Estimating VARs & SVARs
Appendix to Advanced Macroeconomics: An Easy Guide

Appendix C in the book discussed the estimation of a DSGE model. These estimations are known as structural estimations, because they are built on the basis of a ”structural model”. An alternative way of doing macroeconomic forecasting is to use an approach that is agnostic about the model and works through unveiling the statistical properties of the data itself. This is the approach that was pioneered by Chris Sims and earned him, eventually, the Nobel Prize in Economics.

This appendix will introduce you to the estimation of VARs and SVARs. In order to do so we will do three things. First, we will introduce the concepts from a theoretical point of view. Then we will review the concepts of reduced form vector autoregressions (VARs) and structural vector autoregressions (SVARs). To do so we will refer to a specific case, which is the estimation of these models applied to a two variable system including US GDP and US inflation. As we go along, we will provide the codes in R so that you can apply these methods to your own variables of interest.

Reduced form VARs

The VAR approach is, by now, very well incorporated in standard statistical packages [1], so here we will just provide a basic review of the main concepts which may enable to tackle further readings on your own.

We will start by defining the vector autoregression (VAR) representation. It's name makes it look more difficult than it is. Consider two variables (y_t, x_t) with one lag (but you can add as many lags as you like),

$$y_t = \phi_{11} y_{t-1} + \phi_{12} x_{t-1} + u_t,$$
$$x_t = \phi_{21} y_{t-1} + \phi_{22} x_{t-1} + v_t,$$

where u_t and v_t are white noise, possibly correlated with each other. We can write this system with vectors and matrices,

$$\underbrace{\begin{pmatrix} y_t \\ x_t \end{pmatrix}}_{\equiv z_t} = \underbrace{\begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix}}_{\equiv \Phi} \underbrace{\begin{bmatrix} y_{t-1} \\ x_{t-1} \end{bmatrix}}_{\equiv z_{t-1}} + \underbrace{\begin{bmatrix} u_t \\ v_t \end{bmatrix}}_{\equiv \omega_t}.$$

Even more compactly we can write this VAR(1) model (the one in the parenthesis stands for the number of lags) for the vector z_t as

$$z_t = \Phi z_{t-1} + +\omega_t, \tag{1}$$

and we call this as a reduced-form VAR. Notice that while the notation looks complex this is nothing more than running an OLS regression of these two variables on themselves lagged! There is really not much more to it.

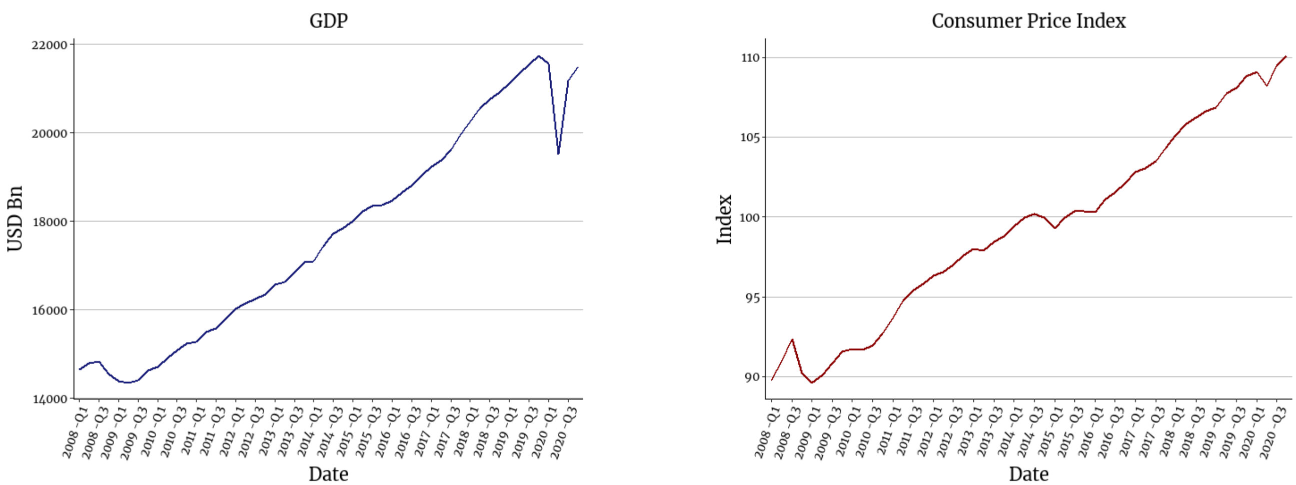
[1] Here we are going to develop our empirical example with RStudio

