Universidade de São Paulo

FFCLRP: Método de Monte Carlo

Profa: Fernando FF

Ribeirão Pretro
Junho - 2023

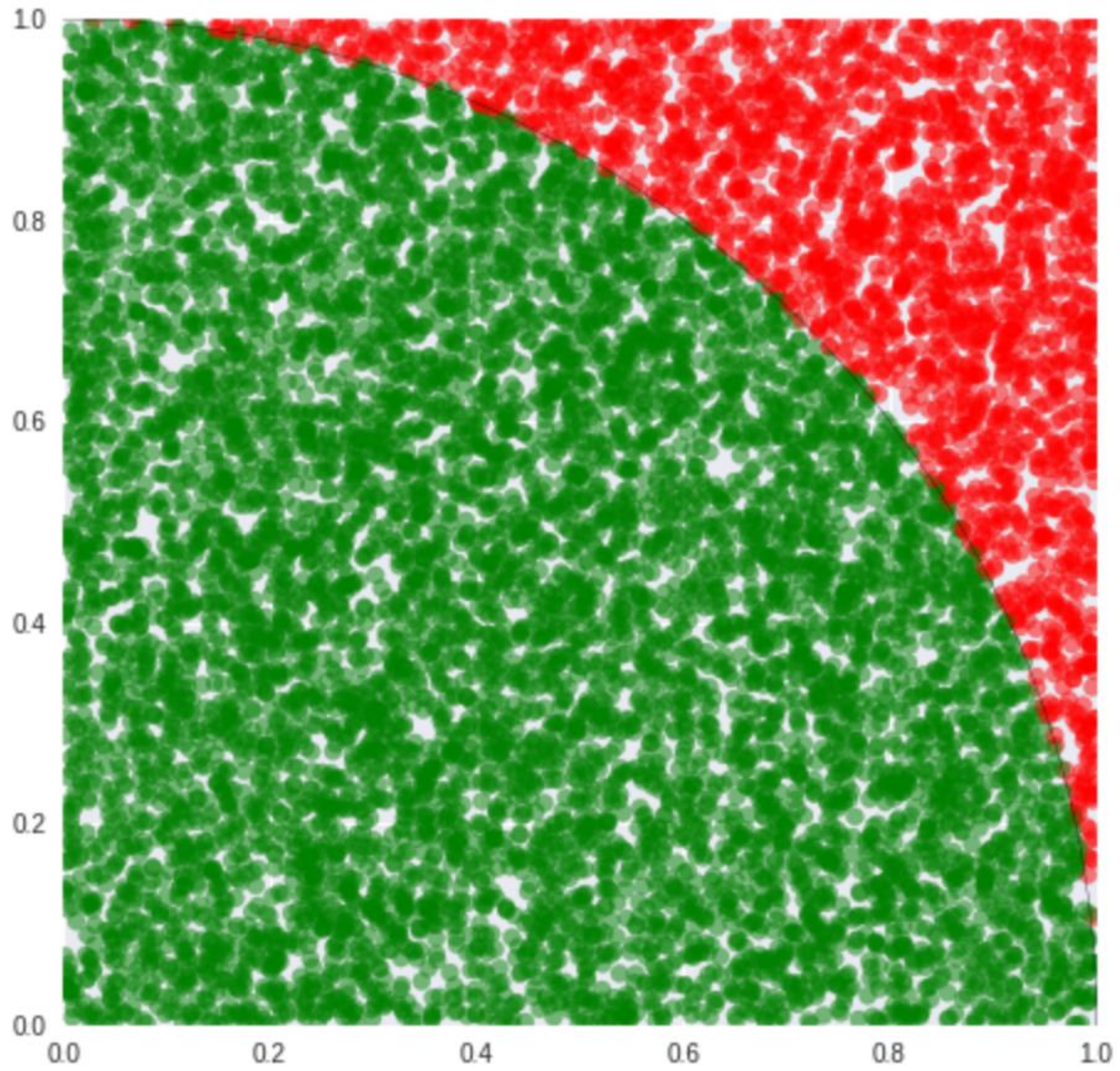
O método de Monte Carlo é uma técnica estatística utilizada para resolver problemas complexos usando amostragem aleatória. Ele foi desenvolvido durante a Segunda Guerra Mundial, como parte do projeto de pesquisa e desenvolvimento do Projeto Manhattan, para auxiliar no desenvolvimento da bomba atômica.

A ideia fundamental do método de Monte Carlo é usar a aleatoriedade para estimar quantidades desconhecidas ou difíceis de calcular. Ele baseia-se na geração de números aleatórios para simular eventos incertos ou estocásticos e, em seguida, utilizar essas simulações para calcular uma aproximação numérica do resultado desejado.

A aplicação básica do método de Monte Carlo envolve os seguintes passos:

1. **Definir o problema: Identificar a quantidade desconhecida ou o problema que se deseja resolver.**
2. **Modelar o problema: Desenvolver um modelo matemático ou computacional que descreva o problema em termos de eventos aleatórios.**
3. **Gerar amostras: Gerar uma sequência de números aleatórios que representem os eventos aleatórios do modelo.**
4. **Simular eventos: Utilizar as amostras geradas para simular os eventos aleatórios e calcular os resultados correspondentes.**
5. **Calcular a estimativa: Realizar cálculos estatísticos com base nas simulações para obter uma estimativa numérica da quantidade desconhecida ou do resultado desejado.**
6. **Refinar a estimativa: Repetir os passos várias vezes, gerando novas amostras e refinando a estimativa com base em um número maior de simulações.**

Queremos gerar uma figura como esta no Python. Os pontos coloridos são gerados aleatoriamente!



