4. Simulação de Monte Carlo

Ref.: Winston, Operations Research, 4.ed., cap.21

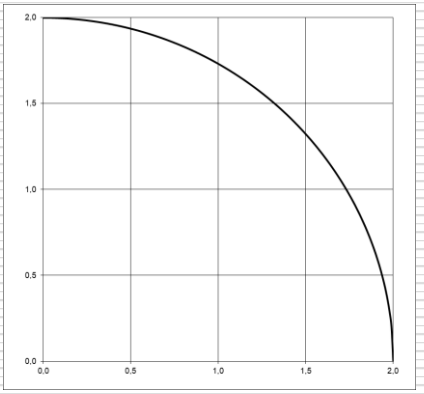
Seções: 21.3, 21.4 e 21.5

Simulação de Monte Carlo

- Os modelos de simulação podem ser classificados em **estáticos** e **dinâmicos**.
- Os modelos de simulação **estáticos** consistem na repetição de um mesmo experimento aleatório um grande número de vezes, para resolver um problema.
- Já os modelos de simulação **dinâmicos** representam a evolução dos estados de um sistema ao longo do tempo.
- Os modelos estáticos são usualmente chamados de **Simulação de Monte Carlo** e comumente são implantados em planilha, com possibilidade de uso de suplementos para expandir a capacidade de geração de números aleatórios e análise estatística dos resultados.
- https://en.wikipedia.org/wiki/Monte_Carlo_method

4

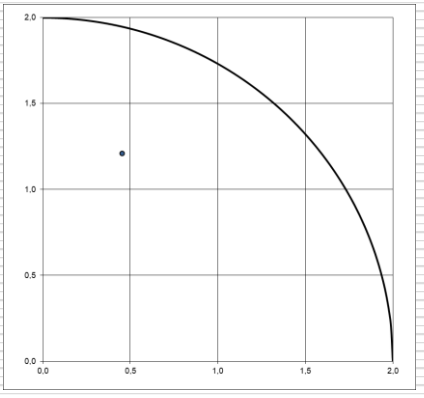
Exemplo 1: Cálculo do π



- $A = \frac{\pi r^2}{4} = \pi = 3{,}1416$
- Gerar dois números aleatórios (x,y) entre 0 e 2
- Verificar se (x,y) cai dentro ou fora do semicírculo
- Calcular a porcentagem de pontos dentro do semicírculo
- Estimar a área, aplicando esta porcentagem no valor da área do quadrado (área igual a 4)

5

Exemplo 1: Cálculo do π (n=1)

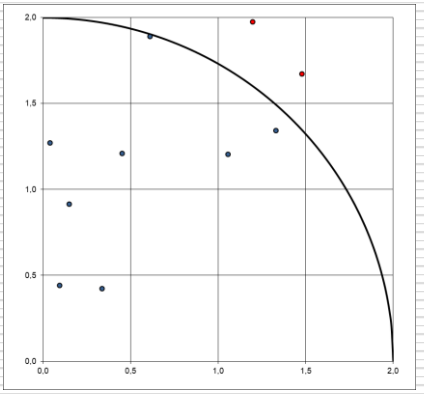


- Área: 3,1416 (π)
- Pontos gerados: 1
- Porcentagem dentro: 100%
- Estimativa da área:

  1,0*4 = 4,0

6

Exemplo 1: Cálculo do π (n=10)

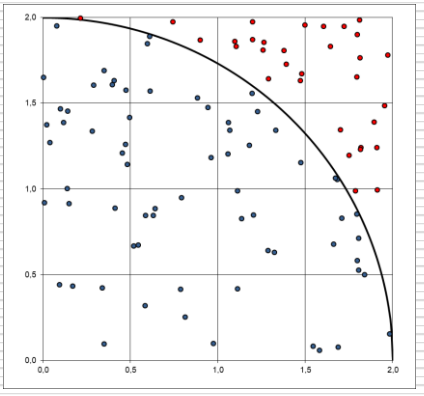


- Área: 3,1416 (π)
- Pontos gerados: 10
- Porcentagem dentro: 80,0%
- Estimativa da área:

  0,80*4 = 3,20

7

Exemplo 1: Cálculo do π (n=100)

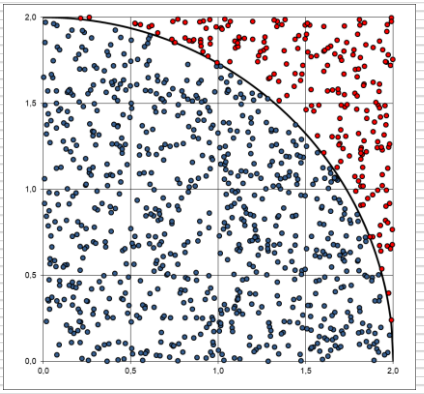


- Área: 3,1416 (π)
- Pontos gerados: 100
- Porcentagem dentro: 68,0%
- Estimativa da área:

  0,680*4 = 2,72

8

Exemplo 1: Cálculo do π (n=1000)

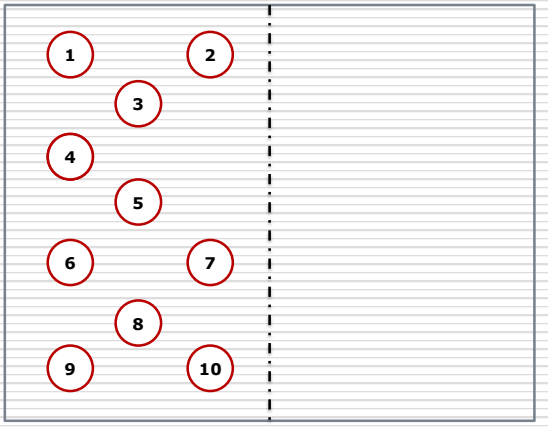


- Área: 3,1416 (π)
- Pontos gerados: 1000
- Porcentagem dentro: 78,5%
- Estimativa da área:

