Tema_11_Distribucion_Uniforme

antonio

3/4/2021

Distribución Uniforme

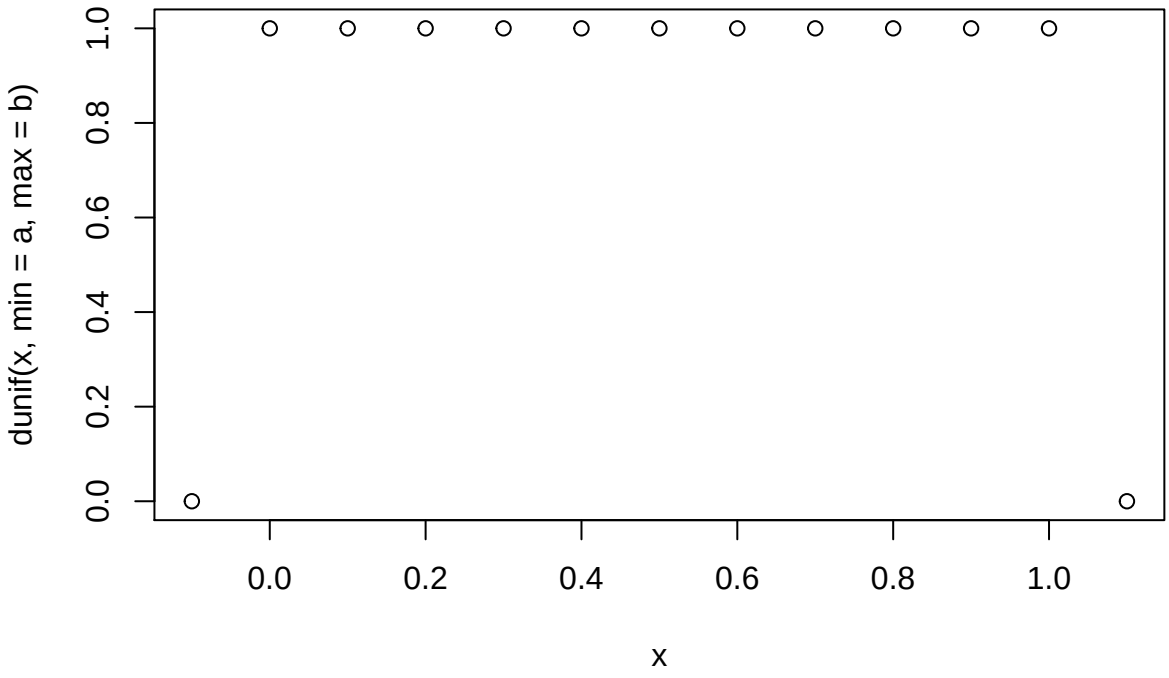
Supongamos que $X \sim U([0,1])$ entonces podemos estudiar sus parámetros

En R

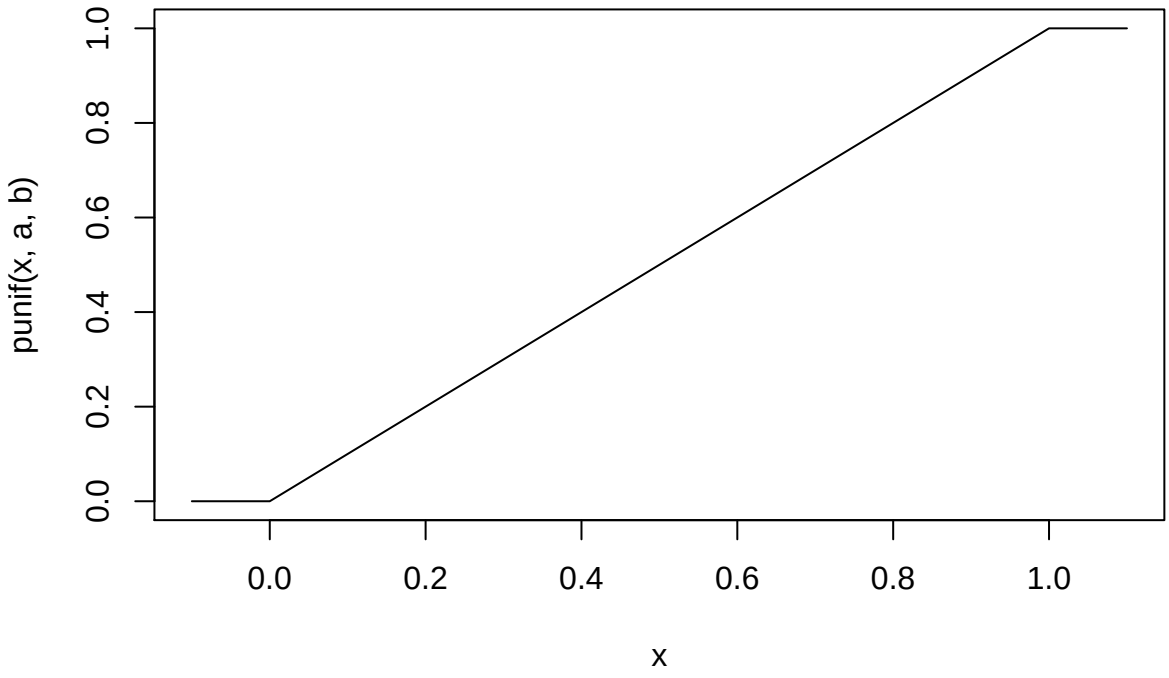
```
a = 0 # Intervalo Minimo
b = 1 #Intervalo Maximo

#donde punif(<probabilidad>,<Intervalo MIN>,<Intervalo MAX>)
dunif(0.5, min=a,max= b)
```

```
## [1] 1
```

```
x = seq(-0.1, 1.1, 0.1) #generamos la secuencia de valores en intervalo -0.1: 1.1 (probabilidad siempre
plot(x, dunif(x, min = a, max = b)) #plot funcion de densidad
```



