AAG03

July 7, 2020

Atividade prática aula 04

Aluno: Kristtopher Kayo Coelho Matrícula: ES95031

A atividade consiste em desenvolver exemplos (enunciados e resoluções) que expliquem cada uma das distribuições apresentadas no conjunto de slides da aula quatro. A seção I corresponde à importação das bibliotecas necessária e a implementação das funções de apresentação gráfica. As seções seguintes descreves as principais distribuições consideradas neste trabalho.

0.1 I - Bibliotecas e funções de exibição dos gráficos.

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
import pandas as pd
from scipy.stats import binom
import seaborn as sns
```

Implementação das funções de apresentação gáfica:

[2]:
```
def cdf(np_amostra, xlabel=r'x', ylabel=r'$F(X \leq x)$'):

    x = np.sort(np_amostra)
    ecdf = sm.distributions.ECDF(np_amostra)
    y = ecdf(x)
    plt.step(x, y, where='post')
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.xlim(min(x) - 0.05, max(x) + 0.05)
    plt.ylim(0 - 0.01, 1 + 0.01)
    plt.title('CDF')

def pmf(amostra_discreta, xlabel=r'x', ylabel=r'$p(X = x)$'):
    df_amostra = pd.DataFrame({'': amostra_discreta})
    ax = df_amostra.plot(kind='hist', density=True, histtype='bar', rwidth=0.1,
                         xlim=(min(amostra_discreta) - 0.05,␣
 ↪max(amostra_discreta) + 0.05), ylim=(0, 1),
```

```
                        legend=False, bins=np.arange(len(amostra_discreta)) -␣
↪0.5)
    ax.set_xlabel(xlabel)
    ax.set_ylabel(ylabel)
    plt.title('PMF')


def pdf (amostra_continua, xlabel=r'$Tempo~x$', ylabel=r'$f(X)$'):
    df_amostra = pd.DataFrame({'': amostra_continua})
    ax = df_amostra.plot(kind='hist', density=True, histtype='bar', rwidth=1,
                        xlim=(min(amostra_continua) - 0.05,␣
↪max(amostra_continua) + 0.05), ylim=(0, 1),
                        legend=False, bins=np.arange(len(amostra_continua)) -␣
↪0.5)
    ax.set_xlabel(xlabel)
    ax.set_ylabel(ylabel)

