# Modelos probabilísticos para variáveis aleatórias discretas

## Resolução dos exemplos e exercícios utilizando o Python

### Distribuição Binomial

*Distribuição Binomial para o cálculo de* ***P(X=x)***

*binom.pmf(x,n,p)*

*Distribuição Binomial para o cálculo de* ***P(0≤X≤x) = P(X≤x)***

*binom.cdf(x,n,p)*

*Distribuição Binomial para o cálculo de* ***P(X>x)***

*binom.sf(x,n,p)*

**onde:**

n = tamanho da amostra

p= probabilidade de sucesso

**Importante:**

Para usar as funções de cálculo de probabilidade para a distribuição binomial no Python é necessário primeiramente que você importe a função *binom:*

**from** scipy.stats **import** binom

*Exemplo 1*

Suponha que 5% de todas as peças que saiam de uma linha de produção sejam defeituosas. Se 10 dessas peças forem escolhidas e inspecionadas, pede-se:

Observe que temos:

- ✓ Um experimento com somente duas opções de resposta (peças defeituosas ou não defeituosas);
- ✓ Um número fixo e independente de vezes que o experimento será repetido (10 amostras);
- ✓ A probabilidade de peças defeituosas é constante p=0,05 e, consequentemente, a probabilidade de peças não defeituosas q=1-p=0,95.

Dessa forma, podemos dizer que o modelo binomial se adapta bem à situação proposta no exemplo, ou seja, $X \sim B(10; 0{,}05)$,

a) Identifique a variável aleatória estudada. Quais valores ela pode assumir?
   X: número de peças defeituosas produzidas
   x= 0, 1, 2,..., 10.

b) Calcule o número médio de peças defeituosas e, também, o desvio padrão.
   Média: E(X) = np = 10.0,05 = 0,5
   Variância: $\sigma^2$ = npq = 10.0,05.0,95 = 0,475
   Desvio padrão: $\sigma = \sqrt{\sigma^2}$ = 0,6892

   **Utilizando o Python:**

```
# Número médio de peças defeituosas
media_defeituosas = n * p_defeituosas

# Desvio padrão
desvio_padrao = (n * p_defeituosas * (1 - p_defeituosas))**0.5

# Resultados
print("Número médio de peças defeituosas:", media_defeituosas)
print("Desvio padrão:", desvio_padrao)

Número médio de peças defeituosas: 0.5
Desvio padrão: 0.7071067811865476
```

c) Qual será a probabilidade de que:
   i. Exatamente 7 sejam defeituosas.

$$P(X=7)=\binom{10}{7}.0{,}05^7.(1-0{,}05)^{10-7}=8{,}03789\times10^{-8}$$

   ii. No máximo 2 sejam defeituosas.

$$P(X \leq 2) = P(X = 0) + P(X = 1) + P(X = 2) = 0{,}9884$$

$$P(X=0)=\binom{10}{0}.0{,}05^0.(1-0{,}05)^{10-0}=0{,}5987$$

$$P(X=1)=\binom{10}{1}.0{,}05^1.(1-0{,}05)^{10-1}=0{,}3151$$

$$P(X=2)=\binom{10}{2}.0{,}05^{2}.(1-0{,}05)^{10-2}=0{,}0746$$

*Observe que a probabilidade diminui à medida que nos afastamos da média.*

iii. No máximo 8 sejam defeituosas.

$P(X \leq 8) = 1 - P(X > 8) = 1 - [P(X = 9) + P(X = 10)]$ = 1

$$P(X=9)=\binom{10}{9}.0{,}05^{9}.(1-0{,}05)^{10-9}=1{,}8555\times10^{-11}$$

$$P(X=10)=\binom{10}{10}.0{,}05^{10}.(1-0{,}05)^{10-10}=9{,}7656\times10^{-14}$$

iv. Mais de 7 sejam <u>não defeituosas</u>.

Observe a correspondência entre peças defeituosas e não defeituosas na amostra, a partir do que é pedido nesse item:

| | | |
|---|---|---|
| Nº de peças não defeituosas | >7 | Então seriam: 8, 9, 10 peças não defeituosas. |
| Nº de peças defeituosas | <3 | Então seriam: 2, 1, 0 peças defeituosas. |

Dessa forma, para obter a probabilidade de mais de 7 peças não defeituosas deveríamos calcular a probabilidade de menos de três peças defeituosas na amostra:

