

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 3 de la Guía N° 6

Problema 3: Teorema central del límite:

Sean X_1, X_2, \dots, X_r un conjunto de r variables aleatorias independientes, con la misma densidad de probabilidad gaussiana $P_X(x)$, con promedio 0 y varianza σ_x^2 . La suma Y , de estas variables, tiene la densidad de probabilidad

$$P_Y(y) = \frac{1}{\sqrt{2\pi r\sigma_x^2}} \exp\left(-\frac{y^2}{2r\sigma_x^2}\right);$$

que también es una gaussiana pero con varianza $r\sigma_x^2$.

Existen entonces dos sumas naturales para variables aleatorias idénticas:

$$Y_1 = \frac{X_1 + X_2 + \dots + X_r}{\sqrt{r}},$$

lo que da una variable estocástica con la misma varianza que las originales pero que su valor crece con r , ó

$$Y_2 = \frac{X_1 + X_2 + \dots + X_r}{r},$$

que tiene una varianza que disminuye con r pero que su valor medio coincide con el de las variables originales.

El teorema central del límite afirma que aún en el caso que la distribución $P_X(x)$ no sea gaussiana, sino alguna otra distribución con promedio 0 y varianza σ_x^2 , vale que

$$P_{Y_1}(y) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left(-\frac{y^2}{2\sigma_x^2}\right);$$

en el límite $r \rightarrow \infty$. Es por este resultado que la distribución gaussiana tiene un rol protagónico en los campos de la estadística.

a) Use los comandos preliminares:

```

1 numbines = 50
2 pe = 0.8
3 ene = 400
4
5 # elección de la semilla -----
6 np.random.seed(11)
7
8 # elección de Nrep -----
9 Nrep = 1000
10
11 pro = ene*pe
12 anc = np.sqrt(ene*pe*(1-pe) )

```

- b) Defina las dos sumas de variables aleatorias binomiales usando los siguientes comandos en PYTHON:

```
1 numysumas = 15000
2 ysuma1 = np.zeros(numysumas)
3 ysuma2 = np.zeros(numysumas)
4 xsuma = np.arange(numysumas)
5
6 for i in range(numysumas):
7     y = np.random.binomial(ene,pe,Nrep)
8     ysuma1[i] = y.sum()/np.sqrt(Nrep)
9     ysuma2[i] = y.sum()/Nrep
```

- c) Genere un gráfico del histograma correspondiente y de una gaussiana usando los siguientes comandos:

```
1 n, bins, patches = plt.hist(ysuma1, numbins, density=False,
2                             facecolor='g', alpha=0.75,
3                             label='histograma')
4 suma = n.sum()
5 hbin = (bins[-1]-bins[0])/numbins
6 integ = suma*hbin
7 plt.plot(bins, (integ*gau(bins,pro*np.sqrt(Nrep),anc)), 'r',
8          label='integ * Gaussiana mu='+str(pro)+
9          ', sigma='+str(anc) )
```

guardando en el archivo: 'graficos/p3-suma1-'+str(numysumas)+'-'+str(ene)+'-'+str(Nrep)+' .png';
y

```
1 n, bins, patches = plt.hist(ysuma2, numbins, density=False,
2                             facecolor='g', alpha=0.75, label='histograma')
3 suma = n.sum()
4 hbin = (bins[-1]-bins[0])/numbins
5 integ = suma*hbin
6 plt.plot(bins, (integ*gau(bins,pro,anc/np.sqrt(Nrep))), 'r',
7          label='integ * Gaussiana mu='+str(pro)+' , sigma='
8          +str(anc/np.sqrt(Nrep)) )
```

guardando en el archivo: 'graficos/p3-suma2-'+str(numysumas)+'-'+str(ene)+'-'+str(Nrep)+' .png'

- d) Repita lo anterior para `numysumas = 45000`. Compruebe que se cumple la afirmación del teorema central del límite.

Tutorial:

- Guarde en el archivo `p3.py` las siguientes instrucciones:

```
1 """
2 definimos una variable con random y
3 hacemos el histograma correspondiente
4 """
5 import numpy as np
6 import matplotlib.pyplot as plt
```

```

7
8 def gau(xx,prom,ancho):
9     return np.exp(-(xx-prom)**2/(2.*ancho**2))/( np.sqrt(2.*np.pi)*ancho )
10
11
12 numbines = 50
13 xmin = 0.
14 xmax = 1.4
15 pro = 0.
16 anc = 1.
17
18 pe = 0.8
19 ene = 400
20
21 # elección de la semilla -----
22 np.random.seed(11)
23
24 # elección de Nrep -----
25 Nrep = 1000
26 x = np.linspace(xmin, xmax, Nrep)
27 y = np.random.binomial(ene,pe,Nrep)
28
29 print('x =',x)
30 print('y =',y)
31 print('len(x) =',len(x))
32 print('len(y) =',len(y))
33 print('min(y) =',min(y))
34 print('max(y) =',max(y))
35 print('max(y)-min(y) =',(max(y)-min(y)) )
36 numbines = (max(y)-min(y))
37 print('      numbines =',numbines)
38
39
40 pro = ene*pe
41 anc = np.sqrt(ene*pe*(1-pe) )
42 print('  pro =',pro)
43 print('  anc =',anc)
44
45 print(' sumas -----')
46 numysumas = 15000
47 ysuma1 = np.zeros(numysumas)
48 ysuma2 = np.zeros(numysumas)
49 xsuma = np.arange(numysumas)
50
51 for i in range(numysumas):
52     y = np.random.binomial(ene,pe,Nrep)
53     ysuma1[i] = y.sum()/np.sqrt(Nrep)
54     ysuma2[i] = y.sum()/Nrep
55
56
57 plt.figure( figsize=(10, 7.5) )
58 plt.title('Histograma de la suma1 de variables binomiales con '
59         +str(Nrep)+' repeticiones y '
60         +str(numysumas)+' términos')
61 plt.xlabel('y')
62 plt.ylabel('dn/dy')

```

```

63 plt.grid()
64 n, bins, patches = plt.hist(ysuma1, numbins, density=False,
65                             facecolor='g',
66                             alpha=0.75, label='histograma')
67 suma = n.sum()
68 hbin = (bins[-1]-bins[0])/numbins
69 integ = suma*hbin
70 plt.plot(bins, (integ*gau(bins,pro*np.sqrt(Nrep),anc)), 'r',
71          label='integ * Gaussiana mu='+str(pro*np.sqrt(Nrep))+
72          ', sigma='+str(anc) )
73 plt.legend(loc="best")
74 plt.savefig('graficos/p3-suma1-'+str(numysumas)+'-'+str(ene)+
75            '-'+str(Nrep)+'.png', dpi=100)
76 plt.show()
77
78
79 plt.figure(figsize=(10, 7.5) )
80 plt.title('Histograma de la suma2 de variables binomiales con '
81          +str(Nrep)+' repeticiones y '
82          +str(numysumas)+' términos')
83 plt.xlabel('y')
84 plt.ylabel('dn/dy')
85 plt.grid()
86 n, bins, patches = plt.hist(ysuma2, numbins, density=False,
87                             facecolor='g',
88                             alpha=0.75, label='histograma')
89 suma = n.sum()
90 hbin = (bins[-1]-bins[0])/numbins
91 integ = suma*hbin
92 plt.plot(bins, (integ*gau(bins,pro,anc/np.sqrt(Nrep))), 'r',
93          label='integ * Gaussiana mu='+str(pro)+' , sigma='
94          +str(anc/np.sqrt(Nrep)) )
95 plt.legend(loc="best")
96 plt.savefig('graficos/p3-suma2-'+str(numysumas)+'-'+str(ene)+
97            '-'+str(Nrep)+'.png', dpi=100)
98 plt.show()
99
100
101 print('-----')
102
103 # elección de Nrep
104 -----
104 np.random.seed(11)
105
106 Nrep = 1000
107 x = np.linspace(xmin, xmax, Nrep)
108 y = np.random.binomial(ene,pe,Nrep)
109 numbins = (max(y)-min(y))
110 print('      numbins =',numbins)
111
112 print(' sumas -----')
113
114 print(' ----- ')
115 print('      Aguarde por favor, estamos trabajando para Ud.      ')
116 print(' ----- ')
117

