1ª Lista de Exercícios

Aluno: Luiz Renault Leite Rodrigues

1. Mostre, usando análise e simulação, que o gerador de números aleatórios definido por $X_{i+1}= 5x_i \bmod (7)$ é um gerador de período completo. Determine a sequência gerada para sementes $x_0=4$ e $x_0= 7$. Compare as sequências e comente os resultados.

Como é um GCM($x_{i+1} = ax_i\text{mod}\ (m)$) com $m=7\neq 2^k$, para ser de período completo com ρ=m-1=6:

- a deve ser raiz primitiva de m, como m=7 é número primo, ou seja, $a^n\text{mod}(m)\neq 1$ para $n=1,2,\ldots,(m-2=5)$:

$5^1\text{mod}7=5\text{mod}7=5$ $\quad$ $5^2\text{mod}7=25\text{mod}7=4$ $\quad$ $5^3\text{mod}7=125\text{mod}7=6$

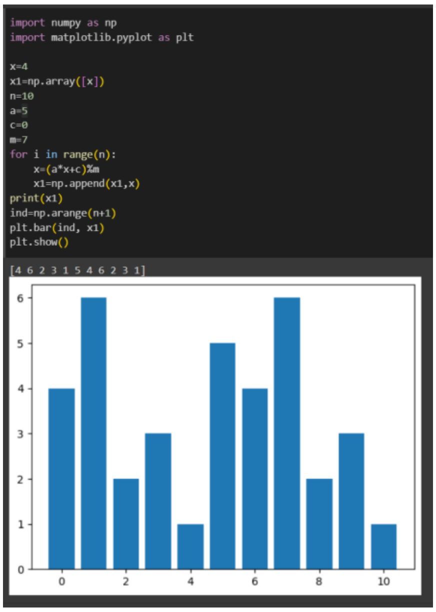
$5^4\text{mod}7=625\text{mod}7=2$ $\quad$ $5^5\text{mod}7=3125\text{mod}7=3$

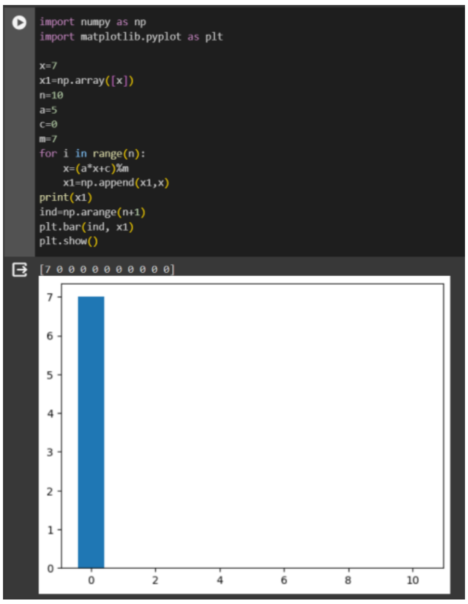
Assim,

para	$x_0=4$	$x_0=7$
	$x_1=5*4\text{mod}7=6$	$x_1=5*7\text{mod}7=0$
	$x_2=5*6\text{mod}7=2$	$x_2=5*0\text{mod}7=0$
	$x_3=5*2\text{mod}7=3$	
	$x_4=5*3\text{mod}7=1$	
	$x_5=5*1\text{mod}7=5$	
	$x_6=5*5\text{mod}7=4$	

para $x_0=4$, período=m-1=6 $\qquad$ para $x_0=7$, período=1

Para a simulação:





2.O número de chamadas para o help-desk de uma empresa tem uma distribuição de Poisson com 60 chamadas por um período de 10 horas. Se C = a variável aleatória para o número de chamadas por hora, encontre:

a. A probabilidade de que o suporte técnico não receba chamadas em uma determinada hora.

Distribuição de Poisson com $\lambda=60/10=6$

$P(X=x)= \lambda^x e^{-\lambda}/x!$

$P(x=0)=6^0 e^{-6}/0!=e^{-6}=0.00247875$

```
import numpy as np


lambda1=6 #Número médio de requisições
N=10000000 #Numero de amostras
value=2
count=0
av=np.array([])
x=np.random.uniform(0,1,N)
for ix in x:
    i = 0
    pr = np.exp(-lambda1)
    F=pr
    while ix>=F:
        pr=lambda1/(i+1)*pr
        F = F + pr
        i = i + 1;
    a1=i
    if i==0:
      count = count+1
    #av=np.append(av,a1)

#print(sum(av==0)/N)
print (count/N)
```

```
0.0024774
```

b. a probabilidade de que o suporte técnico receba menos de oito chamadas em uma determinada hora.