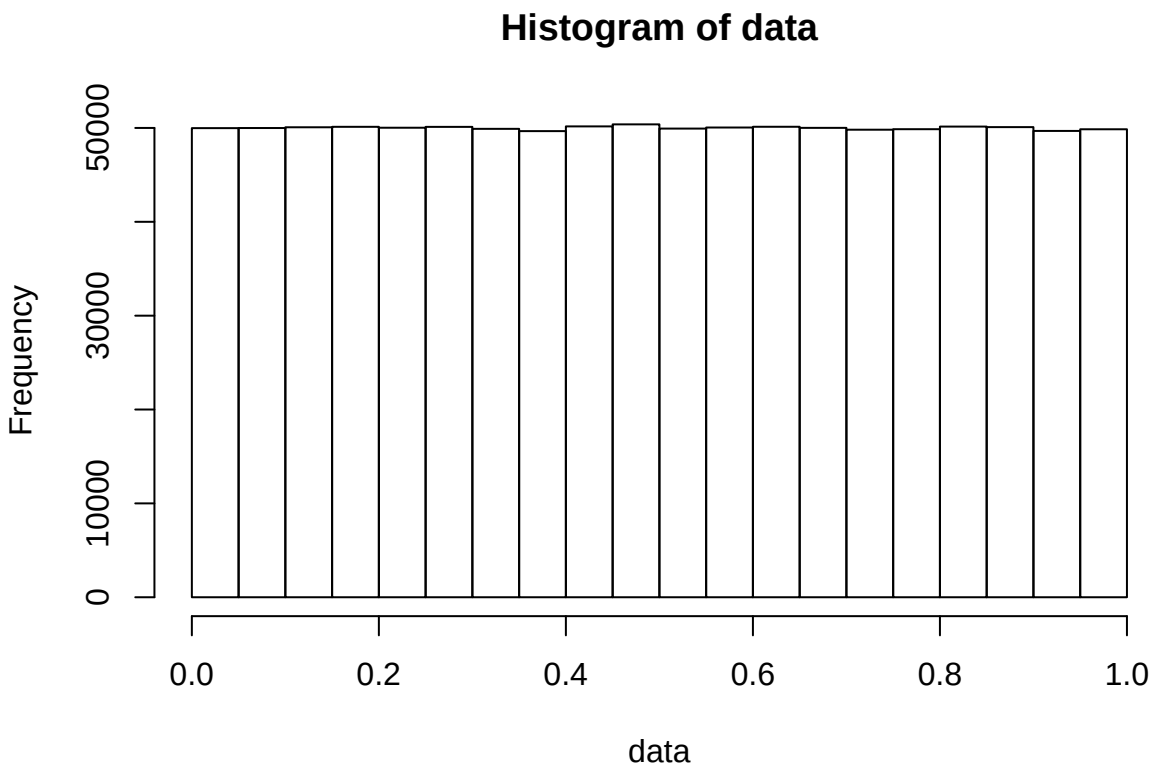

```r
plot(x, punif(x, a, b), type = "l") #plot de funcion de distribucion acumulada,. type = "l"(Pinta como
```




```
qunif(0.5, a, b)
```

```
## [1] 0.5
```

```
#histograma  que pint alos intervalos y las frecuencia de los numeros aletorios
runif(1000000, a, b) -> data   #Generamos valores aleatorios uniformes
hist(data)
```