```

```

118 numysumas = 45000
119 ysuma1 = np.zeros(numysumas)
120 ysuma2 = np.zeros(numysumas)
121 xsuma = np.arange(numysumas)
122
123 for i in range(numysumas):
124     y = np.random.binomial(ene,pe,Nrep)
125     ysuma1[i] = y.sum()/np.sqrt(Nrep)
126     ysuma2[i] = y.sum()/Nrep
127
128
129 plt.figure( figsize=(10, 7.5) )
130 plt.title('Histograma de la suma1 de variables binomiales con '
131         +str(Nrep)+' repeticiones y '
132         +str(numysumas)+' términos')
133 plt.xlabel('y')
134 plt.ylabel('dn/dy')
135 plt.grid()
136 n, bins, patches = plt.hist(ysuma1, numbins, density=False,
137                             facecolor='g', alpha=0.75,
138                             label='histograma')
139 suma = n.sum()
140 hbin = (bins[-1]-bins[0])/numbins
141 integ = suma*hbin
142 plt.plot(bins, (integ*gau(bins,pro*np.sqrt(Nrep),anc)), 'r',
143          label='integ * Gaussiana mu='+str(pro)+
144          ', sigma='+str(anc) )
145 plt.legend(loc="best")
146 plt.savefig('graficos/p3-suma1-'+str(numysumas)+'-'+str(ene)+
147           '-'+str(Nrep)+'.png', dpi=100)
148 plt.show()
149
150
151 plt.figure( figsize=(10, 7.5) )
152 plt.title('Histograma de la suma2 de variables binomiales con '
153         +str(Nrep)+' repeticiones y '
154         +str(numysumas)+' términos')
155 plt.xlabel('y')
156 plt.ylabel('dn/dy')
157 plt.grid()
158 n, bins, patches = plt.hist(ysuma2, numbins, density=False,
159                             facecolor='g', alpha=0.75, label='histograma')
160 suma = n.sum()
161 hbin = (bins[-1]-bins[0])/numbins
162 integ = suma*hbin
163 plt.plot(bins, (integ*gau(bins,pro,anc/np.sqrt(Nrep))), 'r',
164          label='integ * Gaussiana mu='+str(pro)+' , sigma='
165          +str(anc/np.sqrt(Nrep)) )
166 plt.legend(loc="best")
167 plt.savefig('graficos/p3-suma2-'+str(numysumas)+'-'+str(ene)+
168           '-'+str(Nrep)+'.png', dpi=100)
169 plt.show()
170
171 print('-----')
172
173 #quit()

```

- Desde la terminal ejecute:
`python3 p3.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:

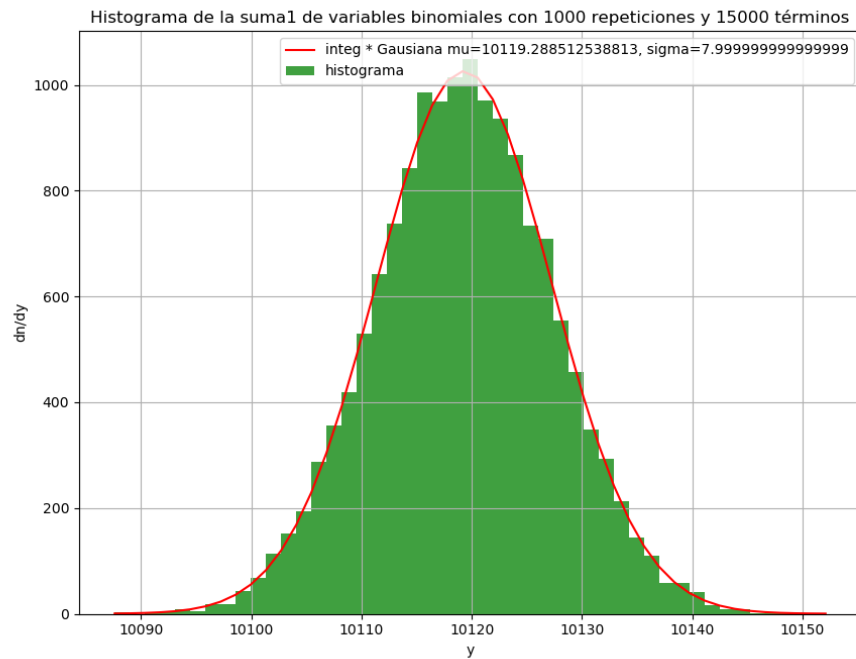


Figura 1: Suma 1 con 15000 términos, de la distribución binomial de 400 pasos con 1000 puntos.

