Prática 4 – MS

Grupo:

Andressa Alvilino Ferreira Silva

Matheus Cunha Reis

Weuler Borges

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

a) Ver código.

b) Ver código.

c) No caso de k = 2, a Desigualdade de Chebyshev nos diz que pk >= 0.75 e conforme o código, o resultado para todas as streams conferem.

No caso de k = 3, a Desigualdade de Chebyshex nos diz que pk>=0.89 e também, conforme o código, o resultado para todas as streams conferem.

4.1.8) 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.

Valor converge para $\mu = 18$.

Onde as bolinhas laranjas são os valores de 'x', e as azuis são a média no primeiro gráfico e desvio padrão no segundo.

4.1.11) Calculate $\bar{x}$ 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 $x_3 = 2$. (b) The sample path $x(t) = 3$ for $0 < t \le 2$, and $x(t) = 8$ for $2 < t \le 5$, over the time interval $0 < t \le 5$.

a)

$n = 3;$

$x1 = 1;$

$x2 = 6;$

$x3 = 2;$

- 2-pass-algorithm

  $\bar{x} = 1 + 6 + 1 / 3 = 3$

  $s = \sqrt{(1-3)^2+(6-3)^2+(2-3)/3)} = \sqrt{(4+9+1/3)} = \sqrt{(4{,}667)}$

- 1-pass-algorithm

  $\sqrt{s^2} = \sqrt{(((1^2+6^2+1^2)/ 3) - ((1+6+2)/3)^2)}$

  $\sqrt{(41-27)/3} = \sqrt{(14/3)}$

  $\bar{x} = \sqrt{9} = 3$

- Welford

  $\bar{x}1 = 0+1 = 1$

  $\bar{x}2 = 1+ (6-1)/2 = 3{,}5$

  $\bar{x}3 = 3{,}5 + (2-3{,}5)/3 = 3$

  $V1 = 0$

  $V2 = 0+(6-1)/2 = 5/2$

  $V3 = 2{,}5 + 2 * (2 - 3{,}5) /3 = 1{,}5$

b)

$x(t) = 3 \quad 0< t <= 2;$

$x(t) = 8 \quad 2< t <=5;$

- Two pass

  $x = 1/5 * [\ 3* (2-0) + 8 *(5 -2)] = 30/5 = 6$

  $s^2 = [$ integral de 0 a 2 $(3-6)$ dt + integral de 2 a $5(8-6)^2$ dt$] = 1/5 * (18+12) = 6$

  $s^2 = 6$

  $\sqrt{6} = 2,44$

- One pass

  $s^2 = [1/5\ (9*(2-0) + 64\ (5-2)$

  $s^2 = 1/5 * (18+192) – 6 = 210/5 – 6 = 42 -36$

  $= \sqrt{6} = 2,44$

  $X = 6$

- Welford

  $x1 = 0 + 1 * (3-0)/2 = 3$

  $x2 = 3 + 3*(8-3)/5 = 3+3 = 6$

  $v1 = 0 + 0*(3-0)^2 = 0$

  $v2 = 0 + (3*2) /5 * (8-0)^2 = 6.64/5 = \sqrt{6}$

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.

a) Com o Código generate_data_exponential.c nós geramos os dados necessários para este experimento. O output foi jogado para um arquivo “data.txt” que será usado a frente.

Executaremos o código conventional_pass.c (implementa o Conventional One pass) e o código uvs.c (Algorithm 4.1.1).

Segue print das execuções:

```
PS C:\Users\Weuler\Desktop\ufu\ms\code> Get-Content data.txt | ./uvs

for a sample of size 1000
mean ................. = 6.9478804
standard deviation ... = 6.9375442
minimum .............. = 0.0007050
maximum .............. = 55.3840780
PS C:\Users\Weuler\Desktop\ufu\ms\code> Get-Content data.txt | ./conventional

for a sample of size 1000
mean ................. = 6.9478804
standard deviation ... = 6.9375442
```