Ejemplo Examen MAT

```
#El tiempo en que tarda un estudiante en solucionar  un examen  de MAT, esta entre 40 y 120 min
#¿Cual es la probabilidad que un estudiante se demore menos de 60 min en solucionar  el examen?

a=40
b=120
x = seq(1, 120, 1)
#donde punif(<probabilidad>,<Intervalo MIN>,<Intervalo MAX>)
dunif(x, min=a,max= b)
```

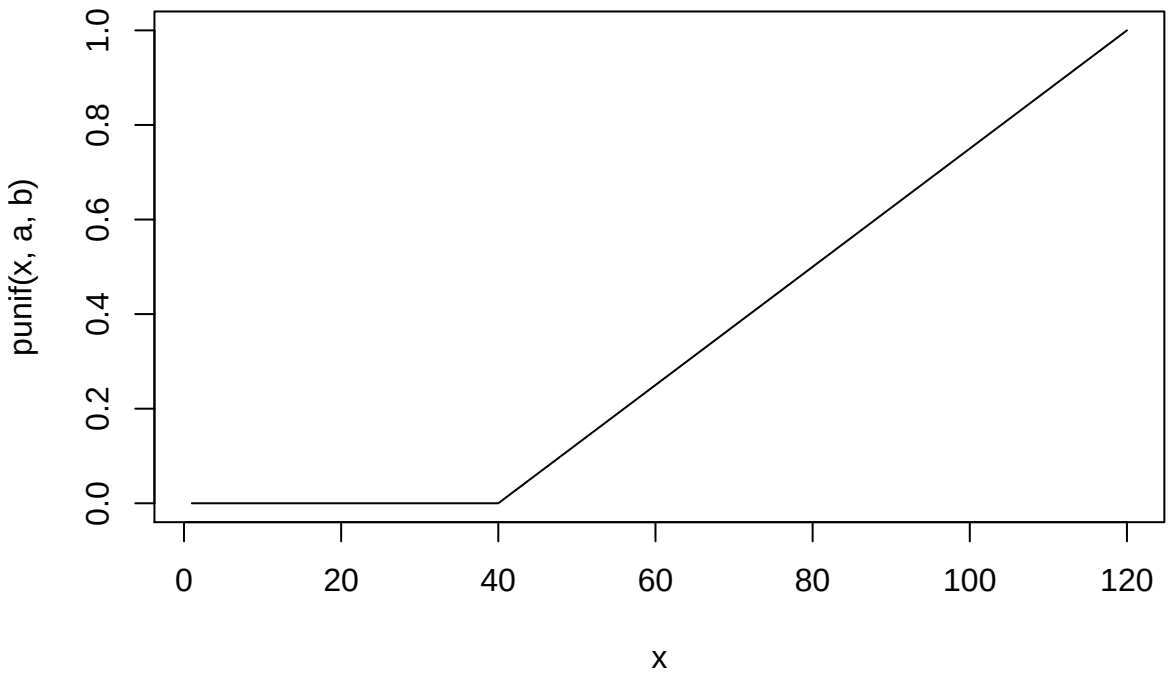
```
##   [1] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
##  [11] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
##  [21] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
##  [31] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0125
##  [41] 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125
##  [51] 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125
##  [61] 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125
##  [71] 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125
##  [81] 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125
##  [91] 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125
## [101] 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125
## [111] 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125 0.0125
```

```
punif(x, min=a,max= b)
```

```
##   [1] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

```
##  [11] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
##  [21] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
##  [31] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
##  [41] 0.0125 0.0250 0.0375 0.0500 0.0625 0.0750 0.0875 0.1000 0.1125 0.1250
##  [51] 0.1375 0.1500 0.1625 0.1750 0.1875 0.2000 0.2125 0.2250 0.2375 0.2500
##  [61] 0.2625 0.2750 0.2875 0.3000 0.3125 0.3250 0.3375 0.3500 0.3625 0.3750
##  [71] 0.3875 0.4000 0.4125 0.4250 0.4375 0.4500 0.4625 0.4750 0.4875 0.5000
##  [81] 0.5125 0.5250 0.5375 0.5500 0.5625 0.5750 0.5875 0.6000 0.6125 0.6250
##  [91] 0.6375 0.6500 0.6625 0.6750 0.6875 0.7000 0.7125 0.7250 0.7375 0.7500
## [101] 0.7625 0.7750 0.7875 0.8000 0.8125 0.8250 0.8375 0.8500 0.8625 0.8750
## [111] 0.8875 0.9000 0.9125 0.9250 0.9375 0.9500 0.9625 0.9750 0.9875 1.0000
```

