Solutions to Problem Set 3

1. The variance is $\begin{pmatrix} 4.42 & 1.88 \\ 1.88 & 1.64 \end{pmatrix}$ and the first autocovariance is $\begin{pmatrix} 2.2 & 1.1 \\ 0.8 & 0.8 \end{pmatrix}$ in both cases. Also, the higher order autocovariances are zeros in both cases. The representation for x_t is not fundamental because its AR form is

$$x_t = \begin{pmatrix} 1.1 & 0 \\ 0 & 0.8 \end{pmatrix} x_{t-1} + \begin{pmatrix} 1.1 & 0 \\ 0 & 0.8 \end{pmatrix}^2 x_{t-2} + \begin{pmatrix} 1.1 & 0 \\ 0 & 0.8 \end{pmatrix}^3 x_{t-2} ... + e_t$$

which does not converge. Meanwhile, the other representation is fundamental.

2. For the other percentile bootstrap, the coverage is 34.2 percent (the bias-adjusted bootstrap gives higher coverage). Here is the program for simulating the coverage of the plain bootstrap:

```
replics=1000; bigt=100;  %Number of replications and sample size
phi=[0.97 0; 0.5 0.5];
sigs=[1 0.3; 0.3 1];
a=(phi^12)*(chol(sigs)');
trueimp=a(2,1);
for imc=1:replics;
    imc
    u=randn(100,2)*chol(sigs);
    y(:,1)=u(1,:)';
    for t=2:bigt; y(:,t)=(phi*y(:,t-1))+u(t,:)'; end;
    ds=boot(y',1,2,12);
    ci(imc,:)=[ds(12,13) ds(12,487)];
end;
g1=ci(:,1)<trueimp; g2=ci(:,2)>trueimp;
disp(mean(g1+g2==2));
```

This calls the bootstrap program:

```
function ds=boot(y,i,j,kmax);
bigt=size(y,1);
ndraws=500; %Number of Bootstrap Draws
%----------------------------------------%
x=[ones(bigt-1,1) y(1:bigt-1,:)];
phi=y(2:bigt,:)'*x*inv(x'*x);
res=y(2:bigt,:)-(x*(phi'));
sigs=(res'*res)/(bigt-1);
for it=1:ndraws;
yb=zeros(bigt,2); yb(1,:)=y(ceil(rand(1,1)*bigt),:);
s=ceil(rand(bigt-1,1)*(bigt-1));
resb=res(s,:);
for t=2:bigt;
yb(t,:)=phi(:,1)'+(phi(:,2:3)*(yb(t-1,:)'))'+resb(t-1,:);
end;
x=[ones(bigt-1,1) yb(1:bigt-1,:)];
phiboot=yb(2:bigt,:)'*x*inv(x'*x);
```

```
sigsboot=((yb(2:bigt,:)-(x*(phiboot')))'*(yb(2:bigt,:)-
(x*(phiboot'))))/(bigt-1);
for k=1:kmax;
dsmat=(phiboot(:,2:3)^k)*chol(sigsboot)';
ds(k,it)=dsmat(j,i);
end;
end; %close it=1:ndraws loop
ds=sort(ds')';
```

3. Here is the main program.

```
m=xlsread('VARdata.xls','Sheet1','B3:K198');
sigs=arvars(m);
lambda=0.2;
vcovminn=kron(lambda*lambda*[1:1:4].^-2,kron(1./sigs,sigs')');
vcovminnx=[100000*ones(10,1) vcovminn];
vcovminnx=diag(reshape(vcovminnx',410,1));
minnesota(m,[1;3],4,4,zeros(410,1),vcovminnx)
```

It calls the following two programs:

- The first program estimates the variance of an AR(1) for each series

```
function sigs = arvars(y);
[bigt,n]=size(y);
for j=1:n;
    q=[ones(bigt-1,1) y(1:end-1,j)]; w=y(2:end,j);
    bhat=inv(q'*q)*q'*w;
    res=w-(q*bhat);
    sigs(j,1)=sqrt(mean(res.^2));
end;
```

