Facultad de Matemática, Astronomía, Física y Computación, U.N.C.

Computación

Aula Virtual: https://famaf.aulavirtual.unc.edu.ar/course/view.php?id=747

Resguardo tutoriales: https://www.famaf.unc.edu.ar/∼moreschi/docencia/Computacion/

Tutorial Problemas 1 de la Guía N° 6

Problema 1:

Cuando se estudia probabilidad y procesos estocásticos, aparece naturalmente la función gaussiana o de Gauss o normal, definida por:

$$g(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{(y-\mu)^2}{2\sigma^2}};$$

que está centrada en μ con ancho característico σ, que es su desviación estándar. A σ^2 se la denomina la varianza de la distribución.

Suponga un proceso estocástico que se repite N veces y tiene una distribución de probabilidad $P(y)$ dada por la función de Gauss, que asignamos a pasos en el espacio de las $x \in [x_{\min}, x_{\max}]$.

a) Usando PYTHON, simule una distribución para este proceso con los comandos:

```
import numpy as np

xmin = 0.
xmax = 1.4

# elección de la semilla ----------------------------------------
np.random.seed(11)

# elección de Nrep ----------------------------------------------
Nrep = 1001
x = np.linspace(xmin, xmax, Nrep)
y = np.random.randn(Nrep)
```

donde estamos usando la función RANDN del paquete RANDOM de NUMPY y asignaremos la distribución al espacio de las x.

b) Defina una función $gau(yy, prom, ancho)$ que sea la función gaussiana evaluada en yy, centrada en $prom$ y con desviación estandar $ancho$.

c) Genere un gráfico de y vs x, mostrando el resultado en pantalla y guardando el mismo en un archivo con nombre 'graficos/p1-normal-'+str(Nrep)+'.png'

d) Genere un gráfico del histograma de este proceso y de la función de Gauss, con la elección de valores $pro = 0.$ y $anc = 1.$, haciendo uso de los comandos:

```
numbines = 50

n, bins, patches = plt.hist(y, numbines, density=False, facecolor='g',
alpha=0.75, label='histograma')
suma = n.sum()
hbin = (bins[-1]-bins[0])/numbines
```

```
integ = suma*hbin
plt.plot(bins , (integ*gau(bins,pro,anc)),'r',
         label='integ * Gausiana mu='+str(pro)+', sigma='+str(anc) )
```

muéstrelo en pantalla y guarde el gráfico en un archivo con el nombre 'graficos/p1-histograma-'+str(Nrep)+'.png'

e) Genere un gráfico del histograma relativo de este proceso y de la función de Gauss, con la elección de valores $pro = 0.$ y $anc = 1.$, haciendo uso de los comandos:

```
n, bins, patches = plt.hist(y, numbines, density=True, facecolor='g',
alpha=0.75, label='histograma')
plt.plot(bins , (gau(bins,pro,anc)),'r',
         label='Gausiana mu='+str(pro)+', sigma='+str(anc) )
```

muéstrelo en pantalla y guarde el gráfico en un archivo con el nombre 'graficos/p1-histograma-relativo-'+str(Nrep)+'.png'

f) Genere un gráfico del histograma de la variable x con el comando:

```
n, bins, patches = plt.hist(x, numbines, density=False, facecolor='g',
alpha=0.75, label='histograma')
```

muéstrelo en pantalla y guarde el gráfico en un archivo con el nombre 'graficos/p1-histograma-x-'+str(Nrep)+'.png'

g) Repita todo lo anterior ahora con la elección $N = 10001$ y compare las simulaciones con la función gaussiana.

Los gráficos deben ser completados con título, etiquetas de curvas, etiquetas de ejes, etc.