```r
plot(x, punif(x, a, b), type = "l")
```



En Python

```python
#Importamos las librerias
from scipy.stats import uniform
import matplotlib.pyplot as plt
import numpy as np

a= 0 #Intervalo Minimo
b= 1 #Intervalo Maximo
```

```
loc= a # Minimo
scale= b-a # Maximo


fig, ax = plt.subplots(1,1) #Creamos la instacia de figura y los ajes

rv = uniform(loc = loc, scale = scale) #Creamos la uniforme en 0:1 como parametro
mean, var, skew, kurt = rv.stats(moments = 'mvsk') #Calculamos los momentos para una variables continua

print("Media %f"%mean)
```

```
## Media 0.500000
```

```
print("Varianza %f"%var)
```

```
## Varianza 0.083333
```

```
print("Sesgo %f"%skew)
```

```
## Sesgo 0.000000
```

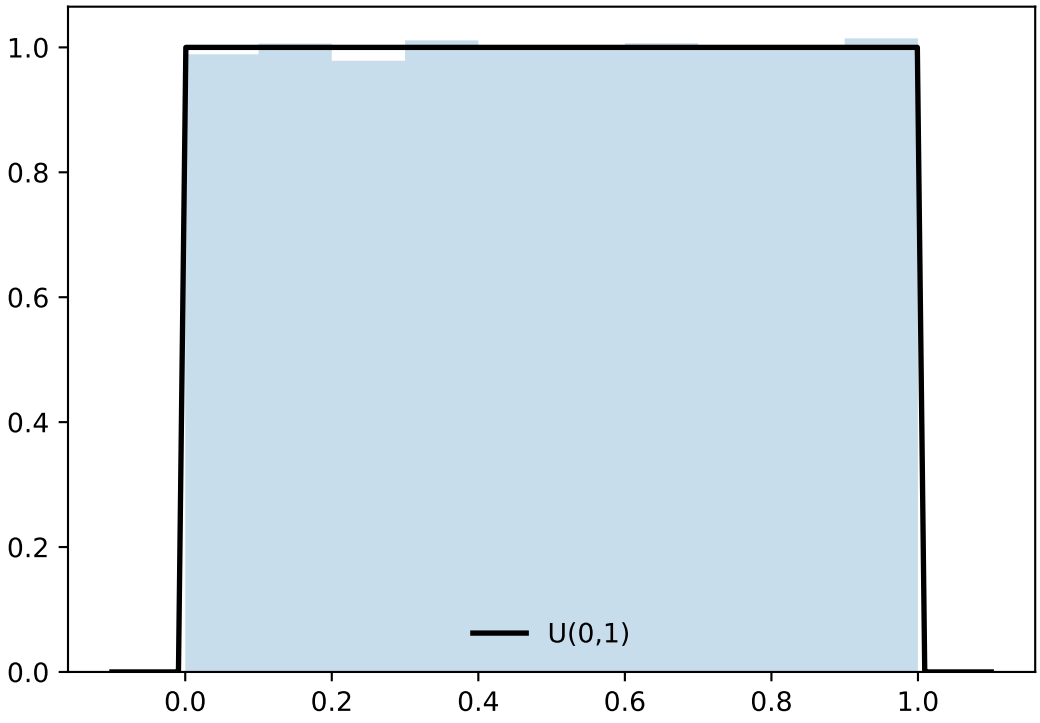
```
print("Curtosis %f"%kurt)

#Funcion de Densidad
```

```
## Curtosis -1.200000
```

```
x = np.linspace(-0.1, 1.1, 120) #Con np.linspace genermos datos que van del intervalo -0.1:1.1, 120 val
ax.plot(x, rv.pdf(x), 'k-', lw = 2, label = "U(0,1)")  #Pintamos el grafico de funcionn de densidad de

#probabilidad
r = rv.rvs(size = 100000) #Geneamos una  distribucion aleatoria para integrarla al grafic y con escala
#Histrograma en formato densidad, <muestar>,density=True, histtype=Escalonada, alpha =<Opacidad>
ax.hist(r, density = True, histtype = "stepfilled", alpha = 0.25)
```

```
## (array([0.98894928, 1.00625015, 0.97864877, 1.01125039, 0.99804974,
##         0.99774972, 1.00675017, 0.99784973, 1.00054986, 1.01445055]), array([9.54134448e-06, 1.000045
##         3.99989608e-01, 4.99984624e-01, 5.99979641e-01, 6.99974658e-01,
##         7.99969674e-01, 8.99964691e-01, 9.99959708e-01]), [<matplotlib.patches.Polygon object at 0x00
```

```
ax.legend(loc = 'best', frameon = False)#Pintamos las leyendas
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
U(0,1)