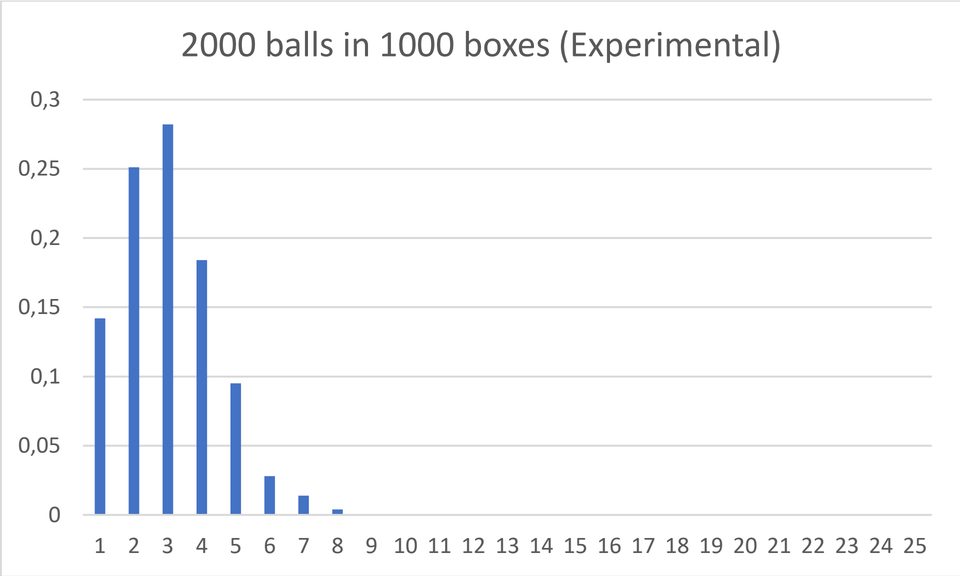
É interessante notar dessas execuções que apesar de o Algoritmo 4.1.1 calcular de forma mais robusta e com menos erros de precisão o desvio padrão, em situações onde o desvio é próximo da média erros não devem ocorrer frequentemente. Por isso, ambos algoritmos calcularam exatamente o mesmo valor de desvio padrão.

4.2.2) (a) Generate the 2000-ball histogram in Example 4.2.2. (b) Verify that the resulting relative frequencies f(x) satisfy the equation

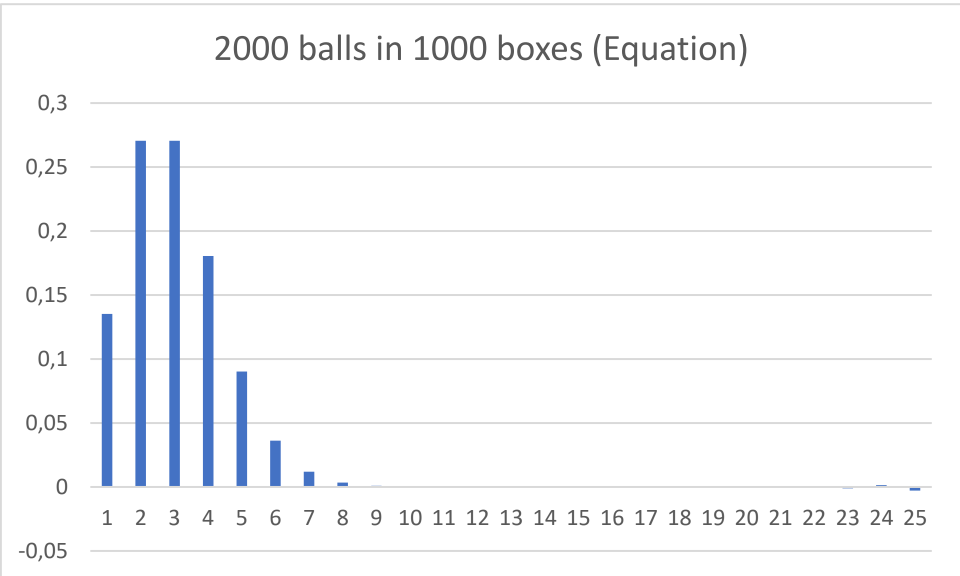
f(x) ~ (2^x * exp(−2))/x!   x = 0, 1, 2, . . .

(c) Then generate the corresponding histogram if 10 000 balls are placed, at random, in 1000 boxes. (d) Find an equation that seems to fit the resulting relative frequencies well and illustrate the quality of the fit.

