UNIVERSIDADE FEDERAL DE UBERLÂNDIA – UFU
Graduação em Ciência da Computação

Atividade Prática 04

GBC065 – Modelagem e Simulação

Uberlândia
2018

Atividade Prática 04

Trabalho apresentado à disciplina de Modelagem e Simulação (GBC065), ministrada pelo professor Anderson Rodrigues dos Santos, para o curso de Bacharelado em Ciência da Computação, no período 2018-2, na Universidade Federal de Uberlândia.

Grupo 02 – Integrantes:

Antonio Carlos Neto
11611BCC054

Ronistone Gonçalves dos Reis Júnior
11521BCC018

Uberlândia
2018

Exercise 4.1.7:

(a) Generate an Exponential (9) random variate sample of size n = 100 and compute the proportion of points in the sample that fall within the intervals $\bar{x} \pm 2s$ and $\bar{x} \pm 3s$. Do this for 10 different rngs streams.

R: Considerações:

- O programa utilizado é uma edição do **uvs.c**, utilizando a função Exponential e a biblioteca **rngs.h**. Portanto, utilizamos **Welford's algorithm**.
- Para saber se Xi pertence ao intervalo dado, foi considerado as seguinte fórmulas:
  - $p_k = |S_k|/n$;
  - $X_i \in S_k$, se $| X_i - (\bar{X}) | < ks$;
    - s : desvio padrão;
    - $(\bar{X})$: média;
    - k : 2 e 3.

```
Welford of size 100
Seed ................................ = 123456789
Mean ............................... =   7.785
Standard Deviation ................. =   9.147
Proportion of Points if k =   2 .... =   0.910
Proportion of Points if k =   3 .... =   0.980

Welford of size 100
Seed ................................ = 2010924726
Mean ............................... =  10.512
Standard Deviation ................. =  10.327
Proportion of Points if k =   2 .... =   0.980
Proportion of Points if k =   3 .... =   0.980

Welford of size 100
Seed ................................ = 417893401
Mean ............................... =   8.476
Standard Deviation ................. =   8.308
Proportion of Points if k =   2 .... =   0.940
Proportion of Points if k =   3 .... =   0.990

Welford of size 100
Seed ................................ = 281668658
Mean ............................... =   9.223
Standard Deviation ................. =   8.582
Proportion of Points if k =   2 .... =   0.940
Proportion of Points if k =   3 .... =   1.000

Welford of size 100
Seed ................................ = 1918141768
Mean ............................... =   9.651
Standard Deviation ................. =   8.681
Proportion of Points if k =   2 .... =   0.970
Proportion of Points if k =   3 .... =   0.980

Welford of size 100
Seed ................................ = 1524875428
Mean ............................... =   8.782
Standard Deviation ................. =   8.299
Proportion of Points if k =   2 .... =   0.980
Proportion of Points if k =   3 .... =   0.990

Welford of size 100
Seed ................................ = 1030381034
Mean ............................... =   7.766
Standard Deviation ................. =   8.599
Proportion of Points if k =   2 .... =   0.920
Proportion of Points if k =   3 .... =   0.970

Welford of size 100
Seed ................................ = 1312571097
Mean ............................... =   8.892
Standard Deviation ................. =   8.478
Proportion of Points if k =   2 .... =   0.970
Proportion of Points if k =   3 .... =   0.980

Welford of size 100
Seed ................................ = 151536961
Mean ............................... =   8.713
Standard Deviation ................. =  10.272
Proportion of Points if k =   2 .... =   0.970
Proportion of Points if k =   3 .... =   0.980

Welford of size 100
Seed ................................ = 1503773726
Mean ............................... =   9.033
Standard Deviation ................. =  10.396
Proportion of Points if k =   2 .... =   0.940
Proportion of Points if k =   3 .... =   0.980
```

(b) In each case, compare the results with Chebyshev's inequality.

R:

- Sabendo que:

Chebyshev's inequality

$$p_k \geq 1 - \frac{1}{k^2} \qquad\qquad (k > 1).$$

- Temos:
  - $p_2 \geq 1 - 1/(2)^2 \Rightarrow p_2 \geq 1 - 0{,}25 \Rightarrow p_2 \geq 0{,}75$;
  - $p_3 \geq 1 - 1/(3)^2 \Rightarrow p_3 \geq 1 - 0{,}11 \Rightarrow p_3 \geq 0{,}89$.