$P(X < 3) = P(X \leq 2) = P(X = 0) + P(X = 1) + P(X = 2)$ = 0,9884

**Utilizando o Python:**

```
from scipy.stats import binom

# Parâmetros
p_defeituosas = 0.05
n = 10

# a) P(X=7)
prob_a = binom.pmf(7, n, p_defeituosas)

# b) P(X<=2)
prob_b = binom.cdf(2, n, p_defeituosas)

# c) P(X<=8)
prob_c = binom.cdf(8, n, p_defeituosas)

# d) P(X<3)
prob_d = binom.cdf(2, n, p_defeituosas)

# Resultados
print("a) P(X=7):", prob_a)
print("b) P(X<=2):", prob_b)
print("c) P(X<=8):", prob_c)
```

```
print("d) P(X<3):", prob_d)
```

```
a) P(X=7): 8.037890624999999e-08
b) P(X<=2): 0.9884964426207031
c) P(X<=8): 0.9999999999813477
d) P(X<3): 0.9884964426207031
```

Essa implementação usa a função pmf para calcular a probabilidade de um valor específico de X (letra a), e a função cdf para calcular a probabilidade acumulada até certo valor de X (letras b, c, d).

*Exercício 1*

Cada amostra de ar tem 10% de chance de conter uma certa molécula rara. Considere que as amostras sejam independentes com relação à presença da molécula rara. Encontre a probabilidade de que nas próximas 18 amostras:

a) Exatamente 2 contenham a molécula rara.
b) No mínimo 4 amostras contenham a molécula rara.
c) De 3 a 7 amostras contenham a molécula rara.
d) O número médio e a variância de moléculas raras.

Respostas: a) 0,2835 b) 0,0982 c) 0,2660 d) 1,8 e 1,62.

Iniciando o raciocínio:

X: Número de moléculas raras

n=18

p=0,10

q=0,90

X pode ser descrita pela distribuição Binomial.

a) Exatamente 2 contenham a molécula rara. → $P(X=2)$
b) No mínimo 4 amostras contenham a molécula rara. → $P(X \geq 4) = 1 - P(X<4) = 1 - P(X \leq 3)$
c) De 3 a 7 amostras contenham a molécula rara. → $P(3 \leq X \leq 7) = P(X \leq 7) - P(X \leq 2)$
d) O número médio e a variância de moléculas raras. → $E(X) = n.p$ ; $Var(X) = n.p.q$

**Utilizando o Python:**

```
from scipy.stats import binom

# Parâmetros
p_molecula_rara = 0.10
n_amostras = 18

# a) Exatamente 2 contenham a molécula rara
prob_a = binom.pmf(2, n_amostras, p_molecula_rara)

# b) No mínimo 4 amostras contenham a molécula rara
prob_b = 1 - binom.cdf(3, n_amostras, p_molecula_rara)

# c) De 3 a 7 amostras contenham a molécula rara
prob_c = binom.cdf(7, n_amostras, p_molecula_rara) - binom.cdf(2, n_amostras, p_molecula_rara)

# Calculando número médio de moléculas raras (esperança)
media = n_amostras * p_molecula_rara

# Calculando variância
variancia = n_amostras * p_molecula_rara * (1 - p_molecula_rara)

# Resultados
print("a) Probabilidade de exatamente 2 amostras conterem a molécula rara:", prob_a)
```

```
print("b) Probabilidade de no mínimo 4 amostras conterem a molécula rara:", prob_b)
print("c) Probabilidade de 3 a 7 amostras conterem a molécula rara:", prob_c)
print("d) Número médio de moléculas raras:", media)
print("   Variância de moléculas raras:", variancia)
```

```
a) Probabilidade de exatamente 2 amostras conterem a molécula rara: 0.2835120888943317
b) Probabilidade de no mínimo 4 amostras conterem a molécula rara: 0.09819684142543739
c) Probabilidade de 3 a 7 amostras conterem a molécula rara: 0.2660305478747673
d) Número médio de moléculas raras: 1.8
   Variância de moléculas raras: 1.62
```

*Exercício 2*

Se 20% dos parafusos produzidos por uma máquina são defeituosos, determine qual a probabilidade de, entre 4 parafusos selecionados ao acaso, no máximo 2 deles serem defeituosos.

Resposta:0,9728

Iniciando o raciocínio:

X: Número de parafusos defeituosos

n=4

p=0,20

q=0,80

X pode ser descrita pela distribuição Binomial.

Pergunta: "...determine qual a probabilidade de, entre 4 parafusos selecionados ao acaso, no máximo 2 deles serem defeituosos."