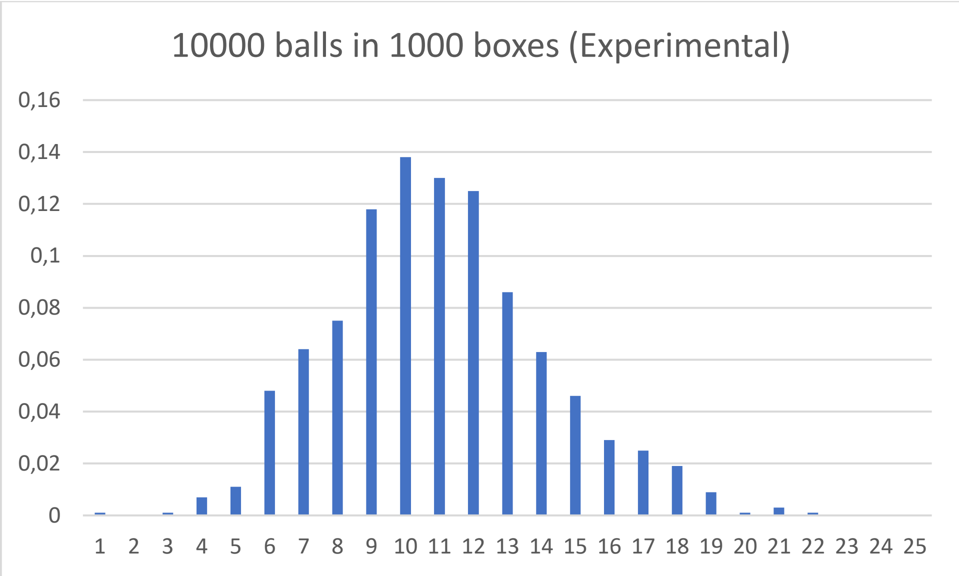
a) Com o programa **generate_ball.c** geramos usando a função Equilikely(1,n) lançamentos conforme o exemplo 4.2.2. Alem disso, nesse mesmo código, calculamos as frequências em cada caixa e com essas frequências, calculamos as frequências relativas. Usamos um valor ate 25 intencionalmente alto para servir também para o próximo item.



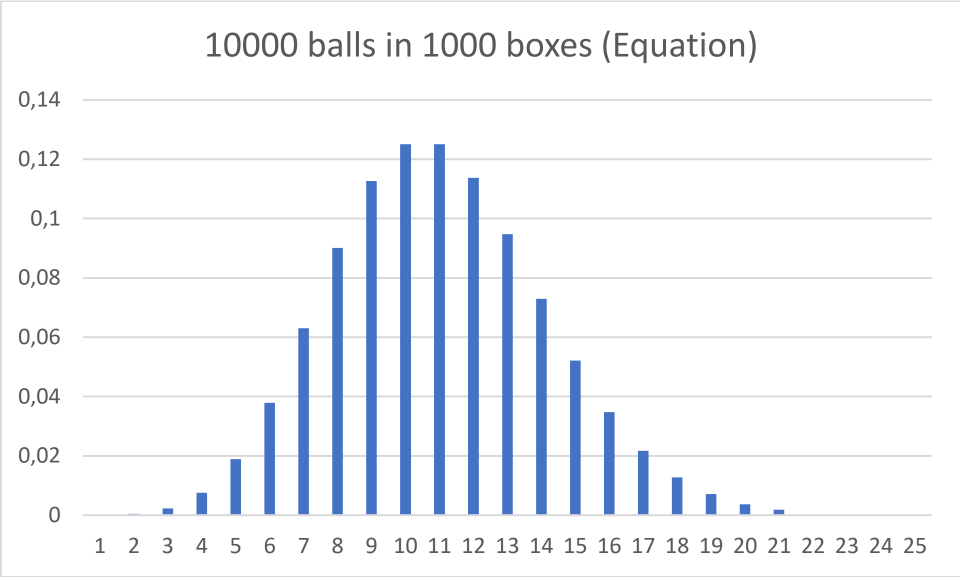
b) Ainda no código **generate_ball.c** calculamos a função f(x) e printamos os valores para calcular o seguinte histograma. Como esperado, ele tem o mesmo formato e valores próximos do resultado aleatório.



c) Histograma gerado também usando o generate_ball.c modificando apenas o numero de lançamentos feitos de 2000 para 10000.