Tutorial:

- Guarde en el archivo p1.py las siguientes instrucciones:

```
"""
definimos una variable con random y
hacemos el histograma correspondiente
"""
import numpy as np
import matplotlib.pyplot as plt

def gau(yy,prom,ancho):
    return np.exp(-(yy-prom)**2/(2.*ancho**2))/( np.sqrt(2.*np.pi)*ancho )


numbines = 50
xmin = 0.
xmax = 1.4
pro = 0.
anc = 1.

# elección de la semilla ----------------------------------------
```

```
np.random.seed(11)

# elección de Nrep ----------------------------------------------------
Nrep = 1000
x = np.linspace(xmin, xmax, Nrep)
y = np.random.randn(Nrep)
print('x =',x)
print('y =',y)
print('len(x) =',len(x))
print('len(y) =',len(y))

plt.figure( figsize=(10, 7.5) )
plt.title('Distribución normal con '+str(Nrep)+' puntos')
plt.xlabel('x')
plt.ylabel('y')
plt.grid()
plt.plot(x , y,'o-', label='normal')
plt.legend(loc="best")
plt.savefig('graficos/p1-normal-'+str(Nrep)+'.png', dpi=100)
plt.show()


plt.figure( figsize=(10, 7.5) )
plt.title('Histograma de la distribución normal con '+str(Nrep)+' puntos')
plt.xlabel('y')
plt.ylabel('dn/dy')
plt.grid()
n, bins, patches = plt.hist(y, numbines, density=False, facecolor='g',
                            alpha=0.75, label='histograma')
suma = n.sum()
hbin = (bins[-1]-bins[0])/numbines
integ = suma*hbin
plt.plot(bins , (integ*gau(bins,pro,anc)),'r',
         label='integ * Gausiana mu='+str(pro)+', sigma='+str(anc) )
plt.legend(loc="best")
plt.savefig('graficos/p1-histograma-'+str(Nrep)+'.png', dpi=100)
plt.show()


plt.figure( figsize=(10, 7.5) )
plt.title('Histograma relativo de la distribución normal con '
    +str(Nrep)+' puntos')
plt.xlabel('y')
plt.ylabel('(1/N) dn/dy')
plt.grid()
n, bins, patches = plt.hist(y, numbines, density=True, facecolor='g',
                            alpha=0.75, label='histograma')
plt.plot(bins , (gau(bins,pro,anc)),'r',
         label='Gausiana mu='+str(pro)+', sigma='+str(anc) )
plt.legend(loc="best")
plt.savefig('graficos/p1-histograma-relativo-'+str(Nrep)+'.png', dpi=100)
plt.show()

#print('     suma =',suma)
#print('     hbin =',hbin)
#print('suma*hbin =',(suma*hbin))
```

```
#print('      n =',n)
#print('   bins =',bins)
#print('patches =',patches)

plt.figure( figsize=(10, 7.5) )
plt.title('Histograma de x')
plt.xlabel('x')
plt.ylabel('dn/dx')
plt.grid()
n, bins, patches = plt.hist(x, numbines, density=False, facecolor='g',
                            alpha=0.75, label='histograma')
plt.legend(loc="best")
plt.savefig('graficos/p1-histograma-x-'+str(Nrep)+'.png', dpi=100)
plt.show()

# elección de Nrep ------------------------------------------------------
Nrep = 40000
x = np.linspace(xmin, xmax, Nrep)
y = np.random.randn(Nrep)
print('x =',x)
print('y =',y)
print('len(x) =',len(x))
print('len(y) =',len(y))

plt.figure( figsize=(10, 7.5) )
plt.title('Distribución normal con '+str(Nrep)+' puntos')
plt.xlabel('x')
plt.ylabel('y')
plt.grid()
plt.plot(x , y,'o-', label='normal')
plt.legend(loc="best")
plt.savefig('graficos/p1-normal-'+str(Nrep)+'.png', dpi=100)
plt.show()


plt.figure( figsize=(10, 7.5) )
plt.title('Histograma de la distribución normal con '+str(Nrep)+' puntos')
plt.xlabel('y')
plt.ylabel('dn/dy')
plt.grid()
n, bins, patches = plt.hist(y, numbines, density=False, facecolor='g',
                            alpha=0.75, label='histograma')
suma = n.sum()
hbin = (bins[-1]-bins[0])/numbines
integ = suma*hbin
plt.plot(bins , (integ*gau(bins,pro,anc)),'r',
         label='integ * Gausiana mu='+str(pro)+', sigma='+str(anc) )
plt.legend(loc="best")
plt.savefig('graficos/p1-histograma-'+str(Nrep)+'.png', dpi=100)
plt.show()

#quit()
```

- Desde la terminal ejecute:

  python3 p1.py

e interprete el resultado.