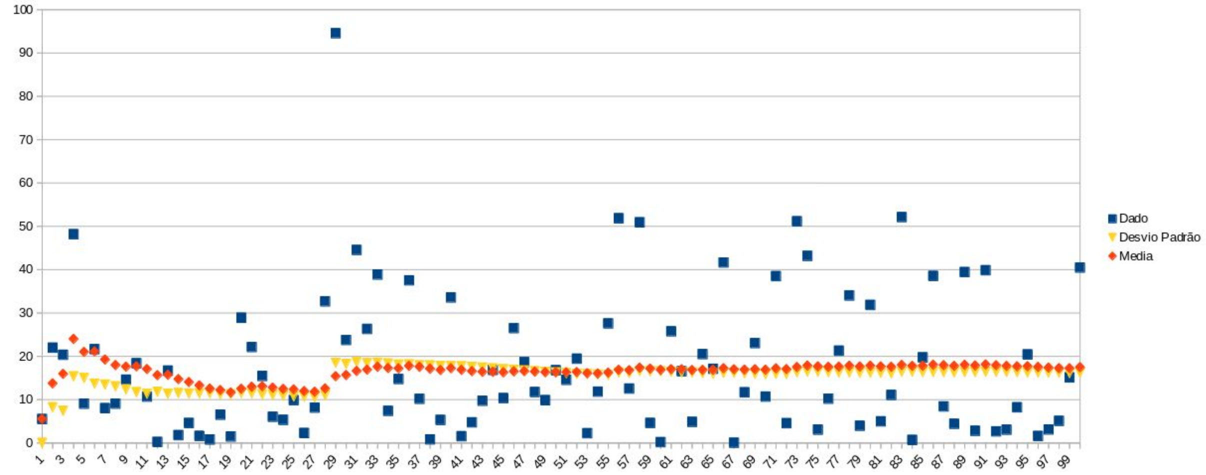
- Comparando os dados, temos que todas as proporções calculadas são válidas.

(c) Comment.
R: Como as probabilidades calculadas estão acima de 0,90, temos que a dispersão dos dados é baixa, sendo que menos de 10% dos dados estão distantes da média.

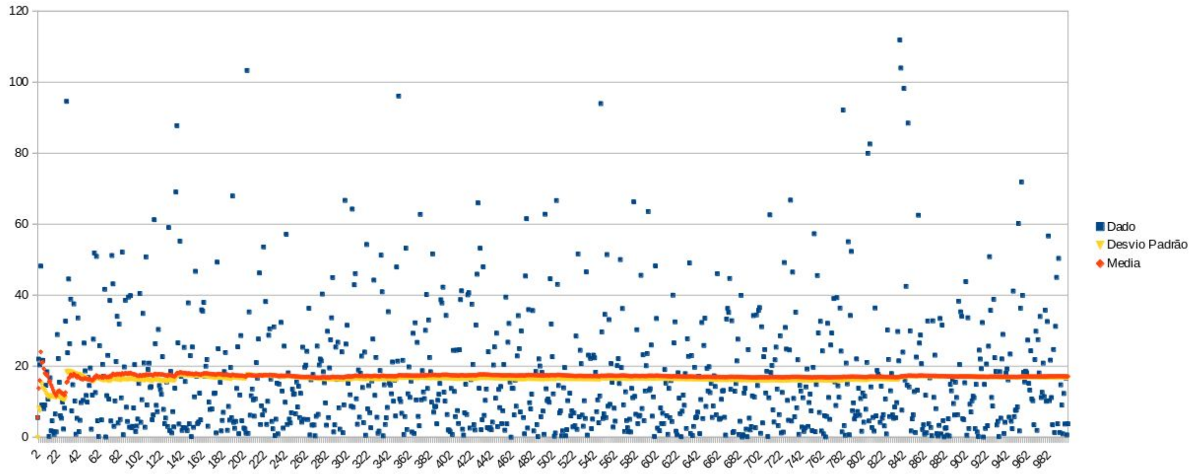
Exercise 4.1.8:

(a) Generate a plot similar to that in Figure 4.1.2 with calls to Exponential (17), rather than Random to generate the variates. Indicate the values to which the sample mean and sample standard deviation will converge.

Com 100 Amostras:



Com 1000 Amostras



- Considerando:

Similarly, if an $\mathit{Exponential}(\mu)$ random variate sample is generated then

$$\bar{x} \to \mu \quad \text{and} \quad s \to \mu$$

- Temos que a média e o desvio padrão convergem para o número 17.

Exercise 4.1.11:

Calculate x̄ and s by hand using the 2-pass algorithm, the 1-pass algorithm, and Welford's algorithm in the following two cases.

