

---

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Examine behavior of the MC under different starting values
% and sample sizes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Set random number generator, start stop watch, open output file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rand('state',37); % set arbitrary seed for uniform draws
randn('state',37); % set arbitrary seed for normal draws

% Load & prepare data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%first, let's use all 10,000 observations
load c:\klaus\AAEC6564\mlab\worksp\mod2_sim_data;
n=length(y);
k=size(X,2);

% general elements
r1=2000; % burn-in
r2=10000; % keepers
R=r1+r2;

% priors for beta
mu0=zeros(k,1); %diffuse prior for mean of betas
V0=eye(k)*100; % diffuse prior for varcov of beta

%priors for sig2
v0=1/2;
tau0=1/2; % diffuse prior shape and scale

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% starting values, set 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% generic OLS
bols=inv(X'*X)*X'*y;
res=y-X*bols;
s2=(res'*res)/(n-k);
betadraw=bols; %use OLS values as starting draws for Gibbs Sampler
% note: not needed here, since we are starting the GS with draws of beta
sig2draw=s2; % use OLS variance as starting draw for Gibbs Sampler

% run GS
[betamat,sig2mat]=...
gs_normal_independent_keepall(X,y,n,k,r1,r2,mu0,V0,betadraw,v0,tau0,sig2draw);
sig2chain1=[sig2mat];
b3chain1=[betamat(end,:)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% starting values, set 2 - completely arbitrary
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% generic OLS
sig2draw=3000; % extremely far from posterior mean...

```

---

```

% run GS
[betamat,sig2mat]=...

gs_normal_independent_keepall(X,y,n,k,r1,r2,mu0,V0,betadraw,v0,tau0,sig2draw);
sig2chain2=[sig2mat];
b3chain2=[betamat(end,:)];

figure(1);
x=(1:1:R);
plot(x,b3chain1,'k',x,b3chain2,'b',x,btrue(3)*ones(R,1),'r');
axis([min(x) max(x) -5 5]);
title('convergence plots for beta3, n=10,000');
xlabel('iteration');
ylabel('drawn values')
legend('chain1','chain2','true value')
hold on;
plot(1,b3chain2(1),'o');
hold off;

% note: in the figure window, you can use the ZOOM tool to take a closer
% look at individual draws, or use the edit/axis property feature to change
% the limits of the y-axis

%Save figure in Matlab format
to = 'c:\klaus\AAEC6564\mlab\figures\mod2_convergence_plots_fig1'; %designate
destination
saveas(gcf,to);%"gcf" means "get current figure"

figure(2);
x=(1:1:R);
plot(x,sig2chain1,'k',x,sig2chain2,'b',x,sig2true*ones(R,1),'r');
axis([min(x) max(x) 0 20]);
title('convergence plots for sig2, n=10,000');
xlabel('iteration');
ylabel('drawn values')
legend('chain1','chain2','true value')
hold on;
plot(1,sig2chain2(1),'o');
hold off;

to = 'c:\klaus\AAEC6564\mlab\figures\mod2_convergence_plots_fig2'; %designate
destination
saveas(gcf,to);%"gcf" means "get current figure"

% note: the sample variance for the simulated error is 1.4122

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Now let's truncate our data to 100 observations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

y=y(1:100);
X=X(1:100,:);

```

---

---

```

n=length(y);
k=size(X,2);

% general elements
r1=2000; % burn-in
r2=10000; % keepers
R=r1+r2;

% priors for beta
mu0=zeros(k,1); %diffuse prior for mean of betas
V0=eye(k)*100; % diffuse prior for varcov of beta

%priors for sig2
v0=1/2;
tau0=1/2; % diffuse prior shape and scale

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% starting values, set 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% generic OLS
bols=inv(X'*X)*X'*y;
res=y-X*bols;
s2=(res'*res)/(n-k);
betadraw=bols; %use OLS values as starting draws for Gibbs Sampler
sig2draw=s2; % use OLS variance as starting draw for Gibbs Sampler

% run GS
[betamat,sig2mat]=...

gs_normal_independent_keepall(X,y,n,k,r1,r2,mu0,V0,betadraw,v0,tau0,sig2draw);
sig2chain1=[sig2mat];
b3chain1=[betamat(end,:)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% starting values, set 2 - completely arbitrary
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% generic OLS
sig2draw=3000; % extremely far from posterior mean...

