IRF

CLMENT CARRIER

Shock response

Consider a VAR(1) in difference where the parameters have been estimated:

$$\Delta Y_t = \Delta Y_{t-1}\hat{A} + \epsilon_t,$$

To compute the response to an exogenous shock of size s to the first variable in Y, construct the vector $\delta = [s, 0, ..., 0]$. The response at after h periods is given by $\delta\hat{A}^h$.

- For models with multiple lags the companion form (or simply recursive computations) can be used.
- For models with exogenous variables, at path for the exogenous variables has to be specified. In general it should be equal to zero or to a random walk forecast, at least in periods following an initial shock.
- The deterministics can be omitted.
- For VECM models, things are more complicated but not much more difficult.

Conditional forecasts

The options so far:

- Create a model for the exogenous variable.
- Use naïve methods (i.e. RW).
- Compute prediction density for the exogenous variable and plug in the model. Gaussian + linear = Gaussian.
- Condition on true value.

Application sur R

```
require(lassovar)
## Loading required package:  lassovar
require(ggplot2)
## Loading required package:  ggplot2
require(reshape2)
## Loading required package:  reshape2
require(urca)
## Loading required package:  urca
require(MSBVAR)
## Loading required package:  MSBVAR
## ##
## ## MSBVAR Package v.0.9-2
## ## Build date:  Wed Jul  1 12:01:35 2015
## ## Copyright (C) 2005-2015, Patrick T. Brandt
## ## Written by Patrick T. Brandt
```

Date: July 1, 2015.

```
## ##
## ## Support provided by the U.S. National Science Foundation
## ## (Grants SES-0351179, SES-0351205, SES-0540816, and SES-0921051)
## ##
##
##
## Attaching package:  'MSBVAR'
##
## The following object is masked from 'package:urca':
##
##     summary
```

```
load("vardata")
```

I keep variables from Q4 1997 :

```
data<-subset(vardataframe[116:180,])
```

First difference of all series :

```
difdata <- tail(data,-1) - head(data,-1)
```

Shock Response

Function for computing the IRF :

```
forecast<-function(data,lag,horizon,choc){
  fore<-matrix(0,nrow=dim(data)[2],ncol=horizon)
  lv<-lassovar(dat=data,lags=lag, ic="BIC")
  coeff<-lv$coefficients[-1,]
  for (i in 1:horizon){
    fore[,i]<-coeff^i%*%choc
  }
return(t(fore))
}
```

Conditional Shock

Function to generate RW for exogonous variables

```
rw<-function(t,x){
  y<-matrix(0,dim(x)[2],t)
  y[,1]<-t(x)
  for(i in 2:t){
    y[,i] <- y[,i-1] + matrix(rnorm(dim(x)[2],0,1),dim(x)[2],1)
  }
  return(y)
}
```

What do we choose for the variances ? Variances of variables in the data set used for estimation ?