Atividade:

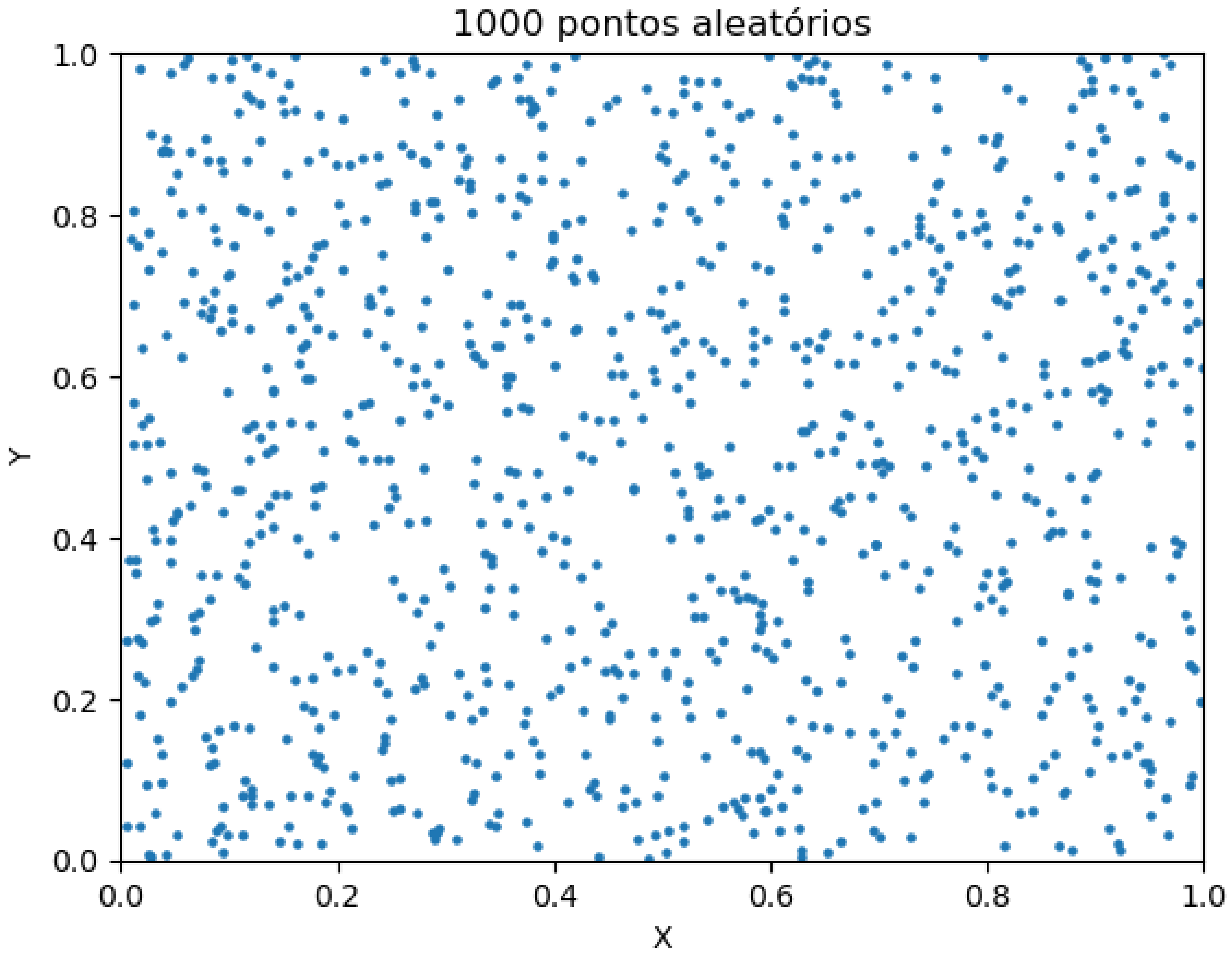
1) Gere uma matriz aleatória com duas colunas para armazenar uma coleção de pontos (x,y). Os valores de x e y devem ser gerados com uma distribuição uniforme no intervalo (0,1)

2) Faça o scatter plot dos pontos gerados acima.