Alternativamente ejecute:

python3

y vaya agregando uno a uno los bloques del programa.

- Estudie cada paso del programa y agregue comentarios explicativos.
- Altere el programa para probar distintas cosas.

Se deberían generar los siguientes gráficos:

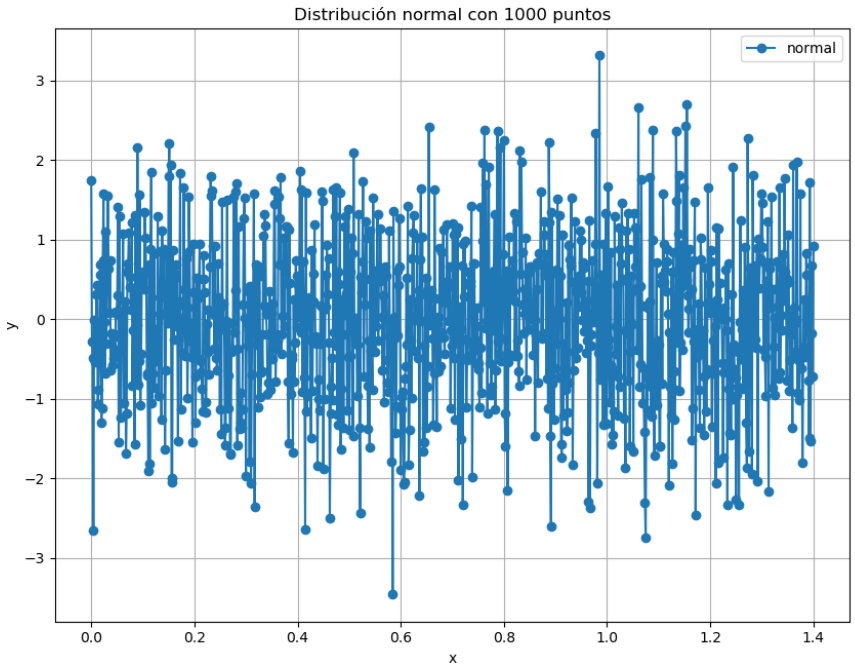

Figura 1: Distribución normal con 1000 puntos.

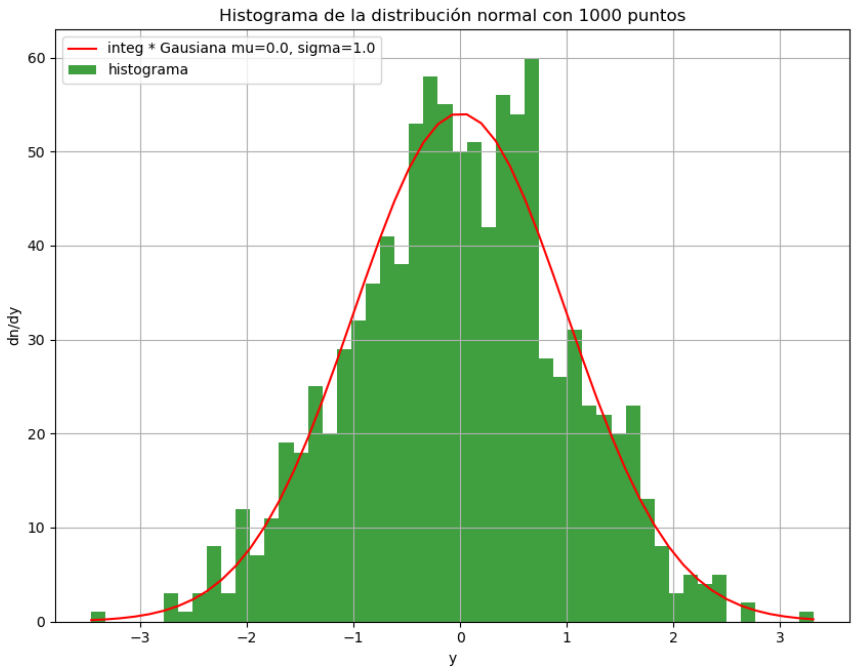

Figura 2: Histograma de la distribución con 1000 puntos.