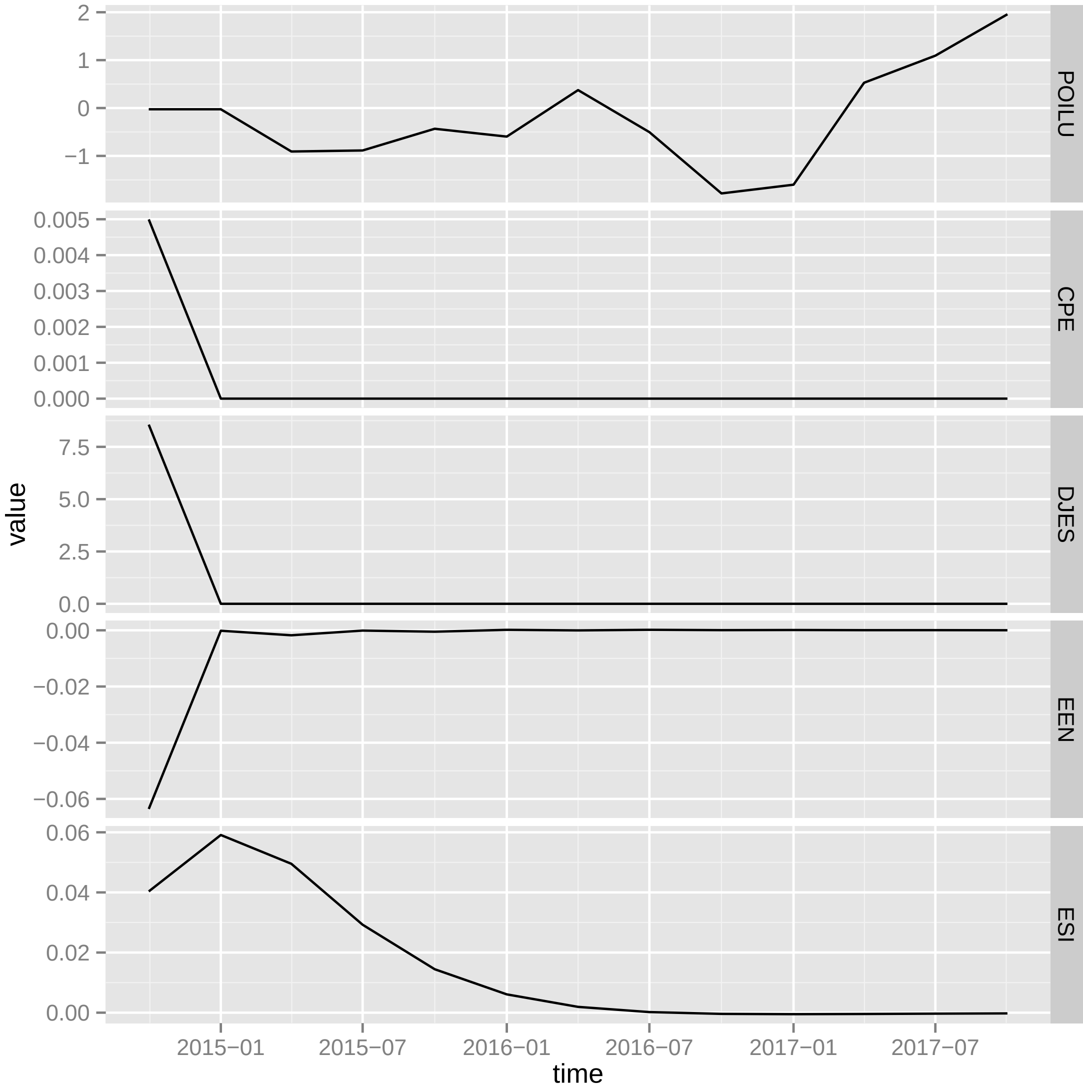
Function to compute IRF to RW evolution of exogeneous variables

```
conditional<-function(exogen,data,lag,horizon){
  all=data.frame(exogen,data)
  fore<-matrix(0,nrow=dim(data)[2]+dim(exogen)[2],ncol=horizon+1)
  fore[,1]<-t(all[dim(all)[1],])
  fore[1:dim(exogen)[2],-1]<-rw(horizon,as.matrix(exogen[dim(exogen)[1],]))
  lv<-lassovar(dat=all,lags=lag, ic="BIC")
  coeff<-as.matrix(lv$coefficients[-1,],26,26)

  for (i in 2:(horizon+1)){
    fore[-dim(exogen)[2],i]<-(coeff%*%fore[,i-1])[-dim(exogen)[2]]
  }
  rownames(fore)<-names(all)
  return(t(fore))
}
```