    lim_min = int(round(min(amostra_continua)))
    if lim_min >= 0:
        lim_min = 0
    else:
        ax.axvline(0, color='black', lw=1, linestyle=':')
    ax.set_xticks(range(lim_min, int(round(max(amostra_continua)))))
    plt.title('PDF')
```

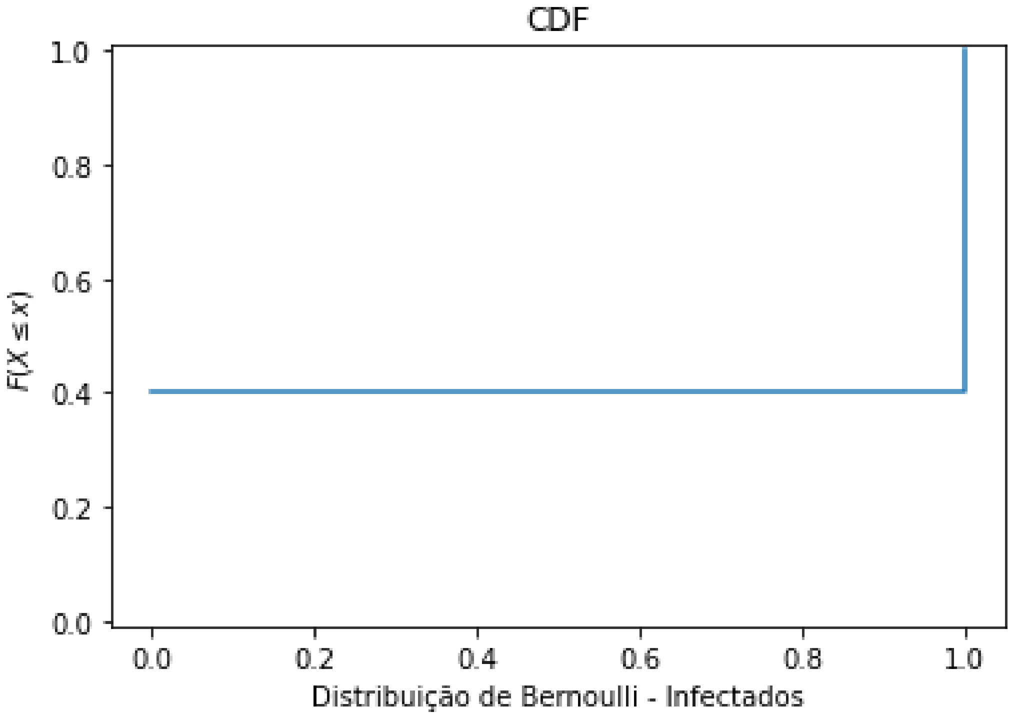
0.2 II - Bernoulli.

O resultado de exames médicos para detecção de corona vírus em 100 pessoas:

Fórmula da CDF

$$F(x) = \begin{cases} 0 & x < 0 \\ 1-p & 0 \leq x < 1 \\ 1, & x \geq 1 \end{cases}$$

[3]:
```
amostras = np.random.randint(0,2,10)
cdf(amostras, xlabel= 'Distribuição de Bernoulli - Infectados')
```

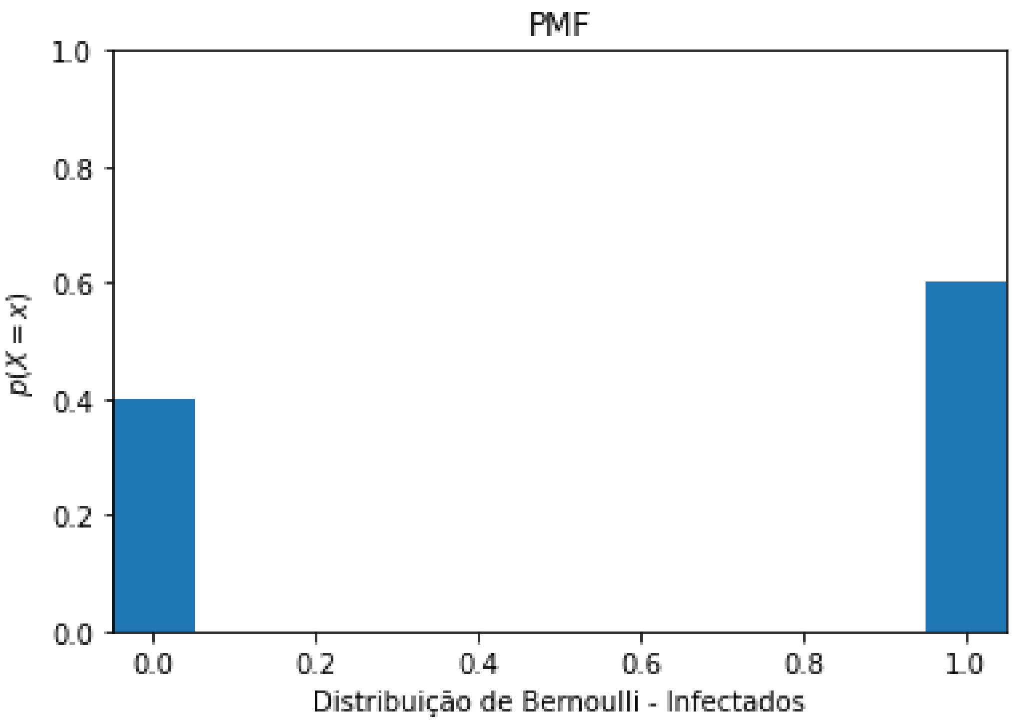


Fórmula da PMF

$$p(x) = \begin{cases} 1 - p = q, & X = 0 \\ p, & X = 1 \end{cases}$$