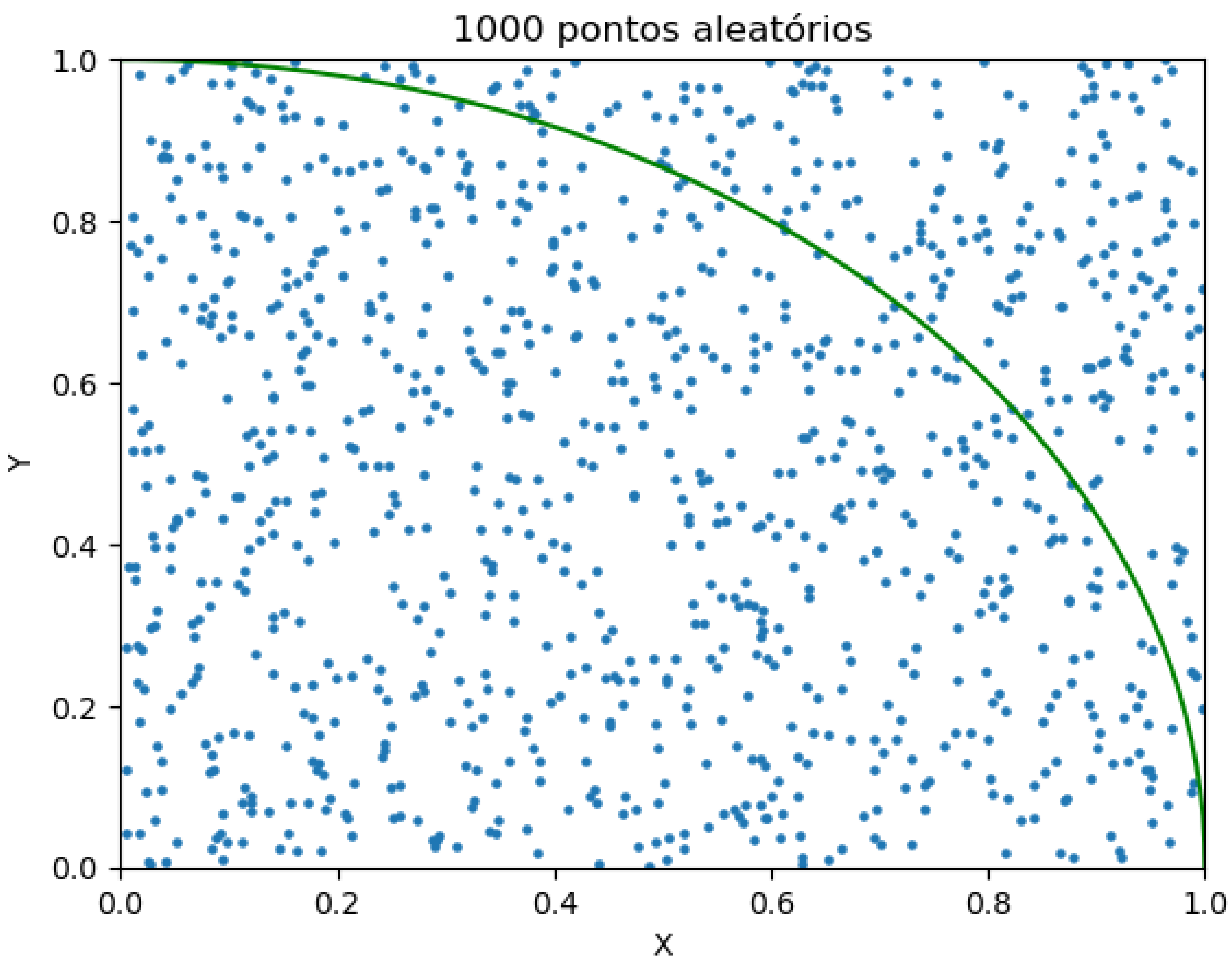
3) Plot sobre este scatter plot um seguimento de reta constante horizontal (y=1 e x no intervalo (0,1)) bem como a transposta, ou seja, x=1 e y no intervalo (0,1).

4) Plot um arco de círculo com raio 1 no quadrante positivo, no gráfico anterior

```python
import numpy as np
import matplotlib.pyplot as plt

# Gerando 1000 pontos aleatórios
np.random.seed(42)  # Define uma semente para reproduzir os resultados
num_points = 1000
x = ???
y = ???

# Plot dos pontos
plt.scatter(x, y, s=5)  # 's' define o tamanho dos pontos
plt.xlim(0, 1)  # Define os limites do eixo x
plt.ylim(0, 1)  # Define os limites do eixo y
plt.xlabel('X')
plt.ylabel('Y')
plt.title('1000 pontos aleatórios')
plt.show()
```

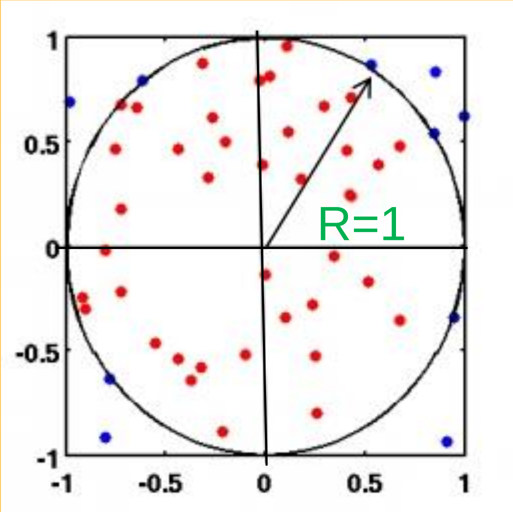

1000 pontos aleatórios
1.0
0.8
0.6
0.4
0.2
0.0
Y
0.0
0.2
0.4
0.6
0.8
1.0
X

```python
# Plot da curva de arco do círculo unitário

theta = np.linspace(0, np.pi, 100)  # ângulos de 0 a pi


x_arc = np.cos(theta)
y_arc = np.sin(theta)

plt.plot(x_arc, y_arc, color='green')
```


1000 pontos aleatórios
1.0
0.8
0.6
0.4
0.2
0.0
Y
0.0
0.2
0.4
0.6
0.8
1.0
X

Faça o código para estimar o valor do PI

Método de Monte Carlo: pseudo-código (π)



L=2

Cont=0;

para j=1 até N:

gerar random values para x e y no intervalor [0,1]

x= rand(0,1)

y=rand(0,1)

calculer f(x)

$$fx = x^2 + y^2$$

Se fx<1:

cont=cont+1;