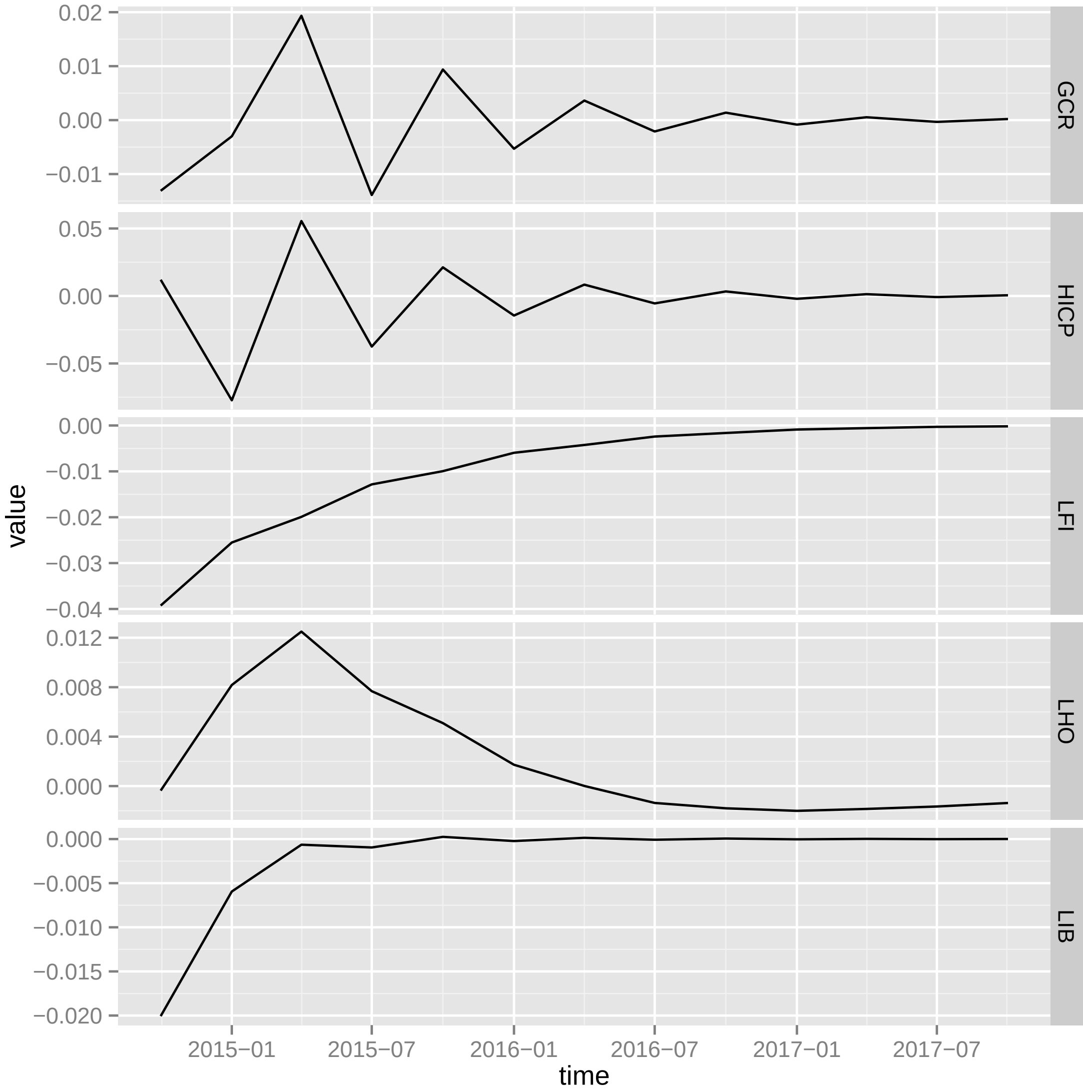
Application with Oil Price as exogenous variable

```
exo<-data.frame(difdata$POILU)
colnames(exo)<-'POILU'
end<-subset(difdata[,-which(names(difdata) %in% c("POILU"))])
IRF<-conditional(exo,end,1,12)
IRF<-data.frame(IRF)
```

