**NOME: GUSTAVO HAMMERSCHMIDT**

**Geração de variáveis aleatórias**

A inversa da função de distribuição acumulada pode ser utilizada na geração de números aleatórios de acordo com certa distribuição de probabilidade.

O MatLab possui funções que que geram números aleatórios distribuídos de acordo com as distribuições de probabilidade.

1. **Geração de números aleatórios de acordo com a distribuição de Poisson**
   1. **Uso da função**

Para gerar um vetor com 10 posições contendo números aleatórios distribuídos de acordo com a distribuição de Poisson com média 4, usando a função do MatLab usamos o comando

poissrnd(4, 1, 10)

No ambiente de aprendizado foi colocada a função rndpoiss, programada de acordo com o algoritmo da inversa da função de distribuição acumulada.

Para gerar um vetor com 10 posições contendo números aleatórios distribuídos de acordo com a distribuição de Poisson com parâmetro 4, usando a função rndpoiss usamos o comando

rndpoiss (4, 10)

Execute esses comandos, um após o outro. É possível, a partir dos resultados, saber se as funções estão gerando os valores corretamente? Responda sim ou não.

É possível saber que as funções estão gerando os valores corretos pois, quando você as testa, não é possível ter uma conclusão matemática de como os resultados foram obtidos, ou seja, são randômicas.

* 1. **Verificação poissrnd**

A maneira de verificar se os números estão de acordo com certa distribuição é fazer um teste de aderência (procedimento estatístico). Mas podemos também comparar visualmente os histogramas para um número muito grande de amostras, que devem ficar muito semelhantes.

A função pdfpoiss usa a função poissrnd do MatLab para gerar um vetor com nElm valores, e usa a função hist para colocar as frequências de ocorrência na variável y1. Divide os valores das frequências por nElm para calcular a frequência relativa. Depois usa a própria função de probabilidade (poisspdf) para gerar os valores de y2, e em seguida plota as duas curvas.

function [ ] = pdfpoiss(L, nElm)

aux = poissrnd(L,1,nElm);

x = 0:max(aux);

[y1 x] = hist(aux, x);

y1 = y1/nElm;

y2 = poisspdf(x,L);

plot(x,y1,'-blue', x,y2,'-red')

end

Execute a função pdfpoiss para L = 4 (parâmetro da distribuição) com 10000 valores (nElm).

Copie o gráfico aqui.

Uma imagem contendo mapa, céu, texto

Descrição gerada automaticamente

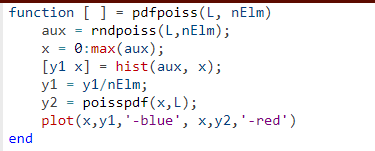
É possível, a partir do gráfico, saber se a função poissrnd está gerando os valores de números aleatórios corretamente? Por quê?

Teoricamente falando, ele está gerando valores corretos, pois a curva das amostras não acompanha a curva da função para todos os valores. Contudo, sabendo que, para essa função, a amostra de 10000 elementos é pequena, a curva das amostras se aproxima bastante da curva da função de Poisson, ou seja, não é tão aleatória e não está gerando valores tão aleatórios. No entanto, as duas curvas não são iguais e, por isso, pode-se dizer que ela é uma função “aleatória”, pois, gera valores randômicos.

* 1. **Verificação rndpoiss**

Para verificar se a função rndpoiss foi programada corretamente, podemos usar a mesma função pdfpoiss, alterando a função de geração de números aleatórios usada. No arquivo está sendo chamada a função poissrnd. Basta substituir a chamada por rndpoiss, alterando para apenas 2 parâmetros como mostrado acima.

Cole o código da função alterada aqui.



Execute a função pdfpoiss para L = 4 (parâmetro da distribuição) com 10000 valores (nElm).

Copie o gráfico aqui.

Uma imagem contendo mapa

Descrição gerada automaticamente

É possível, a partir do gráfico, saber se a função rndpoiss está gerando os valores de números aleatórios corretamente? Por quê?

Teoricamente falando, ele está gerando valores corretos, pois a curva das amostras não acompanha a curva da função para todos os valores. Contudo, sabendo que, para essa função, a amostra de 10000 elementos é pequena, a curva das amostras se aproxima bastante da curva da função de Poisson, ou seja, não é tão aleatória e não está gerando valores tão aleatórios. No entanto, as duas curvas não são iguais e, por isso, pode-se dizer que ela é uma função “aleatória”, pois, gera valores randômicos.