- The second program contains the Gibbs sampling

```
function forc = minnesota(x,forvar,k,maxhor,theta,omega);
nburnin=200; nsim=500;
randn('seed',123);
[bigt,p]=size(x);
replics=nburnin+nsim;
%---------Initialize the Gibbs sampler
q=ones(bigt-k,1);
for j=1:k; q=[q x(k+1-j:bigt-j,:)]; end;
pi=x(k+1:bigt,:)'*q*inv(q'*q);  %p*pk
w=x(k+1:bigt,:);
res=w-(q*pi');

for imc=1:replics;
%---First Gibbs step
w=x(k+1:bigt,:);
e=w-(q*pi');
sigma=iwishrnd(e'*e,bigt);
%---Second Gibbs step
omega_post=inv(kron(inv(sigma),q'*q)+inv(omega));
theta_post=omega_post*(reshape(q'*w*inv(sigma),(p*p*k)+p,1)+(inv(omega)*theta));
pi=reshape(theta_post+(chol(omega_post)'*randn((p*p*k)+p,1)),(p*k)+1,p)';
%---Forecasting
```

```
w=[];
for j=1:k; w=[w x(k+2-j:bigt+1-j,:)]; end;
w=w';
pi_companion=[pi(:,2:end);eye(p*k)]; pi_companion=pi_companion(1:p*k,:);
for j=1:maxhor;
    w(:,end+1)=[pi(:,1);zeros(p*(k-1),1)]+(pi_companion*w(:,end));
end;
forcmat(:,:,imc)=w(forvar,end-maxhor+1:end);
end;
forc=mean(forcmat(:,:,nburnin+1:end),3)';
```

The program generates the forecasts of real GDP growth and CPI inflation at horizons out to four quarters ahead.

4. Plugging into the formula, the price of the bond is $94.21. The duration is 4.35 years.

5. I first put the FF1 and FF2 prices in Matlab (ff1, ff2) with the corresponding dates in d. Then I put the FOMC dates in dfomc. And the Treasury yields on yield10 corresponding to the dates in dtreasury.

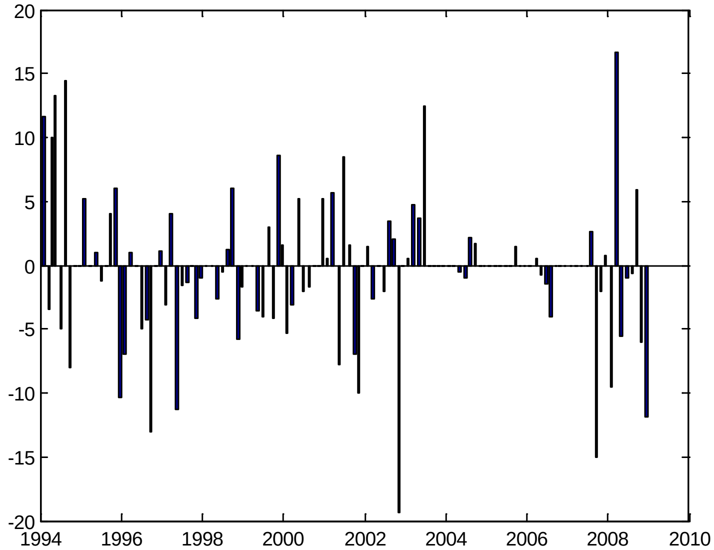
Then I ran this program

```
load fomcdata
ff1=100-ff1; ff2=100-ff2;
for j=1:length(dfomc);
    k=find(dfomc(j)==d);
    [q1,q2,q3]=datevec(d(k));
    e=eomday(q1,q2);
    if q3>22; surp(j,1)=100*(ff2(k)-ff2(k-1)); end;
    if q3==1; surp(j,1)=100*(ff1(k)-ff2(k-1))*e/(e-1); end;
    if q3>1 & q3<=22; surp(j,1)=100*(ff1(k)-ff1(k-1))*e/(e-q3);end;
    k=find(dfomc(j)==dtreasury);
    y(j,1)=100*(yield10(k)-yield10(k-1));
end;
bar(dfomc,surp);  %Surp are the monetary policy surprises
datetick('x','yyyy');
[bhat,vcov]=olswhite(y,surp);
disp([bhat sqrt(vcov)]);
```

