

Lab13

November 14, 2024

1 Laboratorio 13: PRINCIPALES FUNCIONES DE DISTRIBUCIÓN DE PROBABILIDAD. FUNCIONES PARA VARIABLE DISCRETA. FUNCIONES PARA VARIABLE CONTINUA

1.1 Integrante:

- Aranda Huerta, Milene Natalia
- Escriba Flores, Daniel Agustin

```
[ ]: # Importamos las librerias

from scipy import stats

import numpy as np
import matplotlib.pyplot as plt
plt.style.use('seaborn-v0_8-ticks')

# Configuración warnings
import warnings
warnings.filterwarnings('ignore')
```

```
[ ]: ['Solarize_Light2',
      '_classic_test_patch',
      '_mpl-gallery',
      '_mpl-gallery-nogrid',
      'bmh',
      'classic',
      'dark_background',
      'fast',
      'fivethirtyeight',
      'ggplot',
      'grayscale',
      'seaborn-v0_8',
      'seaborn-v0_8-bright',
      'seaborn-v0_8-colorblind',
      'seaborn-v0_8-dark',
```

```
'seaborn-v0_8-dark-palette',
'seaborn-v0_8-darkgrid',
'seaborn-v0_8-deep',
'seaborn-v0_8-muted',
'seaborn-v0_8-notebook',
'seaborn-v0_8-paper',
'seaborn-v0_8-pastel',
'seaborn-v0_8-poster',
'seaborn-v0_8-talk',
'seaborn-v0_8-ticks',
'seaborn-v0_8-white',
'seaborn-v0_8-whitegrid',
'tableau-colorblind10']
```

1.2 Pregunta 1 :

Sea una variable aleatoria, tal que $X \sim P(3)$. En un notebook de Jupyter y utilizando la librería 'stats' de 'Scipy', haga lo siguiente:

1.2.1 a. Genere un array para con valores de 0 a 9 y otro que contenga los valores de $P(X)$.

[41]:

```
# X ~ P(3)

n = 3

fd = stats.poisson(n)
x = np.linspace(0, 9, 10)

print("x =", x)
fd_x = fd.pmf(x)
print("f(x) =", fd_x)
```

```
x = [0. 1. 2. 3. 4. 5. 6. 7. 8. 9.]
f(x) = [0.04978707 0.14936121 0.22404181 0.22404181 0.16803136 0.10081881
0.05040941 0.02160403 0.00810151 0.0027005 ]
```

1.2.2 b. Halle la media y la varianza para la variable X .

