```
function mkv = MarkovProcess(rho,sigma,r,N,mu)
% Autor: Bruno Tebaldi Q Barbosa
%
% Modela um processo AR(1) por um modelo de cadeia de markov.
%  y(t)= (1-rho)*mu + rho*y(t-1) + e(t)
%
% Imputs:
% rho: Fator de correlação do AR(1)
% sigma: Variancia dos erros no AR(1)
% r: Quantidade de desvios padores que os estados mais distantes do
%    processo de Markov deve ter
% N: Quantidade total de estados no processo de Markov.
% mu: Média de longo prazo do processo AR(1)


% Validade\Fill in unset optional values.
switch nargin
    case {0,1,2,3}
        error('função necessita pelo menos de 4 argumentos');
    case 4
        mu=0;
end

% Valida rho
if abs(rho) >= 1
    error('processo ar não estacionario. (rho = %f)', rho);
end

% Discretizacao do espaco de estado
dpy = sqrt(sigma^2/(1-rho^2));    % desvio padrao de y_t (não condicional)
z_N = mu + r*dpy;                 % limite superior
z_1 = mu - r*dpy;                 % limite inferior

d = (z_N-z_1)/(N-1);              % tamanho do intervalo
z = z_1:d:z_N;                    % grid de estados

% Inicia a matriz de tranzicao
PI = nan(N);

% Calculate the transition matrix - see Tauchen
for lin=1:N
    for col=2:N-1
        PI(lin,col)= normcdf(z(col)+d/2-rho*z(lin) -mu*(1-rho), 0, sigma)...
            - normcdf(z(col)-d/2-rho*z(lin) -mu*(1-rho), 0, sigma);
    end

    % Casos de j={1 ,N}
    PI(lin,1) = normcdf(z(1)+d/2-rho*z(lin)-mu*(1-rho),0,sigma);
    PI(lin,N) = 1 - normcdf(z(N)-d/2-rho*z(lin)-mu*(1-rho),0,sigma);
end
```

```
% Verifica se as probabilidade somam um por linha
if sum(PI,2) ~= ones(N,1)
    % find rows not adding up to one
    rowNotSumTo1 = find(sum(PI,2) < 1);  % find rows not adding up to one
    fprintf('Error in transition matrix\n');
    fprintf('row %d does not sum to one\n', rowNotSumTo1);
end

% Cria structure de resposta
mkv.AR.form = 'y(t)= (1-rho)*mu + rho*y(t-1) + e(t)';
mkv.AR.mu = mu;
mkv.AR.rho = rho;
mkv.AR.sigma2 = sigma;
mkv.AR.sigma2_y = dpy^2;
mkv.TransitionMatrix = PI;
mkv.StateVector = z;
mkv.QtdStates = N;
mkv.d = d;
mkv.StateBorder = z(1:end-1) + d/2;

end % end of function MarkovProcess
```