$P(X\leq x)= e^{-\lambda}\sum_{i=0}^{x}\lambda^i/i!$

$P(X<8)= P(X\leq 7)=e^{-}\lambda[1+\lambda+\lambda^2/2+\lambda^3/6+\lambda^4/24+\lambda^5/120+\lambda^6/720+\lambda^7/5040]=e^{-6}[1+6+18+36+54+64.8+64.8+55.5428]=0.74398$

```python
import numpy as np



lambda1=6 #Número médio de requisições
N=10000000 #Numero de amostras
value=2
count=0
av=np.array([])
x=np.random.uniform(0,1,N)
for ix in x:
    i = 0
    pr = np.exp(-lambda1)
    F=pr
    while ix>=F:
        pr=lambda1/(i+1)*pr
        F = F + pr
        i = i + 1;
    a1=i
    if i<8:
      count = count+1
    #av=np.append(av,a1)

#print(sum(av==0)/N)
print (count/N)
```

```
0.744048
```

c. O número médio de chamadas por hora E (C).

A média é o próprio parâmetro λ=60/10=6 chamadas por hora

```python
import numpy as np



lambda1=6 #Número médio de requisições
N=10000000 #Numero de amostras
value=2
count=0
av=np.array([])
x=np.random.uniform(0,1,N)
for ix in x:
    i = 0
    pr = np.exp(-lambda1)
    F=pr
    while ix>=F:
        pr=lambda1/(i+1)*pr
        F = F + pr
        i = i + 1;
    a1=i
    count = count+i
    #av=np.append(av,a1)

#print(sum(av==0)/N)
print (count/N)
```

```
6.0007846
```

d. A variância de C.

A variância é o próprio parâmetro λ=60/10=6

```
import numpy as np


lambda1=6 #Número médio de requisições
N=1000000 #Numero de amostras
value=2
count=0
count2=0
av=np.array([])
x=np.random.uniform(0,1,N)
for ix in x:
    i = 0
    pr = np.exp(-lambda1)
    F=pr
    while ix>=F:
        pr=lambda1/(i+1)*pr
        F = F + pr
        i = i + 1;
    a1=i
    count = count+i
    count2 = count2 + i*i
    #av=np.append(av,a1)

#print(sum(av==0)/N)
print (count2/N - pow(count/N,2))
```

```
5.996895605616004
```

e. O desvio padrão de C

O desvio padrão é a raiz quadrada da variância: $\sqrt{6} = 2.4495$

```
import numpy as np


lambda1=6 #Número médio de requisições
N=1000000 #Numero de amostras
value=2
count=0
count2=0
av=np.array([])
x=np.random.uniform(0,1,N)
for ix in x:
    i = 0
    pr = np.exp(-lambda1)
    F=pr
    while ix>=F:
        pr=lambda1/(i+1)*pr
        F = F + pr
        i = i + 1;
    a1=i
    count = count+i
    count2 = count2 + i*i
    #av=np.append(av,a1)

#print(sum(av==0)/N)
print (np.sqrt(count2/N - pow(count/N,2)))
```

```
2.4507542596676237
```

3)Um fabricante de pistões de metal descobre que, em média, 15% de seus pistões são rejeitados porque são superdimensionados ou subdimensionados. Qual é a probabilidade de um lote de 8 pistões conter

(a) não mais que 2 rejeitados?

Distribuição Binomial com p=0.15, n=8

P(x<=2)=$(\genfrac{}{}{0pt}{}{8}{0}(1-\mathrm{p})^8 + (\genfrac{}{}{0pt}{}{8}{1}(1-\mathrm{p})^7\mathrm{p} + \{(\genfrac{}{}{0pt}{}{8}{2})(1-\mathrm{p})^6\mathrm{p}^2$ =0.8947

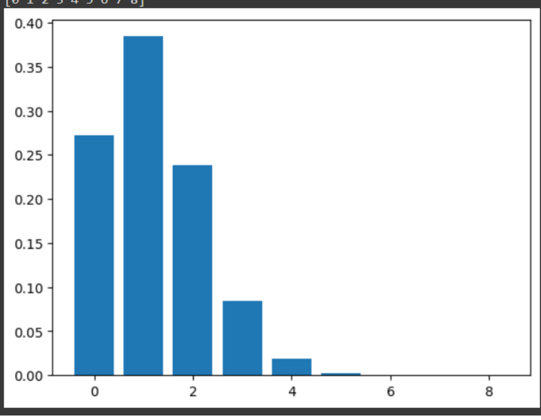
(b) pelo menos 6 rejeitados?

$\mathrm{P}(\mathrm{X} \geq 6) = (\genfrac{}{}{0pt}{}{8}{6}(1-\mathrm{p})^2\mathrm{p}^6 + (\genfrac{}{}{0pt}{}{8}{7}(1-\mathrm{p})\mathrm{p}^7 + (\genfrac{}{}{0pt}{}{8}{8}\mathrm{p}^8$=0.0002423

-Traçar o histograma da variável analisada.