[42]:

```
# Media y varianza de x

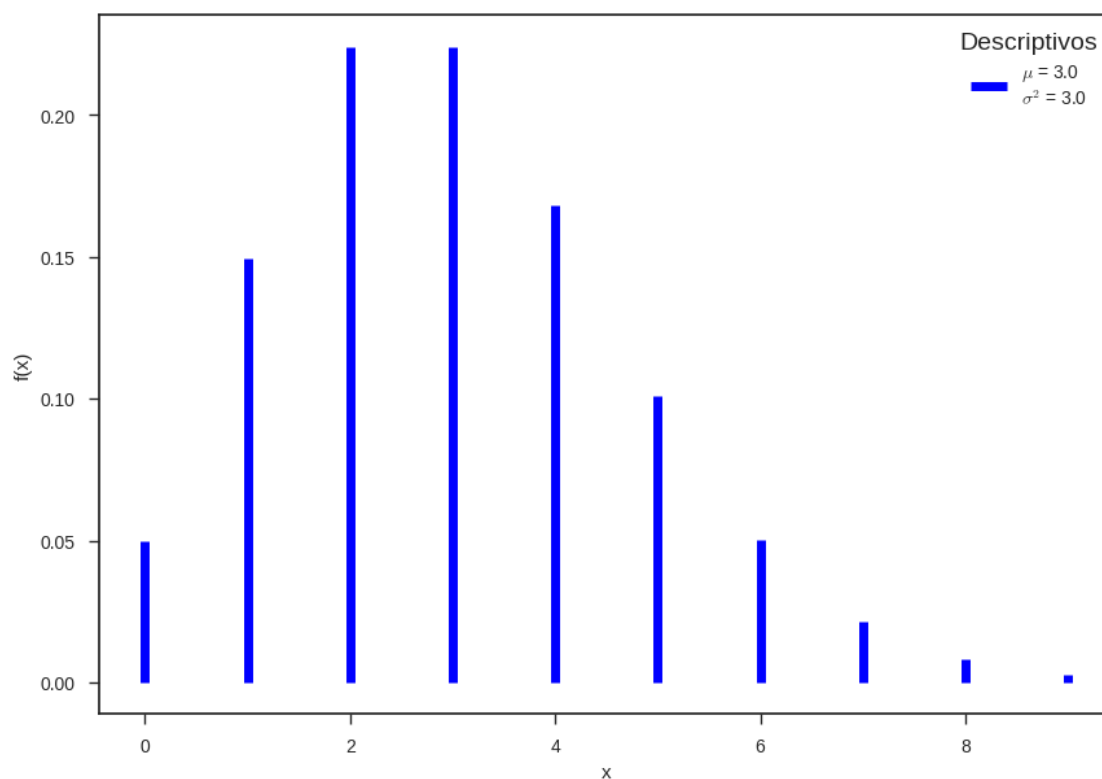
media, varianza = fd.stats(moments='mv')
print("Media =", media)
print("Varianza =", varianza)
```

```
Media = 3.0
Varianza = 3.0
```

1.2.3 c. Utilizando la librería ‘matplotlib.pyplot’, grafique la función de distribución ().

```
[43]: # Gráfico de f(x)
plt.vlines(x,0,fd_x,colors="blue",lw=5,
          label='$\mu$ = {:.1f}\n$\sigma^2$ = {:.1f}'.
          format(float(media),float(varianza)))
plt.xlabel("x")
plt.ylabel("f(x)")
plt.legend(facecolor="orange",edgecolor="red",title="Descriptivos",loc=0)
plt.show
```

```
[43]: <function matplotlib.pyplot.show(close=None, block=None)>
```



1.3 Pregunta 2: Sea una variable aleatoria, tal que $\sim (0,1)$. En un notebook de Jupyter y utilizando la librería ‘stats’ de ‘Scipy’, haga lo siguiente:

1.3.1 a. Genere un array para con 100 valores que vayan de -4 a 4 y otro que contenga los valores de $()$.