  0,785*4 = 3,140

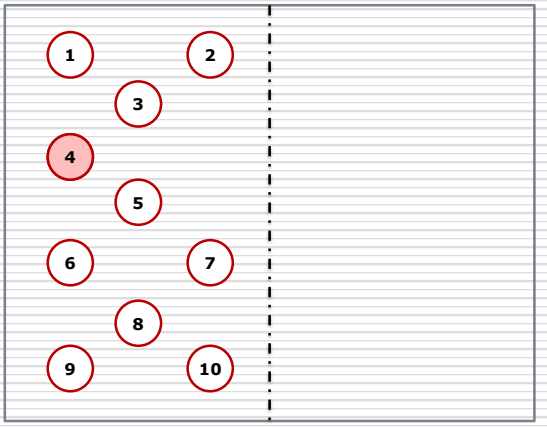
9

Exemplo 2: Difusão com m=10



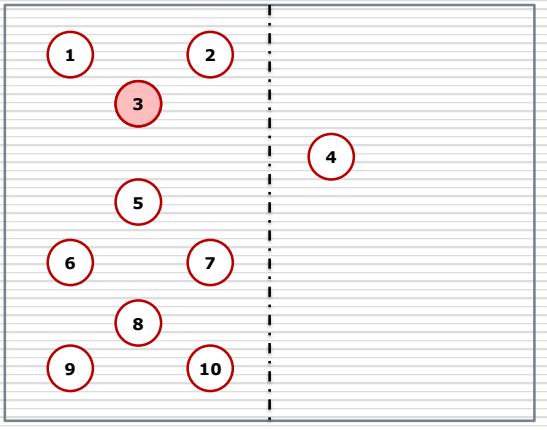
12

Iteração #1



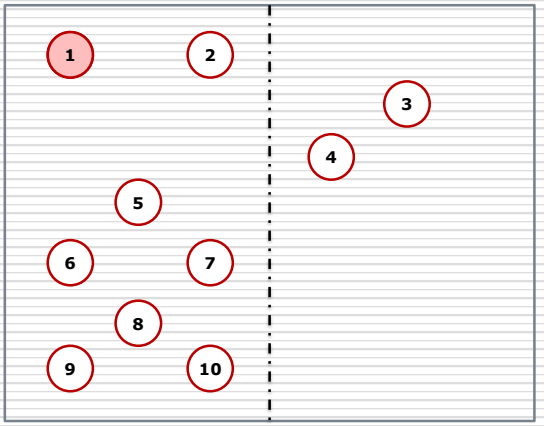
13

Iteração #2



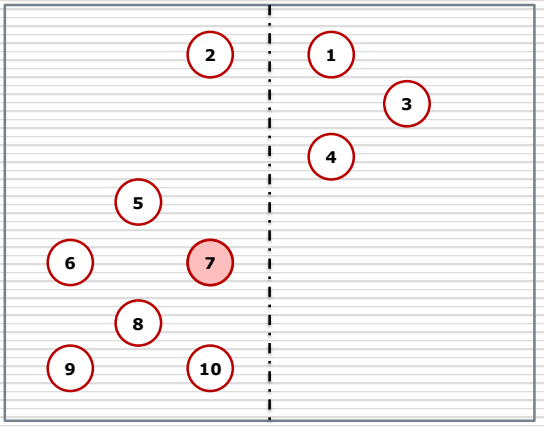
14

Iteração #3



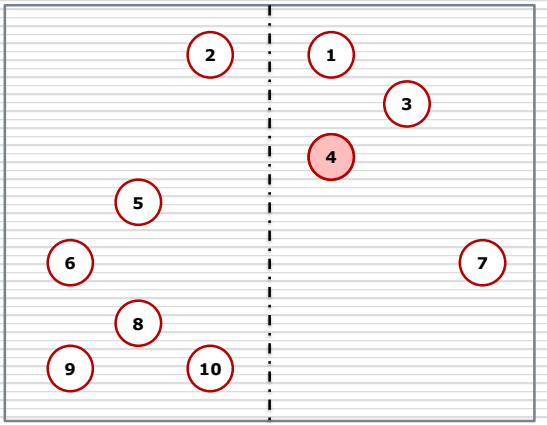
15

Iteração #4



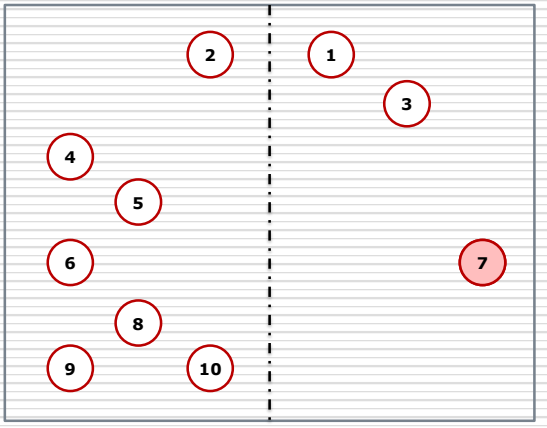
16

Iteração #5



17

Iteração #6

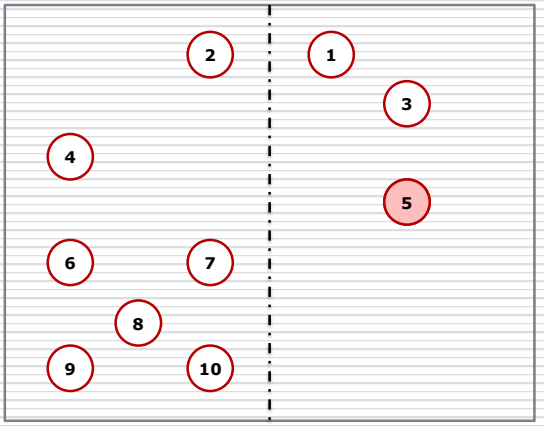


18

Iteração #7

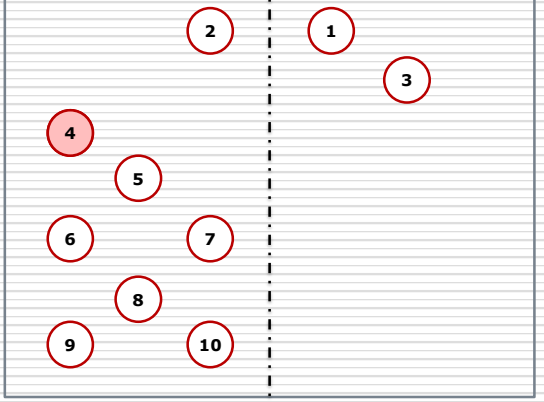
19

Iteração #8



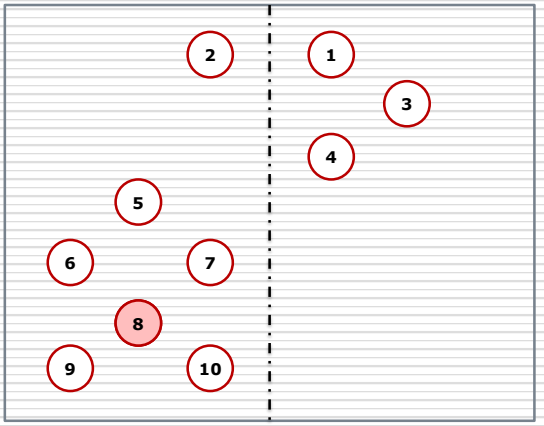
20

Iteração #9



21

Iteração #10

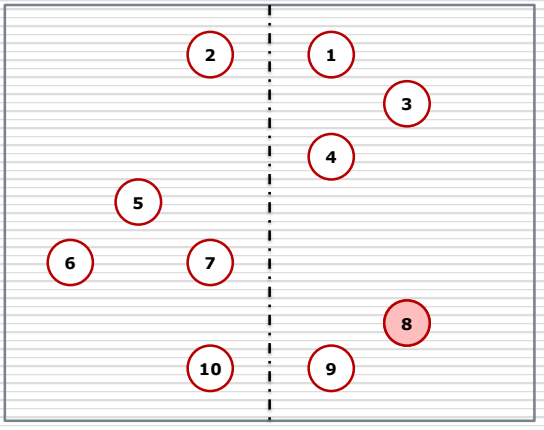


22

Iteração #11

23

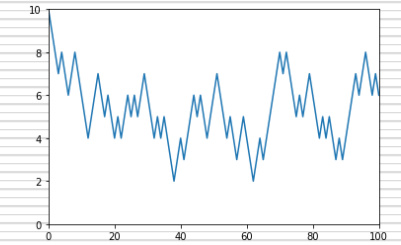
Iteração #12

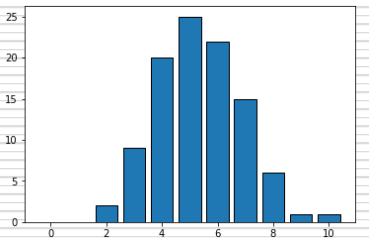


24

Simulação de Monte Carlo – Python

```
In [ ]: import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline

In [ ]: n=100
        a=list(range(1,11))      # caixa 1
        b=[]                     # caixa 2

In [ ]: y=[len(a)]     # lista com o estado da caixa 1

In [ ]: for j in range(n):
            k=np.random.randint(1,11)
            if k in a:
                a.remove(k)
                b.append(k)
            else:
                b.remove(k)
                a.append(k)
            print('# {}: {}'.format(j+1,k))
            print(a)
            print(b)
            y.append(len(a))

In [ ]: plt.plot(y)
        plt.axis([0,n,0,10])
        plt.show()

In [ ]: xs=list(range(11))
        freq=[y.count(x) for x in xs]
        plt.bar(xs,freq,edgecolor='black')
        plt.show()

In [ ]: sum(freq)
```

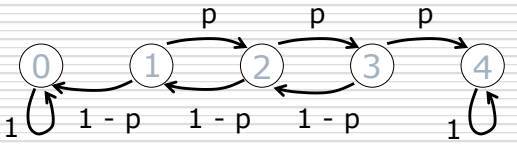




25

Exemplo 3: Problema do Apostador

Diagrama de Transição entre Estados



Para p=0,25, tem-se:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0{,}75 & 0 & 0{,}25 & 0 & 0 \\ 0 & 0{,}75 & 0 & 0{,}25 & 0 \\ 0 & 0 & 0{,}75 & 0 & 0{,}25 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P^{\infty} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0{,}975 & 0 & 0 & 0 & 0{,}025 \\ 0{,}900 & 0 & 0 & 0 & 0{,}100 \\ 0{,}675 & 0 & 0 & 0 & 0{,}325 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

26

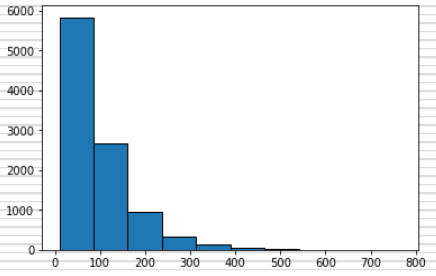
Exemplo 3: Problema do Apostador

```
In [ ]: import numpy as np

In [ ]: def joga(x,p,xmax):
            y=x     # saldo
            k=0     # número de jogadas
            while y>0 and y<xmax:
                k+=1
                r=np.random.rand()
                if r<=p:
                    y+=1    # ganha
                else:
                    y-=1    # perde
            return k,y

In [ ]: n=10000
        x=10
        p=0.48
        xmax=20
        resultados=[]
        for i in range(n):
            resultados.append(joga(x,p,xmax))

In [ ]: print(resultados)

In [ ]: a=[k for (k,y) in resultados]     # número de jogadas por jogo
        print(np.mean(a))

In [ ]: b=[y>0 for (k,y) in resultados]     # 0 ou 1, perdeu ou ganhou
        print(np.mean(b))

In [ ]: c=[y for (k,y) in resultados]     # resultado por jogo (0 ou 20)
        print(np.mean(c))

In [ ]: import matplotlib.pyplot as plt
        %matplotlib inline

In [ ]: plt.hist(a,edgecolor='black')
        plt.show()

In [ ]: p1=np.mean(b)
        p2=1-p1
        plt.pie([p1,p2],labels=['S','F'],explode=[0.1,0],autopct='%.1f%%')
        plt.show()

In [ ]:
```

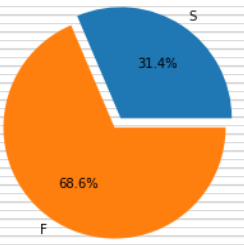
- valor inicial = $ 10
- teto = $ 20
- probabilidade de sucesso = 0.48
- número de repetições = 10000
- % de sucesso = 0.305
- número médio de jogadas = 95.45
- ganho médio = $ 6.106

27

Exemplo 3: Problema do Apostador



k – número de jogadas por jogo



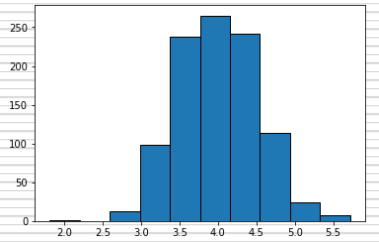
28

Exemplo 4: Soma de v.a.i.