$$0 \leq p \leq 1$$

[4]:
```
pmf(amostras, xlabel= 'Distribuição de Bernoulli - Infectados')
```



0.3 III - Binomial.

Considerando pacientes do grupo sanguineo tipo A com maior propensão a agravar a crise respiratória por corona virus. Suponha que foi constatado que a probabilidade de ter sangue tipo A é 0,4. Para 4 exames aleatórios e seja X o número com o tipo sanguíneo A: A representação da distribuição Binomial foi realizada com auxilio da função pronta "binom.pmf", a qual já apresenta a Função Massa de Probabilidade.

Fórmula da PMF

$$p(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

$$x = 0, 1, \ldots, n$$

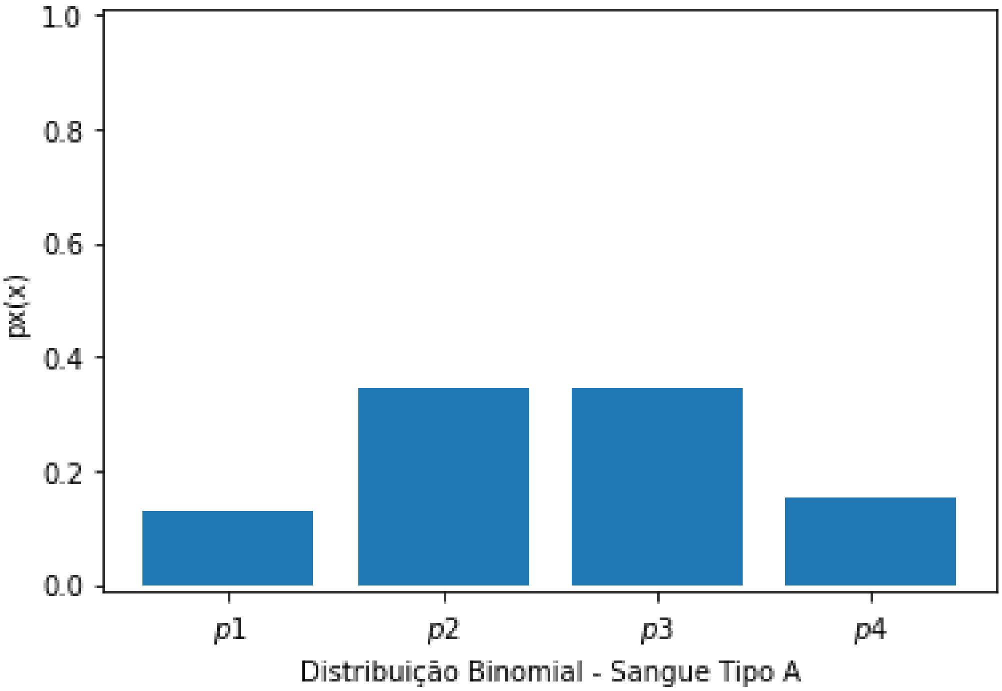
[5]:

```
x = np.random.randint(0,4,4)  # valores de x =0,1,2,3 ou 4
# distribuição dos resultados com probabilidade de um experimento resultar em␣
 ↪sucesso
p_x = binom.pmf(x, 4, 0.4)

plt.xlabel('Distribuição Binomial - Sangue Tipo A')
plt.ylabel('px(x)')
plt.ylim(0 - 0.01, 1 + 0.01)
plt.xticks([0,1,2, 3],['$p1$','$p2$','$p3$','$p4$'])
```

```
plt.bar(x,p_x)
```

[5]: <BarContainer object of 4 artists>



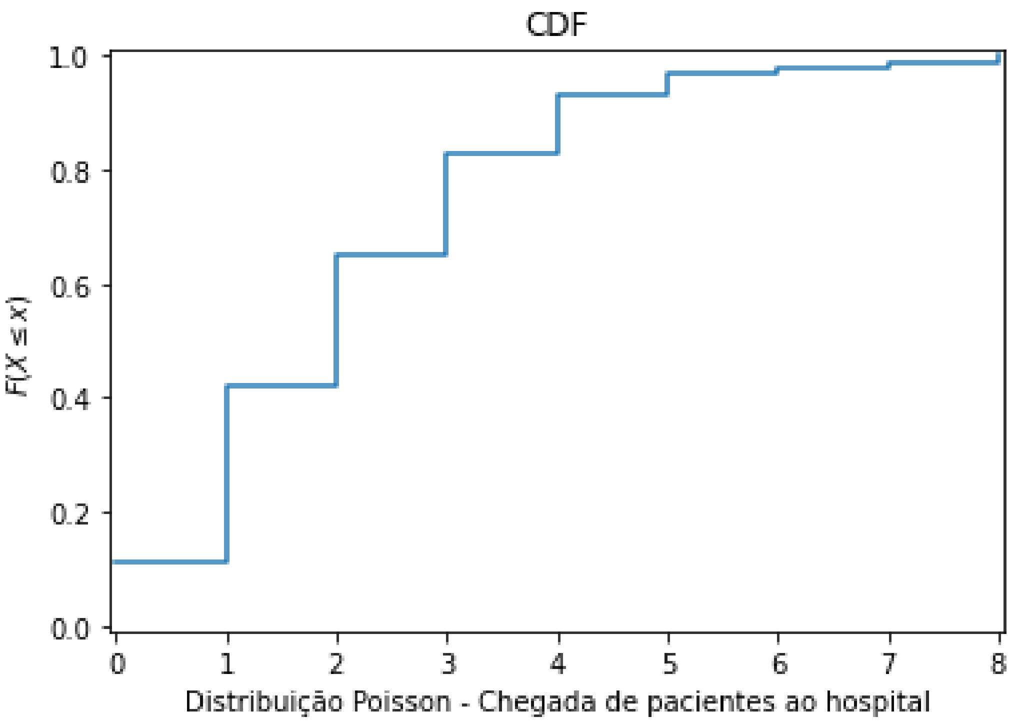
0.4 IV - Poisson.

Probabilidade de pacientes chegando para atendimento em um pronto socorro.

Fórmula da CDF

$$F(x) = \sum_{k=0}^{\infty} \frac{\alpha^k e^{-\alpha}}{k!} u(x-k)$$

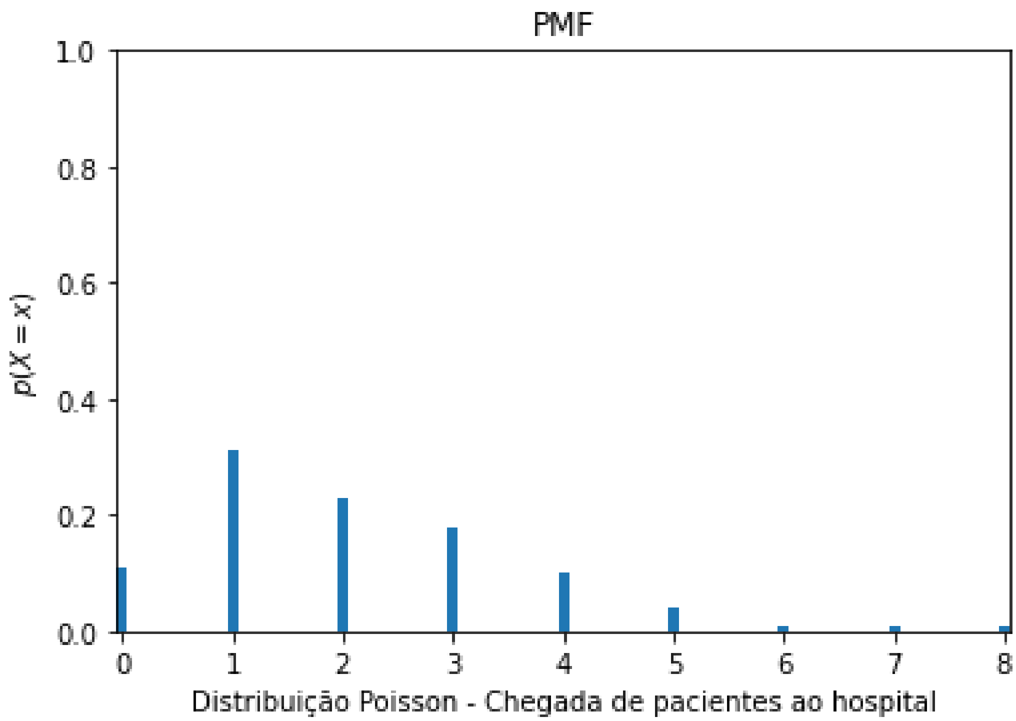
[6]:
```
x = np.random.poisson(lam=2, size=100)