[50]: # $X \sim N(m, var)$

```
a= 0
b= 1
```

```
fd = stats.norm(a,b)
x = np.linspace(-4,4,100)
print("x =",x)
fd_x = fd.pdf(x) # función de densidad evaluada en x
print("f(x) =",fd_x)
fd_rvs = fd.rvs(1000) #
```

```
x = [-4.          -3.91919192 -3.83838384 -3.75757576 -3.67676768 -3.59595956
-3.51515152 -3.43434343 -3.35353535 -3.27272727 -3.19191919 -3.11111111
-3.03030303 -2.94949495 -2.86868687 -2.78787879 -2.70707071 -2.62626263
-2.54545455 -2.46464646 -2.38383838 -2.3030303  -2.22222222 -2.14141414
-2.06060606 -1.97979798 -1.8989899  -1.81818182 -1.73737374 -1.65656566
-1.57575758 -1.49494949 -1.41414141 -1.33333333 -1.25252525 -1.17171717
-1.09090909 -1.01010101 -0.92929293 -0.84848485 -0.76767677 -0.68686869
-0.60606061 -0.52525253 -0.44444444 -0.36363636 -0.28282828 -0.2020202
-0.12121212 -0.04040404  0.04040404  0.12121212  0.2020202  0.28282828
 0.36363636  0.44444444  0.52525253  0.60606061  0.68686869  0.76767677
 0.84848485  0.92929293  1.01010101  1.09090909  1.17171717  1.25252525
 1.33333333  1.41414141  1.49494949  1.57575758  1.65656566  1.73737374
 1.81818182  1.8989899  1.97979798  2.06060606  2.14141414  2.22222222
 2.3030303  2.38383838  2.46464646  2.54545455  2.62626263  2.70707071
 2.78787879  2.86868687  2.94949495  3.03030303  3.11111111  3.19191919
 3.27272727  3.35353535  3.43434343  3.51515152  3.59595956  3.67676768
 3.75757576  3.83838384  3.91919192  4.          ]
```

```
f(x) = [1.33830226e-04 1.84295302e-04 2.52138056e-04 3.42709873e-04
4.62784614e-04 6.20862299e-04 8.27514755e-04 1.09577221e-03
1.44154732e-03 1.88408981e-03 2.44646147e-03 3.15601632e-03
4.04486639e-03 5.15030799e-03 6.51517825e-03 8.18810653e-03
1.02236211e-02 1.26820683e-02 1.56292995e-02 1.91360817e-02
2.32771927e-02 2.81301641e-02 3.37736510e-02 4.02854146e-02
4.77399263e-02 5.62056185e-02 6.57418315e-02 7.63955298e-02
8.81978860e-02 1.01160853e-01 1.15273870e-01 1.30500851e-01
1.46777638e-01 1.64010075e-01 1.82072870e-01 2.00809396e-01
2.20032535e-01 2.39526659e-01 2.59050772e-01 2.78342808e-01
2.97125003e-01 3.15110210e-01 3.32008980e-01 3.47537175e-01
3.61423830e-01 3.73418974e-01 3.83301094e-01 3.90883931e-01
3.96022313e-01 3.98616779e-01 3.98616779e-01 3.96022313e-01]
```

```

3.90883931e-01 3.83301094e-01 3.73418974e-01 3.61423830e-01
3.47537175e-01 3.32008980e-01 3.15110210e-01 2.97125003e-01
2.78342808e-01 2.59050772e-01 2.39526659e-01 2.20032535e-01
2.00809396e-01 1.82072870e-01 1.64010075e-01 1.46777638e-01
1.30500851e-01 1.15273870e-01 1.01160853e-01 8.81978860e-02
7.63955298e-02 6.57418315e-02 5.62056185e-02 4.77399263e-02
4.02854146e-02 3.37736510e-02 2.81301641e-02 2.32771927e-02
1.91360817e-02 1.56292995e-02 1.26820683e-02 1.02236211e-02
8.18810653e-03 6.51517825e-03 5.15030799e-03 4.04486639e-03
3.15601632e-03 2.44646147e-03 1.88408981e-03 1.44154732e-03
1.09577221e-03 8.27514755e-04 6.20862299e-04 4.62784614e-04
3.42709873e-04 2.52138056e-04 1.84295302e-04 1.33830226e-04]

```

1.3.2 b. Halle la media y la varianza para la variable .

[51]: *# Media y varianza de x*

```

media, varianza = fd.stats(moments='mv')
print("Media =",media)
print("Varianza =",varianza)

```

Media = 0.0

Varianza = 1.0

1.3.3 c. Utilizando la librería ‘matplotlib.pyplot’, grafique la función de distribución () que incluya también un histograma con 500 valores muestreados con distribución $\sim (0,1)$

[53]: *# Gráfico de f(x)*

```

plt.plot(x,fd_x,lw=5,color="red",
         label='$\mu$ = {:.1f}\n$\sigma^2$ = {:.2f}'.
         ↪format(float(media),float(varianza)))
plt.hist(fd_rvs,density=True)
plt.xlabel("x")
plt.ylabel("f(x)")
plt.legend(facecolor="skyblue",edgecolor="purple",title="Descriptivos",loc=0)
plt.show

```

[53]: <function matplotlib.pyplot.show(close=None, block=None)>