(a) The data based on n = 3 observations: $x_1 = 1$, $x_2 = 6$, and $x_3 = 2$.
R:

- **2-pass algorithm:**

Definition 4.1.1 Given the sample $x_1, x_2, \ldots, x_n$ (either continuous or discrete data) the *sample mean* is

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

The *sample variance* is the average of the squared differences about the sample mean

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2.$$

The *sample standard deviation* is the positive square root of the sample variance, $s = \sqrt{s^2}$.

  - **$\bar{x}$: (1 + 6 + 2)/3 = 9/3 = 3**
  - **s^2: ($(1 - 3)^2 + (6 - 3)^2 + (2 - 3)^2$)/3 = (4 + 9 + 1)/3 = 4.667**
  - **s: 2.16**

- **1-pass algorithm:**

$$
\begin{aligned}
s^2 &= \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(x_i^2 - 2\bar{x}x_i + \bar{x}^2) \\
&= \left(\frac{1}{n}\sum_{i=1}^{n} x_i^2\right) - \left(\frac{2}{n}\bar{x}\sum_{i=1}^{n} x_i\right) + \left(\frac{1}{n}\sum_{i=1}^{n} \bar{x}^2\right) \\
&= \left(\frac{1}{n}\sum_{i=1}^{n} x_i^2\right) - 2\bar{x}^2 + \bar{x}^2 \\
&= \left(\frac{1}{n}\sum_{i=1}^{n} x_i^2\right) - \bar{x}^2.
\end{aligned}
$$

- **$\bar{x}$: (1 + 6 + 2)/3 = 9/3 = 3**
- **s^2: $(1^2 + 6^2 + 2^2)/3 - 3^2$ = 41/3 - 9 = 4.667**
- **s: 2.16**

- **Welford's algorithm:**

Theorem 4.1.2 For $i = 1, 2, 3, \ldots$, the variables $\bar{x}_i$ and v_i in Definition 4.1.2 can be computed recursively by

$$
\begin{aligned}
\bar{x}_i &= \bar{x}_{i-1} + \frac{1}{i}(x_i - \bar{x}_{i-1}) \\
v_i &= v_{i-1} + \left(\frac{i-1}{i}\right)(x_i - \bar{x}_{i-1})^2
\end{aligned}
$$

with the initial conditions $\bar{x}_0 = 0$ and $v_0 = 0$.

- **$\bar{x}$: $\bar{x}_3$ = 3**
  - $\bar{x}_0$ = 0;
  - $\bar{x}_1 = \bar{x}_0$ + (1/1)*(1 - $\bar{x}_0$) = 0 + 1 = 1;
  - $\bar{x}_2 = \bar{x}_1$ + (½)*(6 - $\bar{x}_1$) = 1 + 5/2 = 3.5;
  - $\bar{x}_3 = \bar{x}_2$ + (⅓)*(2 - $\bar{x}_2$) = 3.5 - 0.5 = 3.
- **v: v_3 = 14**

- $v_0 = 0$;
- $v_1 = v_0 + (0/1)*(1 - \bar{x}_0)^2 = 0 + 0 = 0$;
- $v_2 = v_1 + (½)*(6 - \bar{x}_1)^2 = 0 + 25/2 = 12.5$;
- $v_3 = v_2 + (⅔)*(2 - \bar{x}_2)^2 = 12.5 + 1.5 = 14$;

○ **s: $\sqrt{(v_3/3)} = 2.16$**

(b) The sample path x(t) = 3 for 0 < t ≤ 2, and x(t) = 8 for 2 < t ≤ 5, over the time interval 0 < t ≤ 5.
R:

- **2-pass algorithm:**

Definition 4.1.3 Given a function $x(t)$ defined for all $0 < t < \tau$ as a realization (sample path) of a stochastic process the *sample-path mean* is

$$\bar{x} = \frac{1}{\tau}\int_0^\tau x(t)\,dt.$$

The associated *sample-path variance* is

$$s^2 = \frac{1}{\tau}\int_0^\tau \big(x(t) - \bar{x}\big)^2 dt$$

and the *sample-path standard deviation* is $s = \sqrt{s^2}$.