cdf(x, xlabel = 'Distribuição Poisson - Chegada de pacientes ao hospital')
```



Fórmula da PMF

$$p(x) = \frac{\alpha^x}{x!}e^{-\alpha}$$
$$x = 0, 1, \ldots \text{ e } \alpha > 0$$

[7]:
```
pmf(x, xlabel = 'Distribuição Poisson - Chegada de pacientes ao hospital')
```



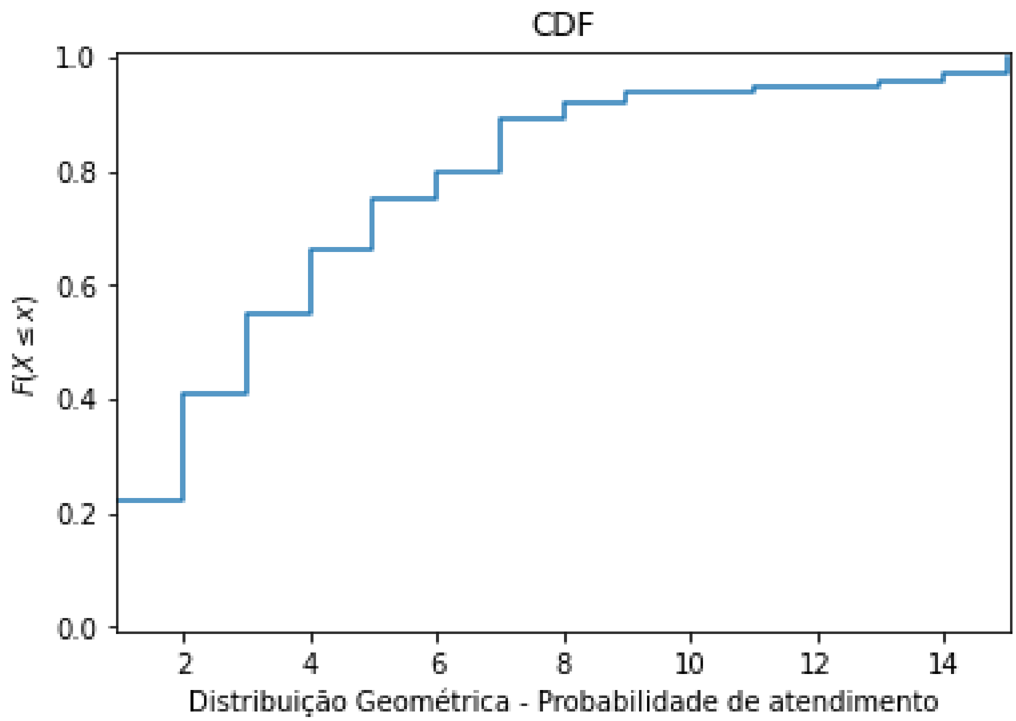
0.5 V - Geométrica.

Distribuição geométrica para 100 pacientes, com a probabilidade de um sucesso individual para atendimento igual a 0,25:

Fórmula da CDF

$$F(x) = \sum_{k=0}^{\infty} p(1-p)^k u(x-k)$$

[8]:
```
x = np.random.geometric(p=0.25, size=100)
cdf(x, xlabel = 'Distribuição Geométrica - Probabilidade de atendimento')
```



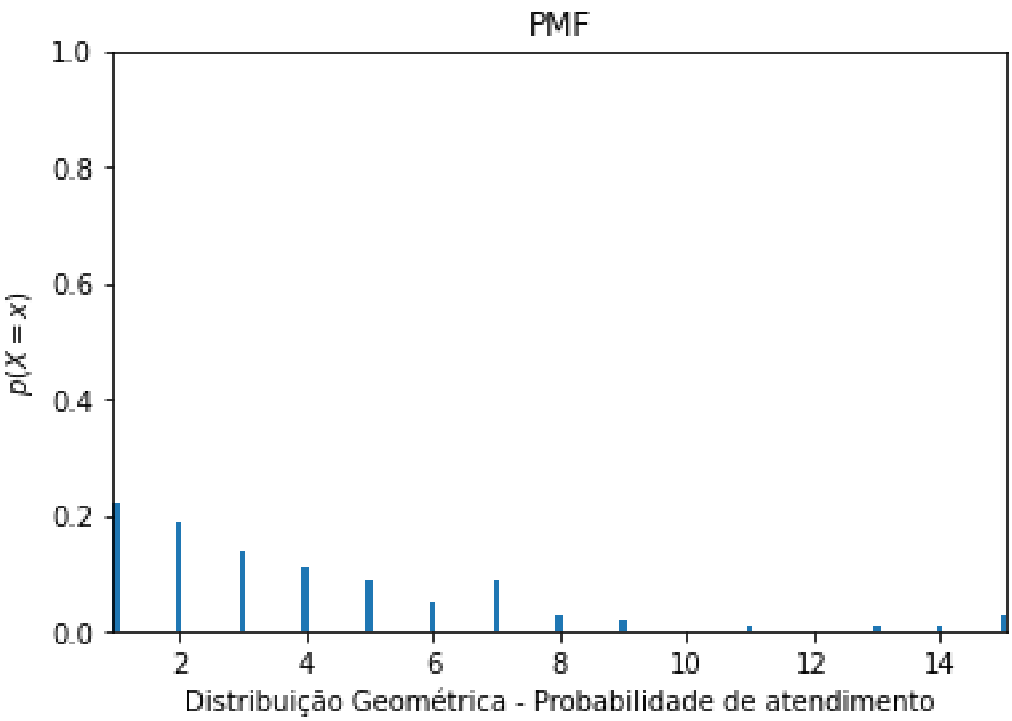
Fórmula da PMF

$$p(x) = p(1-p)^x$$

$$x = 0, 1, 2, \ldots$$

[9]:

```
pmf(x, xlabel = 'Distribuição Geométrica - Probabilidade de atendimento')
```



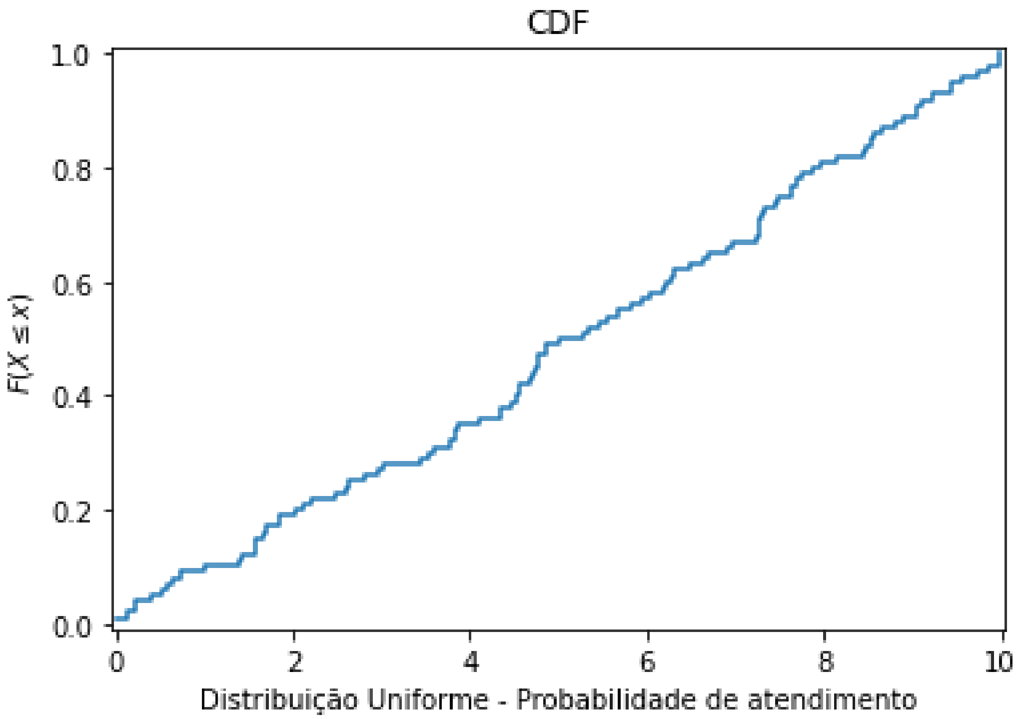
0.6 VI - Uniforme.

A distribuição uniforme é um número finito de resultados com chances iguais de acontecer. Portanto representa a probabilidade de cada paciente (10) ser atendido. A população é 100.

Fórmula da CDF

$$F(x)\begin{cases}0, & x < a\\ \frac{x-a}{b-a}, & a \leq x \leq b\\ 1, & x > b\end{cases}$$

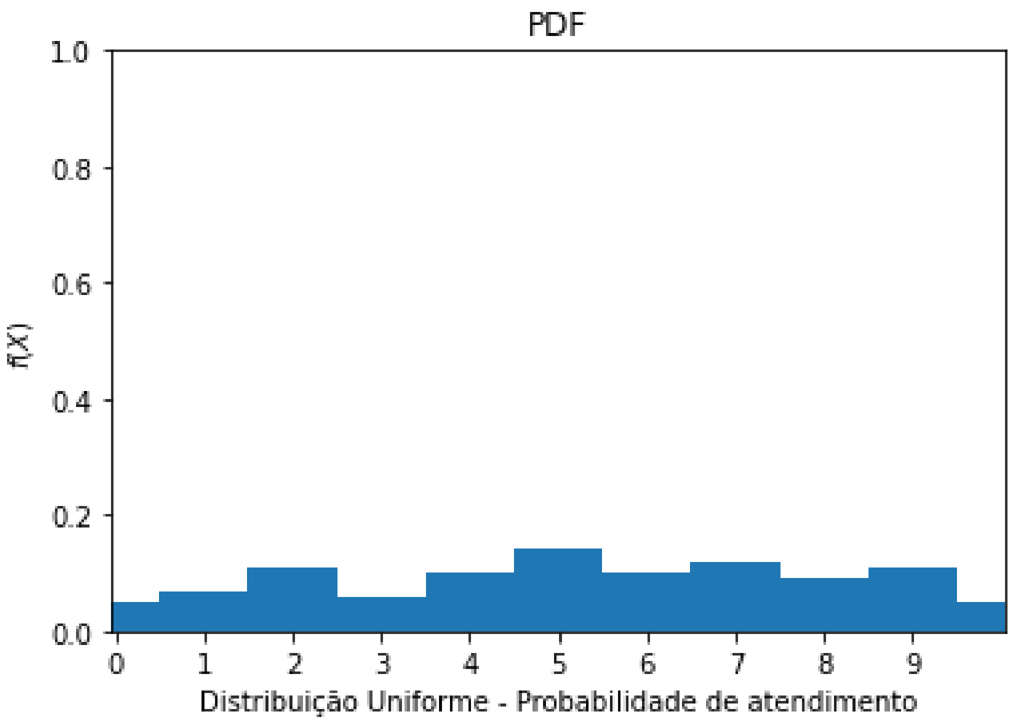
[10]:
```
x = np.random.uniform(0.0, 10.0, 100)
cdf(x, xlabel = 'Distribuição Uniforme - Probabilidade de atendimento')
```



Fórmula da PDF

$$f(x) \begin{cases} \frac{1}{b-a,} & a \leq x \leq b \\ 0, & \text{caso contrário} \end{cases}$$

[11]:
```
pdf(x, xlabel = 'Distribuição Uniforme - Probabilidade de atendimento')
```



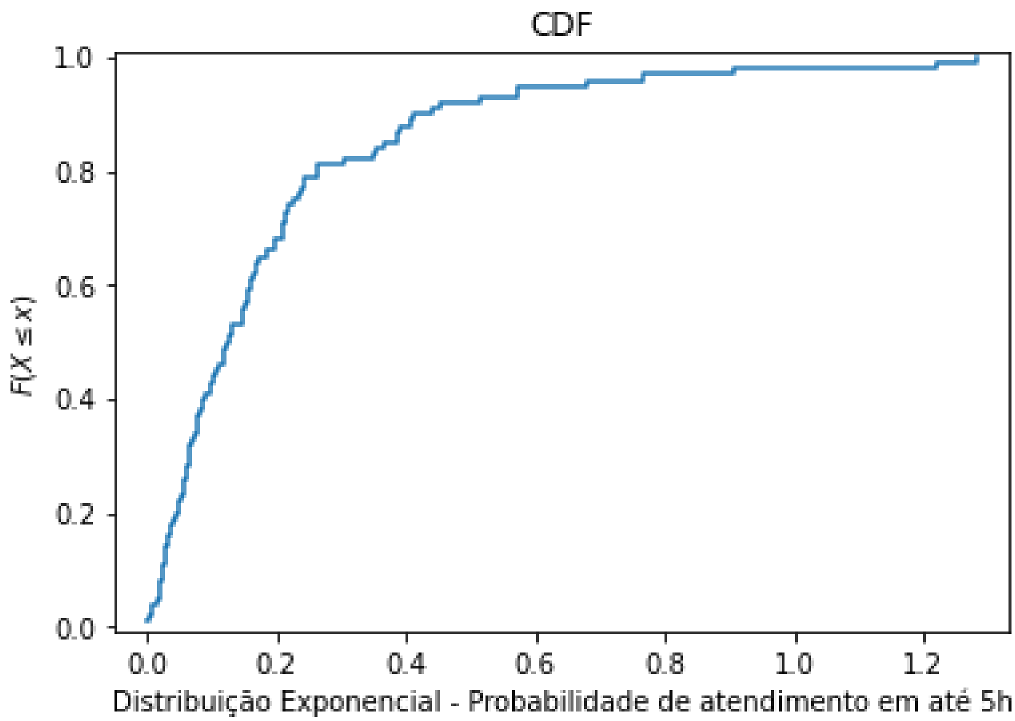
0.7 VII - Exponencial.