Notice that the matrix of variance and covariances is

$$\Omega = E\,\omega_t\,\omega_t^{'} = \begin{pmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{vu} & \sigma_v^2 \end{pmatrix}, \tag{2}$$

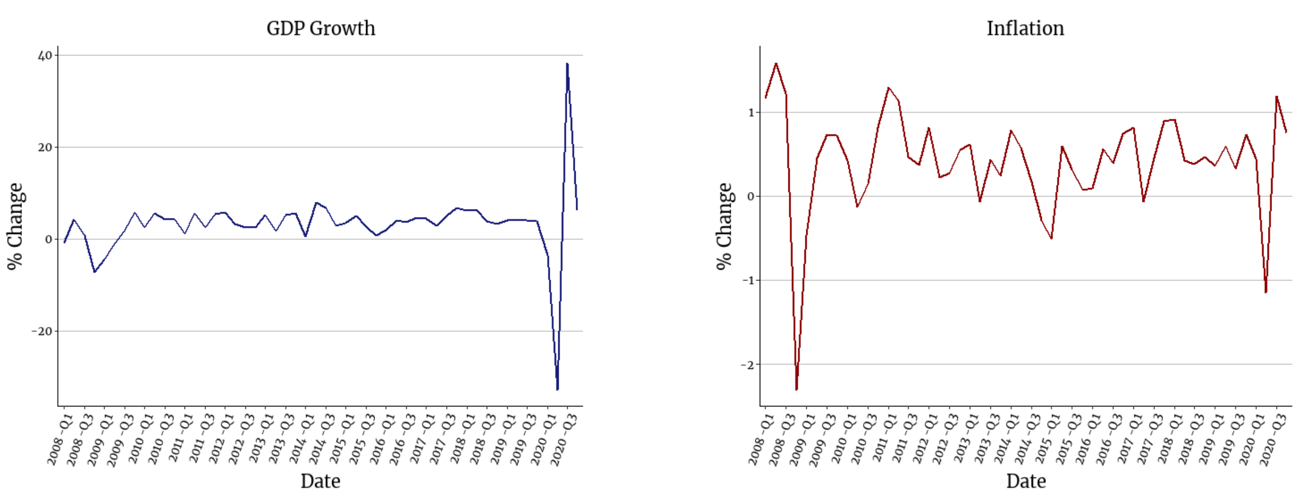
where the critical feature is that the off diagonal elements may not be zero. This means that the shocks may be correlated. This will be a source of problem as we will see quickly. Let us show you an example of this. Consider the GDP and the CPI for United States since 2008 to 2020. The data will look something like

Figure 1: **Endogenous Variables**



As can be seen both variables have a trend, which can complicate the estimations, so it becomes necessary to compute the growth rates to make the variables stationary. After this transformation GDP growth and inflation will look something as in Figure 2, where you can see that the data is stationary. Notice the big swing at the end resulting from the Covid pandemic.

Figure 2: **Growth Rates of Endogenous Variables**



```
#Removing our environment and setting the working directory
rm(list=ls())
setwd(dirname(rstudioapi::getActiveDocumentContext()$path)) #With this command you set the directory
    where the script is saved

#First, we have to load the libraries that we will use in this exercise. If you have never used it, you
    must first run the command install.packages() and inside the argument put the argument in quotes

library(openxlsx)
library(ggplot2)
library(tidyr)
library(dplyr)
library(zoo)
library(tidyverse)
library(vars)
library(stargazer)


#Once you imported the series of interest, it is easy to plot them with the ggplot package. For
    example, we create an object with the graph of the GDP, but you can replicate it for the case of
    CPI or growth rates as well (here we put the basic command, but you can explore other variants of
    format and design).

gdp.graph <- dataset %>%
  ggplot(aes(x=Date3, y=GDP,group=1)) +
  geom_line()
```

If we carry out the estimation of a VAR(1), the coefficient values are as follows.