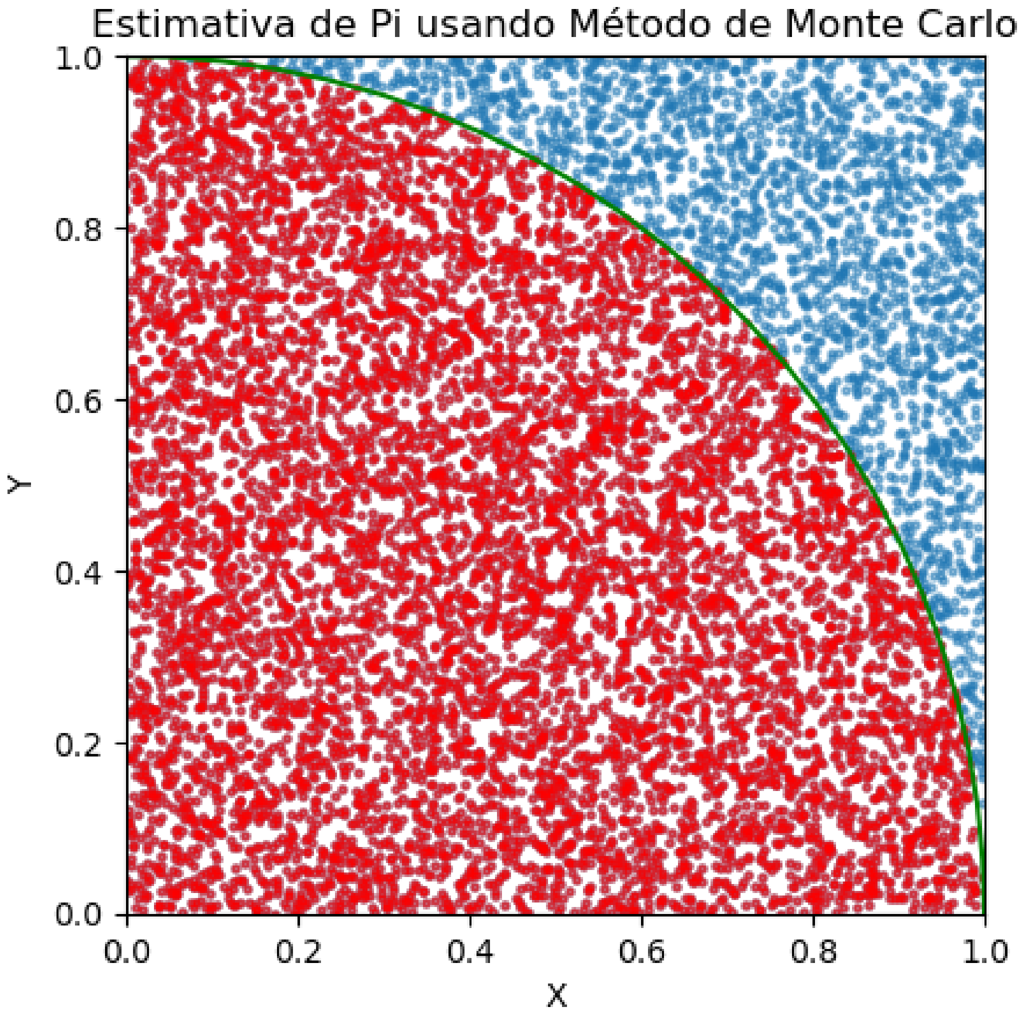
fim se

fim para

pi=4*cont/N

```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(42)

# Geração de pontos aleatórios
num_points = 10000
x = np.random.uniform(0, 1, num_points)
y = np.random.uniform(0, 1, num_points)

# Verificação dos pontos dentro do círculo
Calcule a distância ate a origem : dist_squared = x**2 + y**2
Conte numero de pontos dentro do circulo

# Estimativa de PI
pi_estimate = 4 * num_points_inside_circle / num_points
print("Estimativa de Pi:", pi_estimate)
```

