```
function [betamat,sig2mat]= ...
    gs_normal_independent(X,y,n,k,r1,r2,mu0,V0,betadraw,v0,tau0,sig2draw);

R = r1+r2;
betamat=zeros(k,r2); %will collect draws of beta;
sig2mat=zeros(1,r2); %will collect draws of sig2;

thoucount=1000; % shows every 1000th iteration in the command window
% you may want to change this to 100 for slower applications

% start main loop
i=1;
for i=1:R

    %%%%%%%%%%%%%
    % draw betas
    %%%%%%%%%%%%%
    V1=inv(inv(V0)+(1/sig2draw)*X'*X);
    mu1=V1*(inv(V0)*mu0 +(1/sig2draw)*X'*y);
    betadraw=mvnrnd(mu1,V1)';
    if i>r1
       betamat(:,i-r1)= betadraw;
    end

    %%%%%%%%%%%%%%%%%%
    % draw sig2
    %%%%%%%%%%%%%%%%%%
    v1=(n+2*v0)/2;
    tau1=(1/2)*((y-X*betadraw)'*(y-X*betadraw)+2*tau0);
    sig2draw=1/gamrnd(v1,1/tau1);
    % Matlab defines the ig scale as 1/tau, thus the inversion for the last
    % term
    if i>r1
       sig2mat(:,i-r1)=sig2draw;
    end


    if i== thoucount
        i
        thoucount=thoucount+1000;
    end


 end
```

Published with MATLAB® R2022b