

TEFE 2017

Suplemento da 1ª aula do tópico 2 versão Octave e Matlab

Esse suplemento aborda geração de números pseudo-aleatórios com distribuição uniforme, o uso do desvio-padrão para determinar a incerteza dos resultados de simulações, o uso de condições lógicas em matrizes e a criação de funções.

Geração de pseudo-aleatórios com distribuição uniforme ¶

A geração de números pseudo-aleatórios com distribuição uniforme entre 0 e 1 é feita com o comando `rand`. Note que `randn` gera números pseudo-aleatórios com distribuição gaussiana de média 0 e desvio-padrão 1.

>> **% versão 1 - calcular cada elemento do vetor x dentro de um loop**

```
>> N = 1000;
>> L = 1;
>> x = zeros(N,1);
>> for i=1:N
x(i) = L*( rand - 0.5 );
end
```

>> **% versão 2 - calcular todo o vetor x ao mesmo tempo**

```
>> N = 1000;
>> L = 1;
>> x = L*( rand(N,1) - 0.5 );
```

>> **% desvio-padrão calculado a partir do vetor x**

```
>> sx = std( x )
sx = 0.29450
```

>> **% desvio-padrão de uma função densidade de probabilidade uniforme entre -L/2 e L/2**

```
>> s0 = L/sqrt(12)
s0 = 0.28868
```

>> **% média do vetor x**

```
>> xm = mean( x )
xm = -0.0011897
```

>> **% desvio-padrão da média d**

```
>> sxm = sx / sqrt( N )
sxm = 0.0093128
```

>> **% para ver o histograma dos dados gerados distribuídos em 20 canais**

```
>> figure
>> hist( x, 20 )
```

Calculando a incerteza do desvio padrão obtido nas simulações ¶

```
>> % Vamos fazer nREP=50 repetições de conjuntos com N=1000 dados
>> clear
>> N=1000;
>> L = 1;
>> nREP = 50;
>> x = L*( rand(N,nREP) - 0.5 );

>> % Em seguida, vamos calcular os desvios-padrões, sx, de cada um desses conjuntos
>> % e depois o desvio-padrão dos valores de sx, que é a incerteza de cada valor de sx.
>> sx = std( x );

>> fprintf( 'N=%d, nREP=%d, L=%g\n', N, nREP, L )
N=1000, nREP=50, L=1
>> fprintf('Desvio-padrao dos desvios-padroes: %g\n', std(sx) )
Desvio-padrao dos desvios-padroes: 0.00399947

>> fprintf('Media dos desvios padroes: %g\n', mean(sx) )
Media dos desvios padroes: 0.288976
>> fprintf('Incerteza da media dos desvios-padroes: %g\n', std(sx)/sqrt(nREP) )
Incerteza da media dos desvios-padroes: 0.00056561
```

Contagem de elementos dado uma condição ¶

```
>> N = 1000;
>> L = 1;
>> x = L*( rand(N,1) - 0.5 );

>> % valores_dentro é o número de valores de x com módulo menor ou igual a L/sqrt(12):
>> valores_dentro = 0;
>> for i=1:N
if abs(x(i)) <= L/sqrt(12)
    valores_dentro = valores_dentro + 1;
end
end

>> % valores_dentro também pode ser calculado pela soma de um vetor em que a condição
>> % de verdadeiro ocorre para cada valor de x que tenha módulo menor ou igual a L/sqrt(12),
>> % pois cada item verdadeiro contribui em 1 para a soma, e cada falso contribui em 0:
>> valores_dentro = sum( abs(x)<=L/sqrt(12) )
valores_dentro = 565

>> % A frequência relativa de valores de x nessas condições pode ser calculada por:
>> mean( abs(x)<=L/sqrt(12) )
ans = 0.56500
```

Funções ¶

É possível fazer funções com argumentos de entrada e de saída. No exemplo abaixo, a função **calcular** tem que ser criada no editor e salva com o nome "calcular.m". Se necessário, é possível controlar o número de argumentos de entrada (**nargin**) e de saída (**nargout**) com que a função foi chamada.

No Editor: (salvar com o nome de "calcular.m")

```
function [media_sx, s_media_sx] = calcular( N, nREP, L )

if nargin==0
    N = 1000;
    nREP = 50;
    L = 1;
end

x = L*( rand( N, nREP ) - 0.5 );
sx = std( x );

media_sx = mean( sx );
s_media_sx = std( sx ) / sqrt( nREP );

if nargout==0
    fprintf( 'N=%d, nREP=%d, L=%g\n', N, nREP, L )
    fprintf( 'media de sx = %g com inc. %g\n', media_sx, s_media_sx )
    clear media_sx
end
```

Na Janela de Comandos:

```
>> % exemplos de utilizacao:
>> % 1 - sem especificar parâmetros de entrada (usa os valores padrões definidos quando nargin==0) nem de saída
>> calcular
N=1000, nREP=50, L=1
media de sx = 0.288096 com inc. 0.000613504

>> % 2 - especificando os parâmetros de entrada, mas sem especificar variáveis para a saída
>> calcular(2000,100,1)
N=2000, nREP=100, L=1
media de sx = 0.289013 com inc. 0.000307368

>> % 3 - sem especificar parâmetros de entrada, mas registrando os valores obtidos na função em variáveis
>> [m_sx, s_m_sx]=calcular
m_sx = 0.28919
s_m_sx = 5.7859e-004
```

Observação sobre parâmetros de entrada no Octave ¶

No Octave (assim como em Python), é possível fazer funções com argumentos opcionais ao igualar ele a um valor padrão após o nome da variável na definição da função. No exemplo abaixo, a função **calcular** tem com argumentos padrões **N=1000**, **nREP=50** e **L=1**, mas estes valores podem ser escolhidos na chamada na função.

No Editor: (salvar com o nome de “calcular2.m”)

```
function [media_sx, s_media_sx] = calcular2( N=1000, nREP=50, L=1 )
```

```
x = L*( rand( N, nREP ) - 0.5 );  
sx = std( x );
```

```
media_sx = mean( sx );  
s_media_sx = std( sx ) / sqrt( nREP );
```

```
if nargin==0  
  fprintf( 'N=%d, nREP=%d, L=%g\n', N, nREP, L )  
  fprintf( 'media de sx = %g com inc. %g\n', media_sx, s_media_sx )  
  clear media_sx  
end
```

Na Janela de Comandos:

```
>> % exemplos de utilizacao:
```

```
>> % 1 - sem especificar parâmetros de entrada
```

```
>> calcular2
```

```
N=1000, nREP=50, L=1
```

```
media de sx = 0.28867 com inc. 0.000607172
```

```
>> % 2 - especificando os parâmetros de entrada
```

```
>> calcular2(2000,100,1)
```

```
N=2000, nREP=100, L=1
```

```
media de sx = 0.288434 com inc. 0.000298538
```

```
>> % 3 - Pode-se especificar apenas parte dos parâmetros de entrada (em ordem)
```

```
>> calcular2( 100 )
```

```
N=100, nREP=50, L=1
```

```
media de sx = 0.289973 com inc. 0.00173106
```

```
>> % 4 - Para especificar parâmetros de entrada fora de ordem, escreva : nos parâmetros precedentes
```

```
>> calcular2( : , 100 )
```

```
N=1000, nREP=100, L=1
```

```
media de sx = 0.289501 com inc. 0.000421531
```