- Seja X uma variável aleatória normal com média 1,0 e desvio-padrão 0,25. Qual a probabilidade da soma de 4 observações superar o valor de 5,0?

```
In [ ]: import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline

In [ ]: mu=1.0
        sigma=0.25
        n=1000
        m=4
        amostra=[sum(np.random.normal(mu,sigma,4)) for i in range(n)]
        plt.hist(amostra,edgecolor='black')
        plt.show()

In [ ]: sum([x>5 for x in amostra])/len(amostra)

In [ ]: print('n:', len(amostra))
        print('xbar: {:.4f}'.format(np.mean(amostra)))
        print('d.p.: {:.4f}'.format(np.std(amostra,ddof=1)))
        print('min:  {:.4f}'.format(np.min(amostra)))
        print('Q1:   {:.4f}'.format(np.percentile(amostra,0.25)))
        print('med:  {:.4f}'.format(np.percentile(amostra,0.5)))
        print('Q3:   {:.4f}'.format(np.percentile(amostra,0.75)))
        print('max:  {:.4f}'.format(np.max(amostra)))
```

- Estimativa de $P(Y_4>5,0) = 2,4\%$



n: 1000 xbar: 3.9999 d.p.: 0.5091

min: 1.8105 Q1: 2.7783 med: 2.8490 Q3: 2.8958 max: 5.7211

29

Exemplo 4: Solução Exata

- Seja X uma variável aleatória normal com média 1,0 e desvio-padrão 0,25. Qual a probabilidade da soma de 4 observações superar o valor de 5,0?

$$m = 4 \quad X \sim N(1{,}0; 0{,}25)$$

$$Y \sim N(4{,}0; \sqrt{4} \cdot 0{,}25)$$

$$P(Y > 5{,}0) = 1 - G\left(\frac{5{,}0 - 4{,}0}{0{,}5}\right) = 1 - G(2{,}0) = 2{,}28\%$$

30

21.3 Números Aleatórios

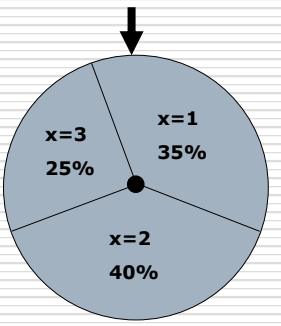
- A simulação computacional, seja estática ou dinâmica, está fundamentada na geração de números aleatórios.
- O procedimento para geração de variáveis aleatórias a partir de uma distribuição de probabilidades é denominado "**geração de números aleatórios**".
- Há métodos de geração para variáveis discretas e para variáveis contínuas.
- O método geração deve ser exato (aderente), eficiente (rápido), "sem memória" (sequência independente) e repetível (semente)

33

Roleta de Monte Carlo

- Geração de uma distribuição discreta a partir da "roleta"

x	p(x)
1	35%
2	40%
3	25%



34

- Pode-se realizar a geração de números aleatórios, utilizando-se uma roleta com 100 números, de 0 a 99.
- No exemplo anterior, gira-se a roleta uma vez. Se parar entre 0 e 34, o número gerado será 1; entre 35 e 74, será 2 e, entre 75 e 99, será 3.
- A roleta de 100 números pode ser substituída por outro processo de geração de números aleatório entre 0 e 99, por exemplo, uma **tabela de números aleatórios**.

35

Tabela de Números Aleatórios

21	96	47	59	97	56	33	24	87	36	17	18	16	90	46	75	27	28	52	13
93	41	69	96	07	97	50	81	79	59	42	37	13	81	83	82	42	85	04	31
40	24	74	36	42	40	33	04	46	24	35	63	02	31	61	34	59	43	36	96
06	06	16	25	98	17	78	80	36	85	26	41	77	63	37	71	63	94	94	33
97	81	26	03	89	39	46	67	21	17	98	10	39	33	15	61	63	00	25	92
65	99	59	97	84	90	14	79	61	55	56	16	88	87	60	32	15	99	67	43
16	91	21	32	41	60	22	66	72	17	31	85	33	69	07	68	49	20	43	29
62	03	89	26	32	35	27	99	18	25	78	12	03	09	70	50	93	19	35	56
92	27	73	40	38	37	11	05	75	16	98	81	99	37	29	92	20	32	39	67
45	51	94	69	04	00	84	14	36	37	95	66	39	01	09	21	68	40	95	79
00	81	06	28	48	12	08	05	75	26	56	16	88	87	60	13	81	20	67	58
05	06	42	24	07	60	60	29	99	93	31	85	33	69	07	25	76	01	54	03
12	68	46	55	89	60	09	71	87	89	78	12	03	09	70	83	79	68	20	66
07	79	26	69	61	67	85	72	37	41	98	81	99	37	29	61	58	87	08	05
52	16	16	23	56	62	85	80	97	63	32	25	34	03	36	48	84	60	37	65

36

Geradores de Números Aleatórios (*Random Number Generators*)

- Para desenvolver modelos de simulação computacionais eficientes, necessitam-se de mecanismos computacionais de geração de números aleatórios.
- Um elemento básico deste processo são as funções *random*, que geram números aleatórios entre 0 e 1.
- Exemplo de função *random*:

$$x_i = resto\left(\frac{a \cdot x_{i-1} + c}{m}\right) \qquad u_i = \frac{x_i}{m}$$

- O método acima produz uma sequência de números pseudoaleatórios entre 0 e 1.

37

Exemplo

x_0 =	31
a =	13
c =	65
m =	100

$$x_i = \text{resto}\left(\frac{13 \cdot x_{i-1} + 65}{100}\right)$$

$$u_i = \frac{x_i}{100}$$

0	31	
1	68	0,68
2	49	0,49
3	2	0,02
4	91	0,91
5	48	0,48
6	89	0,89
7	22	0,22
8	51	0,51
9	28	0,28
10	29	0,29

38

Exemplo (cont.)

10	29	0,29
11	42	0,42
12	11	0,11
13	8	0,08
14	69	0,69
15	62	0,62
16	71	0,71
17	88	0,88
18	9	0,09
19	82	0,82
20	**31**	**0,31**

- Parâmetros anteriores resultaram em um ciclo (n=20)
- Ajustar valores dos parâmetros, por exemplo:
  - x_0 = 327.680
  - a = 1.140.671.485
  - c = 12.820.163
  - m = 2^{24}
- Referências:
  - McCullough (2008)
  - Wikipedia: "random number generator" e "linear congruential generator"

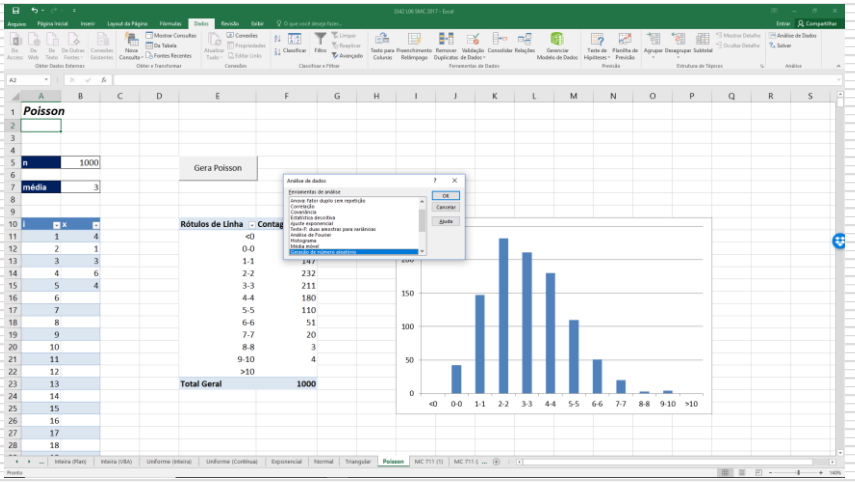
39

Geradores de Números Aletórios

- A partir da geração de números aleatórios *U*(0,1), o passo seguinte é obter amostras aleatórias de distribuições mais gerais, discretas ou contínuas.
- Propriedades desejáveis de um GNA:
  1. Eficiência computacional (velocidade),
  2. "Exatidão" (aderência e independência),
  3. Devem permitir a replicação, e
  4. Devem apresentar um ciclo longo antes da repetição.
- A grande maioria das linguagens de programação apresentam funções de geração de números (pseudo) aleatórios.
- Exemplos: Excel/VBA, Python, C++ etc.