$P(X<2) = ?$

**Utilizando o Python:**

```
from scipy.stats import binom

# Parâmetros
p_defeituosos = 0.20  # Probabilidade de um parafuso ser defeituoso
n = 4  # Número total de parafusos selecionados

# Calculando a probabilidade de no máximo 2 serem defeituosos
prob_max_2_defeituosos = binom.cdf(2, n, p_defeituosos)

# Resultado
print("Probabilidade de no máximo 2 parafusos serem defeituosos:", prob_max_2_defeituosos)

Probabilidade de no máximo 2 parafusos serem defeituosos: 0.9728
```

*Exercício 3*

Um fabricante de certas peças de automóveis garante que uma caixa de suas peças conterá no máximo 2 itens defeituosos. Se a caixa contém 20 peças e a experiência tem demonstrado que esse processo de fabricação produz 2 por cento de itens defeituosos, qual a probabilidade de que uma caixa de suas peças não vá satisfazer a garantia?

Resposta: 0,0071

Iniciando o raciocínio:

X: Número de itens defeituosos

$n=20$

$p=0,02$

$q=0,98$

X pode ser descrita pela distribuição Binomial.

Pergunta: "...qual a probabilidade de que uma caixa de suas peças não vá satisfazer a garantia?"

Qual é a garantia? "Um fabricante de certas peças de automóveis garante que uma caixa de suas peças conterá no máximo 2 itens defeituosos..."

Para não satisfazer a garantia: $P(X>2) = 1 - P(X\leq 2) = ?$

**Utilizando o Python:**

```
from scipy.stats import binom

# Parâmetros
p_defeituosos = 0.02  # Probabilidade de um item ser defeituoso
n = 20  # Número total de itens na caixa

# Probabilidade de no máximo 2 itens defeituosos
prob_max_2_defeituosos = binom.cdf(2, n, p_defeituosos)

# Probabilidade de que uma caixa não satisfaça a garantia
prob_nao_satisfaz_garantia = 1 - prob_max_2_defeituosos

# Resultado
print("Probabilidade de uma caixa não satisfazer a garantia:", prob_nao_satisfaz_garantia)

Probabilidade de uma caixa não satisfazer a garantia: 0.007068693403814108
```

## Distribuição Poisson

*Distribuição Poisson para o cálculo de* ***P(X=x)***

*poisson.pmf(x,media)*

*Distribuição Poisson para o cálculo de* ***P(0≤X≤x) = P(X≤x)***

*poisson.cdf(x,media)*

*Distribuição Poisson para o cálculo de* ***P(X>x)***

*poisson.sf(x,media)*

onde:

Media= lambda ou número médio de eventos ocorrendo no intervalo considerado.

**Importante:**

Para usar as funções de cálculo de probabilidade para a distribuição binomial no Python é necessário primeiramente que você importe a função *poisson:*

**from** scipy.stats **import** poisson

*Exemplo 2*

Uma central telefônica recebe, em média, cinco chamadas por minuto.

a) Defina a variável aleatória.
X: Número de chamadas recebidas por minuto.
x: 0, 1, 2, ...
X ~ Po(5)

b) Calcule a probabilidade de que durante um intervalo de um minuto:

i. A central telefônica não receba chamada.

$$P(X=0)=\frac{e^{-5}.5^{0}}{0!}=0{,}0067$$

ii. Receba, no máximo, uma chamada.

$$P(X \leq 1) = P(X = 0) + P(X = 1) = 0{,}0067 + 0{,}0337 = 0{,}0404$$

$$P(X=1)=\frac{e^{-5}.5^{1}}{1!}=0{,}0337$$

iii. Receba mais de duas chamadas.

$$P(X > 2) = 1 - P(X \leq 2) = 1 - [P(X = 0) + P(X = 1) + P(X = 2)]$$

$$P(X > 2) = 1 - 0{,}1246 = 0{,}8754$$

$$P(X=2)=\frac{e^{-5}.5^{2}}{2!}=0{,}0842$$

c) Durante um intervalo de quatro minutos, qual a probabilidade de que ocorram 15 chamadas? Nesse caso, como o intervalo de tempo foi alterado, o parâmetro da distribuição também irá sofrer alteração proporcional, ou seja, $\lambda$= 5 chamadas por minuto passará a ser $\lambda$= **20 chamadas por quatro minutos**. Assim:

$$P(X=15)=\frac{e^{-20}.20^{15}}{15!}=0{,}0516$$

**Utilizando o Python:**

```
from scipy.stats import poisson

# Parâmetros
media_chamadas_por_minuto = 5
intervalo_tempo_minuto = 1
intervalo_tempo_minutos = 4
num_chamadas_4_minutos = 15

# a) i. Probabilidade de não receber chamada em um minuto
prob_nenhuma_chamada = poisson.pmf(0, media_chamadas_por_minuto * intervalo_tempo_minuto)

# a) ii. Probabilidade de receber no máximo uma chamada em um minuto
prob_max_uma_chamada = poisson.cdf(1, media_chamadas_por_minuto * intervalo_tempo_minuto)

# a) iii. Probabilidade de receber mais de duas chamadas em um minuto
prob_mais_de_2_chamadas = 1 - poisson.cdf(2, media_chamadas_por_minuto * intervalo_tempo_minuto)

# b) Probabilidade de ocorrerem 15 chamadas em quatro minutos
prob_15_chamadas_4_minutos = poisson.pmf(num_chamadas_4_minutos, media_chamadas_por_minuto *
intervalo_tempo_minutos)

# Resultados
print("a) i. Probabilidade de não receber chamada em um minuto:", prob_nenhuma_chamada)
print("a) ii. Probabilidade de receber no máximo uma chamada em um minuto:", prob_max_uma_chamada)
```

```
print("a) iii. Probabilidade de receber mais de duas chamadas em um minuto:", prob_mais_de_2_chamadas)
print("b) Probabilidade de ocorrerem 15 chamadas em quatro minutos:", prob_15_chamadas_4_minutos)
```

```
a) i. Probabilidade de não receber chamada em um minuto: 0.006737946999085467
a) ii. Probabilidade de receber no máximo uma chamada em um minuto: 0.04042768199451279
a) iii. Probabilidade de receber mais de duas chamadas em um minuto: 0.8753479805169189
b) Probabilidade de ocorrerem 15 chamadas em quatro minutos: 0.05164885353175814
```

Essa implementação usa a função pmf para calcular a probabilidade de um número específico de chamadas (a i) e (b) em um dado intervalo de tempo, e a função cdf para calcular a probabilidade acumulada até um certo número de chamadas (a ii e iii).

*Exercício 4*

Falhas ocorrem, ao acaso, ao longo do comprimento de um fio delgado de cobre. Suponha que o número de falhas siga a distribuição de Poisson, com uma média de 2,3 falhas por milímetro.

a) Determine a probabilidade de existir exatamente 2 falhas em 1 milímetro de fio.
b) Determine a probabilidade de existir entre 2 e 4 falhas em 1 milímetro de fio.
c) Determine a probabilidade de 10 falhas em 5 milímetros de fio.
d) Determine a probabilidade de existir, no mínimo, uma falha em 2 milímetros de fio.

Respostas: a) 0,2652 b)0,2033 c) 0,1129 d) 0,99

Iniciando o raciocínio:

X: Número de falhas por milímetro

$\lambda$=2,3

X pode ser descrita pela distribuição Poisson.

**Utilizando o Python:**

```
from scipy.stats import poisson

# Parâmetros
media_falhas_por_mm = 2.3
comprimento_mm = 1
comprimento_mm_5 = 5
comprimento_mm_2 = 2

# a) Probabilidade de existir exatamente 2 falhas em 1 milímetro de fio
prob_2_falhas_1_mm = poisson.pmf(2, media_falhas_por_mm * comprimento_mm)

# b) Probabilidade de existir entre 2 e 4 falhas em 1 milímetro de fio
prob_entre_2_e_4_falhas_1_mm = poisson.pmf(3, media_falhas_por_mm * comprimento_mm)

# c) Probabilidade de 10 falhas em 5 milímetros de fio
prob_10_falhas_5_mm = poisson.pmf(10, media_falhas_por_mm * comprimento_mm_5)

# d) Probabilidade de existir, no mínimo, uma falha em 2 milímetros de fio
prob_minimo_uma_falha_2_mm = 1 - poisson.cdf(0, media_falhas_por_mm * comprimento_mm_2)

# Resultados
print("a) Probabilidade de existir exatamente 2 falhas em 1 milímetro de fio:", prob_2_falhas_1_mm)
print("b) Probabilidade de existir entre 2 e 4 falhas em 1 milímetro de fio:", prob_entre_2_e_4_falhas_1_mm)
print("c) Probabilidade de 10 falhas em 5 milímetros de fio:", prob_10_falhas_5_mm)
print("d) Probabilidade de existir, no mínimo, uma falha em 2 milímetros de fio:",
prob_minimo_uma_falha_2_mm)
```

a) Probabilidade de existir exatamente 2 falhas em 1 milímetro de fio: 0.2651846416468159
b) Probabilidade de existir entre 2 e 4 falhas em 1 milímetro de fio: 0.20330822526255884
c) Probabilidade de 10 falhas em 5 milímetros de fio: 0.11293507088124335
d) Probabilidade de existir, no mínimo, uma falha em 2 milímetros de fio: 0.9899481642553665

*Exercício 5*

O número de falhas em parafusos de máquinas da indústria têxtil segue distribuição de Poisson, com uma média de 0,1 falha por metro quadrado.

a) Qual é a probabilidade de que haja duas falhas em 1 metro quadrado de tecido?

b) Qual é a probabilidade de que haja uma falha em 10 metros quadrados de tecido?

c) Qual é a probabilidade de que não haja falhas em 20 metros quadrados de tecido?

d) Qual é a probabilidade de que haja no mínimo duas falhas em 10 metros quadrados de tecido?