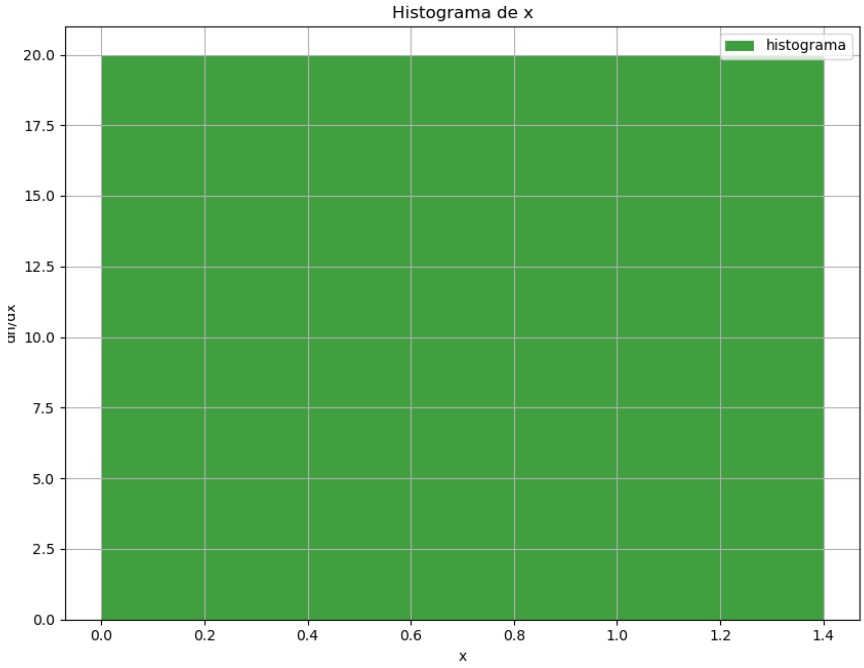

Figura 3: Histograma de la variable x con 1000 puntos.

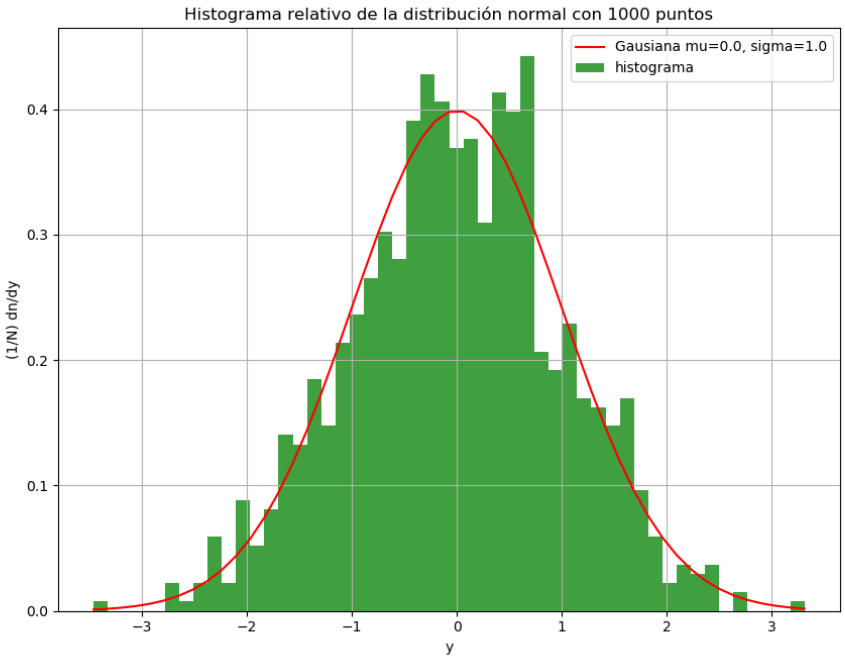

Figura 4: Histograma relativo de la distribución con 1000 puntos.