A distribuição exponencial representa a possibilidade de um paciente ser atendido até 5 horas de espera.

Fórmula da CDF

$$F(x) \begin{cases} = 1 - e^{-\lambda x}, & x \geq 0, \lambda > 0 \\ 0, & \text{caso contrário} \end{cases}$$

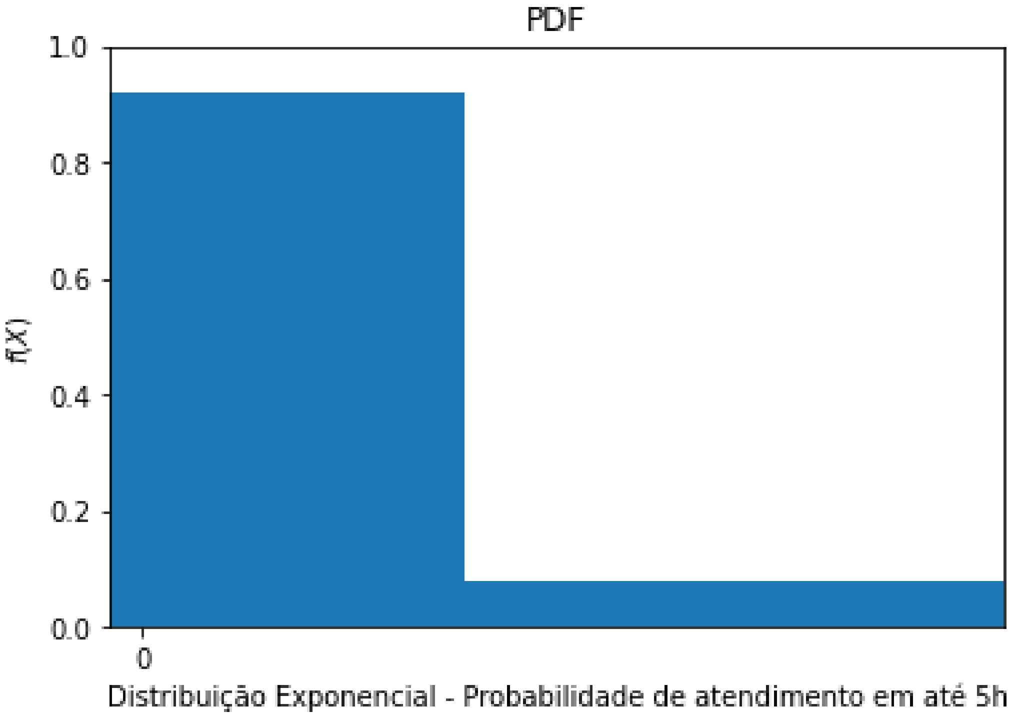
[12]:
```
x = np.random.exponential(1/5, 100)
cdf(x, xlabel = 'Distribuição Exponencial - Probabilidade de atendimento em até␣
↪5h')
```



Fórmula da PDF

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \text{ e } \lambda > 0 \\ 0, & \text{caso contrário} \end{cases}$$

[13]:
```
pdf(x, xlabel = 'Distribuição Exponencial - Probabilidade de atendimento em até␣
 ↪5h')
```



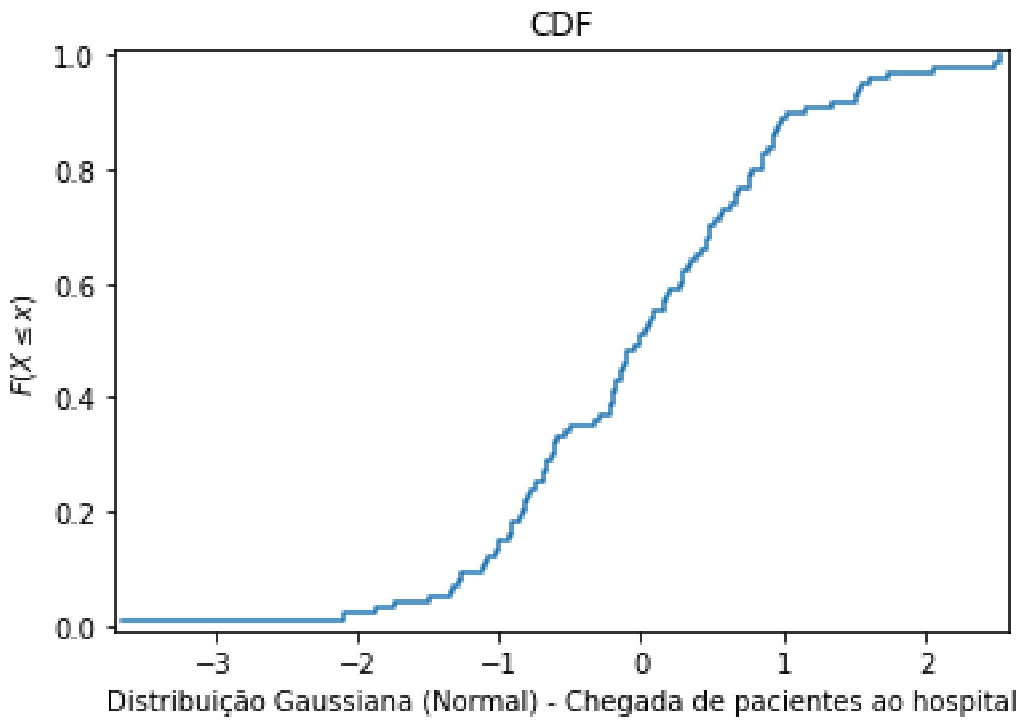
0.8 VIII - Gaussiana (Normal).

Caso a curva de contagio e consequentemente atendimento para pacientes com corona vírus tivesse um comportamento normal, ela poderia ser representadas através de uma distribuição Gausiana. Os gráficos a seguir representam o comportamento dos dados apresentados por uma distribuição normal.