  - **$\bar{x}$: ((2 - 0)*3 + (5 - 2)*8)/5 = (6 + 24)/ 5= 30/5 = 6**
  - **s^2: ((2 - 0)*(3 - 6)² + (5 - 2)*(8 - 6)²)/5 = (18 + 12)/5 = 6**
  - **s: √6 = 2.449**

- **1-pass algorithm:**

The equation for s^2 in Definition 4.1.3 is the two-pass variance equation; the corresponding one-pass equation is

$$s^2 = \left(\frac{1}{\tau}\int_0^\tau x^2(t)\,dt\right) - \bar{x}^2.$$

  - **$\bar{x}$: ((2 - 0)*3 + (5 - 2)*8)/5 = (6 + 24)/ 5= 30/5 = 6**
  - **s^2: ((2 - 0)*3² + (5 - 2)*8²)/5 - 6² = (18 + 192)/5 - 36 = 6**
  - **s: √6 = 2.449**

- **Welford's algorithm:**

inter-event times $\delta_i = t_i - t_{i-1}$ for $i = 1, 2, \ldots, n$

Theorem 4.1.4 For $i = 1, 2, \ldots, n$ the variables $\bar{x}_i$ and v_i in Definition 4.1.4 can be computed recursively by

$$\bar{x}_i = \bar{x}_{i-1} + \frac{\delta_i}{t_i}(x_i - \bar{x}_{i-1})$$

$$v_i = v_{i-1} + \frac{\delta_i t_{i-1}}{t_i}(x_i - \bar{x}_{i-1})^2$$

with the initial conditions $\bar{x}_0 = 0$ and $v_0 = 0$.

- **$\bar{x}$: $\bar{x}_2 = 6$**
  - $\bar{x}_0 = 0$;
  - $\bar{x}_1 = \bar{x}_0 + (2/2)*(3 - \bar{x}_0) = 0 + 3 = 3$;
  - $\bar{x}_2 = \bar{x}_1 + (3/5)*(8 - \bar{x}_1) = 3 + 3 = 6$.
- **v:**
  - $v_0 = 0$;
  - $v_1 = v_0 + ((2*0)/2)*(3 - \bar{x}_0)^2 = 0 + 0 = 0$;
  - $v_2 = v_1 + ((3*2)/5)*(8 - \bar{x}_1)^2 = 0 + (6/5)*25 = 30$.
- **s: $\sqrt{(v_2/5)} = \sqrt{6} = 2.449$**

Exercise 4.1.13:

(a) Generate an Exponential(7) random variate sample of size n = 1000 and compute the mean and standard deviation using the Conventional One pass algorithm and the Algorithm 4.1.1. Comment on the results.
R:

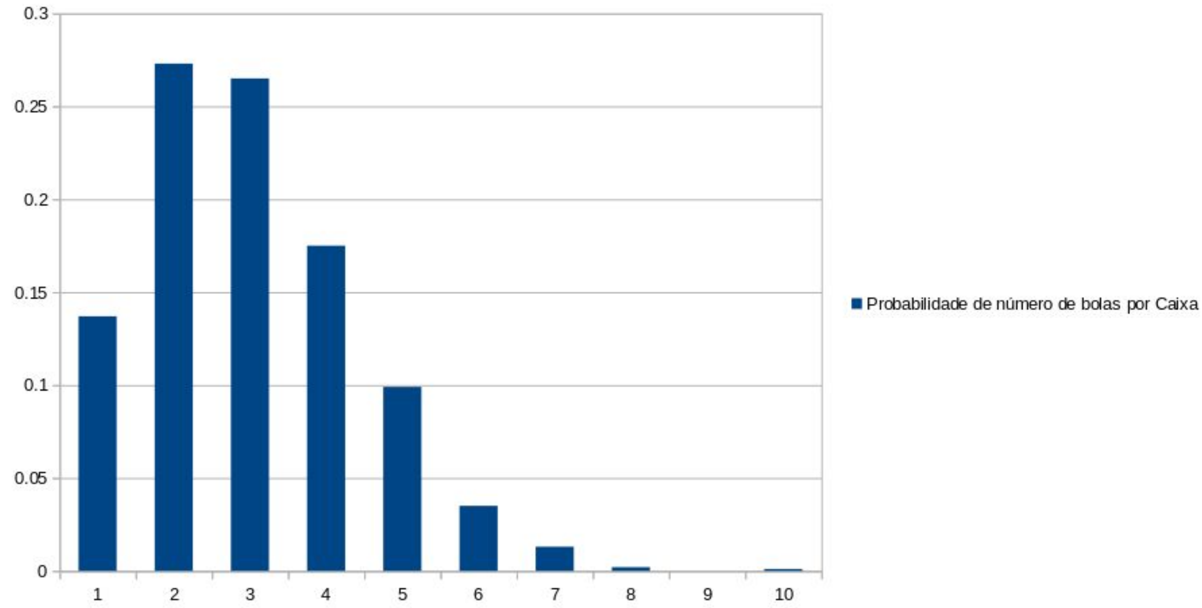
```
One pass algorithm of size 1000
Rng ............................. = 123456789
Mean ............................ =     6.948
Standard Deviation .............. =     6.938

Welford of size 1000
Rng ............................. = 123456789
Mean ............................ =     6.948
Standard Deviation .............. =     6.938
```