% run GS
[betamat,sig2mat]=...

gs_normal_independent_keepall(X,y,n,k,r1,r2,mu0,V0,betadraw,v0,tau0,sig2draw);
sig2chain2=[sig2mat];
b3chain2=[betamat(end,:)];

figure(3);
x=(1:1:R);
plot(x,b3chain1,'k',x,b3chain2,'b',x,btrue(3)*ones(R,1),'r');
axis([min(x) max(x) -5 5]);
title('convergence plots for beta3, n=100');
xlabel('iteration');
ylabel('drawn values');
legend('chain1','chain2','true value')

```

---

---

```

hold on;
plot(1,b3chain2(1),'o');
hold off;

to = 'c:\klaus\AAEC6564\mlab\figures\mod2_convergence_plots_fig3'; %designate
destination
saveas(gcf,to);%"gcf" means "get current figure"

figure(4);
x=(1:1:R);
plot(x,sig2chain1,'k',x,sig2chain2,'b',x,sig2true*ones(R,1),'r');
axis([min(x) max(x) 0 20]);
title('convergence plots for sig2, n=100');
xlabel('iteration');
ylabel('drawn values')
legend('chain1','chain2','true value')
hold on;
plot(1,sig2chain2(1),'o');
hold off;

to = 'c:\klaus\AAEC6564\mlab\figures\mod2_convergence_plots_fig4'; %designate
destination
saveas(gcf,to);%"gcf" means "get current figure"

% note: the sample variance for the simulated error is 1.4122

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Now let's truncate our data to 10 observations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

y=y(1:10);
X=X(1:10,:);
n=length(y);
k=size(X,2);

% general elements
r1=2000; % burn-in
r2=10000; % keepers
R=r1+r2;

% priors for beta
mu0=zeros(k,1); %diffuse prior for mean of betas
V0=eye(k)*100; % diffuse prior for varcov of beta

%priors for sig2
v0=1/2;
tau0=1/2; % diffuse prior shape and scale

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% starting values, set 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% generic OLS

```

---

---

```

bols=inv(X'*X)*X'*y;
res=y-X*bols;
s2=(res'*res)/(n-k);
betadraw=bols; %use OLS values as starting draws for Gibbs Sampler
sig2draw=s2; % use OLS variance as starting draw for Gibbs Sampler

% run GS
[betamat,sig2mat]=...

gs_normal_independent_keepall(X,y,n,k,r1,r2,mu0,V0,betadraw,v0,tau0,sig2draw);
sig2chain1=[sig2mat];
b3chain1=[betamat(end,:)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% starting values, set 2 - completely arbitrary
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% generic OLS

sig2draw=3000; %
% run GS
[betamat,sig2mat]=...

gs_normal_independent_keepall(X,y,n,k,r1,r2,mu0,V0,betadraw,v0,tau0,sig2draw);
sig2chain2=[sig2mat];
b3chain2=[betamat(end,:)];

figure(5);
x=(1:1:R);
plot(x,b3chain1,'k',x,b3chain2,'b',x,btrue(3)*ones(R,1),'r');
axis([min(x) max(x) -5 5]);
title('convergence plots for beta3, n=10');
xlabel('iteration');
ylabel('drawn values');
legend('chain1','chain2','true value');
hold on;
plot(1,b3chain2(1),'o');
hold off;

to = 'c:\klaus\AAEC6564\mlab\figures\mod2_convergence_plots_fig5'; %designate
destination
saveas(gcf,to);%"gcf" means "get current figure"

figure(6);
x=(1:1:R);
plot(x,sig2chain1,'k',x,sig2chain2,'b',x,sig2true*ones(R,1),'r');
axis([min(x) max(x) 0 20]);
title('convergence plots for sig2, n=10');
xlabel('iteration');
ylabel('drawn values');
legend('chain1','chain2','true value');
hold on;
plot(1,sig2chain2(1),'o');
hold off;

```

---

---

```
to = 'c:\klaus\AAEC6564\mlab\figures\mod2_convergence_plots_fig6'; %designate
destination
saveas(gcf,to);%"gcf" means "get current figure"
```

*Published with MATLAB® R2022b*