

Glab15

November 28, 2024

```
[1]: import pandas as pd
import numpy as np
from scipy import stats
import random
```

0.1 Caso Discreto

Si X es el numero de clientes de un total de 5, que alquilan automoviles de tipo A en un tienda y la probabilidad de que cualquier cliente alquile un auto de este tipo es 0.7

- Determine la funcion de distribucion de la variable aleatoria X .
- Simule dicho evento 5 veces con 5 valores aleatorios $u \sim U[0,1]$.

```
[2]: # Defenimos los parametros para la variable X.
#  $X \sim B(5, 0.7)$ 

n=5
p=0.7
```

Definimos con el método '`binom()`' a la variable aleatoria con distribución binomial. Luego, generamos un vector con todos los valores posibles para x con el método '`arange()`'. Con el método '`pmf()`' aplicado a la variable aleatoria que se generó ('fd'), evaluamos a $f(x)$ en cada valor para x y con el método '`cdf()`' aplicado a la variable aleatoria que se generó ('fd'), evaluamos a $F(x)$ en cada valor para x . Para mostrar ambos resultados, hacemos uso de la función '`print()`'.

```
[3]: # hallando  $f(x)$  y  $F(x)$ 

fd = stats.binom(n,p)
x = np.arange(0,n+1)

print("x = ",x)
```

```
x = [0 1 2 3 4 5]
```

```
[4]: fx = fd.pmf(x)
print("f(x) = ",fx)
Fx = fd.cdf(x)
print("F(x) = ",Fx)
```

```
f(x) = [0.00243 0.02835 0.1323 0.3087 0.36015 0.16807]
F(x) = [0.00243 0.03078 0.16308 0.47178 0.83193 1.      ]
```

```
[5]: tabla =pd.DataFrame({"x":x,"f(x)=P(X=x)":fx,"F(x)=P(X<=x)":Fx})
      tabla
```

```
[5]:   x  f(x)=P(X=x)  F(x)=P(X<=x)
0  0      0.00243      0.00243
1  1      0.02835      0.03078
2  2      0.13230      0.16308
3  3      0.30870      0.47178
4  4      0.36015      0.83193
5  5      0.16807      1.00000
```

```
[7]: m=5
      u=np.zeros(m)

      for i in range(m):
          u[i]=random.uniform(0,1)

      print(u)
```

```
[0.36028964 0.58863657 0.69933216 0.85220701 0.95123199]
```

```
[8]: x_sim = np.zeros(m)

      for i in range(m):
          x_sim[i]=fd.ppf(u[i])

      print(x_sim)
```

```
[3. 4. 4. 5. 5.]
```

```
[12]: x_sim = np.zeros(m)

      for j in range(m):
          for i in range(0,n+1):
              if (u[j]>Fx[i])and (u[j]<=Fx[i+1]):
                  x_sim[j]=i+1

      x_sim
```

```
[12]: array([3., 4., 4., 5., 5.])
```

Para el caso continuo

```
[13]: from sympy import integrate
      from sympy.abc import x
      from sympy import exp
      import math
```

```
[30]: fx = (1/5)*(exp(-x/5))
      fx
```

```
[30]:  $0.2e^{-\frac{x}{5}}$ 
```

```
[31]: Fx = integrate(fx,(x,0,x))
      Fx
```

```
[31]:  $1.0 - 1.0e^{-\frac{x}{5}}$ 
```

```
[32]: from sympy import symbols, solve, Eq
      U=symbols('U')
      sol= solve(Eq(U,Fx),x)[0]
      sol
```

```
[32]:  $5.0 \log\left(-\frac{1}{U-1.0}\right)$ 
```

```
[33]: m = 5
      u = np.zeros(m)
      for i in range(m):
          u[i] = random.uniform(0,1)
      print(u)
```

```
[0.52643027 0.54815119 0.98206262 0.01554721 0.86642403]
```

```
[34]: x_sim = np.zeros(m)

      for i in range(m):
          x_sim[i] = sol.subs(U, u[i])
      print(x_sim)
```

```
[ 3.73728056  3.97203823 20.10434288  0.07834669 10.06542462]
```

```
[35]: x_sim = np.zeros(m)

      for i in range(m):
          x_sim[i] = 5*math.log(1/(1-u[i]))
      print(x_sim)
```

```
[ 3.73728056  3.97203823 20.10434288  0.07834669 10.06542462]
```