- Os valores da média e do desvio padrão são o mesmo para os dois casos, comprovando que as fórmulas são equivalentes. Porém, pode existir uma diferença mínima por conta de arredondamento de ponto flutuante, no caso dos valores serem extremamente pequenos ou grandes. Pela fórmula vista anteriormente vemos que os valores convergem para 7.

Exercise 4.2.2:

(a) Generate the 2000-ball histogram in Example 4.2.2.
R:



Selecionar C:\Windows\Syste

```
p[ 0] = 0.137
p[ 1] = 0.273
p[ 2] = 0.265
p[ 3] = 0.175
p[ 4] = 0.099
p[ 5] = 0.035
p[ 6] = 0.013
p[ 7] = 0.002
p[ 8] = 0.000
p[ 9] = 0.001
TOTAL: 1.000000
```

(b) Verify that the resulting relative frequencies f(x) satisfy the equation

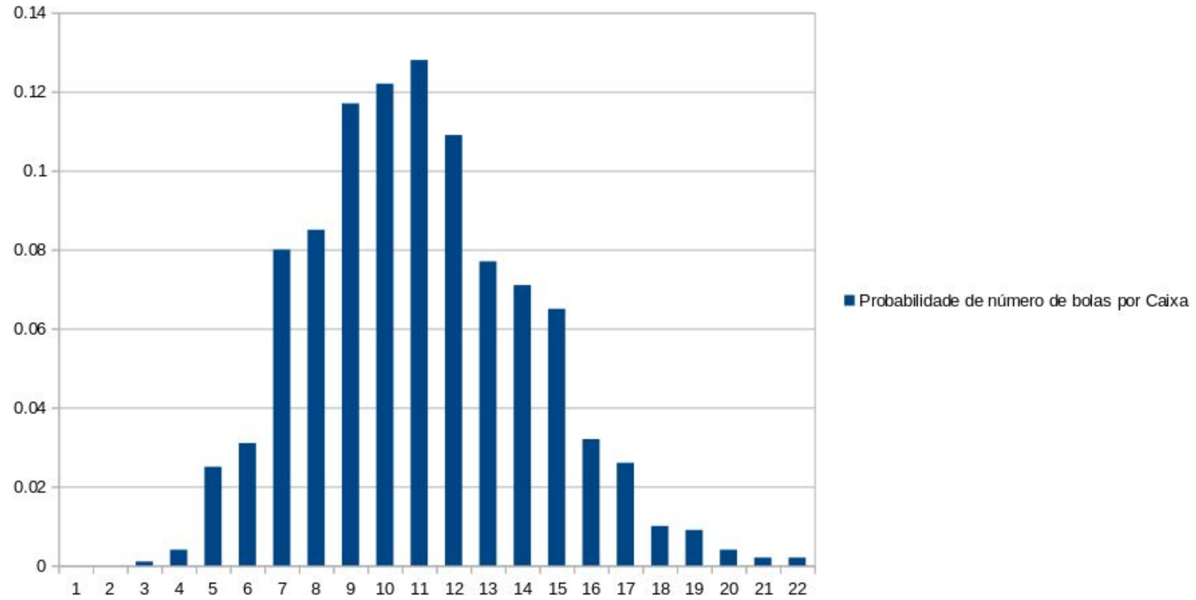
$$\hat{f}(x) \cong \frac{2^x \exp(-2)}{x!} \qquad x = 0, 1, 2, \ldots$$

R:

- **f(0) = (1*0.135)/1 = 0.135 == 0.137;**
- **f(1) = (2*0.135)/1 = 0.27 == 0.273;**
- **f(2) = (4*0.135)/2 = 0.27 == 0.265;**
- **f(3) = (8*0.135)/6 = 0.18 == 0.175;**
- **f(4) = (16*0.135)/24 = 0.09 == 0.099;**
- **f(5) = (32*0.135)/120 = 0.036 == 0.035;**
- **f(6) = (64*0.135)/720 = 0.012 == 0.013;**
- **f(7) = (128*0.135)/5040 = 0.00343 == 0.002;**
- **f(8) = (256*0.135)/40320 = 0.00085 == 0.000;**
- **f(9) = (512*0.135)/362880 = 0.0001904 == 0.001;**