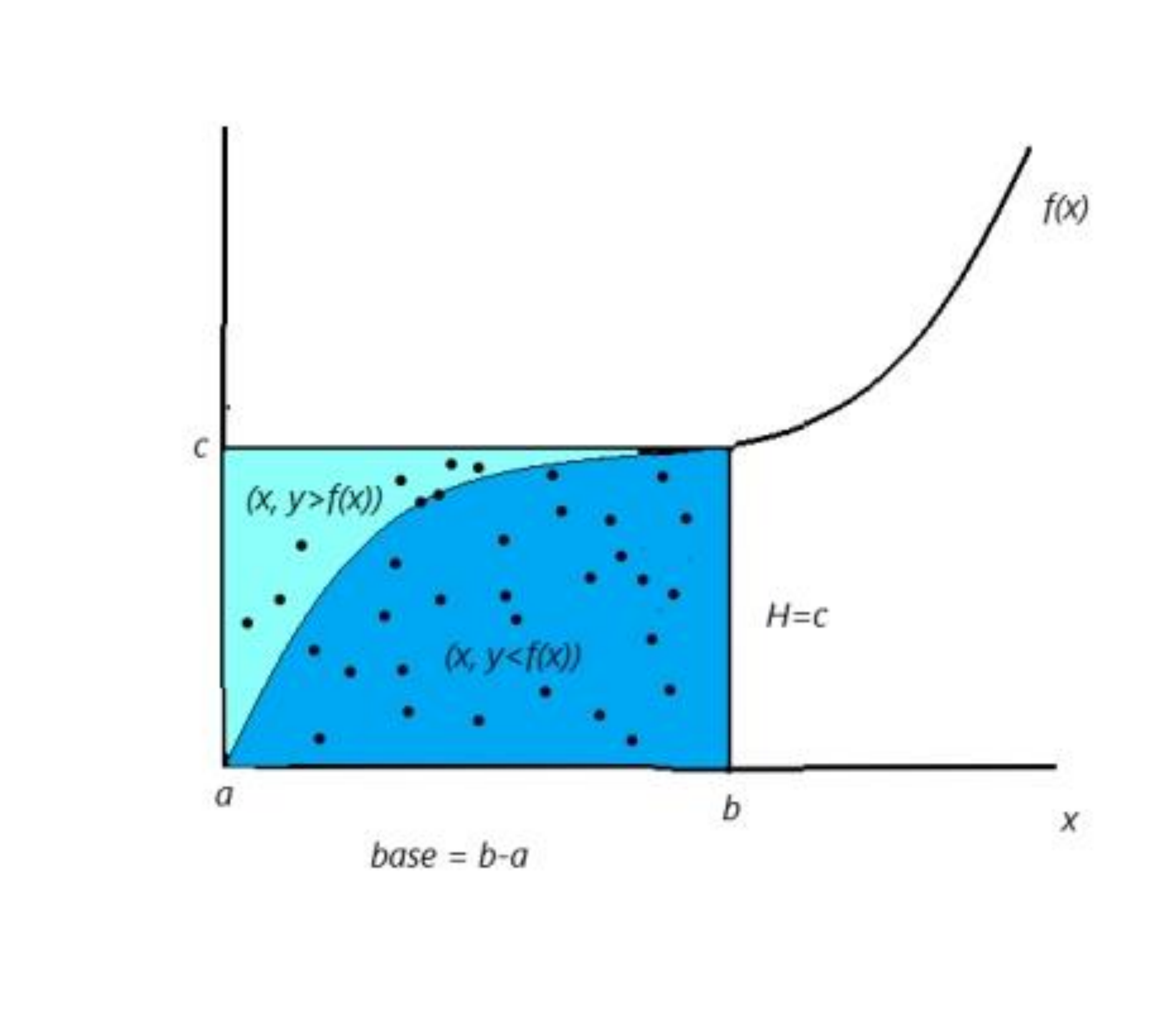
Você poderia pintar os pontos no interior do círculo de vermelho e fora de azul. O plot ficaria assim:



```
# Plot dos pontos e do círculo unitário
plt.scatter(x, y, s=5, alpha=0.5)
plt.scatter(x[points_inside_circle], y[points_inside_circle],
color='r', s=5, alpha=0.5)
theta = np.linspace(0, 2*np.pi, 100)
x_circle = np.cos(theta)
y_circle = np.sin(theta)
plt.plot(x_circle, y_circle, color='green')
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Estimativa de Pi usando Método de Monte Carlo')
plt.gca().set_aspect('equal')
plt.show()
```

Integração com o Método de Monte Carlo

- Podemos usar o método de Monte Carlo na integração numérica. Na figura abaixo, a integral da função é a área entre a função, o eixo dos x, cor azul escuro.



- Note que tem um retângulo
  Com a área

A= (b-a).c

FAÇA UM PROGRAMA PARA CALCULAR A INTEGRAL PARA AS SEGUINTES FUNÇÕES:

$a)\ \ f(x) = \sin(x)\,,\ \ \ldots\,,\ \ para\ \ \ x\, entre\ 0\ e\ 2$

b) f(x)= ln(x) , ..., para x entre 0.1 e 10

c) Para um polinômio de grau 5 , no intervalo 0 a 1

Use Wolfram alpha para ver se o resultado que vc obteve é bom.

```
import random
import math

# Função que queremos integrar
def f(x):
    return math.sin(x)

# Intervalo de integração
a = 0  # Limite inferior
b = math.pi  # Limite superior

# Número de pontos a serem gerados
n = 1000000

resultado = monte_carlo_integration(f, a, b, n)
print("O valor da integral é:", resultado)
```

$$I = (b-a)\frac{\sum_{i=1}^{n} f(xi)}{n}$$

Exercicio da Lista: **The Cauchy distribution**

$$p(x) = \frac{1}{\pi}\frac{1}{1+x^2}, \ -\infty < x < \infty$$

Integração dessa função:

$$c(x) = \frac{1}{\pi}\int_{-\infty}^{x}\frac{dx'}{1+x'^2} = \frac{1}{\pi}\left(\arctan(x) + \frac{\pi}{2}\right)$$

Obtenha uma estimativa par esta integral quando x=10. Calcule a diferença entre o valor exato e o simulado. Escolha número de amostras bem grande.

VALOR ESPERADO OU MÉDIA

O valor esperado de uma variável aleatória, x, é definido como:

$$E[x] = \int_{\Omega} xp(x)dx$$

- **O valor esperado de uma função sobre variáveis aleatórias , *f(x)*, is defined as:**

$$E[f(x)] = \int_{\Omega} f(x)p(x)dx$$

- **A média da amostra, para as amostras xi, é definida como:**

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

$$\overline{g(x)} \approx \frac{1}{n}\sum_{i=1}^{n} g(x_i)$$

Vamos fazer uma aplicação -> VALOR ESPERADO OU MÉDIA

1) Considere uma variável aleatória, x, que segue uma distribuição exponencial com media=2.

Calcule

$$E[x] = \int_{\Omega} x p(x) dx \qquad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

2) Repita o exercício anterior para a distribuição Normal com media= 4 e desvio padrão =2

3) O valor esperado quando função sobre variáveis aleatórias *uniforme*

***f(x),* is cos(x), para x entre 0 e π:**

$$E[f(x)] = \int_{\Omega} f(x) p(x) dx \qquad \overline{f(x)} \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

Faça um código e execute-o para os seguintes problemas:

a) Calcular a média de um variável aleatória lognormal com média 2 e variância 2.
b) Calcular o valor médio da função

$$\overline{\boldsymbol{f}(\boldsymbol{x})} = 0.5 \sin(x))^2$$

no intervalo 0 a Pi, sendo que x segue uma distribuição normal (1,1)

Demanda por Leitos Hospitalares

Quanto a oferta: O número de leitos disponíveis na cidade é igual a 250 sendo 200 de enfermaria e 50 de UTI.