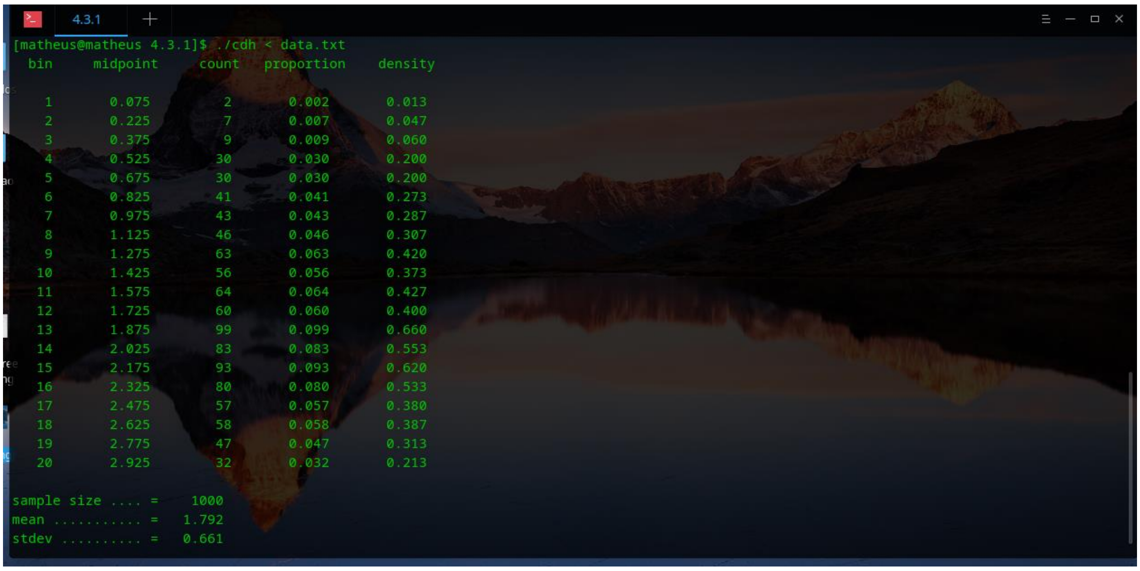


d) A função que fit esses dados é similar a f(x) do item b, com a diferença que usamos 10 como a base da exponenciação e -10 em exp. Fazemos isso pois 10 é a média esperada para este exemplo. O fit é de alta qualidade pois o formato da função experimental é bastante parecido com a esperada, o que demonstra que encontramos provavelmente a melhor função.

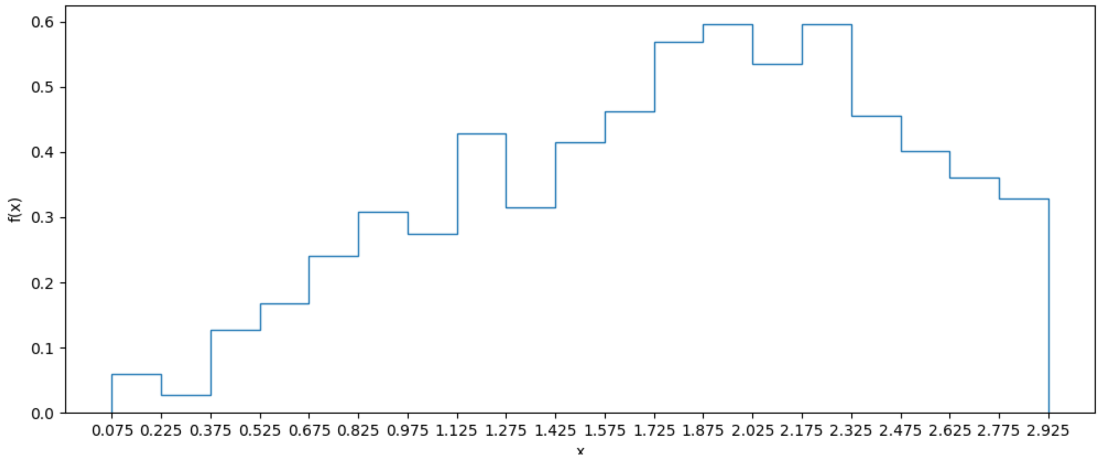


4.3.1) (a) Use program cdh to construct a continuous-data histogram like the one on the left in Example 4.3.1, but corresponding to a needle of length r = 2. (b) Based on this histogram what is the probability that the needle will cross at least one line. (c) What is the corresponding axiomatic probability that a needle of length r = 2 will cross at least one line?