which calls the White standard error function

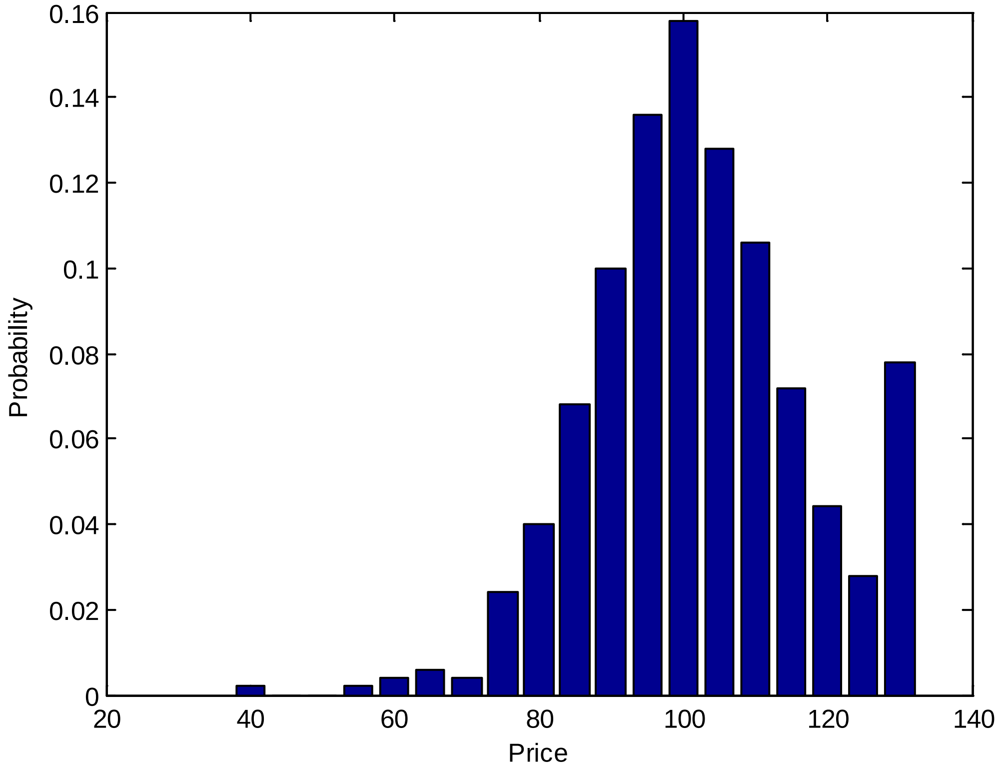
```
function [bhat,vcov]=olswhite(y,x);
%OLS with White standard errors
bhat=inv(x'*x)*x'*y;
res=y-(x*bhat);
[n,p]=size(x);
m1=zeros(p,p);
for i=1:n; m1=m1+((res(i)^2)*(x(i,:)'*x(i,:))); end;
vcov=inv(x'*x)*m1*inv(x'*x);
```

Here are the surprises (in basis points) which you can see are mostly quite small.



In the regression, the slope coefficient is 0.106, with a standard errors of 0.151, which is not significant.

6. Here is the pdf.



and here are the formulas in excel that create this

Strike	Price	Outcome	Probability
		40	=B3/5
45	0.01	45	=(B4-(2*B3)+B2)/5
50	0.02	50	=(B5-(2*B4)+B3)/5
55	0.03	55	=(B6-(2*B5)+B4)/5
60	0.05	60	=(B7-(2*B6)+B5)/5
65	0.09	65	=(B8-(2*B7)+B6)/5
70	0.16	70	=(B9-(2*B8)+B7)/5
75	0.25	75	=(B10-(2*B9)+B8)/5
80	0.46	80	=(B11-(2*B10)+B9)/5
85	0.87	85	=(B12-(2*B11)+B10)/5
90	1.62	90	=(B13-(2*B12)+B11)/5
95	2.87	95	=(B14-(2*B13)+B12)/5
100	4.8	100	=(B15-(2*B14)+B13)/5
105	7.52	105	=(B16-(2*B15)+B14)/5
110	10.88	110	=(B17-(2*B16)+B15)/5
115	14.77	115	=(B18-(2*B17)+B16)/5
120	19.02	120	=(B19-(2*B18)+B17)/5
125	23.49	125	=(B20-(2*B19)+B18)/5
130	28.1	130	=1-SUM(E2:E19)