Já a demanda por leitos no hospital depende do Cenário

- **alto entre 100 e 200, cuja probabilidade ocorrer é 40%**
- **médio entre 200 e 250, cuja probabilidade ocorrer é 35%**
- **baixo entre 250 e 400, cuja probabilidade ocorrer é 25%**
- Historicamente, 20% dos pacientes vão para UTI.

Calcule a distribuição do número de pessoas que demandam UTI bem como de pessoas que ficarão sem acesso ao leito de UTI. (use distr. Uniforme

Demanda por Leitos Hospitalares: Tabela

experimento	Cenário - Gerar aleatorio	Demanda Por leitos Gerar aleatorio Segundo o cenário	Pesoas na UTI	Pessoas enfermaria
1	Alta	300	50	200
2	Baixa	110	22	88
…….	Baixa	180		
100	Média	220		
200	Alta	377		
……				
1000	Baixa	152		

```
import random    # ou import numpy as np e usar np.random.uniform(0,1)ou np.random.randint(a,b)
import matplotlib.pyplot as plt

# Número de leitos de enfermaria e UTI disponíveis
leitos_enfermaria = 200
leitos_uti = 50

# Probabilidade de demanda em cada cenário
prob_demanda_alta = 0.4
prob_demanda_media = 0.35
prob_demanda_baixa = 0.25

# Executar simulação
enfermaria_ocupacao, uti_ocupacao, pacientes_sem_acessohospital =
 simular_ocupacao(leitos_enfermaria, leitos_uti, prob_demanda_alta, prob_demanda_media, prob_demanda_baixa)
```

Vc precisa criar uma função simular_ocupação que entra com variáveis:
leitos_enfermaria, leitos_uti, prob_demanda_alta, prob_demanda_media, prob_demanda_baixa

Retorna: enfermaria_ocupacao, uti_ocupacao, pacientes_sem_acessohospital

Plotar histograma da ocupação de enfermaria, da ocupação de UTI e dos sem leitos

Executar simulação

```
enfermaria_ocupacao, uti_ocupacao, pacientes_sem_acessohospital =
simular_ocupacao(leitos_enfermaria, leitos_uti, prob_demanda_alta, prob_demanda_media,
prob_demanda_baixa)
```

```
# Plotar histograma da ocupação de enfermaria
plt.hist(enfermaria_ocupacao, bins='auto', color='blue',density="True")
plt.title("Distribuição da Ocupação de Leitos de Enfermaria")
plt.xlabel("Número de Pacientes")
plt.ylabel("Frequência")
plt.show()

# Plotar histograma da ocupação de UTI
plt.hist(uti_ocupacao, bins='auto', color='red',density="True")
plt.title("Distribuição da Ocupação de Leitos de UTI")
plt.xlabel("Número de Pacientes")
plt.ylabel("Frequência")
plt.show()

# Plotar histograma dos sem leitos
plt.hist(pacientes_sem_acessohospital, bins='auto', color='k',density="True")
plt.title("Distribuição pacientes sem acessso a  Leitos")
plt.xlabel("Número de Pacientes")
plt.ylabel("Frequência")
plt.show()

# Número de pessoas sem acesso a enfermaria e UTI
print("Número de pessoas sem acesso a enfermaria e UTI:", pacientes_sem_acessohospital)
print("Número de pessoas com acesso a enfermaria:",enfermaria_ocupacao )
print("Número de pessoas com acesso a  UTI:",uti_ocupacao )
```

Exercício 1 da Lista de Simulação de Monte Carlo

Monte Carlo: Tomada de Decisão

A SIMULAÇÃO MONTE CARLO COMO INSTRUMENTO PARA A TOMADA DE DECISÃO: CENÁRIO ALTERNATIVO EM UMA INDÚSTRIA DE CONFFECÇÃO

Valderedo Sedano Fontana1

Ednea Zandonadi Brambila Carletti2

Valquiria Cruz Cereza3

Síndia Pessin Andreon4

Revista Dimensão Acadêmica, v.1, n.1, jan-jun. 2016

Estudo de Caso. A empresa Beta possui 12 funcionários diretos e mais 50 indiretos, tem uma capacidade de produção de 6.000 peças/mês, a empresa Beta é hoje a maior indústria de uniformes profissionais da cidade de Cachoeiro de Itapemirim.

objeto de estudo é a projeção de lucro para as peças de Uniforme, produzidas pela empresa. Destacamos a função objetivo;

Lucro = [(PV – CMP - CMO) x Demanda] – CDF,

PV representa o Preço de Venda,
CMP o Custo da Matéria Prima,
CMO Custo da Mão de Obra,
e CDF Custos e Despesas Fixas .

Custos: A para o menor e,*B* maior custo

De posse destes dados, iniciou-se a aplicação da técnica de simulação, com objetivo de projetar o lucro para o próximo ano.

Tabela 1 Relação de variáveis, matéria prima e mão de obra

	MATERIA PRIMA	MAO DE OBRA
A	12	1
B	19	4

Fonte: Os autores, 2016

O Custo da matéria prima foi calculado por meio da função:

CMP= a+[(b-a)*rand].

"rand" ´uniforme entre (0,1) de acordo com o artigo.
Minha opinião rand deveria ser 0 ou 1 (binário)

O Custo da mão de obra

Tabela 2 Relação de custo/fornecedor

A	B	C	D	E
2	4	4	1	2

Fonte: Os autores (2016)

Tabela 3 Intervalo de Números aleatórios para a variável Custo de Mão de Obra

Fornecedor	Frequência relativa	Frequência acumulada	Limite Inferior	Limite Superior
A	0,10	0,10	0,00	0,10
B	0,20	0,30	0,11	0,30
C	0,40	0,70	0,31	0,70
D	0,20	0,90	0,71	0,90
E	0,10	1,0	0,91	1,0
Total	1,0			

Fonte: Os autores (2016)

Apenas o Preço de Venda e os Custos e Despesas Fixas já estavam determinados.