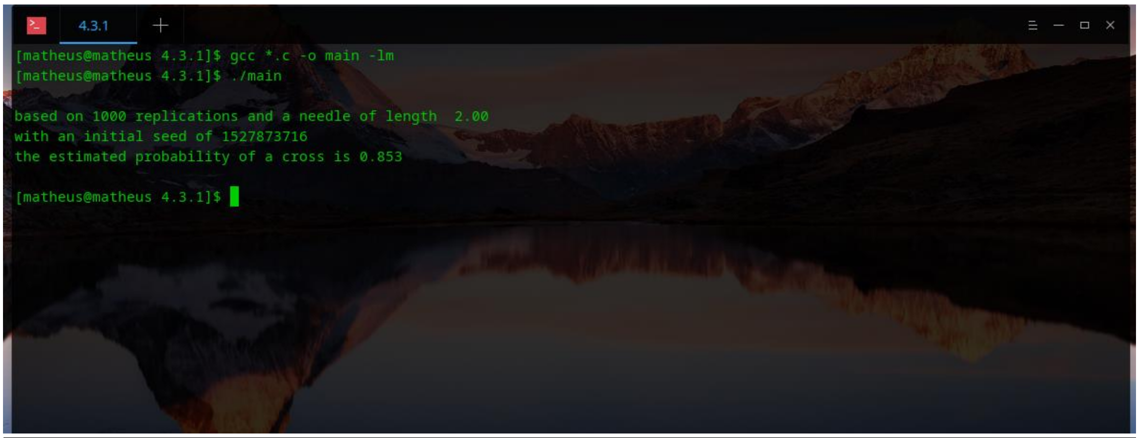
a) Código: generateRandomNumbers.c   cdh.c   generateHistogram.py



```
[matheus@matheus 4.3.1]$ ./cdh < data.txt
  bin     midpoint     count   proportion     density

    1       0.075          2      0.002        0.013
    2       0.225          7      0.007        0.047
    3       0.375          9      0.009        0.060
    4       0.525         30      0.030        0.200
    5       0.675         30      0.030        0.200
    6       0.825         41      0.041        0.273
    7       0.975         43      0.043        0.287
    8       1.125         46      0.046        0.307
    9       1.275         63      0.063        0.420
   10       1.425         56      0.056        0.373
   11       1.575         64      0.064        0.427
   12       1.725         60      0.060        0.400
   13       1.875         99      0.099        0.660
   14       2.025         83      0.083        0.553
   15       2.175         93      0.093        0.620
   16       2.325         80      0.080        0.533
   17       2.475         57      0.057        0.380
   18       2.625         58      0.058        0.387
   19       2.775         47      0.047        0.313
   20       2.925         32      0.032        0.213

sample size .... =     1000
mean ........... =    1.792
stdev .......... =    0.661
```