	Dependent variable:	
	GDP Growth	*Inflation*
GDP_{t-1}	−0.24	−0.01
	(0.16)	(0.01)
$Inflation_{t-1}$	−0.42	0.18
	(1.98)	(0.16)
$Constant$	4.31***	0.35***
	(1.27)	(0.10)
Observations	51	51
R^2	0.07	0.03
Note:	*p<0.1; **p<0.05; ***p<0.01	

```
##### VAR Estimation #####
# First we create the time series objetc for the estimation and then we create a matrix with both series
```

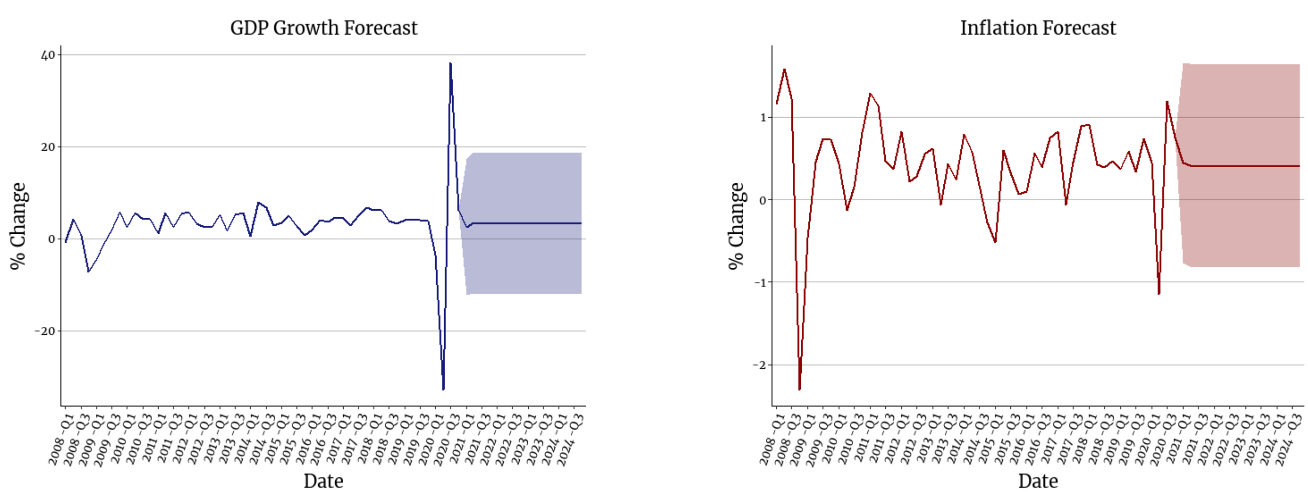
```
GDP <- ts(dataset$GDP.Growth, start = c(2008, 01), frequency = 4)
CPI <- ts(dataset$Inflation, start = c(2008, 01), frequency = 4)
Yd <- cbind(GDP,CPI)


# Now, we estimate a reduced form VAR(1)
VAR <- VAR(Yd, p = 1, type = "const")

fitvar <- VAR$varresult #here we extract the results of the estimation of both equations
stargazer(fitvar$GDP, fitvar$CPI, type = "text" ) #here you can replace the type argument for "latex"
    and you get the latex code for your document
```

Again nothing new here. These are just two OLS equations. However we can use them mechanically to estimate both variables going forward. How? It is very simple. You have the data until today, and the regression estimates the value tomorrow as a function of today's value. So you get an estimate for tomorrow's variables, and then use that to get an estimate for the day after and so on. In doing so you typically assume that the futures errors are zeros, but you could actually add some "exogenous" disturbance if you think there is a reason for doing so.

Figure 3: **Forecast for Endogenous Variables**




```
##### Forecast #####
#Here, we use the command "predict" to compute the forecast n periods ahead for GDP growth
GDP_fcst<- predict(VAR, n.ahead = 16)

#Here, we append the forecast output to the real variable and then we create a data frame object to do
    the plot of interest. The symbol "$" allows you to get inside the forecast object and select the
    part of that you are interested on (It's easy to compute the same results for Inflation)
gdp.fcst<- append(GDP,GDP_fcst$fcst$GDP[,1])
gdplow<-append(GDP,GDP_fcst$fcst$GDP[,2])
gdpupper<-append(GDP,GDP_fcst$fcst$GDP[,3])

#Here we create a sequence of a date for the forecast period, and then, we append this object to the
```

```
    ``in sample'' date.
date.ext<- seq(as.Date("2021-01-01"), by = "quarter", length.out = 16)
date.fcst<- append(dataset$Date, date.ext)

#Finally, we create a data frame with the date and the variable of interest that we will use to
    generate the graphs of interest
fanch_gdp<- data.frame(date.fcst,gdp.fcst,gdplow,gdpupper)
```