Com relação a variável demanda, a mesma possui uma de demanda entre 1000 e 6.000 unidades, com média 3500 e desvio-padrão 3535

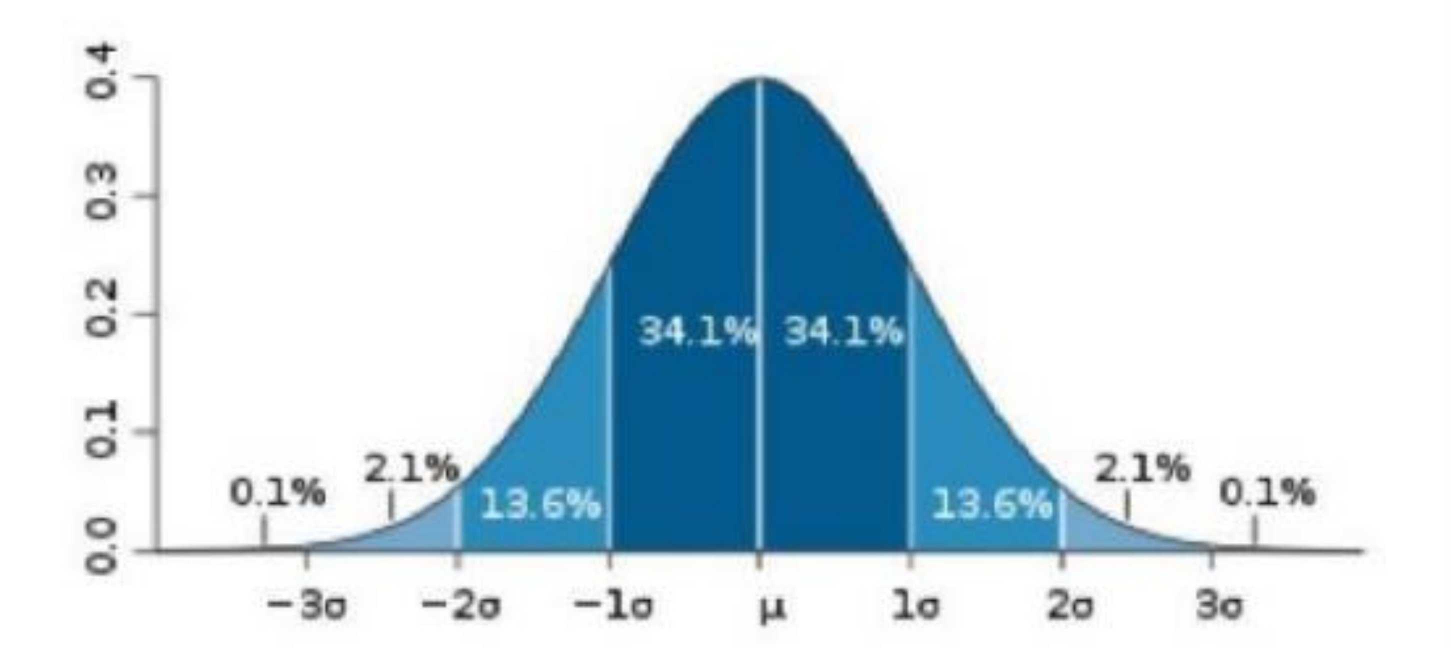


Figura 1 Projeção do resultado das variáveis custo da matéria prima, custo da mão de obra, custo da demanda e projeção do Lucro

