Matlab: R2015a IRIS: 20150527

Monte-Carlo Stochatic Simulations

resample_from_model.m

by Jaromir Benes

May 27, 2015

Summary

Draw random time series from the model distribution, and compare their sample properties against the unconditional model-implied models. Keep in mind that this is a purely simulation exercise, and no observed data whatsoever are involved.

Contents

1 Clear Workspace

Clear workspace, close all graphics figures, clear command window, and check the IRIS version.

```
clear;
close all;
clc;
irisrequired 20140315;
%#ok<*NOPTS>
```

2 Load Solved model

Load the solved model object built in `read_model`. Run `read_model` at least once before running this m-file.

```
load read_model.mat m;
```

3 Define Dates

```
startDate = qq(1991,1);
endDate = qq(2020,4);
```

4 Set Standard Deviations of Shocks

No std deviations or cross-correlation coefficients have been assigned yet – in that case, std devs are 0.01 and corr coeffs are 0 by default. Later on, these will be estimated; now, simply pick some values for them. Note the double underscore deparating the names of shocks when referring to a corr coeff.

In general, after changing some parameters the steady state and model solution need to be recalculated. However, std devs and corr coeff have no impact on the steady state or solution so go ahead without running `sstate` or `solve`.

1 This `get` command returns a database with the currently assigned std deviations.

1 This `get` command returns a database with the currently assigned non-zero cross-correlations.

```
get(m,’std’) 1
get(m,’nonzerocorr’) 2

m.std_Mp = 0.001;
m.std_Mw = 0.001;

m.std_Ey = 0.01;
m.std_Ep = 0.01;
m.std_Ea = 0.001;
m.std_Er = 0.005;
m.corr_Ea__Ep = 0.25;

get(m,’std’) 1
get(m,’nonzerocorr’) 2
```

```
ans =
    std_Mp: 0
    std_Mw: 0
    std_Ey: 0.0100
    std_Ep: 0.0100
    std_Ea: 1.0000e-03
    std_Er: 0.0100
    std_Ew: 0.0100
ans =
struct with no fields.
ans =
    std_Mp: 1.0000e-03
    std_Mw: 1.0000e-03
    std_Ey: 0.0100
    std_Ep: 0.0100
    std_Ea: 1.0000e-03
    std_Er: 0.0050
    std_Ew: 0.0100
ans =
    corr_Ep__Ea: 0.2500
```

5 Draw Random Time Series from Model Distribution

A total of N = 1,000 different time series samples for each variables will be generated from the model distribution, each 30 years (120 quarters) long.

```
J = struct();
```

```
J.std_Ey = tseries();
J.std_Ey(startDate+(1:3)) = 0.02;

N = 1000;
d = resample(m,[],startDate:endDate,N,J,'progress=',true);
```

```
[--IRIS model.resample progress----------]
[****************************************]
```

6 Re-Simulate Data

If the resampled database, `d`, is used as an input database in `simulate`, the simulated database will simply reproduce the paths. Note that only initical condition and shocks are taken from the input database. The paths for the endogenous variables contained in the input database are completely ignored, and not used at all.

Also, remember to set `'anticipate=' false` because `resample` produces unanticipated shocks.

```
d1 = simulate(m,d,startDate:endDate,'anticipate=',false,'progress=',true);

maxabs(d,d1)
```

```
ans = 
      Short: 1.0303e-13
       Infl: 7.3053e-14
     Growth: 1.8119e-13
       Wage: 2.8777e-13
          Y: 1.6431e-14
          N: 5.5511e-16
          W: 1.3145e-13
          Q: 1.9984e-14
          H: 1.6875e-14
          A: 1.0658e-14
          P: 2.3093e-14
          R: 2.2204e-16
         Pk: 1.1369e-13
         Rk: 4.1078e-15
     Lambda: 4.9960e-15
         dP: 2.2204e-16
        d4P: 6.6613e-16
         dW: 6.6613e-16
        RMC: 7.7716e-16
         Mp: 0
         Mw: 0
         Ey: 0
         Ep: 0
         Ea: 0
         Er: 0
         Ew: 0
      alpha: 0
       beta: 0
      gamma: 0
      delta: 0
          k: 0
         pi: 0
        eta: 0
        psi: 0
        chi: 0
        xiw: 0
        xip: 0
       rhoa: 0
       rhor: 0
     kappap: 0
     kappan: 0
     Short_: 0
      Infl_: 0
    Growth_: 0
      Wage_: 0
```

7 Compute Sample Properties of Simulated Time Series

Calculate the sample mean, and use the `acf` function to calculate the std dev and autocorrelation coefficients for the three measurement variables, `Short`, `Infl`, and `Growth`.

```
smean = struct();
sstd = struct();
sauto = struct();

smean.Short = mean(d.Short);
[c,r] = acf(d.Short,Inf,'order',1);
sstd.Short = sqrt(diag(c(:,:,1)).');
sauto.Short = diag(r(:,:,2));

smean.Infl = mean(d.Infl);
[c,r] = acf(d.Infl,Inf,'order',1);
sstd.Infl = sqrt(diag(c(:,:,1)).');
```

```
sauto.Infl = diag(r(:,:,2));

smean.Growth = mean(d.Growth);
[c,r] = acf(d.Growth,Inf,’order’,1);
sstd.Growth = sqrt(diag(c(:,:,1)).’);
sauto.Growth = diag(r(:,:,2));

smean
sstd
sauto
```

```
smean =
     Short: [1x1000 double]
      Infl: [1x1000 double]
    Growth: [1x1000 double]
sstd =
     Short: [1x1000 double]
      Infl: [1x1000 double]
    Growth: [1x1000 double]
sauto =
     Short: [1000x1 double]
      Infl: [1000x1 double]
    Growth: [1000x1 double]
```

