

1_ Calculamos el $P(x)$ y la $Acu(x)$ Nor.

```
datos.sort()

for x in datos:
    sum_acu=sum_acu+x # sumatoria de todos los valores del vector datos historicos

print("Val x ", "P(x) ", "Ac(x) Nor")
for x in datos:
    acu=acu+x # acumulada
    acu_nor = acu / sum_acu # acumulada normalizada
    acu_nor_temp.append(acu/sum_acu) # guarda los datos de la acumulada normalizada en el vector acu_nor_temp
    print(x, "{:.4f}".format(x/250), "{:.4f}".format(acu_nor), sep=" ")
```

fernando@fernando: ~/Escritorio/montecarlo

Archivo Editar Ver Buscar Terminal Ayuda

44 0.4400 1.0000

(base) fernando@fernando:~/Escritorio/montecarlo\$ python montecarlo.py

Val x	P(x)	Ac(x) Nor
6	0.0240	0.0010
6	0.0240	0.0020
7	0.0280	0.0032
9	0.0360	0.0046
10	0.0400	0.0063
10	0.0400	0.0080
11	0.0440	0.0098
11	0.0440	0.0116
11	0.0440	0.0134
11	0.0440	0.0153
11	0.0440	0.0171
12	0.0480	0.0191
12	0.0480	0.0211
12	0.0480	0.0230
13	0.0520	0.0252
13	0.0520	0.0274
14	0.0560	0.0297
14	0.0560	0.0320
14	0.0560	0.0343
14	0.0560	0.0366
14	0.0560	0.0390

2_ Generamos 100 valores random, y se compara cada valor random con el inmediato superior de la acumulada normalizada. Y el valor obtenido de la acumulada, apunta a un valor x de la primer columna, el cual se guarda en un vector para calcular nuevamente promedio y acumulada.

```
# por cada valor random generado, barre el vector de la acumulada normalizada (acu_nor_temp)
print("Val Rand --->", "Val x")
for x in range(len(datos_random)):
    for y in range(len(acu_nor_temp)):
        if datos_random[x] > acu_nor_temp[y]:
            if datos_random[x] < acu_nor_temp[y+1]:
                print("{:.4f}".format(datos_random[x]), datos[y+1], sep=" ")
                datos2.append(datos[y+1])

datos2.sort()
```

fernando@fernando		
Archivo	Editar	Ver
46	0.1840	1.0000
Val Rand ---> Val x		
0.5488		27
0.7152		31
0.6028		29
0.5449		27
0.4237		24
0.6459		30
0.4376		25
0.8918		36
0.9637		39
0.3834		23
0.7917		32
0.5289		26
0.5680		27
0.9256		37
0.0710		16
0.0871		17
0.0202		12
0.8326		33
0.7782		32
0.8700		35
0.9786		43
0.7992		32

fernando@fernando:		
Archivo	Editar	Ver
26	0.1040	0.5211
26	0.1040	0.5255
26	0.1040	0.5298
26	0.1040	0.5341
27	0.1080	0.5386
27	0.1080	0.5430
27	0.1080	0.5475
27	0.1080	0.5520
27	0.1080	0.5565
27	0.1080	0.5609
27	0.1080	0.5654
27	0.1080	0.5699
27	0.1080	0.5744
28	0.1120	0.5790
28	0.1120	0.5837
28	0.1120	0.5883
28	0.1120	0.5929
28	0.1120	0.5976
28	0.1120	0.6022
29	0.1160	0.6070
29	0.1160	0.6118
29	0.1160	0.6166
29	0.1160	0.6215
29	0.1160	0.6263

El valor aleatorio 0,5488 cae entre los valores 0,5475 y 0,5520, elegimos el superior y lo relacionamos con la primer columna de los valores de x, en este caso el valor 27. Este mecanismo se realiza para todos los números aleatorios generados y se genera un nuevo vector.

3_ Con el nuevo vector generado en el anterior paso, se vuelven a calcular los $P(x)$ y $Acu(x)$ Nor, y verificamos que sean valores similares a los que calculamos en el paso 1.

```
datos2.sort()

for x in datos2:
    sum_acu2=sum_acu2+x

# Para el vector nuevo, se vuelven a calcular los promedios y la acumulada
print("Val x (Rand)", "P(x) (Rand)", "Ac(x) Nor (Rand)", sep=" ")
for x in datos2:
    acu2=acu2+x
    acu_nor2 = acu2 / sum_acu2
    print(x, "{:.4f}".format(x/100), "{:.4f}".format(acu_nor2), sep=" ")
```

fernando@fernando: ~/Escritorio/montecarlo

Archivo	Editar	Ver	Buscar	Terminal	Ayuda
Val x (Rand)	P(x) (Rand)	Ac(x) Nor (Rand)			
10	0.1000	0.0039			
12	0.1200	0.0085			
12	0.1200	0.0132			
12	0.1200	0.0179			
15	0.1500	0.0237			
16	0.1600	0.0299			
16	0.1600	0.0361			
16	0.1600	0.0423			
17	0.1700	0.0490			
17	0.1700	0.0556			
17	0.1700	0.0622			
17	0.1700	0.0688			
18	0.1800	0.0758			
18	0.1800	0.0828			
18	0.1800	0.0897			
18	0.1800	0.0967			
18	0.1800	0.1037			
18	0.1800	0.1107			
18	0.1800	0.1177			
18	0.1800	0.1247			
19	0.1900	0.1321			
19	0.1900	0.1395			
19	0.1900	0.1469			