- **Comparando os resultados obtidos pela fórmula e pelo programa feito, temos que são muito semelhantes. Portanto, podemos dizer que os resultados pela fórmula são aceitos.**

(c) Then generate the corresponding histogram if 10 000 balls are placed, at random, in 1000 boxes.
R:



```
p[ 0] = 0.000     p[12] = 0.077
p[ 1] = 0.000     p[13] = 0.071
p[ 2] = 0.001     p[14] = 0.065
p[ 3] = 0.004     p[15] = 0.032
p[ 4] = 0.025     p[16] = 0.026
p[ 5] = 0.031     p[17] = 0.010
p[ 6] = 0.080     p[18] = 0.009
p[ 7] = 0.085     p[19] = 0.004
p[ 8] = 0.117     p[20] = 0.002
p[ 9] = 0.122     p[21] = 0.002
p[10] = 0.128     TOTAL: 1.000000
p[11] = 0.109
```

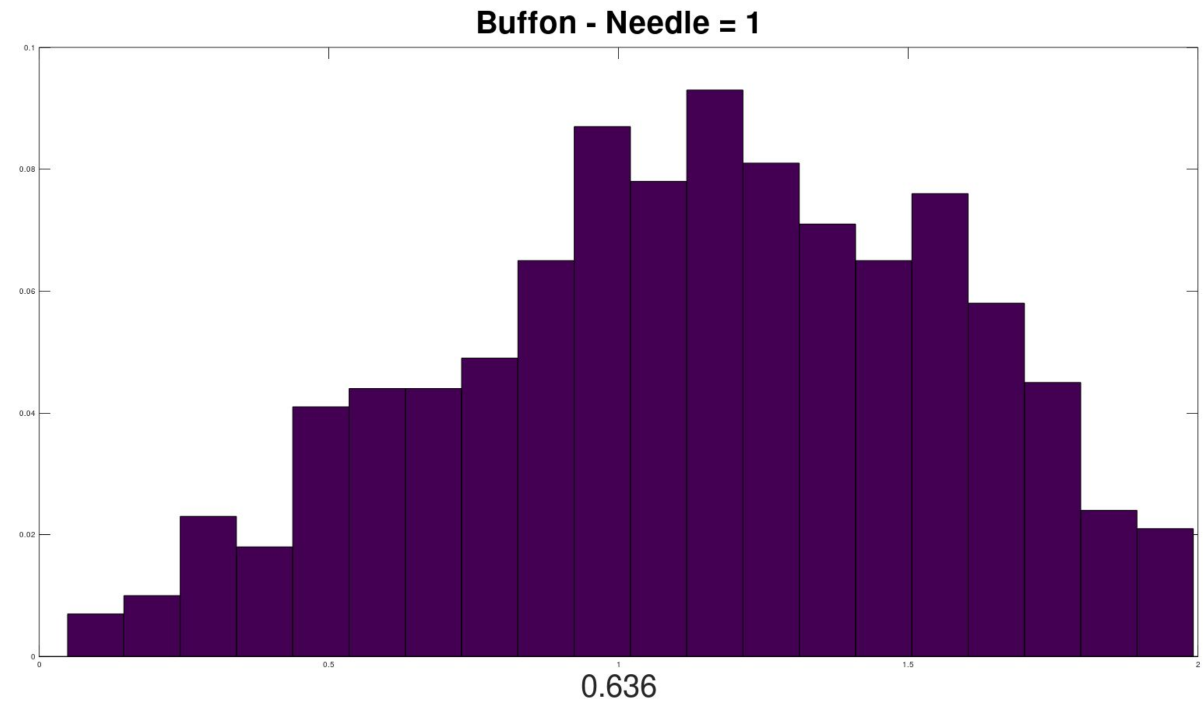
(d) Find an equation that seems to fit the resulting relative frequencies well and illustrate the quality of the fit.
R:

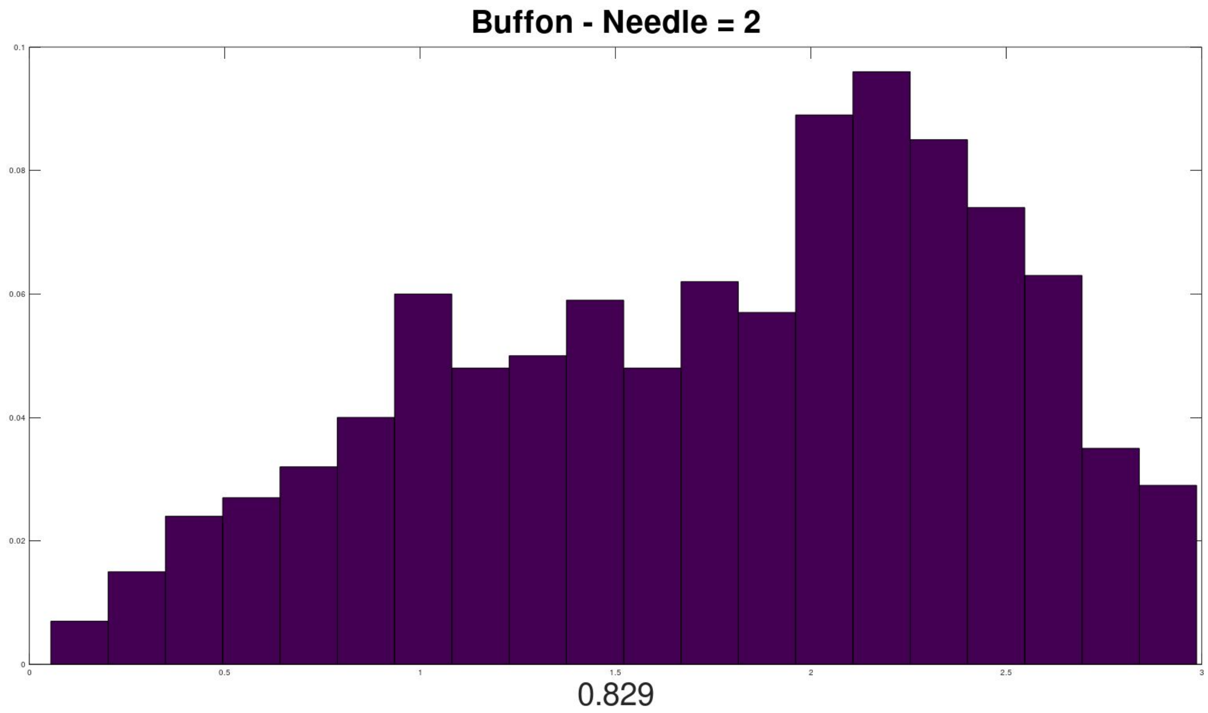
```
f(0) ~=   0.000     f(11) ~=   0.114
f(1) ~=   0.000     f(12) ~=   0.095
f(2) ~=   0.002     f(13) ~=   0.073
f(3) ~=   0.008     f(14) ~=   0.052
f(4) ~=   0.019     f(15) ~=   0.035
f(5) ~=   0.038     f(16) ~=   0.022
f(6) ~=   0.063     f(17) ~=   0.013
f(7) ~=   0.090     f(18) ~=   0.007
f(8) ~=   0.113     f(19) ~=   0.004
f(9) ~=   0.125     f(20) ~=   0.002
f(10) ~=   0.125    f(21) ~=   0.001
```

