DynareR: A Seamless Integration of R and Dynare

Sagiru Mati (PhD)

```
knitr::opts_chunk$set(
  collapse = TRUE,
  comment = "#>",
  eval=F,
  echo = T

)
library(DynareR)
# add_path("some/path")
# set_dynare_version("6-unstable-2022-04-03-0800-700a0e3a")
```

1 About DynareR

DynareR is an R package that can run `Dynare` program from R Markdown.

2 Requirements

Users need the following in order to knit this document:

- Dynare 4.6.1 or above
- Octave 5.2.0 or above

3 Installation

DynareR can be installed using the following commands in R.

```
install.packages("DynareR")

        OR

devtools::install_github('sagirumati/DynareR')
```

4 Usage

Please load the DynareR package as follows:

```
```{r DynareR}
library(DynareR)
```
```

Then create a chunk for **dynare** (adopted from Dynare example file example1) as shown below:

```dynare
```{dynare example1,eval=T,echo=T,comment=NULL,results='hide'}
/*
 * Example 1 from F. Collard (2001): "Stochastic simulations with DYNARE:
 * A practical guide" (see "guide.pdf" in the documentation directory).
 */

/*
 * Copyright (C) 2001-2010 Dynare Team
 *
 * This file is part of Dynare.
 *
 * Dynare is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * Dynare is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with Dynare.  If not, see <http://www.gnu.org/licenses/>.
 */


var y, c, k, a, h, b;
varexo e, u;

parameters beta, rho, alpha, delta, theta, psi, tau;

alpha = 0.36;
rho   = 0.95;
tau   = 0.025;
beta  = 0.99;
delta = 0.025;
psi   = 0;
theta = 2.95;

phi   = 0.1;

model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b(+1))*c(+1)))
    *(exp(b(+1))*alpha*y(+1)+(1-delta)*k));
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;

initval;
y = 1.08068253095672;
c = 0.80359242014163;
```

```
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;

shocks;
var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;

stoch_simul;
```

Dynamic 1.5883

The above chunk creates a Dynare program with the chunk's content, then automatically run Dynare, which will save Dynare outputs in the current directory.

Please note that DynareR uses the chunk name as the model name. So, the outputs of Dynare are saved in a folder with its respective chunk name. Thus a new folder example1 will be created in your current working directory.

5 Plotting the IRF

The Impulse Response Function (IRF) is saved by default in `example1/example1/graphs/` folder with the IRF's name `example1_IRF_u.pdf`, where `example1` is the Dynare model's name.

5.1 The include_IRF function

Use this function to embed the graphs Impulse Response Function (IRF) in R Markdown document. This function supports IRF graph saved in pdf format only. Therefore, you need to add `stoch_simul(graph_format = (pdf))` to change the default saving behaviour of `Dynare` from `eps` to `pdf`.

```
include_IRF(model="example1",IRF="u",path=".")
```

The Impulse Response Function (IRF) of the example1 model can be fetched using the following R chunk. Note that only the last part of the IRF's name (`u`) is needed, that is `example1_IRF_` is excluded. Also note that `out.extra='trim={0cm 7cm 0cm 7cm},clip'` is used to trim the white space above and below the IRF

```
```{r IRF,out.extra='trim={0cm 7cm 0cm 7cm},clip',fig.cap="Another of figure generated from Dynare softw
include_IRF("example1","u")
```
```

However, Dynare figure can only be dynamically included if the output format is pdf as Dynare produces pdf and eps graphs only.

# 6 DynareR functions for base R

The DynareR package is also designed to work with base R. The following functions show how to work with DynareR outside R Markdown.

Figure 1: IRF automatically fetched

## 6.1 The write_dyn function

This function writes a new `dyn` file. Use `write_dyn(model,code)` if you want the `Dynare` file to live in the current working directory. Use `write_dyn(model,code,path)` if you want the Dynare file to live in the path different from the current working directory.