b) A probabilidade é de 85,3 %

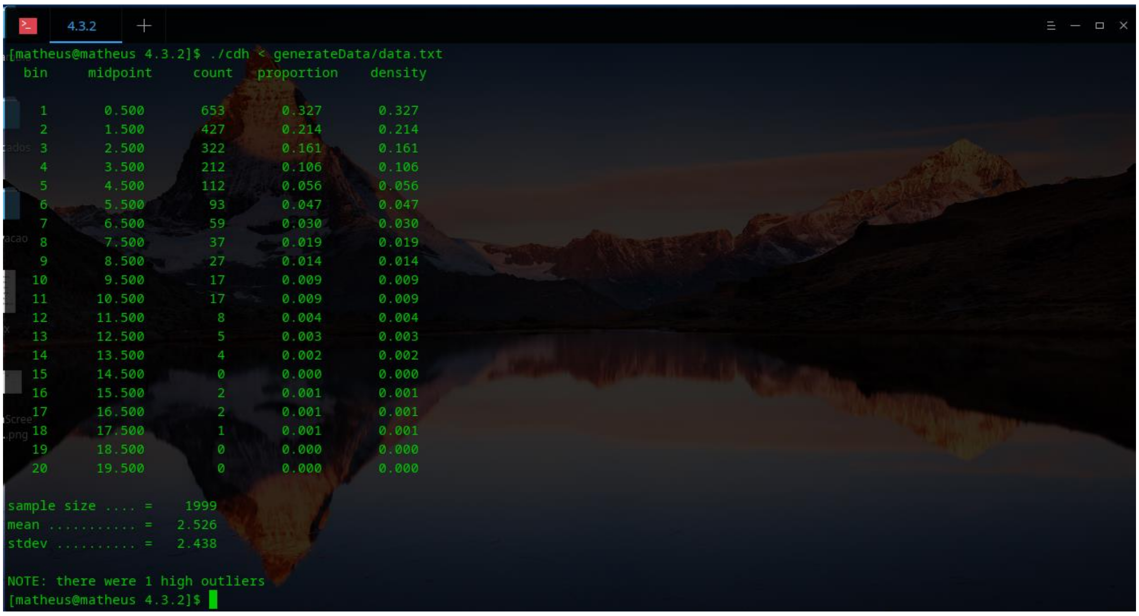


```
[matheus@matheus 4.3.1]$ gcc *.c -o main -lm
[matheus@matheus 4.3.1]$ ./main

based on 1000 replications and a needle of length  2.00
with an initial seed of 1527873716
the estimated probability of a cross is 0.853

[matheus@matheus 4.3.1]$
```

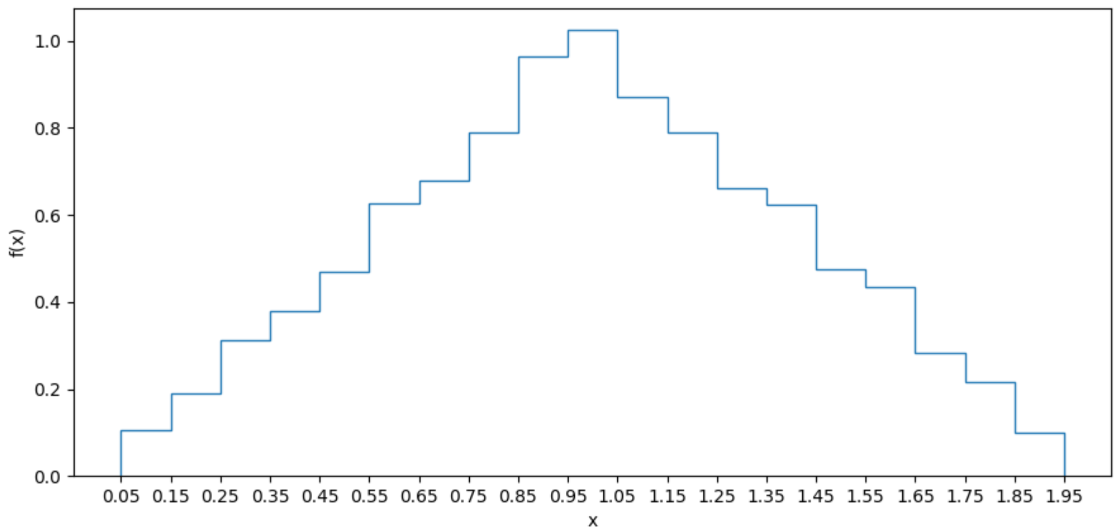
c)

4.3.2) Repeat the experiment in Example 4.3.6 with t = 5000 and n = 2000. Do not use a bubble sort.

Código: generateData.cpp cdh.c   generateHistogram.py



```
[matheus@matheus 4.3.2]$ ./cdh < generateData/data.txt
  bin     midpoint      count   proportion    density

   1       0.500        653       0.327        0.327
   2       1.500        427       0.214        0.214
   3       2.500        322       0.161        0.161
   4       3.500        212       0.106        0.106
   5       4.500        112       0.056        0.056
   6       5.500         93       0.047        0.047
   7       6.500         59       0.030        0.030
   8       7.500         37       0.019        0.019
   9       8.500         27       0.014        0.014
  10       9.500         17       0.009        0.009
  11      10.500         17       0.009        0.009
  12      11.500          8       0.004        0.004
  13      12.500          5       0.003        0.003
  14      13.500          4       0.002        0.002
  15      14.500          0       0.000        0.000
  16      15.500          2       0.001        0.001
  17      16.500          2       0.001        0.001
  18      17.500          1       0.001        0.001
  19      18.500          0       0.000        0.000
  20      19.500          0       0.000        0.000

sample size .... =     1999
mean ........... =    2.526
stdev .......... =    2.438

NOTE: there were 1 high outliers
[matheus@matheus 4.3.2]$
```