40

Excel / VBA (suplementos)



41

Python / Numpy / SciPy

42

Geração de Variáveis Aleatórias Discretas

- Um procedimento geral de geração de v.a. discretas consiste de dois passos:
  1. Determina-se a distribuição acumulada da variável aleatória, e
  2. Utiliza-se a distribuição acumulada para obtenção dos valores da amostra conforme a seguir.

$$x_i = \begin{cases} x_1 & se\ \ u_i \le F(x_1) \\ x_2 & se\ \ F(x_1) < u_i \le F(x_2) \\ \vdots & \vdots \\ x_n & se\ \ u_i > F(x_{n-1}) \end{cases}$$

43

Exemplo 4

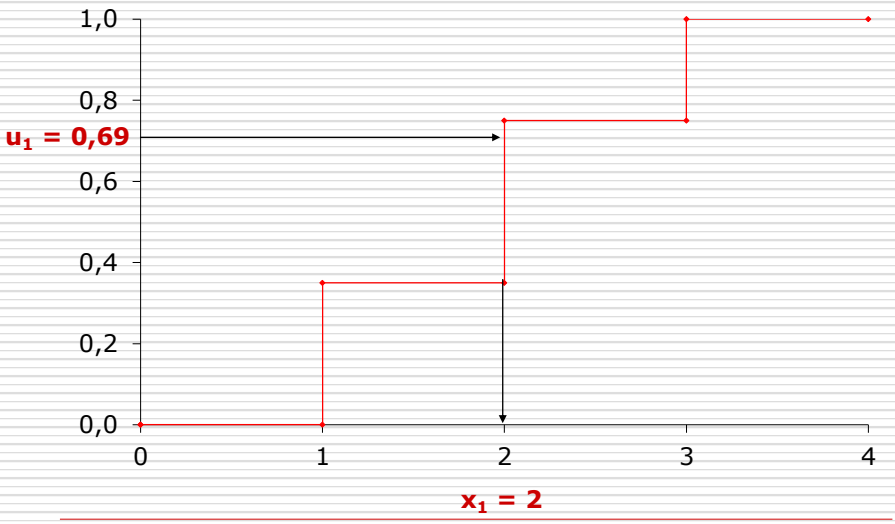
- Obtenha, pelo método de Monte Carlo, uma amostra com cinco valores da seguinte distribuição de probabilidades discreta:

x	p(x)	F(x)
1	0,35	0,35
2	0,40	0,75
3	0,25	1,0

i	u_i	x_i
1	0,69	2
2	0,79	3
3	0,30	1
4	0,32	1
5	0,66	2

44

Representação Gráfica do método de Geração de V.A. Discretas



45

Planilha MS Excel

- Busca em tabelas (função LOOKUP ou PROC)

=PROC(H11;$D$11:$D$13;$B$11:$B$13)

x	p(x)	inf	sup
1	35%	0%	35%
2	40%	35%	75%
3	25%	75%	100%
Total	100%		

i	u_i	x_i
1	0,69	2
2	0,79	3
3	0,30	1
4	0,32	1
5	0,66	2

- Recursos para geração de números aleatórios
  - Funções INV e Linguagem VBA (funções e procedimentos)
- Softwares: Crystal Ball, @Risk etc

46

Exemplo 5:
La Boulangerie de Pierre (p.1159)

- Problema: quantos pães produzir diariamente (Q)?
  - Q = [36, 48, 60, 72, 84, 96]
  - Demanda aleatória
  - Venda = min(Q,D)
  - Custo Variável = Q·c
  - Receita da Venda = V·p
  - Receita marginal com a quantidade não vendida
- Método de Solução
  - Simular vários dias, calculando o resultado por dia e, ao final, a média para o total de dias simulados
  - Variar o Q e verificar o valor Q^* que maximiza o resultado

47

La Boulangerie de Pierre (Demanda)

Cenário	Prob.
Alta	30%
Média	45%
Baixa	25%
	100%

Demanda	Prob (Alta)	Prob (Média)	Prob (Baixa)
36	5%	10%	15%
48	10%	20%	25%
60	25%	30%	35%
72	30%	25%	15%
84	20%	10%	5%
96	10%	5%	5%
	100%	**100%**	**100%**

48

La Boulangerie de Pierre (n=15 dias)

Qtd Produzida	60

Preço 1	R$ 0,40
Preço 2	R$ 0,10
Custo de Produção	R$ 0,25
Custo de Falta	R$ 0,15

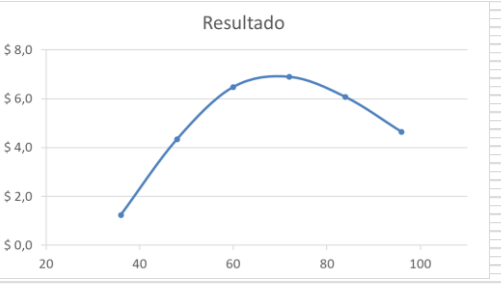
Resultado Médio
R$ 6,48

Dia	rnd1	Cenário	rnd2	Demanda	Venda	Sobra	Falta	Receita	Custo	Custo Falta	Receita Marg	Resultado
1	69%	Média	56%	60	60	0	0	R$ 24,00	R$ 15,00	R$ -	R$ -	R$ 9,00
2	30%	Média	32%	60	60	0	0	R$ 24,00	R$ 15,00	R$ -	R$ -	R$ 9,00
3	66%	Média	79%	72	60	0	12	R$ 24,00	R$ 15,00	R$ 1,80	R$ -	R$ 7,20
4	55%	Média	24%	48	48	12	0	R$ 19,20	R$ 15,00	R$ -	R$ 1,20	R$ 5,40
5	80%	Baixa	35%	48	48	12	0	R$ 19,20	R$ 15,00	R$ -	R$ 1,20	R$ 5,40
6	10%	Alta	98%	96	60	0	36	R$ 24,00	R$ 15,00	R$ 5,40	R$ -	R$ 3,60
7	92%	Baixa	88%	72	60	0	12	R$ 24,00	R$ 15,00	R$ 1,80	R$ -	R$ 7,20
8	82%	Baixa	17%	48	48	12	0	R$ 19,20	R$ 15,00	R$ -	R$ 1,20	R$ 5,40
9	4%	Alta	86%	84	60	0	24	R$ 24,00	R$ 15,00	R$ 3,60	R$ -	R$ 5,40
10	31%	Média	13%	48	48	12	0	R$ 19,20	R$ 15,00	R$ -	R$ 1,20	R$ 5,40
11	23%	Alta	44%	72	60	0	12	R$ 24,00	R$ 15,00	R$ 1,80	R$ -	R$ 7,20
12	93%	Baixa	13%	36	36	24	0	R$ 14,40	R$ 15,00	R$ -	R$ 2,40	R$ 1,80
13	42%	Média	51%	60	60	0	0	R$ 24,00	R$ 15,00	R$ -	R$ -	R$ 9,00
14	16%	Alta	17%	60	60	0	0	R$ 24,00	R$ 15,00	R$ -	R$ -	R$ 9,00
15	29%	Alta	62%	72	60	0	12	R$ 24,00	R$ 15,00	R$ 1,80	R$ -	R$ 7,20

49

La Boulangerie de Pierre (solução)

#	Resultado
	R$ 6,418
36	R$ 1,235
48	R$ 4,343
60	R$ 6,481
72	R$ 6,901
84	R$ 6,074
96	R$ 4,639



50

21.5 Geração de Variáveis Aleatórias Contínuas

- Os dois métodos mais comuns são o método da função inversa e o método de aceitação e rejeição.
- O princípio básico do **método da função inversa** é semelhante ao caso das v.a. discretas.
- O **método da aceitação e rejeição** é utilizado quando a distribuição acumulada não apresenta forma analítica. A geração é feita com base na própria função densidade de probabilidade.
- Além dos métodos gerais, há outros específicos para diferentes distribuições de probabilidade.

51

Método da Função Inversa

- Este método pode ser utilizado quando a distribuição acumulada apresenta função inversa simples.
- Exemplos: uniforme, exponencial, triangular e Weibull.
- O método é relativamente simples, e consiste dos seguintes passos:
  - **Passo 1**: Determinar a distribuição acumulada e sua função inversa.

$$F(x) = \int_{-\infty}^{x} f(y)dy \qquad F^{-1}(y)$$

  - **Passo 2**: Gerar um número aleatório u_i.
  - **Passo 3**: Gerar x_i usando a função inversa ($x_i = F^{-1}(u_i)$)

52

Exemplo

- Seja X uma v.a. com f.d.p. dada por:

$$f(x) = \frac{x}{2} \qquad 0 \le x \le 2$$

- Passo 1: determina-se a função distribuição acumulada:

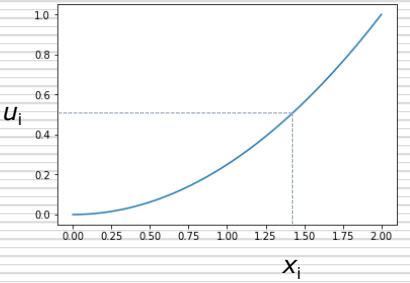
$$F(x) = \int_0^x \frac{t}{2} dt = \frac{x^2}{4} \qquad 0 \le x \le 2$$

- Cuja função inversa é dada por:

$$x = F^{-1}(u) \quad \Rightarrow \quad x = \pm 2\sqrt{u}$$

53

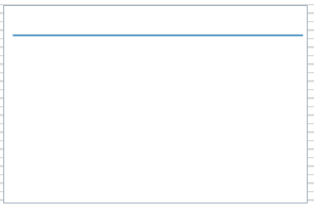
- Passo 2: gera-se uma número aleatório u;
- Passo 3: calcula-se: $x = 2\sqrt{u}$
- Representação Gráfica do método:



- Este exemplo mostra a simplicidade e facilidade de aplicação do método.