```
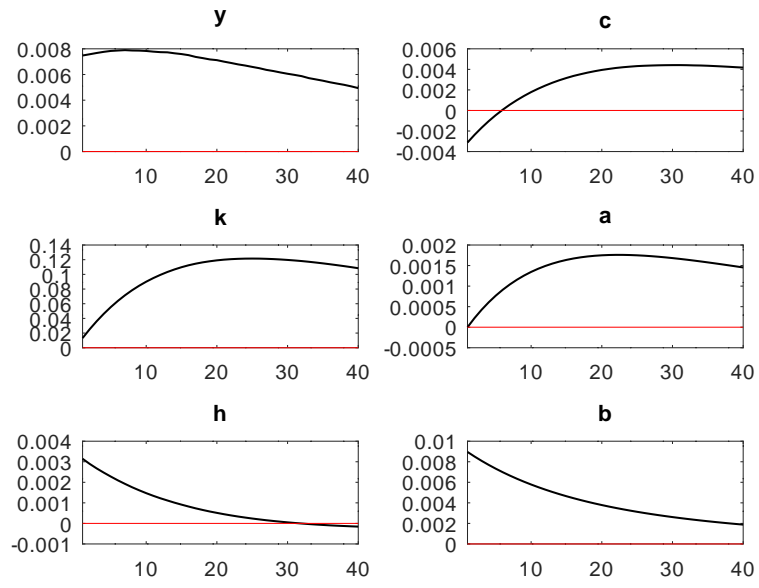DynareCodes='var y, c, k, a, h, b;
varexo e, u;
parameters beta, rho, alpha, delta, theta, psi, tau;
alpha = 0.36;
rho   = 0.95;
tau   = 0.025;
beta  = 0.99;
delta = 0.025;
psi   = 0;
theta = 2.95;
phi   = 0.1;
model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b(+1))*c(+1)))
          *(exp(b(+1))*alpha*y(+1)+(1-delta)*k));
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;
initval;
y = 1.08068253095672;
c = 0.80359242014163;
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;

shocks;
var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;

stoch_simul;'


write_dyn(model="example1",code=DynareCodes)
write_dyn(model="agtrend",code=DynareCodes)
```

6.2 The write_mod function

This function writes a new `mod` file. Use `write_mod(model,code)` if you want the `Dynare` file to live in the current working directory. Use `write_mod(model,code,path)` if you want the Dynare file to live in the path different from the current working directory.

```
DynareCodes='var y, c, k, a, h, b;
varexo e, u;
parameters beta, rho, alpha, delta, theta, psi, tau;
alpha = 0.36;
rho   = 0.95;
tau   = 0.025;
beta  = 0.99;
delta = 0.025;
psi   = 0;
theta = 2.95;
phi   = 0.1;
model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b(+1))*c(+1)))
          *(exp(b(+1))*alpha*y(+1)+(1-delta)*k));
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;
initval;
y = 1.08068253095672;
c = 0.80359242014163;
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;

shocks;
var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;

stoch_simul;'


write_mod(model="example2",code=DynareCodes)
```

6.3   The run_dynare function

Create and run Dynare `mod` file.   Use this function to create and run Dynare mod file.   Use run_dynare(model,code) if you want the Dynare files to live in the current working directory.   Use run_dynare(model,code,path) if you want the Dynare files to live in the path different from the current working directory.

```
DynareCodes='var y, c, k, a, h, b;
varexo e, u;
parameters beta, rho, alpha, delta, theta, psi, tau;
alpha = 0.36;
rho   = 0.95;
```

```
tau   = 0.025;
beta  = 0.99;
delta = 0.025;
psi   = 0;
theta = 2.95;
phi   = 0.1;
model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b(+1))*c(+1)))
          *(exp(b(+1))*alpha*y(+1)+(1-delta)*k));
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;
initval;
y = 1.08068253095672;
c = 0.80359242014163;
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;

shocks;
var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;

stoch_simul;'


run_dynare(model="agtrend",code=DynareCodes)
```

6.4   The run_models function

Run multiple existing `mod` or `dyn` files. Use this function to execute multiple existing Dynare files. Use `run_models(file)` if the Dynare files live in the current working directory. Use `run_models(file,path)` if the Dynare files live in the path different from the current working directory.

Where `example1.mod`, `example2.mod`, `agtrend.mod` and `bkk.mod` are the Dynare model files (with `mod` or `dyn` extension), which live in the current working directory.

```
run_models(model=c("example1","example2","agtrend","bkk"))
```

To run all `Dynare` models that live in the current working directory, use the following:

```
run_models()
```

To run all `Dynare` models that live in particular path (for example 'DynareR/run_models/' folder), use the following:

```
run_models(model = 'DynareR/run_models*',import_log=T)
```

7   Demo

The demo files are included and can be accessed via demo(package="DynareR")

8   Template

Template for R Markdown is created. Go to `file->New File->R Markdown-> From Template->DynareR`.

Please visit my Github for a better explanation and example files.