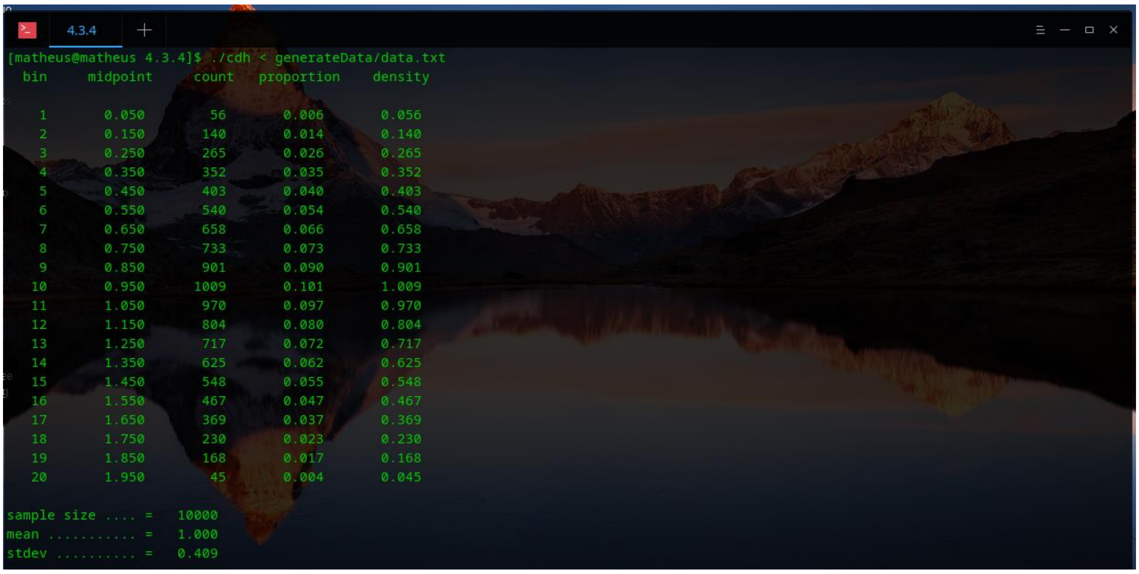


4.3.4) Generate a random variate sample x1 , x 2 , . . . , x n of size n = 10000 as follows:

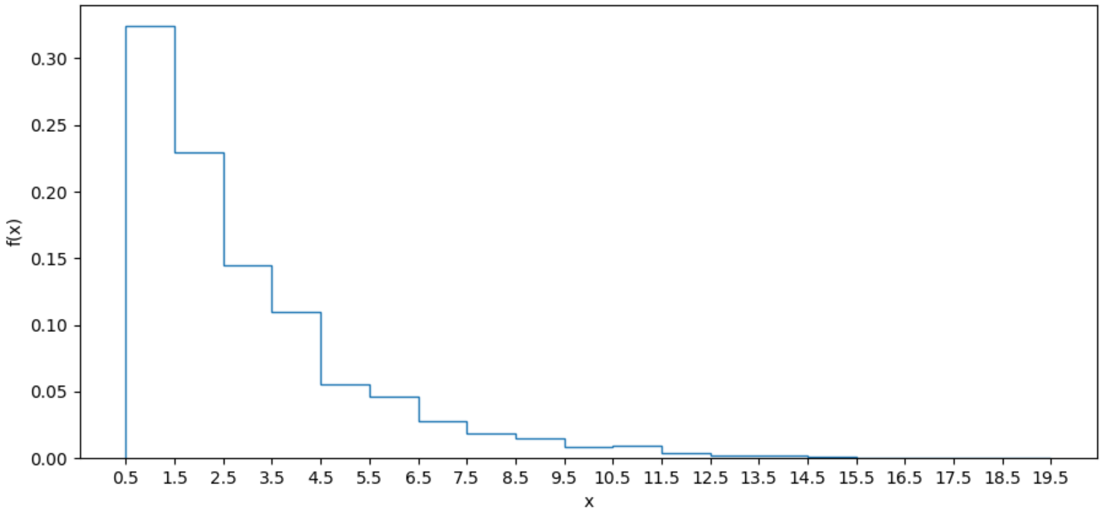
for (i = 1; i <= n; i++) X[i] = Random() + Random();