54

Distribuição Uniforme

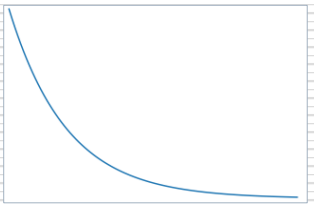


$$f(x) = \frac{1}{b-a} \quad a \le x \le b$$

$$X = a + (b-a) \cdot U$$

57

Distribuição Exponencial

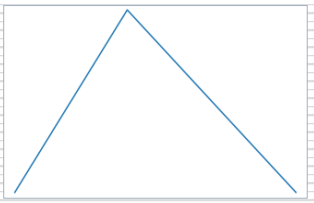


$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \ge 0, \ \lambda > 0 \\ 0 & x < 0 \end{cases}$$

$$X = -\frac{1}{\lambda} \cdot \ln(1-U) \quad \Rightarrow \quad -\frac{1}{\lambda} \cdot \ln(U)$$

58

Distribuição Triangular

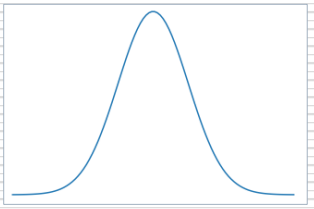


$$f(x) = \begin{cases} \dfrac{2(x-a)}{(b-a)(c-a)} & a < x < b \\ \dfrac{2(c-x)}{(c-b)(c-a)} & b < x < c \end{cases}$$

$$X = \begin{cases} a + \sqrt{U(b-a)(c-a)} & 0 < U < \dfrac{(b-a)}{(c-a)} \\ c - \sqrt{(1-U)(c-b)(c-a)} & \dfrac{(b-a)}{(c-a)} < U < 1 \end{cases}$$

59

Distribuição Normal



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

$$X = \mu + Z \cdot \sigma \qquad Z = \Phi^{-1}(U)$$

60

Método Aceita/Rejeita (MAR)

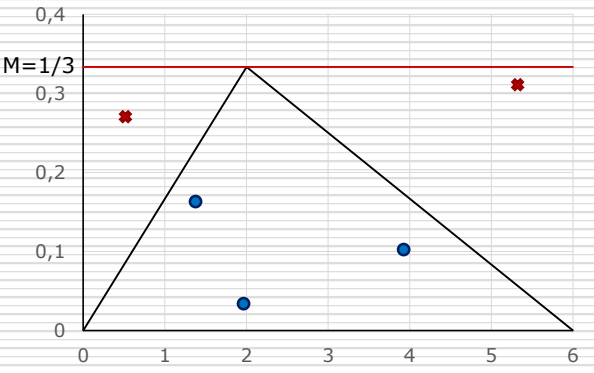
- Há muitas distribuições importantes, incluindo a Erlang (teoria de filas) e a Beta (PERT), cuja distribuição acumulada não tem uma forma analítica
- Para estas distribuições, deve-se buscar métodos alternativa para GNA. Um destes métodos é o da aceitação/rejeição (MAR)
- Este método aplica-se a distribuições definidas em um intervalo finito [*a, b*]

61

- Seja f(x) a fdp de uma v.a. no intervalo $a \leq x \leq b$. O algoritmo consiste nos seguintes passos:
  - **Passo 1:** defina uma constante *M* igual ao maior valor de *f(x)* sobre no intervalo [*a, b*].
  - **Passo 2:** gere dois números aleatórios, u_1 e u_2
  - **Passo 3:** calcule $x = a + (b - a)\cdot u_1$ e $y = M\cdot u_2$.
  - **Passo 4:** calcule *f(x)*
  - **Passo 5:** se $y<f(x)$, aceite x, senão, rejeite
  - **Passo 6:** volte ao Passo 2 até completar a amostra
- Note que o algoritmo pode gastar mais de uma iteração para a geração de um número aleatório. Por esta razão, o algoritmo pode ser relativamente ineficiente.
- Esta ineficiência está altamente relacionada com a forma da distribuição f(x) em questão.

62

Exemplo – Triangular



- (x_i , y_i) => $X_i \sim Unif(0,6)$ e $Y_i \sim Unif(0,1/3)$
- Se $y_i < f(x_i)$, Aceita, senão, Rejeita.

63

Métodos específicos

- Os métodos apresentados até aqui são genéricos, aplicam-se a diferentes distribuições
- Há métodos que são específicos para um dado tipo de distribuição que, em geral, são mais eficientes e exatos
- Por exemplo, apresentamos a seguir alguns métodos para as distribuições Normal e Poisson

64

GNA – Normal

- Geram-se dois valores aleatórios de $U(0,1)$ e, a seguir, duas v.a. normais padrão independentes a partir da seguinte transformação:

$$Z_1 = (-2\ln U_1)^{\frac{1}{2}} \sin 2\pi U_2$$

$$Z_2 = (-2\ln U_1)^{\frac{1}{2}} \cos 2\pi U_2$$

- Tendo as variáveis z, obtém-se duas observações aleatórias de uma variável X~N(μ,σ) pelas expressões:

$$X_1 = \mu + \sigma Z_1 \qquad X_2 = \mu + \sigma Z_2$$

65

Método alternativo

- Gerar uma amostra aleatória $\mathbf{u_1, u_2, \ldots, u_n}$ de uma v.a. $U(0,1)$ com $\mu=0,5$ e $\sigma^2=1/12$.
- Pelo TCL, a variável Z abaixo terá, com n grande, distribuição tendendo à normal.

$$Z = \frac{\sum_{i=1}^{n} U_i - 0{,}5n}{\left(\frac{n}{12}\right)^{1/2}}$$

- Substituindo-se n=12, tem-se: $Z = \sum_{i=1}^{12} U_i - 6$

66

21.X Ajuste de Distribuições de Probabilidades

- Problema 1:
  - Dada uma amostra de números, estimar os parâmetros de uma distribuição que se deseja ajustar aos dados.
  - Testes de Aderência para avaliar a adequação do ajuste
- Problema 2:
  - Dada uma amostra de números, determinar qual é a distribuição teórica que melhor se ajusta aos dados, estimando seus parâmetros.
  - Testar a aderência de diferentes distribuições e selecionar a melhor disponível, diferenciando distribuições discretas e contínuas
  - Stat::Fit

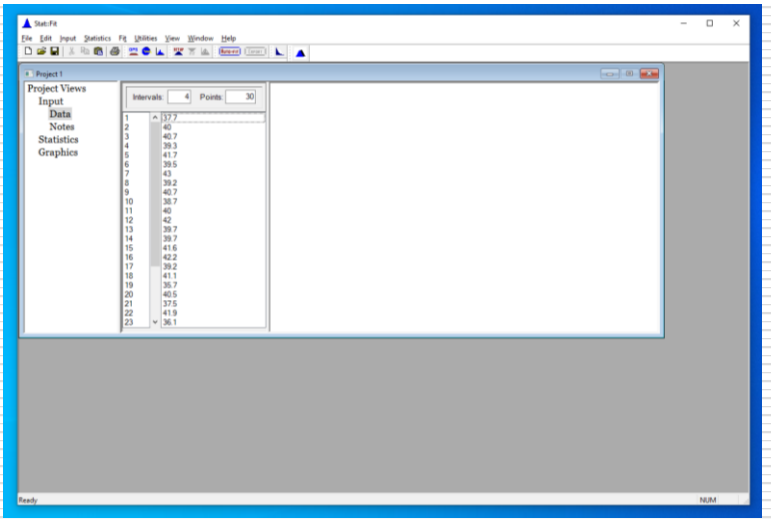
69

Exemplo 1

37.7	43.0	39.7	35.7	37.6
40.0	39.2	39.7	40.5	40.4
40.7	40.7	41.6	37.5	38.9
39.3	38.7	42.2	41.9	45.5
41.7	40.0	39.2	36.1	40.2
39.5	42.0	41.1	40.0	37.9

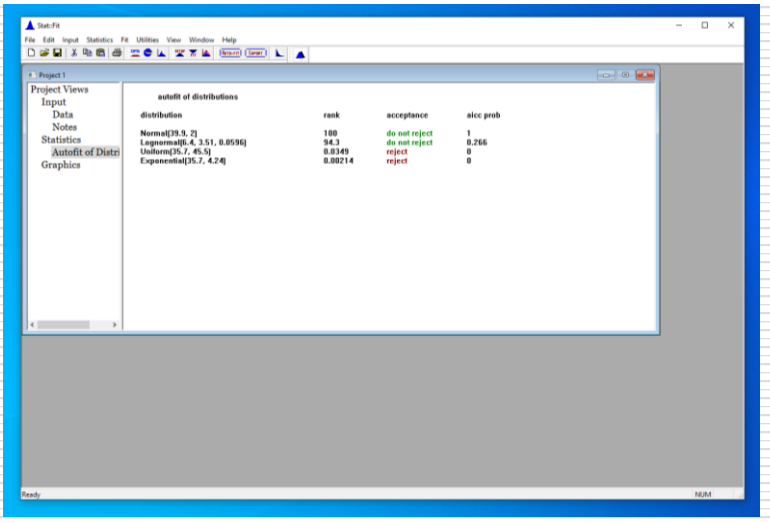
70

Dados 1



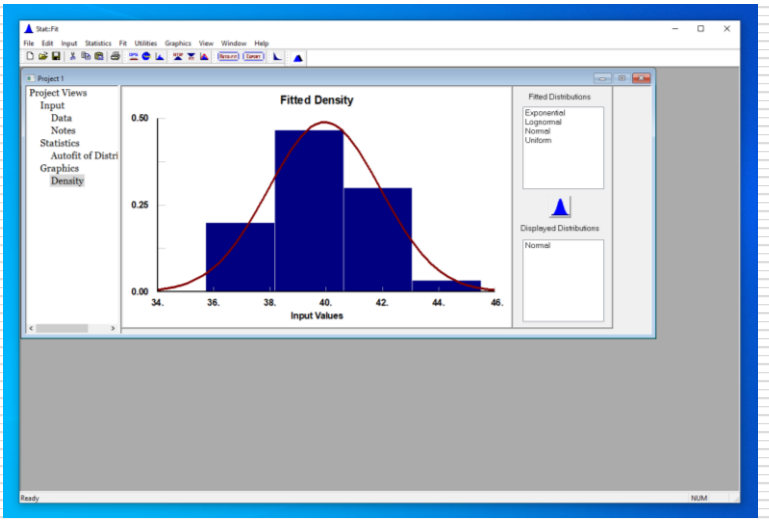
71

Autofit (versão demo, apenas 4 distribuições)