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


q=0.15 #Probabilidade de acerto
n=8 #Número de tentativas
N=1000000 #Número de amostras
c = q/(1-q)
av=np.array([i for i in range(0,n+1)])
count=np.array([0 for i in range(0,n+1)])

for ii in range(1,N):
    ix=np.random.uniform(0,1)
    i = 0
    pr = pow((1 - q),n)
    F = pr
    while ix>=F:
        pr = (c * (n - i) / (i + 1))* pr;
        F = F + pr;
        i = i + 1;
    a1=i
    count = count + (av==i)

prob1=sum(np.multiply(av<=2, count))/N
prob2=sum(np.multiply(av>=6, count))/N
print("item a:",prob1)
print("item b:",prob2)

print(count)
print(av)

plt.bar(av,count/N)
plt.show()
#Código simplificado para rodar mais rápido (e com economia de memória)
```

```
item a: 0.894669
item b: 0.000227
[272132 384385 238152  83949  18562   2592    218      9      0]
[0 1 2 3 4 5 6 7 8]
```



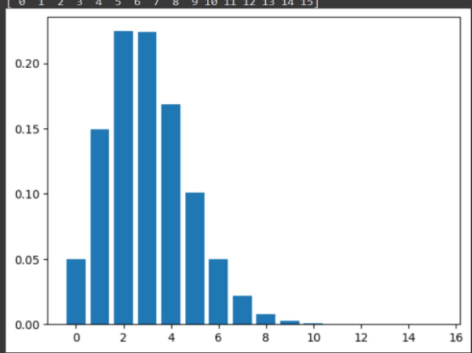
4) Se ocorrerem falhas de energia elétrica de acordo com uma distribuição de Poisson com uma média de 6 falhas a cada duas semanas, calcule a probabilidade de que haverá ao menos 2 falhas durante uma semana específica. Traçar o histograma da variável analisada.

Distribuição de Poisson com λ=6/2=3

P(x≥2)=1-P(X≤1)=1- $e^{-3}\sum_{i=0}^{1} 3^{i}/i! = 1-0.04978[1+3] = 1-0.1991 = 0.800582$

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


lambda1=3 #Número médio de requisições
N=1000000 #Numero de amostras
value=2

av=np.array([i for i in range(0,1)])
count=av*0

for ii in range(1,N):
    ix=np.random.uniform(0,1)
    i = 0
    pr = np.exp(-lambda1)
    F=pr
    while ix>=F:
        pr=lambda1/(i+1)*pr
        F = F + pr
        i = i + 1;
    a1=i
    for ia in range(len(av)-1, i):
        av=np.append(av, ia+1)
        count=np.append(count, 0)

    count = count + (av==i)

print(av)

plt.bar(av,count/N)
plt.show()
print("Probabilidade: ", sum(np.multiply(av>=2, count))/N)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
```




```
Probabilidade:  0.801242
```

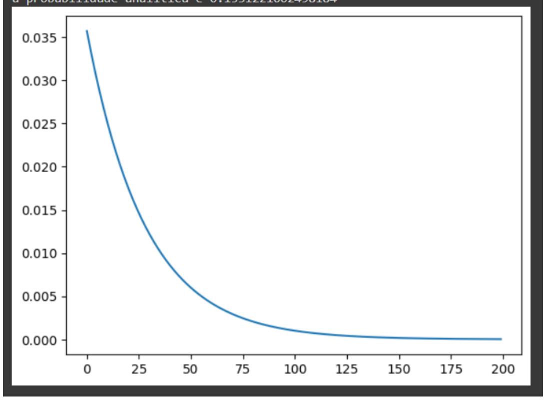
5) O número de dias que os viajantes compram com antecedência suas passagens aéreas pode ser modelado por uma distribuição exponencial com o tempo médio igual a 28 dias. Encontre a probabilidade de um viajante comprar uma passagem com menos de 4dias de antecedência. Traçar a pdf da variável analisada.