(a) Use program cdh to construct a 20-bin continuous-data histogram. (b) Can you find an equation that seems to fit the histogram density well?

a) Código: generateRandomNumbers.c cdh.c

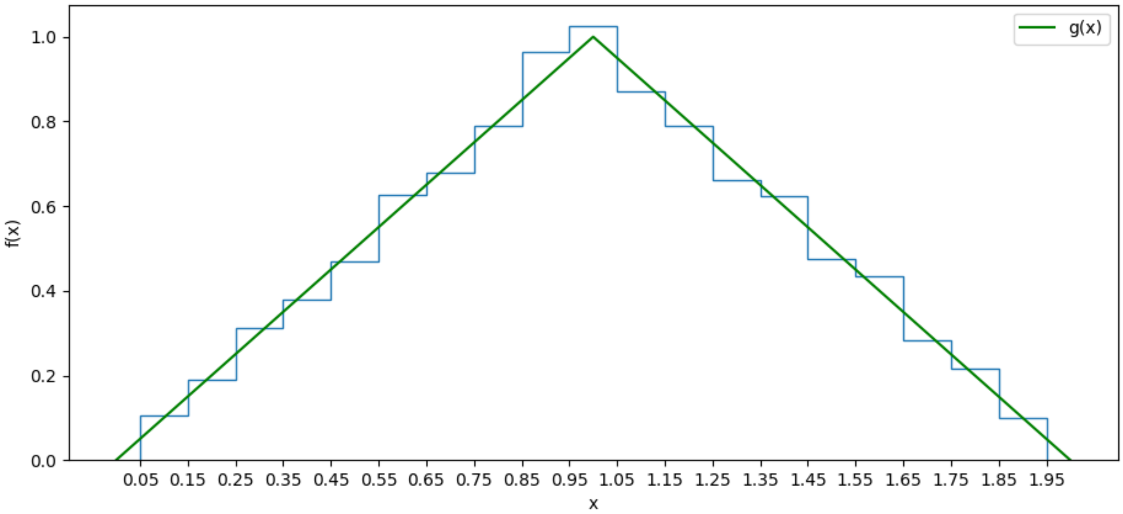


```
[matheus@matheus 4.3.4]$ ./cdh < generateData/data.txt
  bin     midpoint      count   proportion     density

    1      0.050          56      0.006        0.056
    2      0.150         140      0.014        0.140
    3      0.250         265      0.026        0.265
    4      0.350         352      0.035        0.352
    5      0.450         403      0.040        0.403
    6      0.550         540      0.054        0.540
    7      0.650         658      0.066        0.658
    8      0.750         733      0.073        0.733
    9      0.850         901      0.090        0.901
   10      0.950        1009      0.101        1.009
   11      1.050         970      0.097        0.970
   12      1.150         804      0.080        0.804
   13      1.250         717      0.072        0.717
   14      1.350         625      0.062        0.625
   15      1.450         548      0.055        0.548
   16      1.550         467      0.047        0.467
   17      1.650         369      0.037        0.369
   18      1.750         230      0.023        0.230
   19      1.850         168      0.017        0.168
   20      1.950          45      0.004        0.045

sample size .... =    10000
mean ........... =    1.000
stdev .......... =    0.409
```



b) Uma equação que se encaixa razoavelmente bem ao histograma é:

$$g(x) = \begin{cases} x, & if\ 0 \leq x\ \leq 1 \\ 1 - x, & if\ 1 < x \leq 2 \end{cases}$$



g(x)
f(x)
x
1.0
0.8
0.6
0.4
0.2
0.0
0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85 0.95 1.05 1.15 1.25 1.35 1.45 1.55 1.65 1.75 1.85 1.95