72

Distribuição ajustada - Normal



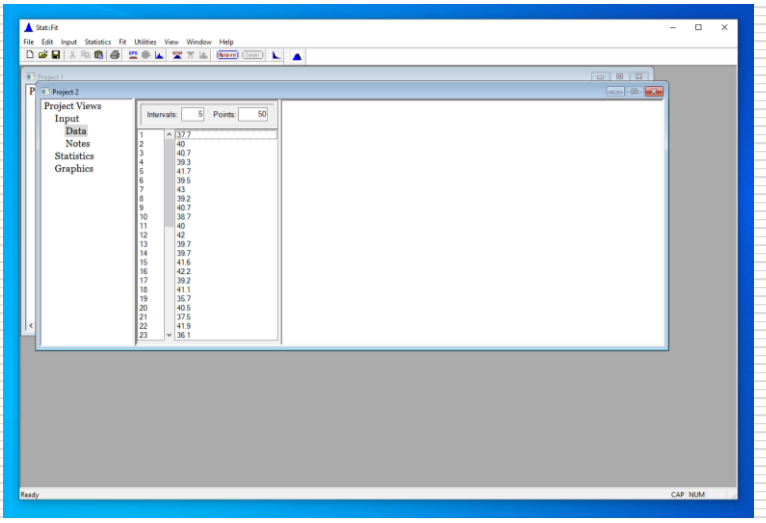
73

Exemplo 2

37.7	40.0	37.5	53.0	49.2
40.0	42.0	41.9	44.7	49.6
40.7	39.7	36.1	42.4	48.3
39.3	39.7	40.0	51.6	49.3
41.7	41.6	37.6	40.1	50.5
39.5	42.2	40.4	50.8	50.9
43.0	39.2	38.9	49.1	39.8
39.2	41.1	45.5	55.9	53.9
40.7	35.7	40.2	47.8	48.8
38.7	40.5	37.9	53.7	44.1

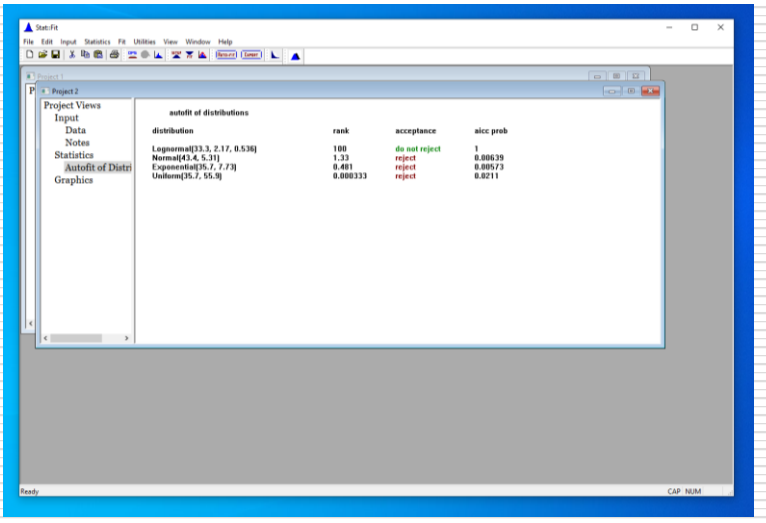
74

Dados 2



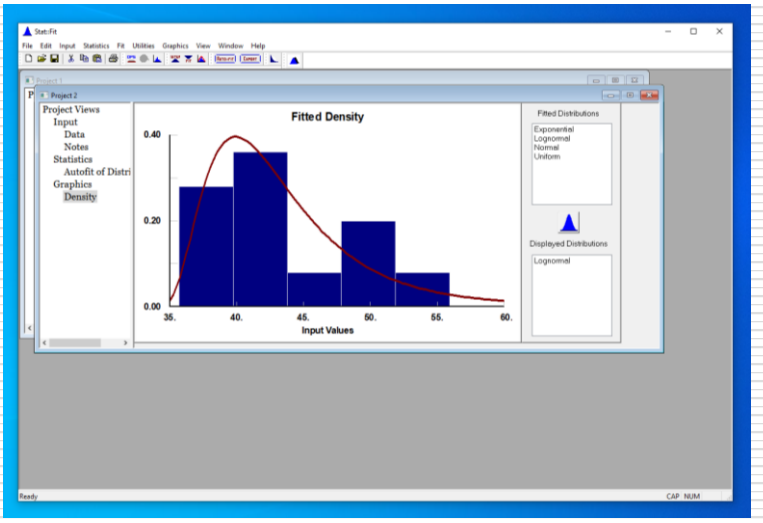
75

Autofit



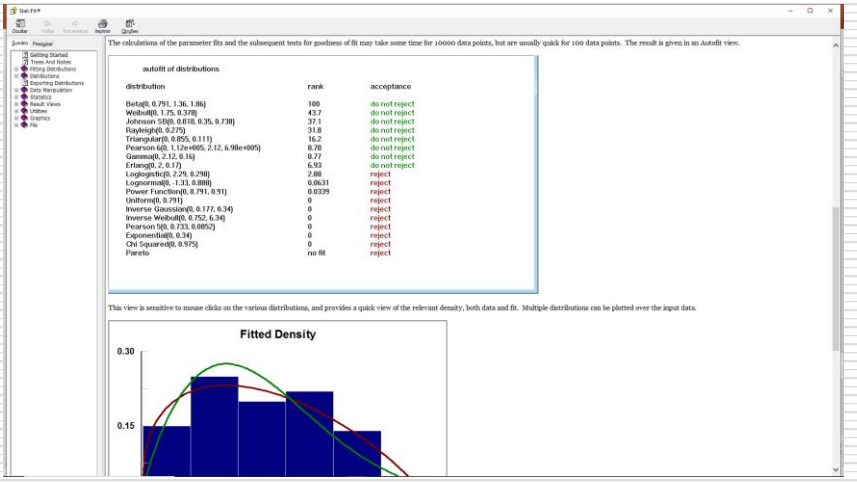
76

Distribuição ajustada - Lognormal



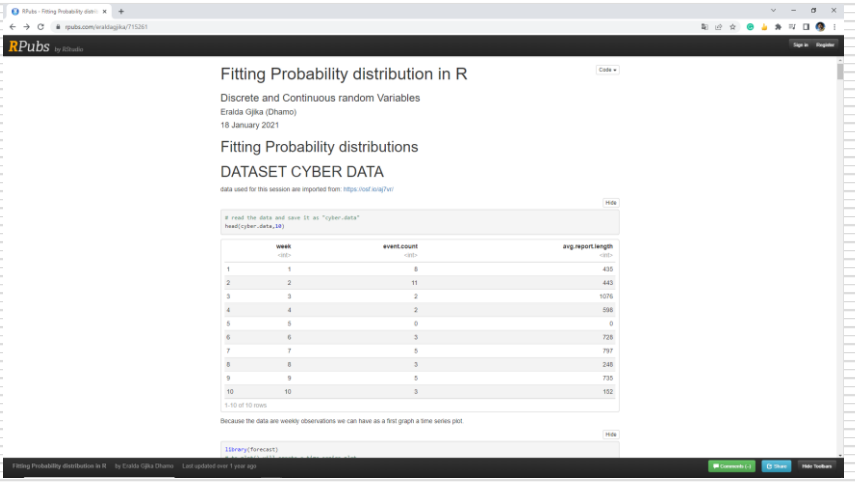
77

Stat::Fit – Help



78

Fitting Probability distribution in R (1)



Fitting Probability distribution in R

Discrete and Continuous random Variables

Eralda Gjika (Dhamo)

18 January 2021

Fitting Probability distributions

DATASET CYBER DATA

data used for this session are imported from: https://osf.io/aj7vr

```
# read the data and save it as "cyber.data"
head(cyber.data,10)
```

	week	event.count	avg.report.length
1	1	8	435
2	2	11	443
3	3	2	1076
4	4	2	598
5	5	0	0
6	6	3	728
7	7	5	797
8	8	3	248
9	9	5	735
10	10	3	152

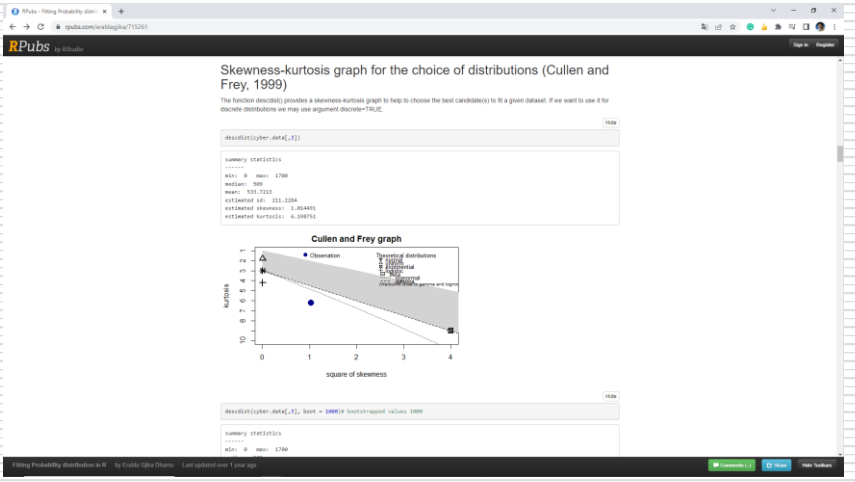
1-10 of 10 rows

Because the data are weekly observations we can have as a first graph a time series plot.

```
library(forecast)
```

