

```

%% Distribuição Gaussiana

%% Entradas

mx = 10 ; % média de x
my = 20 ; % média de y
vx = 1; % variância de x
vy = 1; % variância de y

%% População

X = randn(2000,1);
Y = randn(2000,1);

%% Tratamento para distribuição

xg = (sqrt(vx).*X+mx); % Aplicação da Gaussiana em x
yg = (sqrt(vy).*Y+my); % Aplicação da Gaussiana em y

%% Criar linhas de referência

x1 = [max(xg),min(xg)];
y1 = [max(yg),min(yg)];

v1 = [mx,mx];
v2 = [floor(y1(2)),ceil(y1(1))];

h1 = [floor(x1(2)),ceil(x1(1))];
h2 = [my,my];

%% Plotagem

figure()
title ('Distribuição Gaussiana');
hold on
plot(xg, yg, '*');
plot(h1, h2, 'r', 'LineWidth',2);
plot(v1, v2, 'r', 'LineWidth',2);
axis([h1(1) h1(2) v2(1) v2(2)]);

%% Distribuição Uniforme com desvio

%%

Xu = rand(2000,1);
Yu = rand(2000,1);

```

```

%%

amin = (2*mx - sqrt(12*vx))/2;
amax = (2*mx + sqrt(12*vx))/2;
bmin = (2*my - sqrt(12*vy))/2;
bmax = (2*my + sqrt(12*vy))/2;

xu = amin + (amax - amin).*Xu;
yu = bmin + (bmax - bmin).*Yu;

%% Criar linhas de referência

x1u = [max(xu),min(xu)];
y1u = [max(yu),min(yu)];

v1u = [mx,mx];
v2u = [floor(y1u(2)),ceil(y1u(1))];

h1u = [floor(x1u(2)),ceil(x1u(1))];
h2u = [my,my];

%% Distribuição Uniforme sem desvio

x2= mx-3.5*vx/2+ 3.5*rand(1,2000)*vx;
y2= my-3.5*vy/2+ 3.5*vy*rand(1,2000);

%% Plotagem

figure()
title ('Distribuição Uniforme');
hold on
% plot(xu, yu, '*');
plot(x2, y2, '*');
plot(h1u, h2u, 'r' , 'LineWidth',2);
plot(v1u, v2u, 'r' , 'LineWidth',2);
axis([h1u(1) h1u(2) v2u(1) v2u(2)]);

%% Parte 2 - Distorção

clear all
close all
clc

```

```

%% Criação de duas séries temporais

% x1 = timeseries(rand(10,1));
% x2 = timeseries(rand(10,1));

t = [0:0.01:20];
x1 = sin(t);
x2 = 4*sin(t);

subplot(2,1,1)
plot(x1,'LineWidth',2)
axis([0,2000,-4,4])
title ('sin(t) ');
subplot(2,1,2)
plot(x2,'LineWidth',2)
axis([0,2000,-4,4])
title ('4*sin(t) ');

%% Adicionar dados da série e uma variável

% S1 = x1.data;
% S2 = x2.data;

S1 = x1;
S2 = x2;

%% Aplicando a fórmula para cálculo de distorção

soma_num = 0;
soma_den = 0;

for i = 1 : length (S1)

    num = (S1(i)-S2(i)).^2;
    soma_num = soma_num + num;

    den = (S1(i)).^2;
    soma_den = soma_den + den;

end

media_num = soma_num/ length(S1);
media_den = soma_den/ length(S1);

dxy = 10*log10(soma_num/soma_den)

```