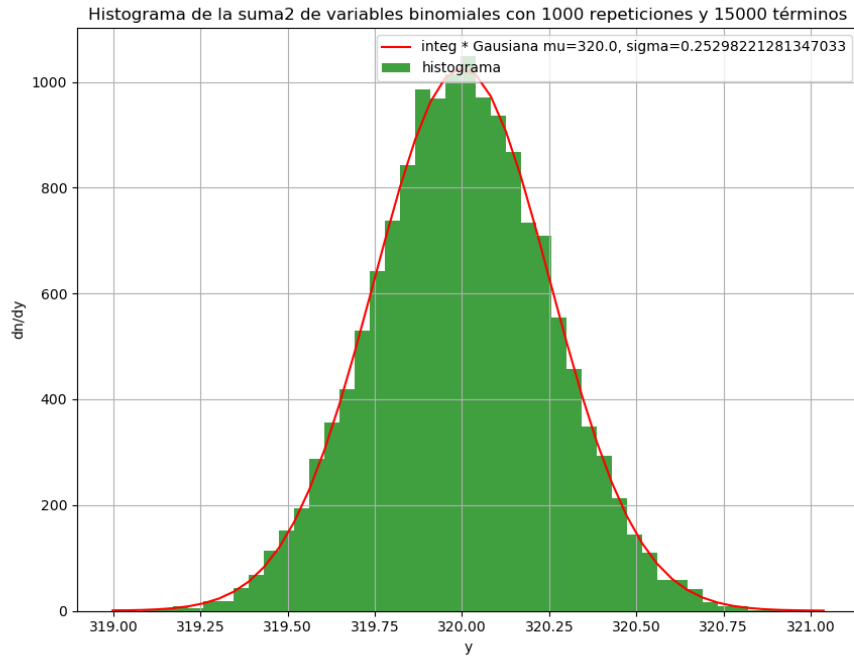


Figura 2: Suma 2 con 15000 términos, de la distribución binomial de 400 pasos con 1000 puntos.

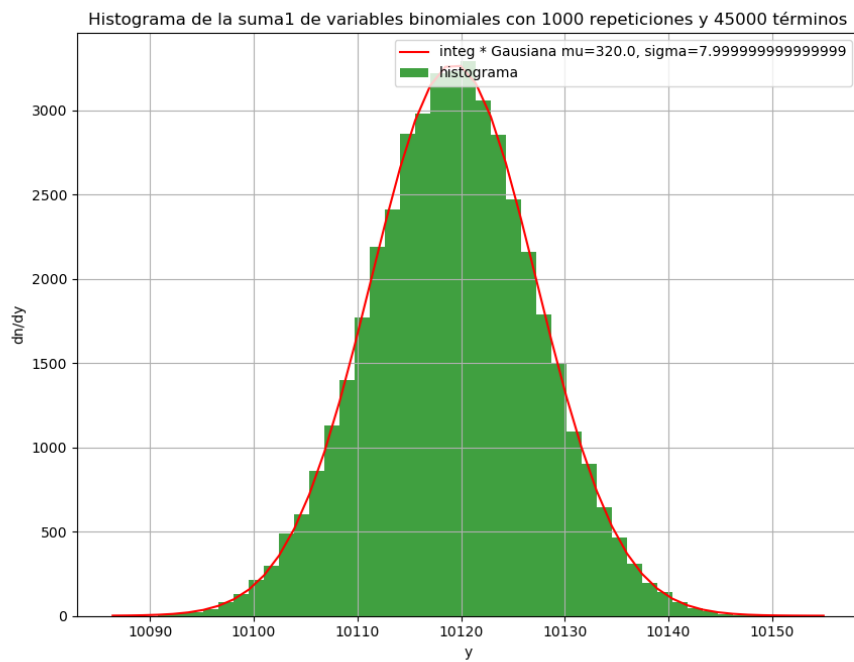


Figura 3: Suma 1 con 45000 términos, de la distribución binomial de 400 pasos con 10000 puntos.

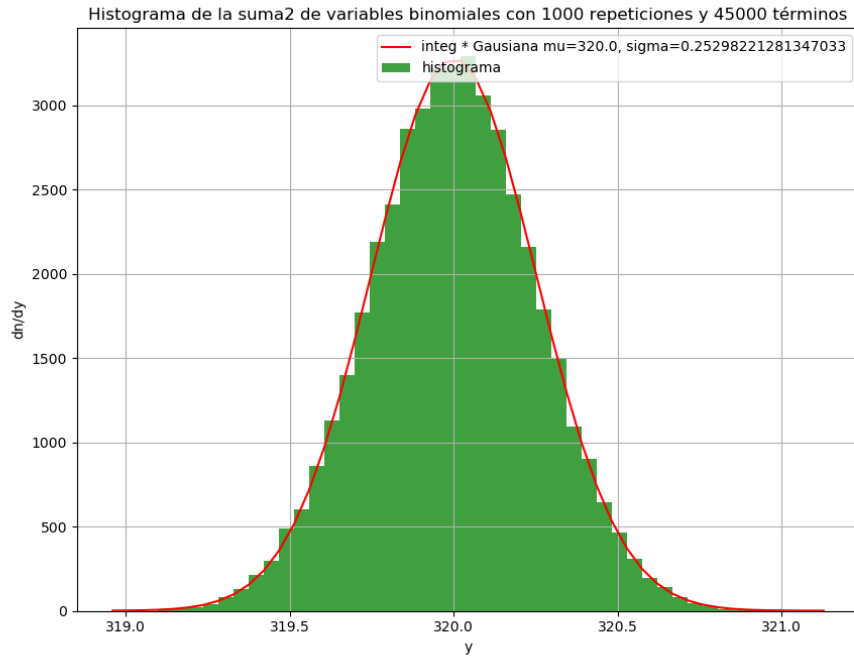


Figura 4: Suma 2 con 45000 términos, de la distribución binomial de 400 pasos con 10000 puntos.