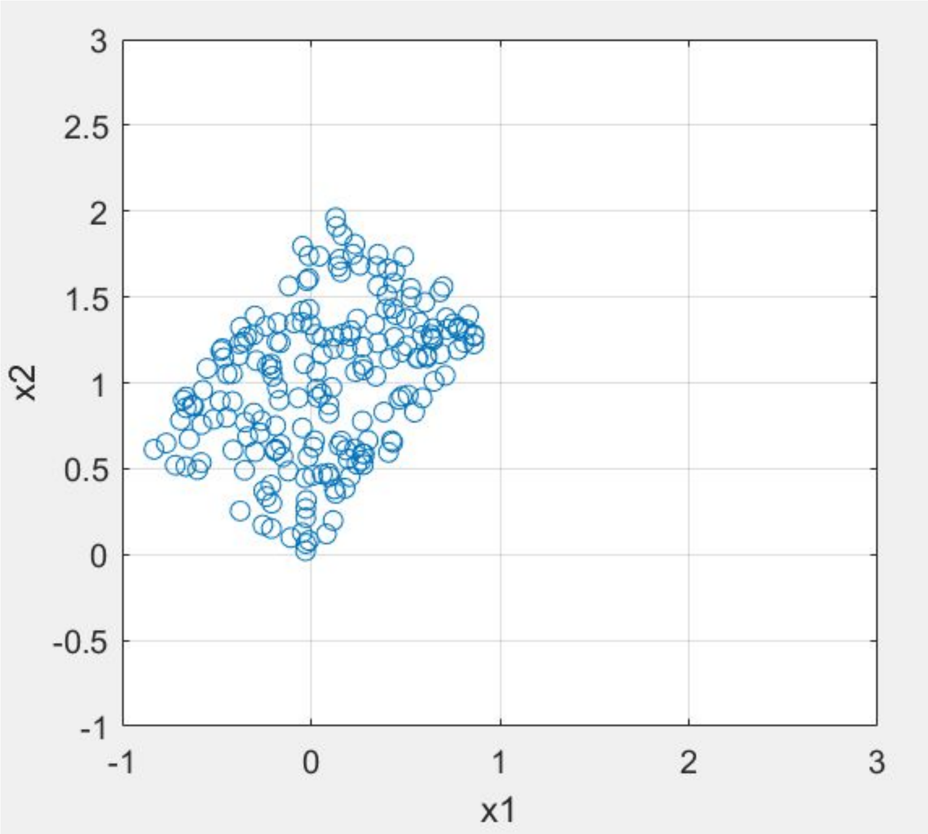
Actividad 4
Vectores aleatorios

$\varrho(X_1, X_2) = 0.3628$

3 palabras q' asustan a todo estudiante:

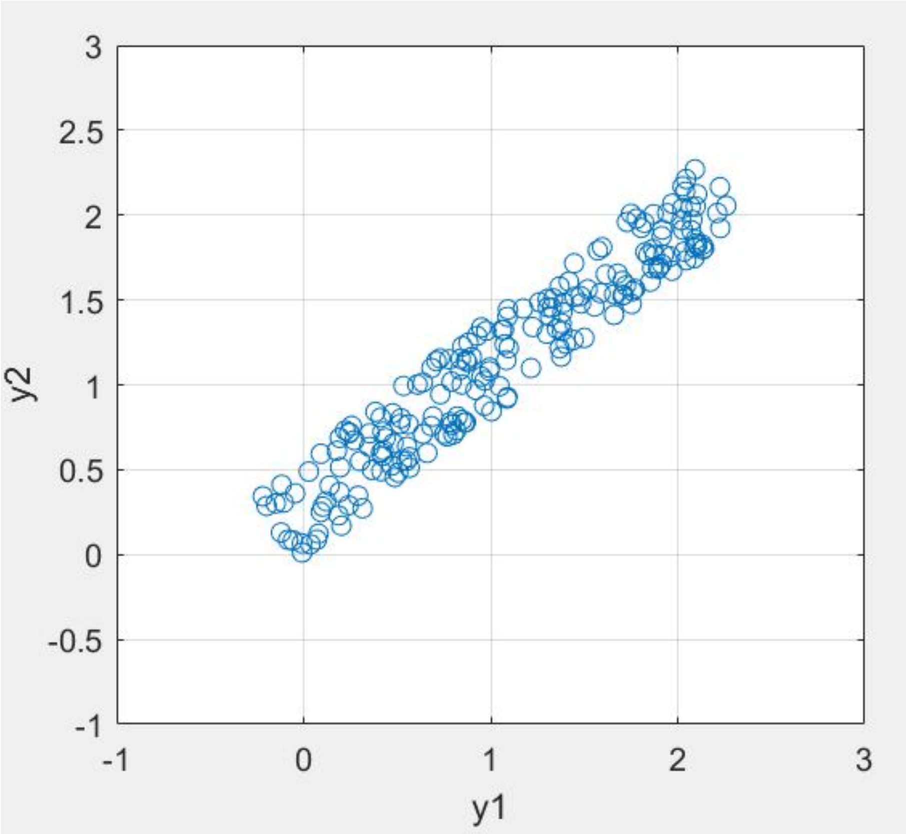
MÉTODO DEL

JACOBIANO



Actividad 4
Vectores aleatorios

$$\varrho(Y_1, Y_2) = 0.9570$$



Ejercicio

Ejercicio (Transformación Box Muller)

Sean U_1, U_2 dos variables aleatorias independientes uniformes ~U(0; 1).

1. Halle la densidad conjunta de las variables:

$$\begin{cases} R = \sqrt{-2\ln(U_1)} \\ \Theta = 2\pi U_2. \end{cases}$$

   Verifique que R tiene distribución Rayleigh, que Θ es uniforme y que son independientes (¿por que?). ´

2. Halle la densidad conjunta de las variables:

$$\begin{cases} Z_1 = R\cos\Theta \\ Z_2 = R\sin\Theta \end{cases}$$

y demuestre que se trata de variables normales estándar independientes. ´