Fórmula da CDF

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{x} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \mathrm{dy} \tag{1}$$

[16]:
```
x = np.random.normal(size=100)
cdf(x, xlabel = 'Distribuição Gaussiana (Normal) - Chegada de pacientes ao␣
 ↪hospital')
```

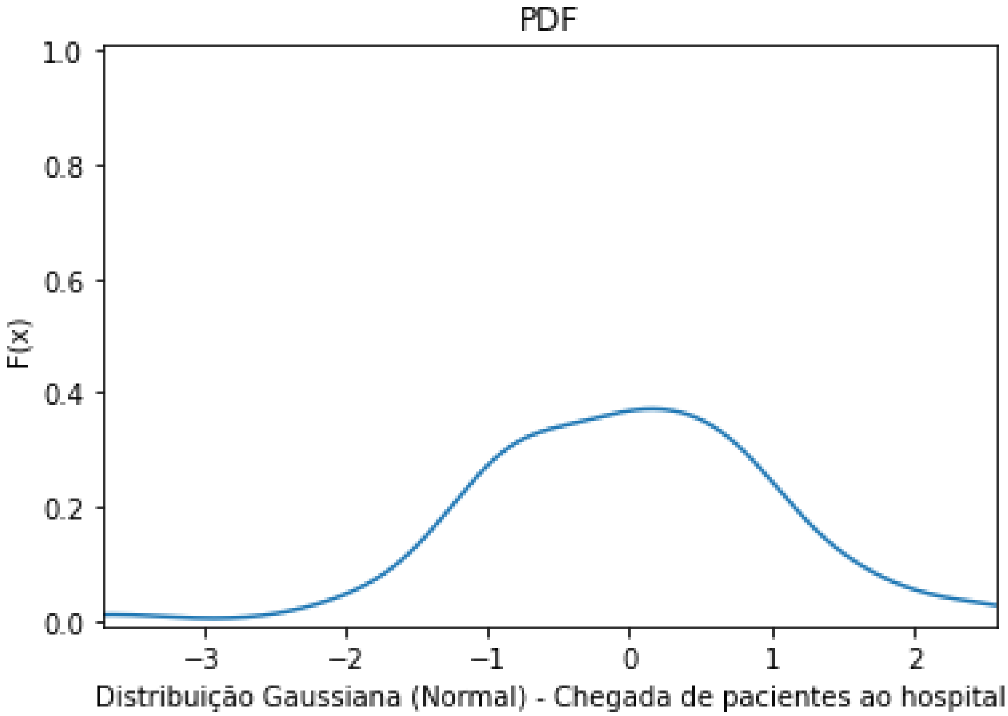


Fórmula da PDF

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

[17]:
```
ser = pd.Series(x)
ser.plot.kde()
plt.xlim(min(x) - 0.05, max(x) + 0.05)
plt.xlabel('Distribuição Gaussiana (Normal) - Chegada de pacientes ao hospital')
plt.title('PDF')
plt.ylabel('F(x)')
plt.ylim(0 - 0.01, 1 + 0.01)
```

[17]:
```
(-0.01, 1.01)
```


PDF
1.0
0.8
0.6
0.4
0.2
0.0
F(x)
−3
−2
−1
0
1
2
Distribuição Gaussiana (Normal) - Chegada de pacientes ao hospital