- Distribuição exponencial com λ=1/28:

$P(X \le x) = 1 - e^{-\lambda x}$ $P(X \le 4) = 1 - e^{-4/28} = 0.1331$

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


N=1000000
lambda1=1/28
x=np.random.uniform(0,1,N)
value=4
Xexp=-np.log(x)/lambda1

count=sum(Xexp<4)
print(Xexp)

prob=count/N
print("a probabilidade e",prob)

Probana=1-np.exp(-lambda1*value)
print("a probabilidade analítica e",Probana)

X=np.arange(0, 200, 1)
fx=lambda1*np.exp(-lambda1*X)
plt.plot(X,fx)
plt.show()
```

```
[34.1468604   3.25507769 11.78598751 ...  3.11848773 11.29779796
  2.83434418]
a probabilidade e 0.132571
a probabilidade analítica e 0.1331221002498184
```



6) A distribuição discreta geométrica conta o número de tentativas até o primeiro sucesso. A pmf é dada por $f(x)=p(1-p)^{x-1}$, onde p representa a probabilidade de sucesso e x o numero de tentativas. Fazer um algoritmo para a geração das variáveis aleatórias geométricas. Com o algoritmo proposto calcular:

Uma urna tem 30 bolas brancas e 20 bolas pretas. Qual a probabilidade de que a 6ª bola retirada com reposição seja a primeira bola preta?

p=30/50=0.6

$P(X=6)=0.6(1-0.6)^5=0.6*0.01024= 0.006144$

Algoritmo: geração de variável aleatória geométrica pelo método da inversa:

$$\mathrm{P}(X \leq x) = 1-(1-p)^{floor(x)}, \mathrm{x} \geq 1$$

$$(1-p)^{floor(x)} = 1-\mathrm{u}$$

$$\left\{log\left((1-p)\right\}^{floor(x)}\right) = \log(1-\mathrm{u})$$

$$\left\{log\left((1-p)\right\}^{floor(x)}\right) / \log(1-\mathrm{p}) = \log(1-\mathrm{u}) / \log(1-\mathrm{p})$$

$$\mathrm{floor}(x) = \log(1-\mathrm{u}) / \log(1-\mathrm{p})$$

$$\mathrm{x} = 1 + \mathrm{floor}\left(\log(1-\mathrm{u}) / \log(1-\mathrm{p})\right)$$

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


N=1000000
p=0.6
x=np.random.uniform(0,1,N)
value=6
Xgeo=np.floor(np.log(x)/np.log(1-p)) + 1

count=sum(Xgeo==value)
print(Xgeo)

prob=count/N
print("a probabilidade e",prob)

plt.show()
```

```
[2. 1. 3. ... 2. 1. 1.]
a probabilidade e 0.006199
```

7) Utilizando o método da inversa, gerar amostras para a distribuição f(x)=e^x/(e^2-1), 0≤x≤2.

$$\mathrm{F}(x) = \int_0^x \frac{e^x}{e^2 - 1}\mathrm{dx}$$

$$\mathrm{F}(x) = \frac{\int_0^x e^x dx}{e^2 - 1}$$

$$\mathrm{F}(x) = \frac{e^x - 1}{e^2 - 1}, 0 \leq x \leq 2$$

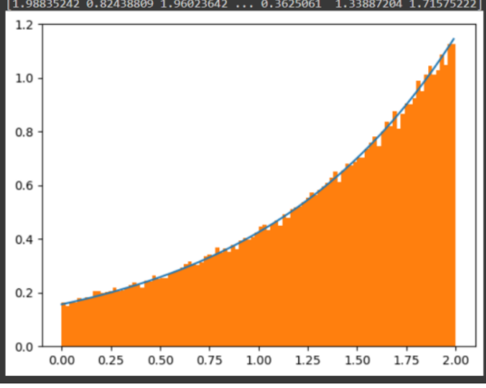
$$\mathrm{u}(\mathrm{e}^2 - 1) = e^x - 1$$

$$e\hat{\ }x = \mathrm{u}(\mathrm{e}^2 - 1) + 1$$

$$x = ln(\mathrm{u}(\mathrm{e}^2 - 1) + 1)$$

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


N=100000
p=0.6
x=np.random.uniform(0,1,N)
value=6
X=np.log(x * (np.e**2-1) +1)

x=np.arange(0,2,0.01)
fx=pow(np.e,x) / (pow(np.e,2) - 1)

print(X)
plt.plot(x,fx)
plt.hist(X, bins=100, density=True)
plt.show()
```

```
[1.98835242 0.82438809 1.96023642 ... 0.3625061  1.33887204 1.71575222]
```



8) Utilizando o método da aceitação/rejeição gerar amostras para a distribuição

$f(x)=1.5x^2$, $-1<x<1$

Plotar a pdf analítica e o histograma normalizado.

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

n=100000
X=np.array([])

for i in range(n):
    x1=np.random.uniform(-1,1)
    x2=np.random.uniform(0,1.5)

    while (x2 >= 1.5*x1**2):
        x1=np.random.uniform(-1,1)
        x2=np.random.uniform(0,1.5)
    X = np.append(X, x1)

x=np.arange(-1,1,0.01)
fx=1.5*x**2

print(X)
plt.plot(x,fx)
plt.hist(X, bins=100, density=True)
plt.show()
```

```
[ 0.93050357 -0.99623093  0.71565217 ...  0.97606112  0.53655301
 -0.79549797]
```