```
function [beta1mat,beta2mat,beta3mat,sig2mat]=...

 gs_normal_blocked_keepall(X,X1,X2,X3,y,n,k,k1,k2,k3,r1,r2,mu01,V01,beta1draw,...
    mu02,V02,beta2draw,mu03,V03,beta3draw,v0,tau0,sig2draw);

R = r1+r2;
beta1mat=zeros(k1,R);
beta2mat=zeros(k2,R);
beta3mat=zeros(k3,R);
sig2mat=zeros(1,R); %will collect draws of sig2;

thoucount=1000;
% start main loop
i=1;
for i=1:R

    %%%%%%%%%%%%
    % draw beta1's
    %%%%%%%%%%%%
    V1=inv(inv(V01)+(1/sig2draw)*X1'*X1);
    mu1=V1*(inv(V01)*mu01 +(1/sig2draw)*X1'*(y-X2*beta2draw-X3*beta3draw));
    beta1draw=mvnrnd(mu1,V1)';
    beta1mat(:,i)= beta1draw;


   %%%%%%%%%%%%
    % draw beta2's
    %%%%%%%%%%%%
    V1=inv(inv(V02)+(1/sig2draw)*X2'*X2);
    mu1=V1*(inv(V02)*mu02 +(1/sig2draw)*X2'*(y-X1*beta1draw-X3*beta3draw));
    beta2draw=mvnrnd(mu1,V1)';

    beta2mat(:,i)= beta2draw;


    %%%%%%%%%%%%%%
    % draw beta3's
    %%%%%%%%%%%%%%
    V1=inv(inv(V03)+(1/sig2draw)*X3'*X3);
    mu1=V1*(inv(V03)*mu03 +(1/sig2draw)*X3'*(y-X1*beta1draw-X2*beta2draw));
    beta3draw=mvnrnd(mu1,V1)';

    beta3mat(:,i)= beta3draw;



    betadraw=[beta1draw;beta2draw;beta3draw];

    %%%%%%%%%%%%%%%%%
    % 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


    sig2mat(:,i)=sig2draw;



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


end
```

Published with MATLAB® R2022b