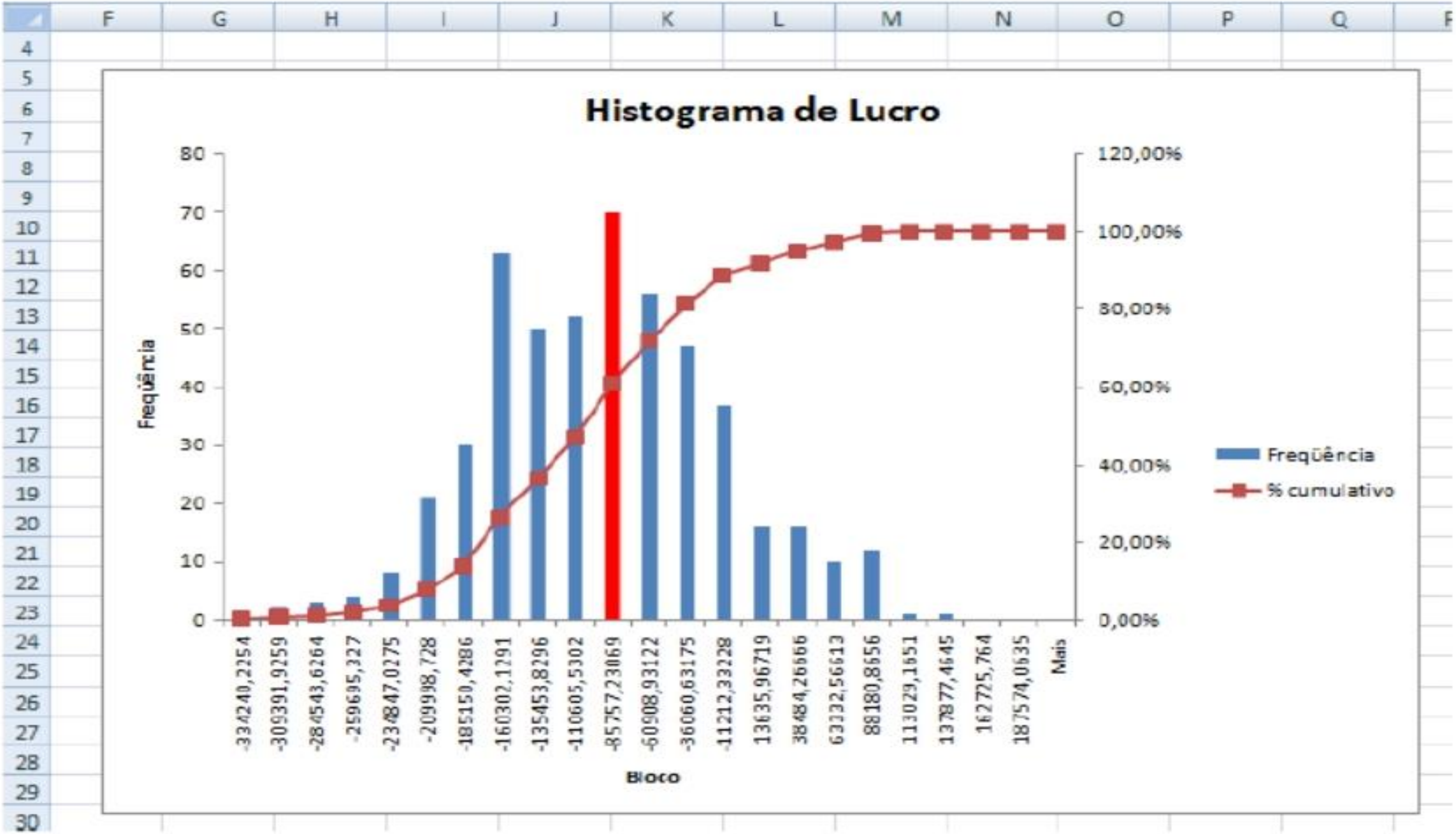
	A	B	C	D	E	F	G
13			PROJEÇÃO DE LUCRO				
14							
15	**EXPERIMENTO**	**PV**	**CMP**	**CMO**	**DEMANDA**	**CDF**	**LUCRO**
16	1	40	17,72477	1	7936,51883	180.000	-11148,7229
17	2	40	18,89147	4	5153,569404	180.000	-91830,0092
18	3	40	15,29969	4	2114,037965	180.000	-136238,7611
19	4	40	12,58722	1	7989,734096	180.000	31031,11185
20	5	40	18,25143	4	1430,711468	180.000	-154606,9169
21	6	40	18,93206	1	2679,124856	180.000	-126235,4809
22	7	40	15,73786	4	6641,921011	180.000	-45420,43514
23	8	40	16,02665	4	7141,907716	180.000	-37352,14759
24	9	40	15,52583	4	8033,746758	180.000	-15515,73811
25	10	40	18,46764	4	4574,058028	180.000	-99805,94607
505	490	40	15,16352	2	873,9345237	180.000	-160042,4151
506	491	40	17,00342	1	1609,107572	180.000	-144605,1335
507	492	40	16,18202	4	11782,33195	180.000	53502,02024
508	493	40	14,00448	4	4700,975956	180.000	-76599,57712
509	494	40	12,18103	1	6955,176917	180.000	6530,653
510	495	40	12,8858	4	4125,818159	180.000	-84635,02854
511	496	40	13,38634	2	9386,382839	180.000	51033,19754
512	497	40	18,08034	1	1876,948294	180.000	-140734,8751
513	498	40	12,4014	2	3666,871177	180.000	-86133,24581
514	499	40	13,67005	4	4536,998713	180.000	-78689,06824
515	500	40	17,73098	4	13607,38864	180.000	68593,64842
516						**Média**	-R$ 104.279,71
517							

Fonte: Os autores (2016)

Figura 2 Análise dos dados obtidos com relação ao Lucro.

	I	J
14	**Análise de dados**	
15	*Coluna1*	
16		
17	Média	-105287,2974
18	Erro padrão	3359,53301
19	Mediana	-104364,6732
20	Desvio padrão	75121,44184
21	Variância da amostra	5643231024
22	Curtose	-0,037112504
23	Assimetria	-0,111867389
24	Intervalo	458775,8692
25	Mínimo	-346451,1169
26	Máximo	112324,7524
27	Soma	-52643648,68
28	Contagem	500
29	Nível de confiança(95,0%)	6600,572964
30		
31		
32		
33	**Intervalo de confiança**	
34	Intervalo Inferior	-111887,8703
35	Intervalo Superior	-98686,7244

Figura 3 Histograma das simulações de lucro da empresa Beta Confecções



Fonte: Os autores (2016)

Assitam ao vídeo da UNIVESP

https://www.youtube.com/watch?v=_dqcIbabEJk

Bibliografia

Video
[1] https://www.youtube.com/watch?v=rFtyUpTz3v0

Artigos
[1] Yoriyaz, Hélio. "Método de Monte Carlo: princípios e aplicações em Física Médica." *Revista Brasileira de Física Médica* 3.1 (2009): 141-149.

[2] https://multivix.edu.br/wp-content/uploads/2018/09/revista-dimensao-academica-v01-n01-artigo-05.pdf

[3]http://www.din.uem.br/sbpo/sbpo2016/pdf/156669.pd

Livros
[1] SCHERER, Claudio. **Métodos computacionais da física**. Editora Livraria da Física, 2005.


$P\ x\ = C\,x^{-\alpha}$
obrigado