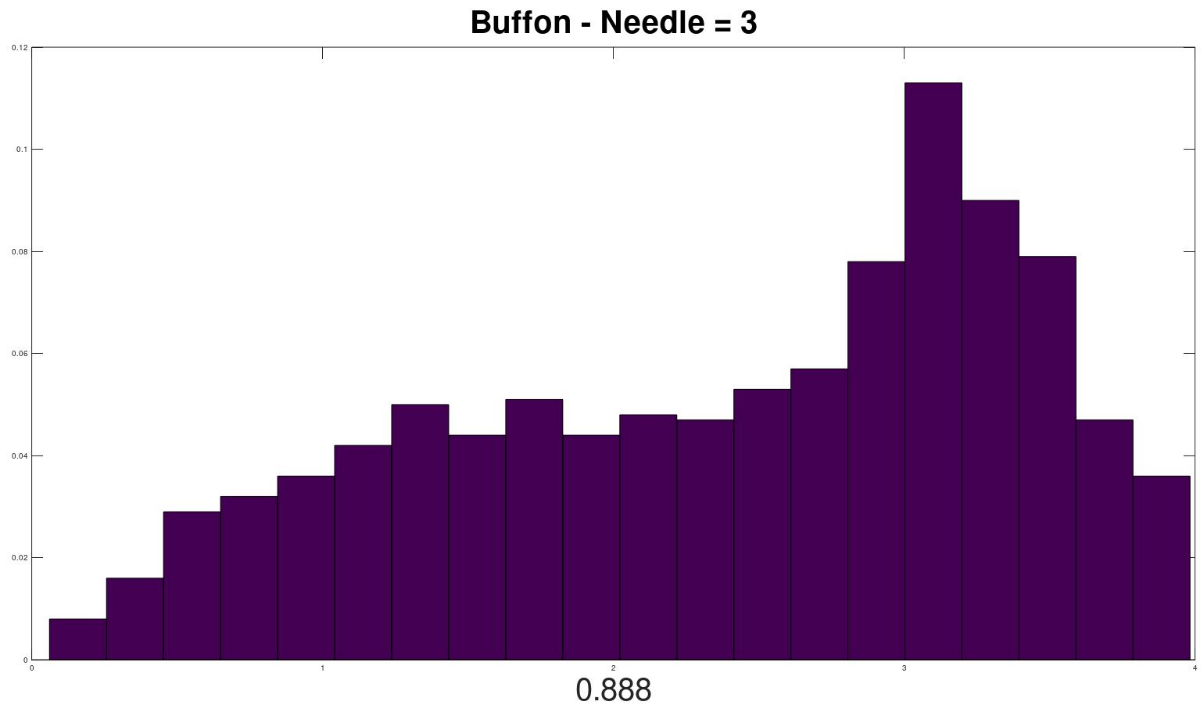
$$\hat{f}(x) \cong \frac{10^{x} \exp(-10)}{x!} \qquad x = 0, 1, 2, \ldots$$

Exercise 4.3.1:

(a) Use program cdh to construct a continuous-data histogram like the one on the left in Example 4.3.1, but corresponding to a needle of length r = 2.
R:







(b) Based on this histogram what is the probability that the needle will cross at least one line.

R: Usando os valores de v do programa buffon.c, com R = 1, 2 e 3, colocamos os dados em três arquivos .txt, com o código desenvolvido em GNU Octave plotamos três histogramas de probabilidade dos dados, depois somamos todos os valores que passam do valor 1, assim temos a probabilidade da agulha cruzar pelo menos uma linha.

Portanto, no histogramas acima tem se a seguinte probabilidade:

- Needle = 1, 0.636;
- Needle = 2, 0.829;
- Needle = 3, 0.888;

(c) What is the corresponding axiomatic probability that a needle of length r = 2 will cross at least one line?
R:

- Para o caso de r=1 a seguinte fórmula calcula a probabilidade de uma agulha aleatoriamente lançada sobre as linhas igualmente espaçadas ficar por sobre alguma linha:

  - **r : 1;**
  - **shadow : %pi - integrate(**
    1-(r*cos(theta)),
    theta,
    -%pi/2,
    %pi/2);
  - **float(shadow/%pi);**

  Probabilidade = 0.6366197723675814

  A fórmula para cálculo da probabilidade varia para cada tamanho de agulha porque depende da interseção da curva u = 1 - r cos (theta) com o eixo x. No caso de r=1, a interseção acontece apenas no ponto (0,0) conforme exibido pelo gráfico anexado a esta imagem. Entretanto, quando utilizamos um valor de r > 1 a curva de u cruza o eixo x em dois pontos e a área abaixo da curva (shadow) que fica com valores de u negativos não podem entrar na soma da área total para cálculo da probabilidade. Por isso, é necessário mapear exatamente onde fica o ponto de interseção, calcular a integral de -%pi/2 até esse ponto, multiplicar por 2 (os pontos de interseção possuem o mesmo módulo), subtrair da área de %pi e dividir por %pi.

- Por exemplo, para r=2: solve(1-2*cos(x)=0,x) resulta em x= %pi/3. Então o cálculo da probabilidade fica assim:

  - **r : 2;**
  - **shadow : %pi-(**
    2*integrate(1-(r*cos(theta)),
    theta,

-%pi/2,
-%pi/3));

- **float(shadow/%pi);**

Probabilidade = 0.8372484205582454

- Por último, para r=3, temos solve(1-3*cos(x)=0,x) resulta em x= acos(1/3). Então o cálculo da probabilidade fica assim:

    - **r : 3;**
    - **shadow : %pi-(**
      2*integrate(1-(r*cos(theta)),
      theta,
      -%pi/2,
      -acos(1/3)));
    - **float(shadow/%pi);**

Probabilidade = 0.8928797888497458