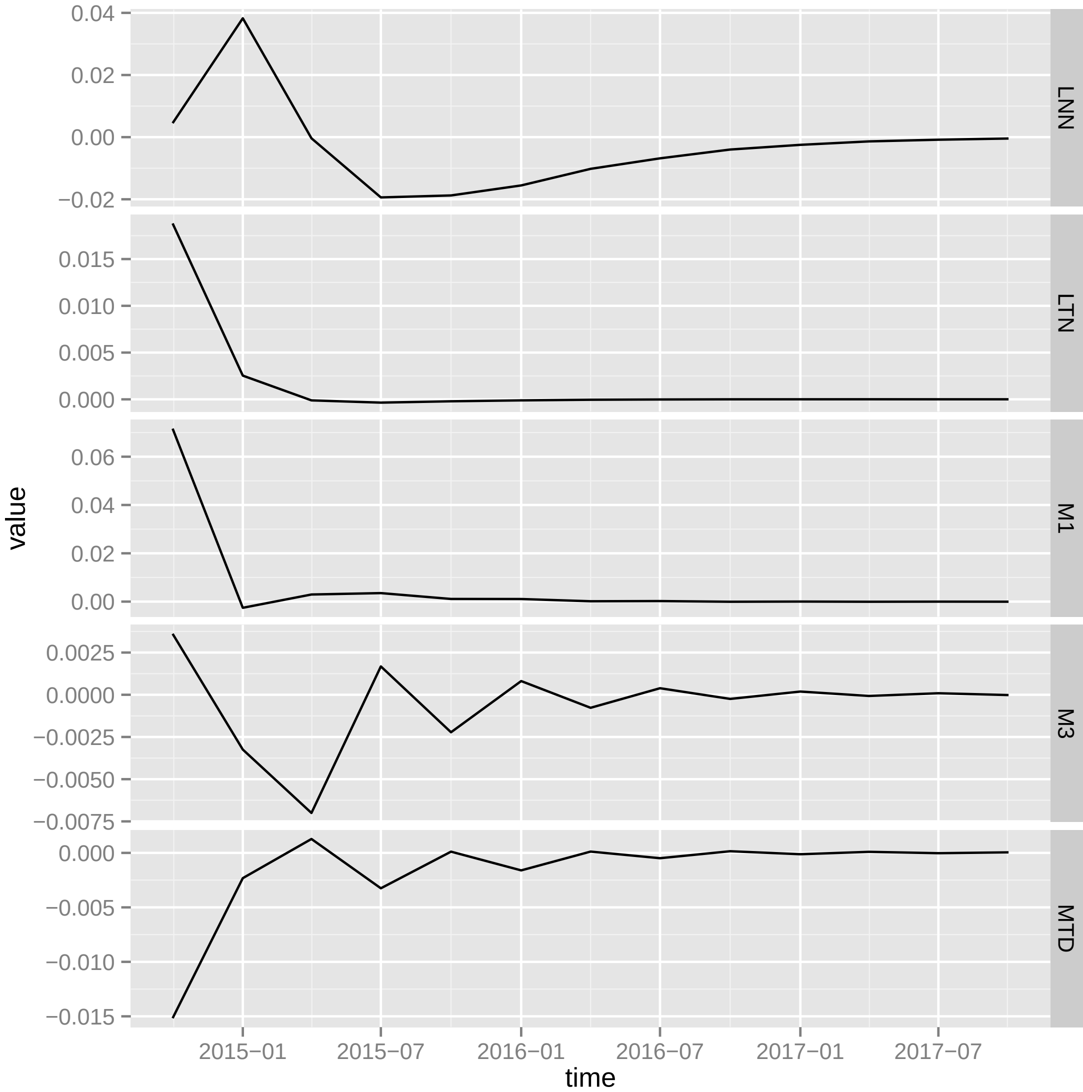
```
var1<-IRF[,1:5]
var2<-IRF[,6:10]
var3<-IRF[,11:15]
var4<-IRF[,16:20]
var5<-IRF[,21:26]

var1$time<-seq(as.Date("2014/10/01"), as.Date("2017/12/31"), by = "quarter")
var2$time<-seq(as.Date("2014/10/01"), as.Date("2017/12/31"), by = "quarter")
var3$time<-seq(as.Date("2014/10/01"), as.Date("2017/12/31"), by = "quarter")
var4$time<-seq(as.Date("2014/10/01"), as.Date("2017/12/31"), by = "quarter")
var5$time<-seq(as.Date("2014/10/01"), as.Date("2017/12/31"), by = "quarter")

mvar1 <- melt(var1,  id = 'time', variable.name = 'series')
mvar2 <- melt(var2,  id = 'time', variable.name = 'series')
mvar3 <- melt(var3,  id = 'time', variable.name = 'series')
mvar4 <- melt(var4,  id = 'time', variable.name = 'series')
mvar5 <- melt(var5,  id = 'time', variable.name = 'series')

ggplot(mvar1, aes(time,value)) + geom_line() + facet_grid(series ~ . ,scales="free")
```