All VARs have what we call a *moving average representation*. This moving average representation is also called the impulse response function. Let's see why by using a very simple example. Consider the following VAR model

$$y_t = \alpha y_{t-1} + \epsilon_t$$

We can rearrange the equation and get,

$$y_t(1 - \alpha L) = \epsilon_t,$$

where L is the lag operator. Furthermore

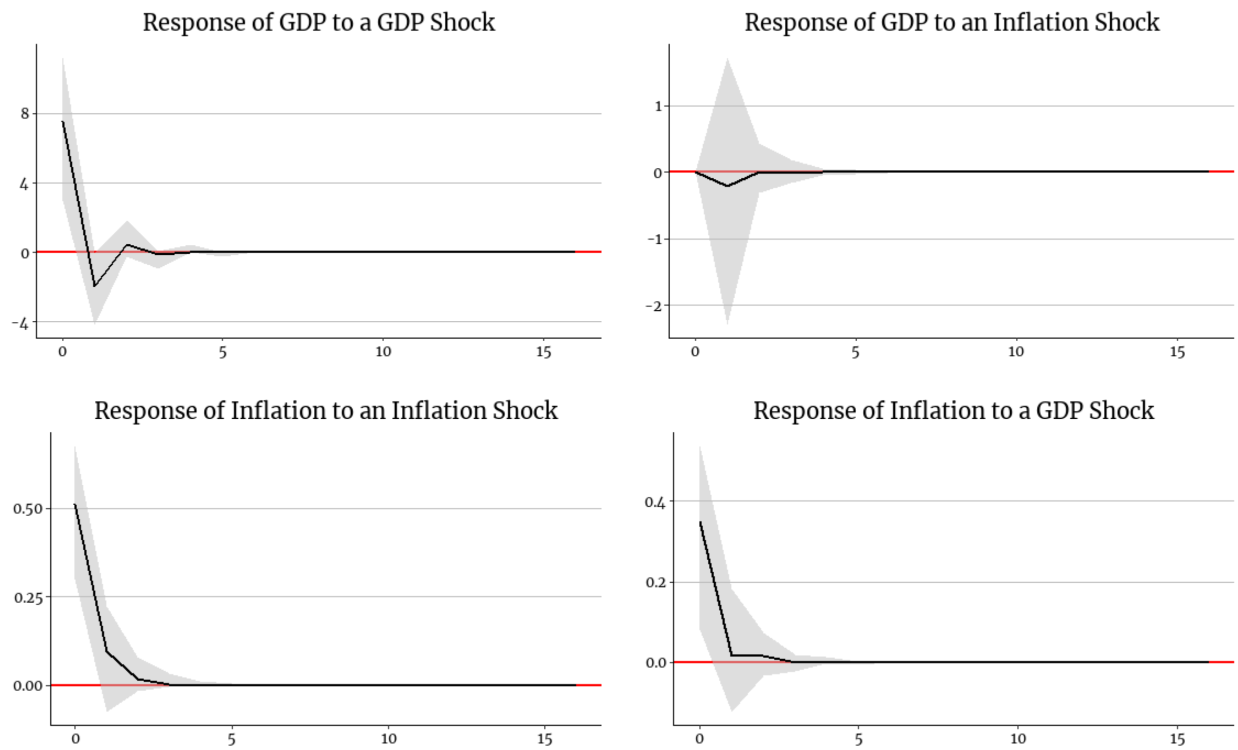
$$y_t = \frac{\epsilon_t}{(1 - \alpha L)} = \epsilon_t + (\alpha L)\epsilon_t + (\alpha L)^2\epsilon_t + ... + (\alpha L)^p\epsilon_t$$

$$= \epsilon_t + \alpha\epsilon_{t-1} + \alpha^2\epsilon_{t-2} + ... + \alpha^p\epsilon_{t-p}.$$

Notice that the coefficients of this representation give the impact of a shock *so many periods before* on the variable today. If we plot the sequence of coefficients we will see how a shock affects the variable over time. As we said this is what is called the *impulse response*. Of course, if the system has more lags and more variables the equivalent to the α above will be a matrix of coefficients. But conceptually it is the same.

