

Palavras-chave: geração de números aleatórios, geração de variáveis aleatórias.

Consulte a informação das aulas Teórico-Práticas e responda às questões seguintes através de pequenos programas em Octave/Matlab:

1. Implemente uma função que gere um vector de números aleatórios com base no método da Congruência.

A função deve permitir controlar o seu comportamento através dos parâmetros necessários à definição da fórmula usada neste método e também permitir definir o comprimento do vector a gerar. Sugestão:
`function y = lcg (X0,a,c,m,N) .`

Utilizando a função:

- (a) gere um vector de comprimento 1000 usando parâmetros à sua escolha e visualize o seu histograma. Quantos números diferentes obteve? Sugestão: experimente a função `unique()`.
 - (b) gere um conjunto de números aleatórios entre 0 e 1 com base nos números gerados na alínea a). Visualize o histograma e conte novamente os números diferentes.
 - (c) repita as alíneas anteriores com os parâmetros utilizados na implementação incluída na biblioteca NAG (ver apresentação da TP).
2. Usando a função Matlab/Octave `rand()`, que permite de uma forma simples obter números aleatórios com características semelhantes aos que obtivemos anteriormente:
 - (a) Simule uma sequência de 10 experiências de Bernoulli com probabilidade de sucesso p . Represente sucesso por 1 ;
 - (b) Simule 15 lançamentos de 1 dado (honesto);
 - (c) Obtenha um conjunto de 20 números reais igualmente distribuídos entre -4 e 10;
 3. Crie uma função que gere um vector de números com uma distribuição de Bernoulli com parâmetro p .
Utilize o histograma (função `hist()`) e a estimativa das probabilidades com base num vector gerado para testar o seu funcionamento.
Exemplo de teste: `hist(Bernoulli(.3, 10000),(0:1)')`
 4. Crie uma função que gere um vector de números com uma distribuição Binomial, dados os parâmetros deste tipo de distribuição (n e p) e N (tamanho do vector a gerar).
Utilize o histograma e a estimativa das probabilidades com base num vector gerado para testar o seu funcionamento. Tente comparar as probabilidades obtidas com base no vector gerado com os valores teóricos.
 5. Crie uma função que gere um vector de números com uma distribuição discreta genérica definida pela sua função de massa de probabilidade (fmp).
A sua função deve receber como parâmetros de entrada dois vectores definindo a fmp, x_i e pX , assim como o número de valores a gerar.
Utilize o histograma e a estimativa das probabilidades com base num vector gerado para testar o seu funcionamento. Sugere-se que para o primeiro teste se use uma fmp de um dado nada honesto em que a probabilidade de sair 6 é bem maior do que a probabilidade de saírem as outras faces, não existindo diferenças de probabilidade entre as outras faces.

6. Utilize o método da transformação para implementar uma função Matlab/Octave capaz de gerar números com uma distribuição exponencial. Use exponencial para o nome dessa função.

Teste a função, usando, por exemplo `hist(exponencial(10000, .5), 20)`.

7. Crie uma função que gere um vector de números com a distribuição normal normalizada utilizando o Método de Box-Müller.

Usando esta função como base, gere as classificações de uma turma de 30 alunos por forma a terem média 14 e variância 2.

Confirme os resultados com a função Matlab `randn()`. Consulte a informação sobre esta função.

8. Repita o exercício anterior criando uma nova versão da função baseada no método da rejeição. Assuma que os valores de $f_X(x)$ da função Gaussiana normalizada é próxima de zero fora do intervalo $(-5,5)$.

Use o histograma para verificar as características principais (forma, média e dispersão em torno da média).

9. (TPC) Crie uma função que permita simular a retirada aleatória de um subconjunto de bolas de uma urna, sem reposição.

Aplique a função ao caso do Totoloto.