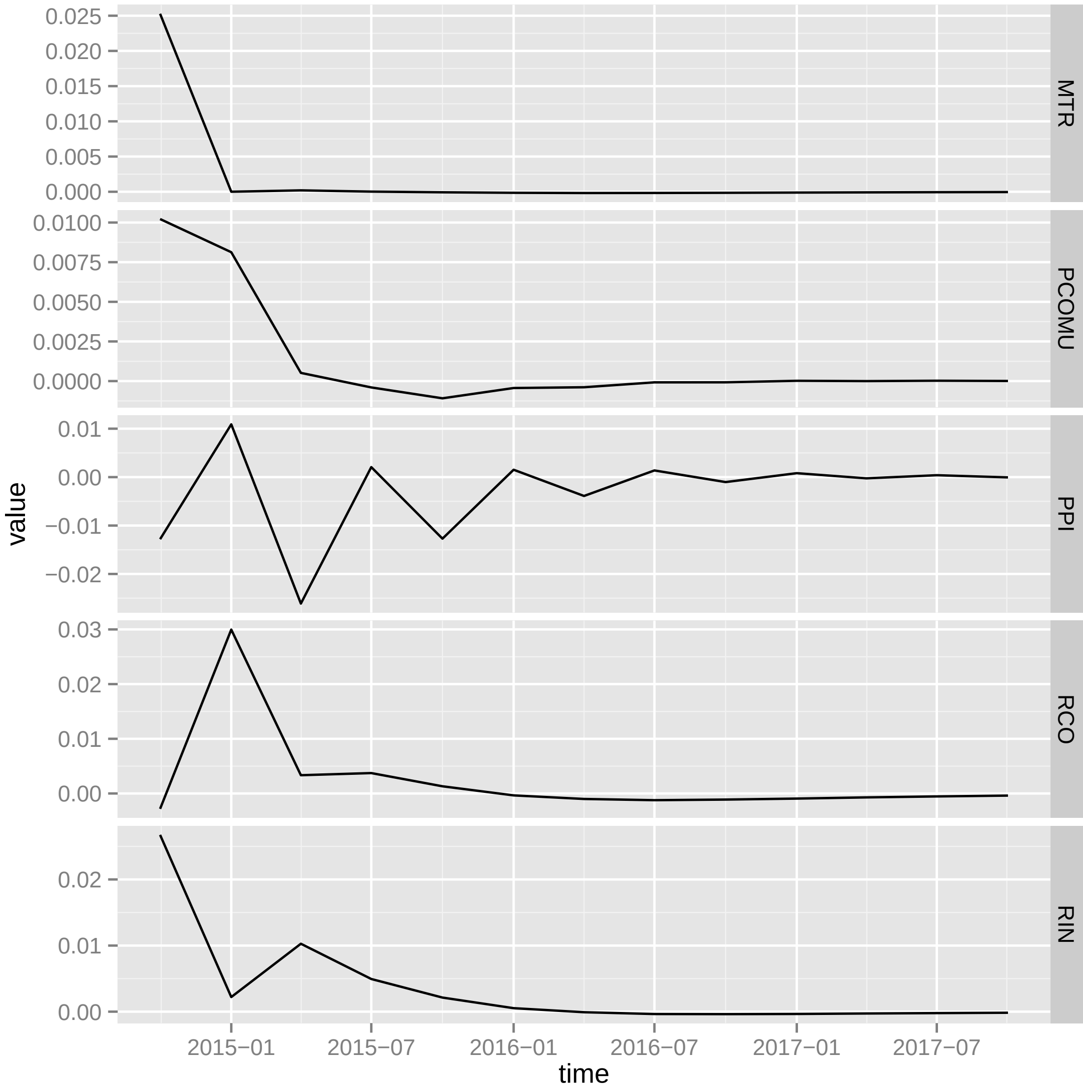

```
ggplot(mvar2, aes(time,value)) + geom_line() + facet_grid(series ~ . ,scales="free")
```