8 Compute Corresponding Asymptotic Properties Analytically

```
amean = struct();
astd = struct();
aauto = struct();

[C,R] = acf(m,’order’,1);
C = select(C,{’Short’,’Infl’,’Growth’});
R = select(R,{’Short’,’Infl’,’Growth’});

amean.Short = real(m.Short);
astd.Short = sqrt(C(1,1,1));
aauto.Short = R(1,1,2);

amean.Infl = real(m.Infl);
astd.Infl = sqrt(C(2,2,1));
aauto.Infl = R(2,2,2);

amean.Growth = real(m.Growth);
```

```
astd.Growth = sqrt(C(3,3,1));
aauto.Growth = R(3,3,2);

amean
astd
aauto
```

```
amean =
     Short: 7.1827
      Infl: 2.5000
    Growth: 3.0000
astd =
     Short: 3.9134
      Infl: 5.6458
    Growth: 3.9849
aauto =
     Short: [1x1 namedmat]
      Infl: [1x1 namedmat]
    Growth: [1x1 namedmat]
```

9 Plot Sample and Asymptotic Properties

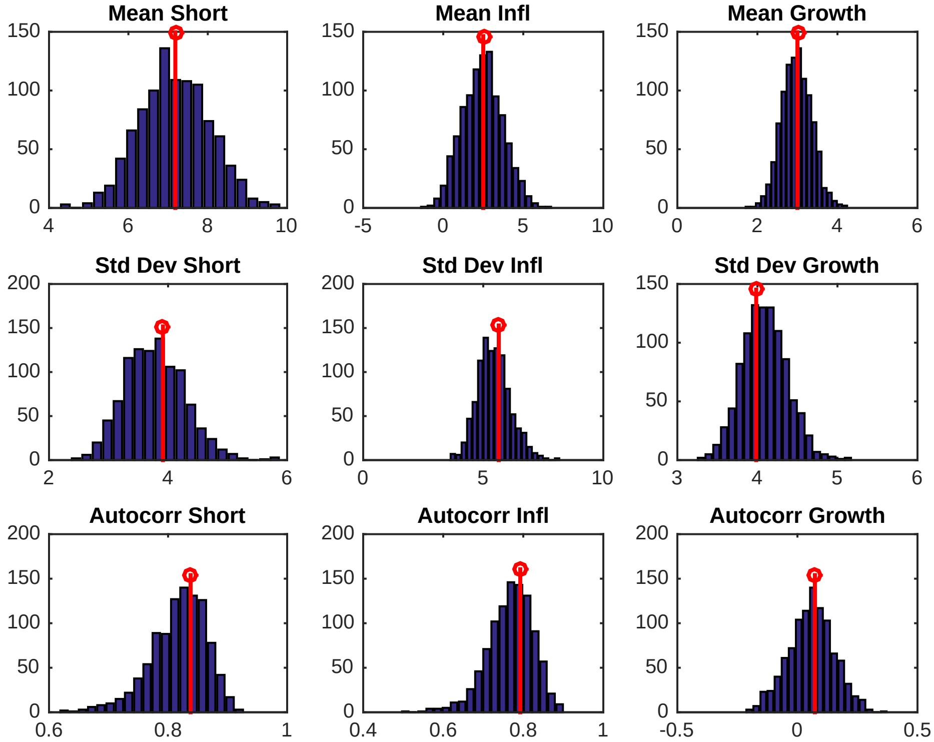
```
list = {’Short’,’Infl’,’Growth’};
figure();

for i = 1 : length(list)
   subplot(3,3,i);
   [y,x] = hist(smean.(list{i}),20);
   bar(x,y);
   hold(’all’);
   stem(amean.(list{i}),1.1*max(y),’color’,’red’,’lineWidth’,2);
   title([’Mean ’,list{i}]);

   subplot(3,3,i+3);
   [y,x] = hist(sstd.(list{i}),20);
   bar(x,y);
   hold(’all’);
   stem(astd.(list{i}),1.1*max(y),’color’,’red’,’lineWidth’,2);
   title([’Std Dev ’,list{i}]);

   subplot(3,3,i+6);
   [y,x] = hist(sauto.(list{i}),20);
   bar(x,y);
```

```
    hold('all');
    stem(aauto.(list{i}),1.1*max(y),'color','red','lineWidth',2);
    title(['Autocorr ',list{i}]);
end
```



10 Help on IRIS Functions Used in This File

Use either `help` to display help in the command window, or `idoc` to display help in an HTML browser window. help model/acf help model/get help model/resample help model/subsasgn help tseries/acf help select