Respostas: a) 0,0045 b)0,3679 c) 0,1353 d) 0,2642

Iniciando o raciocínio:

X: Número de falhas por metro quadrado

$\lambda=0,1$

X pode ser descrita pela distribuição Poisson.

**Utilizando o Python:**

```
from scipy.stats import poisson

# Parâmetros
media_falhas_por_metro_quadrado = 0.1

# a) Probabilidade de existirem duas falhas em 1 metro quadrado de tecido
prob_2_falhas_1_metro_quadrado = poisson.pmf(2, media_falhas_por_metro_quadrado * 1)

# b) Probabilidade de existir uma falha em 10 metros quadrados de tecido
prob_1_falha_10_metros_quadrados = poisson.pmf(1, media_falhas_por_metro_quadrado * 10)

# c) Probabilidade de não haver falhas em 20 metros quadrados de tecido
prob_nenhuma_falha_20_metros_quadrados = poisson.pmf(0, media_falhas_por_metro_quadrado * 20)

# d) Probabilidade de haver no mínimo duas falhas em 10 metros quadrados de tecido
prob_minimo_2_falhas_10_metros_quadrados = 1 - poisson.cdf(1, media_falhas_por_metro_quadrado * 10)

# Resultados
print("a) Probabilidade de existirem duas falhas em 1 metro quadrado de tecido:",
prob_2_falhas_1_metro_quadrado)
print("b) Probabilidade de existir uma falha em 10 metros quadrados de tecido:",
prob_1_falha_10_metros_quadrados)
print("c) Probabilidade de não haver falhas em 20 metros quadrados de tecido:",
prob_nenhuma_falha_20_metros_quadrados)
print("d) Probabilidade de haver no mínimo duas falhas em 10 metros quadrados de tecido:",
prob_minimo_2_falhas_10_metros_quadrados)
```

a) Probabilidade de existirem duas falhas em 1 metro quadrado de tecido: 0.004524187090179801
b) Probabilidade de existir uma falha em 10 metros quadrados de tecido: 0.36787944117144233
c) Probabilidade de não haver falhas em 20 metros quadrados de tecido: 0.1353352832366127
d) Probabilidade de haver no mínimo duas falhas em 10 metros quadrados de tecido: 0.26424111765711533

*Exercício 6*

Um engenheiro de tráfego monitora o fluxo de carros em um cruzamento que tem uma média de 6 carros por minuto. Para estabelecer o tempo de um sinal, as seguintes probabilidades são utilizadas:

a) Qual é a probabilidade de nenhum carro passar pelo cruzamento em 30 segundos?

b) Qual é a probabilidade de três ou mais carros passarem pelo cruzamento em 30 segundos?

Respostas: a) 0,0498 b)0,5768

Iniciando o raciocínio:

X: Número de carros que passam em um cruzamento por minuto

$\lambda=6$

X pode ser descrita pela distribuição Poisson.

**Utilizando o Python:**

```
from scipy.stats import poisson

# Parâmetros
media_carros_por_minuto = 6
tempo_segundos = 30

# Convertendo a média para carros por 30 segundos
media_carros_por_30_segundos = (media_carros_por_minuto / 60) * tempo_segundos

# a) Probabilidade de nenhum carro passar pelo cruzamento em 30 segundos
prob_nenhum_carro = poisson.pmf(0, media_carros_por_30_segundos)

# b) Probabilidade de três ou mais carros passarem pelo cruzamento em 30 segundos
prob_tres_ou_mais_carros = 1 - poisson.cdf(2, media_carros_por_30_segundos)

# Resultados
print("a) Probabilidade de nenhum carro passar pelo cruzamento em 30 segundos:", prob_nenhum_carro)
print("b) Probabilidade de três ou mais carros passarem pelo cruzamento em 30 segundos:",
prob_tres_ou_mais_carros)
```

a) Probabilidade de nenhum carro passar pelo cruzamento em 30 segundos: 0.049787068367863944

b) Probabilidade de três ou mais carros passarem pelo cruzamento em 30 segundos: 0.5768099188731564