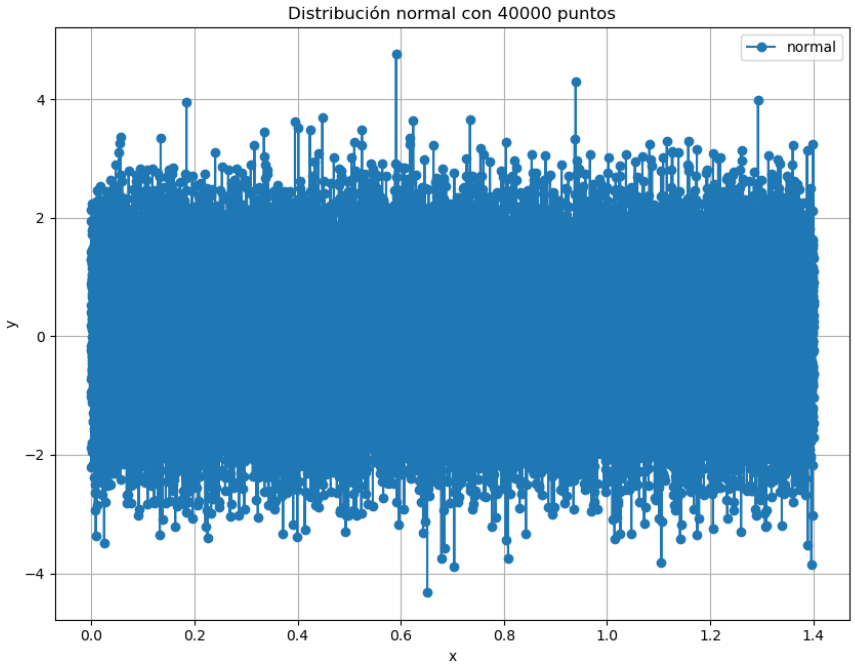

Figura 5: distribución normal con 40000 puntos.

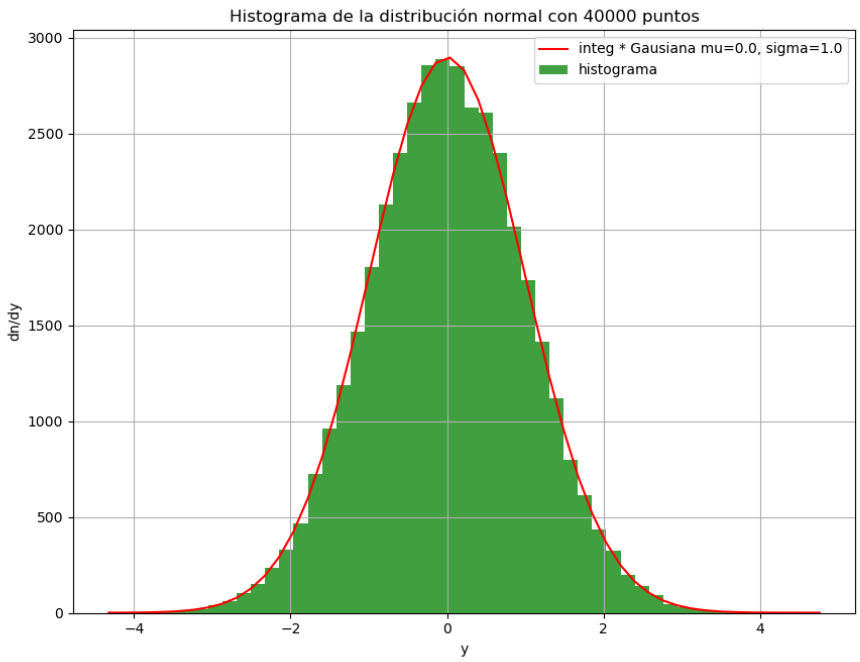

Figura 6: Histograma de la distribución con 40000 puntos.