**Lista de exercícios - Princípios de Simulação de Sistemas de Comunicação**

**João Paulo Silva Dias**

1. Mostre, usando análise e simulação, que o gerador de números aleatórios definido por **Xi+1 = 5Xi mod (7)** é um gerador de período completo. Determine a sequência gerada para sementes **x0 = 4** e **x0 = 7**.

Compare as sequências e comente os resultados.**Xi+1 = 5Xi mod (7)**

**Xn+1 = aXn mod (m)**

**m != 2k**

**a** é a constante multiplicadora

**c** é o incremento

**m** representa o módulo

**X0** é a semente

a = 5; c = 0; m = 7 e X0 = 4 (semente)

A sequência de inteiros aleatórios geradas por essa recorrência é: 4, 6, 2, 3, 1, 5...

O período desse gerador é **6**, logo **ρ = m-1**, pois **a = 5** é uma raiz primitiva de **m = 7** e **m** um número primo.

a = 5; c = 0; m = 7 e X0 = 7 (semente)

A sequência de inteiros aleatórios geradas por essa recorrência é: 7.

O período desse gerador é **1**, devido a escolha da semente **(7)** ser igual a **m**

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  x= x0 #semente  x1=np.array([x])  n=60  a=5 # constante multiplicadora  m=7 #representa o módulo  for i in range(n):  x=(a\*x)%m  x1=np.append(x1,x)  print(x1) |

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:

60 chamadas/10horas

**λ** é igual a **6**

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

= 0,00247

|  |
| --- |
| lambda1=6 #Número médio de requisições  N=1000000 #Número de amostras  value=0  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      av=np.append(av,a1)  print(av)  for poissvalue in av:      if poissvalue==value:          count=count+1  prob=count/N  print("a probabilidade e",prob) |

Uma imagem contendo Gráfico

Descrição gerada automaticamente

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

determinada hora.

0,84716 = 84,716%

|  |
| --- |
| lambda1=6 #Número médio de requisições  N=1000000 #Número de amostras  value=8  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      av=np.append(av,a1)  print(av)  for poissvalue in av:      if poissvalue==value:          count=count+1  prob=count/N  print("a probabilidade e",prob) |

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

**λ** é igual a **6**

1. A variância de C.

**λ** é igual a **6**

1. O desvio padrão de C

**=**  = 2,45

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

**q:** probabilidade de sucesso **q = 0,15**

**p:** probabilidade de sucesso **q = 1- p = 0,85**

**n** = 8

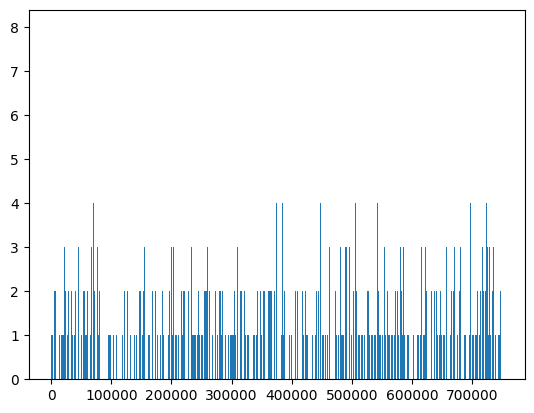
|  |
| --- |
| import numpy as np  q=0.15 #Probabilidade de acerto  n=8 #Número de tentativas  value=0  N=100000 #Número de amostras  c = q/(1-q)  av=np.array([])  count=0  x=np.random.uniform(0,1,N)  for ix in x:  i = 0  pr = pow((1 - q),n)  F = pr  while ix>=F:  pr = (c \* (n - i) / (i + 1))\* pr;  F = F + pr;  i = i + 1;  a1=i  av=np.append(av,a1)  print(av)  for binvalue in av:  if binvalue==value:  count=count+1  prob=count/N  print("a probabilidade e",prob)  ind=np.arange(N)  plt.bar(ind, av)  plt.show() |

1. não mais que 2 rejeitados?

1. pelo menos 6 rejeitados?

**= 0,00128**

- Traçar o histograma da variável analisada.



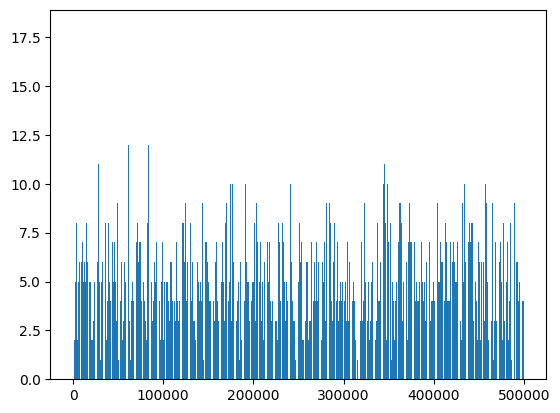
4) Se ocorrerem falhas de energia elétrica de acordo com uma distribuição de Poisson com uma média de 6 falhas a cada duas semanas, calcule a probabilidade de que haverá ao menos 2 falhas durante uma semana específica. Traçar o histograma da variável analisada.

6 Falhas/2 Semanas

**λ** é igual a **3**

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

0,04979 +0,14937 + 0,22381 = 0,57703



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

x = 4

= 0,1331

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  N=100000  lambda1=1/28  x=np.random.uniform(0,1,N)  value=4  Xexp=-np.log(x)/lambda1  count=0  print(Xexp)  for expvalue in Xexp:  if expvalue<=value:  count=count+1  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, 0.1)  fx=lambda1\*np.exp(-lambda1\*X)  plt.plot(X,fx)  plt.hist(Xexp,bins=100,density=True)  plt.show() |

**a probabilidade e 0.13296**

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 número 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?

PBb = 0,6

PBp = 0,4

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  p = 0.4 #Probabilidade de acerto  N = 100000  n = 6 #Número de tentativas  count = 0  X=np.random.geometric(p,N)  xgeom = p\*(1-p)\*\*(X-1)  print(xgeom)  for geomvalue in xgeom:  if geomvalue>=n:  count=count+1  prob=count/N  print("a probabilidade e",prob)  Probana=p\*(1-p)\*\*(n-1)  print("a probabilidade analítica e",Probana)  n = np.linspace(1, n, n+1)  fx = p\*(1-p)\*\*(n-1)  plt.plot(n,fx)  plt.hist(X,bins=10,density=True)  plt.show() |

|  |
| --- |
|  |

7) Utilizando o método da inversa gerar amostras para a distribuição

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  N=1000000  xfunction=np.array([])  u=np.random.uniform(0,1,N)  x=np.log((u\*((np.e\*\*2)-1))+1)  plt.hist(x,bins=1000,density=True)  plt.grid(True) |

|  |
| --- |
|  |

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

Plotar a pdf analítica e o histograma normalizado.

Logo:

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  n=100000  xfunction=np.array([])  #gx = 0.5 1/(a-b)funcao conhecida  #c = 3 maxf(x)/g(x)  for i in range(n):  x1=np.random.uniform(-1,1)  x2=np.random.uniform(0,1)  while (x2 >=(1.5\*(x1\*\*2))/1.5):  x1=np.random.uniform(-1,1)  x2=np.random.uniform(0,1)  xfunction = np.append(xfunction, x1)  #print(xfunction)  X=np.arange(-1, 1, 0.1)  fx=1.5\*(X\*\*2)  plt.hist(xfunction,bins=100,density=True)  plt.grid(True)  plt.plot(X,fx) |
|  |