79

Fitting Probability distribution in R (2)



Skewness-kurtosis graph for the choice of distributions (Cullen and Frey, 1999)

The function descdist() provides a skewness-kurtosis graph to help to choose the best candidate(s) to fit a given dataset. If we want to use it for discrete distributions we may use argument discrete=TRUE.

```
descdist(cyber.data[,3])
```

```
summary statistics
------
min:  0   max:  1700
median:  509
mean:  533.7213
estimated sd:  311.3284
estimated skewness:  1.014491
estimated kurtosis:  6.194751
```

Cullen and Frey graph


```
descdist(cyber.data[,3], boot = 1000)# bootstrapped values 1000
```

```
summary statistics
------
min:  0   max:  1700
```

80

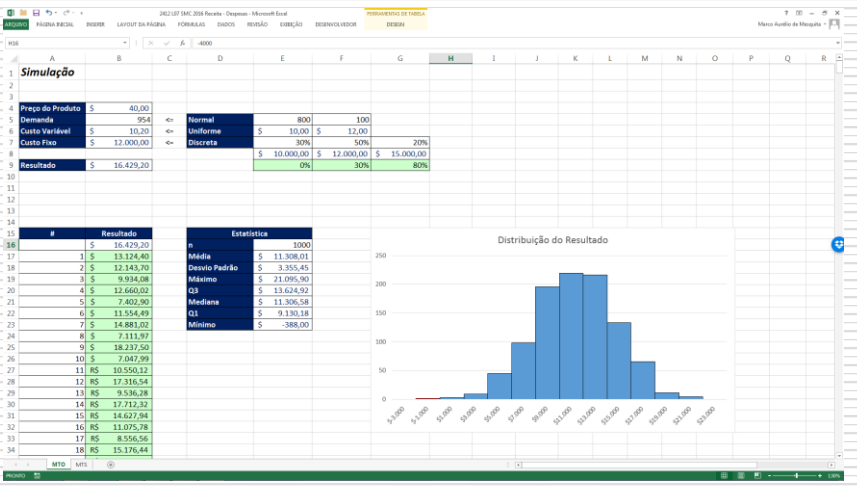
21.4 Exemplos de Simulação de Monte Carlo

- A simulação de Monte Carlo consiste na repetição de um mesmo experimento aleatório um grande número de vezes. Ao contrário da simulação dinâmica, as repetições são independentes umas das outras.

- Ex. 1: **Vendas**. Produto com custos e demanda variável, determinar a expectativa do resultado.
- Ex. 2: **Problema do jornaleiro**. Demanda variável, estoque inicial fixo; determinar venda média, sobra e resultado.
- Ex. 3: Jogo de dados **"7 ou 11"**; determinar as chances de vitória e derrota.
- Ex. 4: **Investimento** com fluxos variáveis; avaliar retorno do investimento.
- Ex. 5: **Risco** de atraso em um projeto (PERT/CPM).

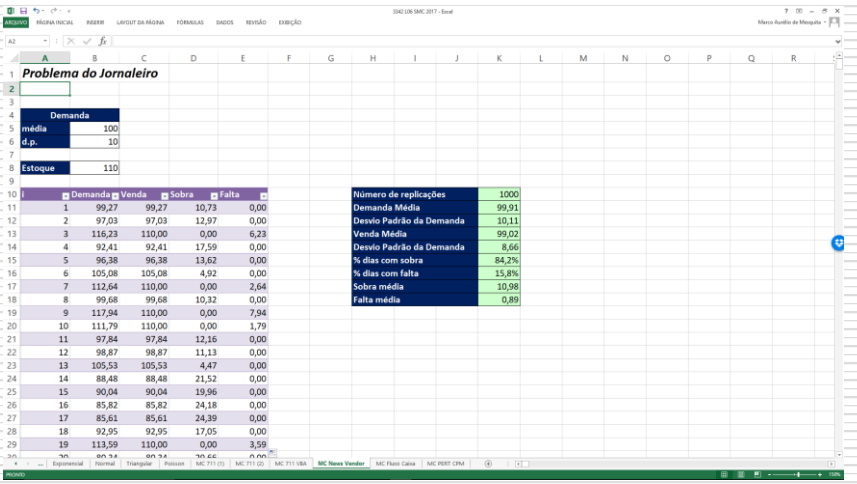
81

Exemplo 1: Resultado = Receita – Despesa



82

Exemplo 2:
Problema do Jornaleiro



Problema do Jornaleiro

Demanda	
média	100
d.p.	10

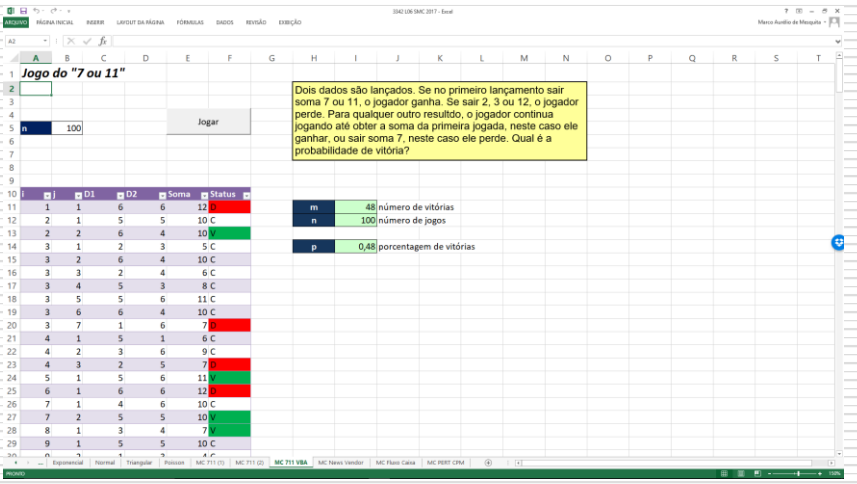
Estoque	110

i	Demanda	Venda	Sobra	Falta
1	99,27	99,27	10,73	0,00
2	97,03	97,03	12,97	0,00
3	116,23	110,00	0,00	6,23
4	92,41	92,41	17,59	0,00
5	96,38	96,38	13,62	0,00
6	105,08	105,08	4,92	0,00
7	112,64	110,00	0,00	2,64
8	99,68	99,68	10,32	0,00
9	117,94	110,00	0,00	7,94
10	111,79	110,00	0,00	1,79
11	97,84	97,84	12,16	0,00
12	98,87	98,87	11,13	0,00
13	105,53	105,53	4,47	0,00
14	88,48	88,48	21,52	0,00
15	90,04	90,04	19,96	0,00
16	85,82	85,82	24,18	0,00
17	85,61	85,61	24,39	0,00
18	92,95	92,95	17,05	0,00
19	113,59	110,00	0,00	3,59

Número de replicações	1000
Demanda Média	99,91
Desvio Padrão da Demanda	10,11
Venda Média	99,02
Desvio Padrão da Demanda	8,66
% dias com sobra	84,2%
% dias com falta	15,8%
Sobra média	10,98
Falta média	0,89

83

Exemplo 3:
Jogo "7 ou 11"



Jogo do "7 ou 11"

Jogar

n 100

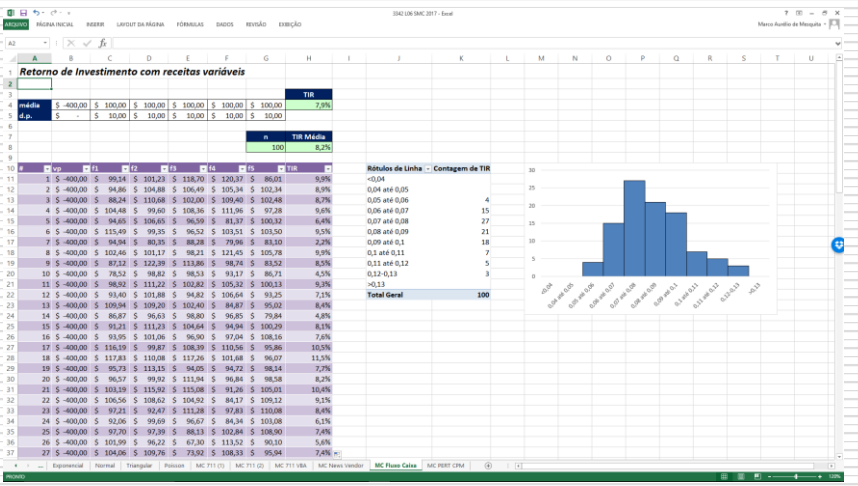
Dois dados são lançados. Se no primeiro lançamento sair soma 7 ou 11, o jogador ganha. Se sair 2, 3 ou 12, o jogador perde. Para qualquer outro resultdo, o jogador continua jogando até obter a soma da primeira jogada, neste caso ele ganhar, ou sair soma 7, neste caso ele perde. Qual é a probabilidade de vitória?