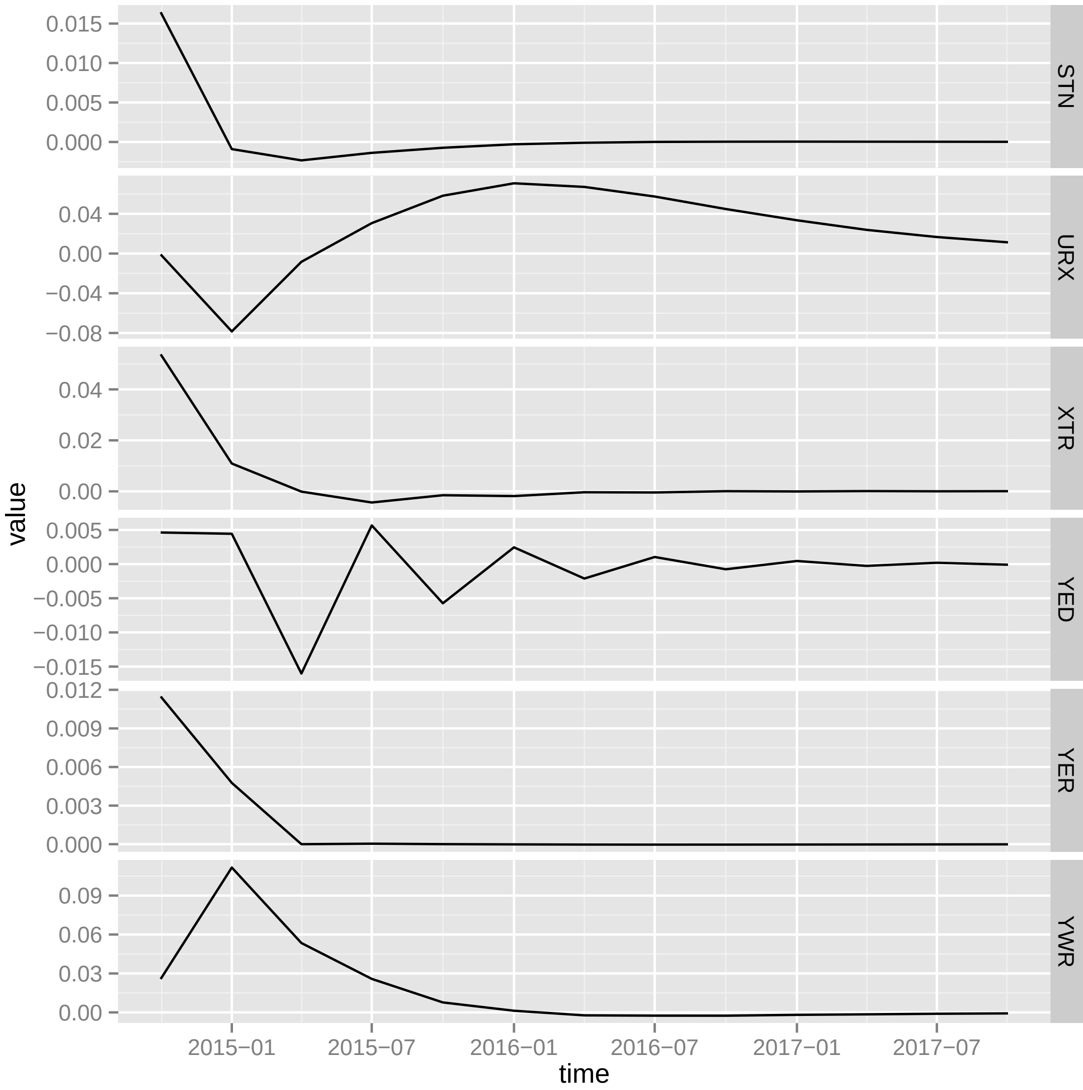

```
ggplot(mvar3, aes(time,value)) + geom_line() + facet_grid(series ~ . ,scales="free")
```




```r
ggplot(mvar4, aes(time,value)) + geom_line() + facet_grid(series ~ . ,scales="free")
```




```
ggplot(mvar5, aes(time,value)) + geom_line() + facet_grid(series ~ . ,scales="free")
```


STN
URX
XTR
YED
YER
YWR
value
time
2015−01
2015−07
2016−01
2016−07
2017−01
2017−07