1. **Geração de números aleatórios de acordo com a distribuição exponencial**
   1. **Uso da função**

Para gerar um vetor com 10 posições contendo números aleatórios distribuídos de acordo com a distribuição de exponencial com média 4, usando a função do MatLab usamos o comando

exprnd(4, 1, 10)

No ambiente de aprendizado foi colocada a função rndexp, programada de acordo com o algoritmo da inversa da função de distribuição acumulada.

Para gerar um vetor com 10 posições contendo números aleatórios distribuídos de acordo com a distribuição de Poisson com parâmetro 4, usando a função rndexp usamos o comando

rndexp (4, 10)

Execute esses comandos, um após o outro. É possível, a partir dos resultados, saber se as funções estão gerando os valores corretamente? Responda sim ou não.

É possível saber que as funções estão gerando os valores corretos pois, quando você as testa, não é possível ter uma conclusão matemática de como os resultados foram obtidos, ou seja, são randômicas.

* 1. **Verificação exprnd**

A maneira de verificar se os números estão de acordo com certa distribuição é fazer um teste de aderência (procedimento estatístico). Mas podemos também comparar visualmente os histogramas para um número muito grande de amostras, que devem ficar muito semelhantes.

A função pdfexp usa a função exprnd do MatLab para gerar um vetor com nElm valores, e usa a função hist para colocar as frequências de ocorrência na variável y1. Antes de gerar o histograma, devemos gerar os pontos centrais onde as frequências vão ser acumuladas. Como a função é contínua, na linha dois calculamos o passo para gerar os centros dos histrogramas (linha 3), como sendo o maior número gerado, menos o menor, divididos por 1000. Na linha 4 a função hist. Acumula as frequências em cada intervalo e na linha 5 divide os valores das frequências por nElm para calcular a frequência relativa. Depois usa a própria função de probabilidade (exppdf) para gerar os valores de y2, e em seguida plota as duas curvas.

function [ ] = pdfexp(mu, nElm)

e=exprnd(mu,1,nElm);

step = (max(e)-min(e))/1000;

r=min(e):step:max(e);

[y1, x] = hist(e,r);

y1 = y1/nElm\*(1/step); %multiplica por 1/step porque foi

%distribuído no histograma a cada step

y2=exppdf(x,mu);

plot(x, y1, '-blue', x, y2, '-red');

end

Execute a função pdfexp para L = 4 (parâmetro da distribuição) com 10000 valores (nElm). Copie o gráfico aqui.

Uma imagem contendo mapa

Descrição gerada automaticamente

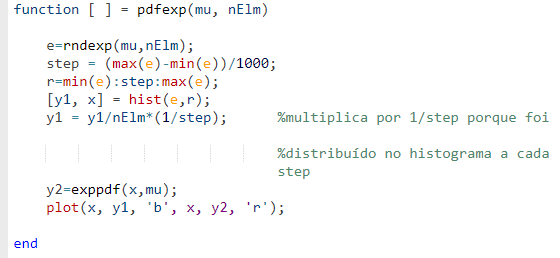
É possível, a partir do gráfico, saber se a função exprnd está gerando os valores de números aleatórios corretamente? Por quê?

Sim, pois a curva das amostras(azul) não consegue acompanhar a curva da função vermelha, o que comprova que, para essa amostra, a função é randômica.

* 1. **Verificação rndexp**

Para verificar se a função rndexp foi programada corretamente, podemos usar a mesma função pdfexp, alterando a função de geração de números aleatórios usada. No arquivo está sendo chamada a função exprnd. Basta substituir a chamada por rndexp, alterando para apenas 2 parâmetros como mostrado acima.

Cole o código da função alterada aqui.



Execute a função pdfexp para L = 4 (parâmetro da distribuição) com 10000 valores (nElm). Copie o gráfico aqui.

Uma imagem contendo mapa

Descrição gerada automaticamente

É possível, a partir do gráfico, saber se a função rndexp está gerando os valores de números aleatórios corretamente? Por quê?

Sim, pois a curva das amostras(azul) não consegue acompanhar a curva da função vermelha, o que comprova que, para essa amostra, a função é randômica.