i	J	D1	D2	Soma	Status
1	1	6	6	12	D
2	1	5	5	10	C
2	2	6	4	10	V
3	1	2	3	5	C
3	2	6	4	10	C
3	3	2	4	6	C
3	4	5	3	8	C
3	5	5	6	11	C
3	6	6	4	10	C
3	7	1	6	7	D
4	1	5	1	6	C
4	2	3	6	9	C
4	3	2	5	7	D
5	1	5	6	11	V
6	1	6	6	12	D
7	1	4	6	10	C
7	2	5	5	10	V
8	1	3	4	7	V
9	1	5	5	10	C

m	48	número de vitórias
n	100	número de jogos
p	0,48	porcentagem de vitórias

84

Exemplo 4:
Retorno de Investimento



Retorno de Investimento com receitas variáveis

média	-400,00	100,00	100,00	100,00	100,00	100,00	TIR 7,9%
d.p.	-	10,00	10,00	10,00	10,00	10,00	

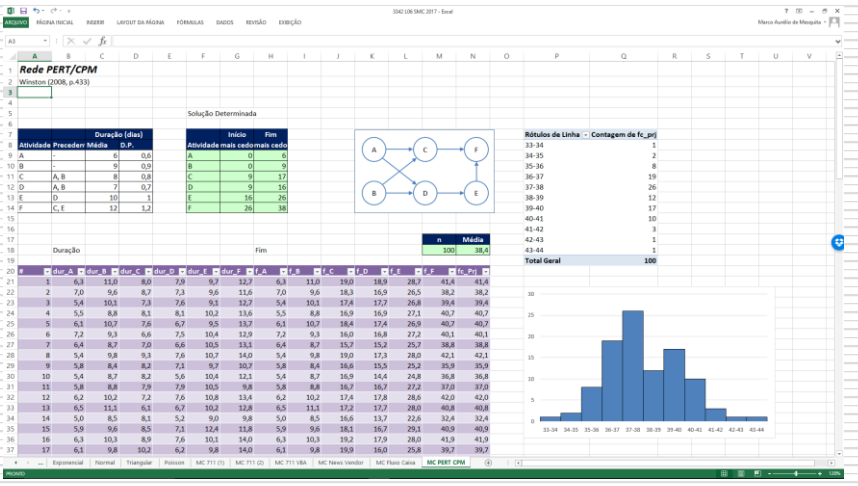
n	TIR Média
100	8,2%

#	vp	f1	f2	f3	f4	f5	TIR
1	-400,00	99,14	101,23	118,70	120,37	86,01	9,9%
2	-400,00	94,86	104,88	106,49	105,34	102,34	8,9%
3	-400,00	88,24	110,68	102,00	109,40	102,48	8,7%
4	-400,00	104,48	99,60	108,36	111,96	97,28	9,6%
5	-400,00	94,65	106,65	96,59	81,37	100,32	6,4%
6	-400,00	115,49	99,35	96,52	103,51	103,50	9,5%
7	-400,00	94,94	80,35	88,28	79,96	83,10	2,2%
8	-400,00	102,46	101,17	98,21	121,45	105,78	9,9%
9	-400,00	87,12	122,39	113,86	98,74	83,52	8,5%
10	-400,00	78,52	98,82	98,53	93,17	86,71	4,5%
11	-400,00	98,92	111,22	102,82	105,32	100,13	9,3%
12	-400,00	93,40	101,88	94,82	106,64	93,25	7,1%
13	-400,00	109,94	109,20	102,40	84,87	95,02	8,4%
14	-400,00	86,87	96,63	98,80	96,85	79,84	4,8%
15	-400,00	91,21	111,23	104,64	94,94	100,29	8,1%
16	-400,00	93,95	101,06	96,90	97,04	108,16	7,6%
17	-400,00	116,19	99,87	108,39	110,56	95,86	10,5%
18	-400,00	117,83	110,08	117,26	101,68	96,07	11,5%
19	-400,00	95,73	113,15	94,05	94,72	98,14	7,7%
20	-400,00	96,57	99,92	111,94	96,84	98,58	8,2%
21	-400,00	103,19	115,92	115,08	91,26	105,01	10,4%
22	-400,00	106,56	108,62	104,92	84,17	109,12	9,1%
23	-400,00	97,21	92,47	111,28	97,81	110,08	8,4%
24	-400,00	92,06	99,69	96,67	84,34	103,08	6,1%
25	-400,00	97,70	97,39	88,13	102,84	108,90	7,4%
26	-400,00	101,99	96,22	67,30	113,52	90,10	5,6%
27	-400,00	104,06	109,76	73,92	108,33	95,94	7,6%

Rótulos de Linha	Contagem de TIR
<0,04	
0,04 até 0,05	
0,05 até 0,06	4
0,06 até 0,07	15
0,07 até 0,08	27
0,08 até 0,09	21
0,09 até 0,1	18
0,1 até 0,11	7
0,11 até 0,12	5
0,12-0,13	3
>0,13	
Total Geral	100

85

Exemplo 5:
Risco em Projeto



Rede PERT/CPM

Winston (2008, p.433)

Solução Determinada

Atividade	Precedentes	Duração (dias) Média	D.P.
A	-	6	0,6
B	-	9	0,9
C	A, B	8	0,8
D	A, B	7	0,7
E	D	10	1
F	C, E	12	1,2

Atividade	Início mais cedo	Fim mais cedo
A	0	6
B	0	9
C	9	17
D	9	16
E	16	26
F	26	38

Duração | Fim

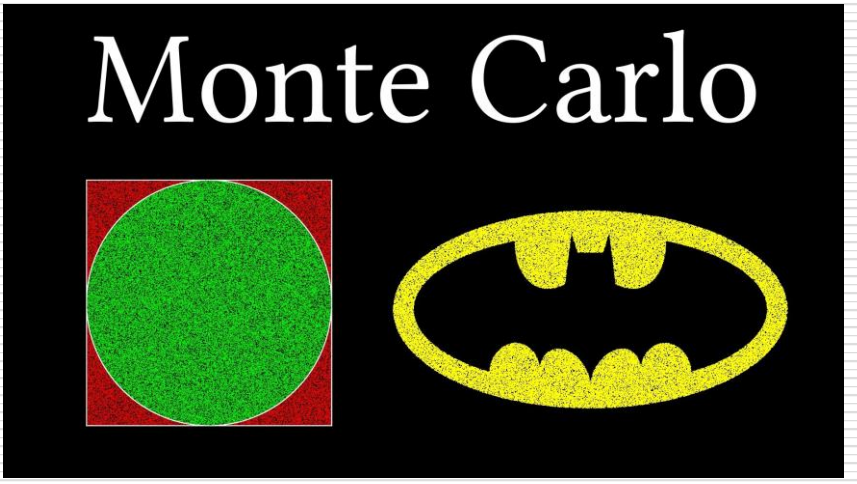
n	Média
100	38,4

#	dur_A	dur_B	dur_C	dur_D	dur_E	dur_E	f_A	f_B	f_C	f_D	f_E	f_E	fc_prj
1	6,3	11,0	8,0	7,9	9,7	12,7	6,3	11,0	19,0	18,9	28,7	41,4	41,4
2	7,0	9,6	8,7	7,3	9,6	11,6	7,0	9,6	18,3	16,9	26,5	38,2	38,2
3	5,4	10,1	7,3	7,6	9,1	12,7	5,4	10,1	17,4	17,7	26,8	39,4	39,4
4	5,5	8,8	8,1	8,1	10,2	13,6	5,5	8,8	16,9	16,9	27,1	40,7	40,7
5	6,1	10,7	7,6	6,7	9,5	13,7	6,1	10,7	18,4	17,4	26,9	40,7	40,7
6	7,2	9,3	6,6	7,5	10,4	12,9	7,2	9,3	16,0	16,8	27,2	40,1	40,1
7	6,4	8,7	7,0	6,6	10,5	13,1	6,4	8,7	15,7	15,2	25,7	38,8	38,8
8	5,4	9,8	9,3	7,6	10,7	14,0	5,4	9,8	19,0	17,3	28,0	42,1	42,1
9	5,8	8,4	8,2	7,1	9,7	10,7	5,8	8,4	16,6	15,5	25,2	35,9	35,9
10	5,4	8,7	8,2	5,6	10,4	12,1	5,4	8,7	16,9	14,4	24,8	36,8	36,8
11	5,8	8,8	7,9	7,9	10,5	9,8	5,8	8,8	16,7	16,7	27,2	37,0	37,0
12	6,2	10,2	7,2	7,6	10,8	13,4	6,2	10,2	17,4	17,8	28,6	42,0	42,0
13	6,5	11,1	6,1	6,7	10,2	12,8	6,5	11,1	17,2	17,7	28,0	40,8	40,8
14	5,0	8,5	8,1	5,2	9,0	9,8	5,0	8,5	16,6	13,7	22,6	32,4	32,4
15	5,9	9,6	8,5	7,1	12,4	11,8	5,9	9,6	18,1	16,7	29,1	40,9	40,9
16	6,3	10,3	8,9	7,6	10,1	14,0	6,3	10,3	19,2	17,9	28,0	41,9	41,9
17	6,1	9,8	10,2	6,2	9,8	14,0	6,1	9,8	19,9	16,0	25,8	39,7	39,7

Rótulos de Linha	Contagem de fc_prj
33-34	1
34-35	2
35-36	8
36-37	19
37-38	26
38-39	12
39-40	17
40-41	10
41-42	3
42-43	1
43-44	1
Total Geral	100

86

Integral de Monte Carlo



87