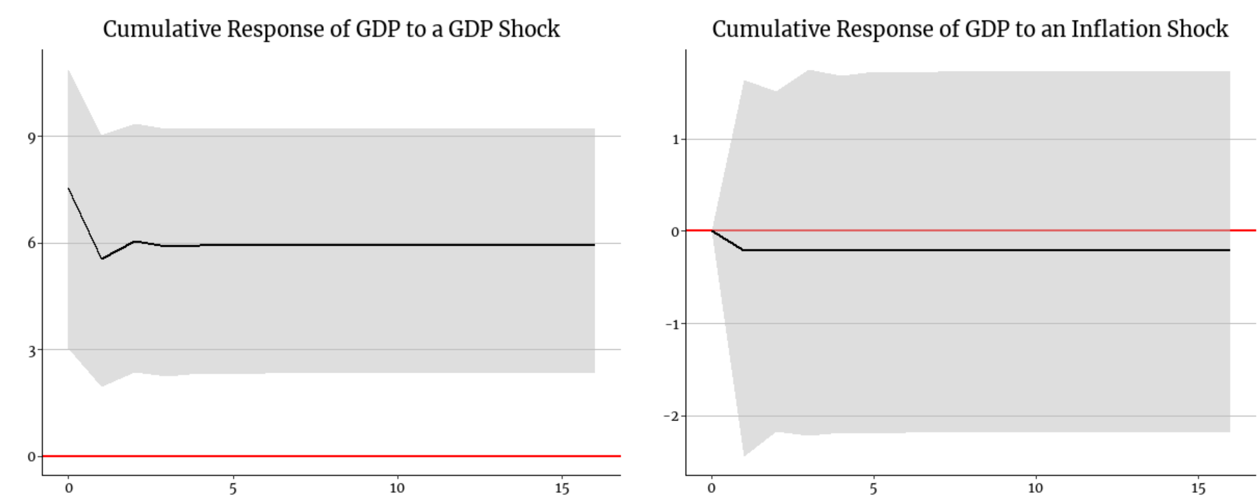
Computer packages will have a command to get these impulse responses. For the case above they take the following form

Figure 4: **Impulse Response Functions**



Remember that our empirical estimate is for the rate of growth. If you want to get the effect on GDP you have to sum up the coefficients to get the accumulated effect, which will give you the effect on the level of the GDP variable.

Figure 5: **Cumulative Impulse Response Functions for GDP Growth**




```
##### Impulse Response Functions #####
#For example we compute the GDP response to an Inflation shock, with a thousand of replications by
```

```
    bootstrap and a traditional confidence interval (and we do the same for the cumulative impulse
    response function)
var_irf <- irf(VAR, impulse = "CPI", response = "GDP", n.ahead = 16,boot = TRUE, runs = 1000, ci=0.95)
var_cum_irf <- irf(VAR, impulse = "CPI", response = "GDP", n.ahead = 16,boot = TRUE, runs = 1000,
    ci=0.95,cumulative = TRUE)
```

In a multivariable problem the MA representation is

$$z_t = \omega_t + \phi_1 \omega_{t-1} + \phi_2 \omega_{t-2} + ... + \phi_p \omega_{t-p}.$$

Now, there is a problem with the impulse response just computed. We can't shock an equation to evaluate the response of the system to that shock because when we shock one we shock we *need* to move the other because $\sigma_{uv} \neq 0$! So it is impossible in this framework to interpret what the impulse response in a simple VAR means. In other words, it is a response to something that does not exist (in the real world these shocks don't move on their own). This is the reason we need to step up to a "structural VAR" which precisely makes the shocks independent from each other.

Structural VARs

How do we do that? What we need to do is to ortogonalize the matrix of variances and covariances. Let's look for a matrix Ω such that

$$\Omega = AA^{'}.$$

Let's now define a transformation of the original shock

$$\tilde{\omega}_t = A^{-1}\omega_t.$$

Notice that if we compute the variance covariance of this modified shock we get

$$\begin{aligned} Var\ \tilde{\omega}_t &= A^{-1} Var\ \omega_t A^{-1'} \\ &= AA^{-1}A^{'}A^{-1'} = I. \end{aligned}$$

The modified shocks are independent of each other! So we can re-write the multivariate impulse response function as

$$z_t = AA^{-1}\omega_t + \phi_1 AA^{-1}\omega_{t-1} + \phi_2 AA^{-1}\omega_{t-2} + ... + \phi_p AA^{-1}\omega_{t-p}.$$

And then:

$$z_t = A\tilde{\omega}_t + \phi_1 A\tilde{\omega}_{t-1} + \phi_2 A\tilde{\omega}_{t-2} + ... + \phi_p A\tilde{\omega}_{t-p}$$

The coefficients A, $\phi_1 A$, $\phi_2 A$ ...$\phi_p A$ ... are called *the structural impulse response functions*, and they can be interpreted as independent shocks to the variable in the corresponding equation or have a deeper economic meaning. The important point is that they move on their own, as they are uncorrelated to the other shock.

The challenge then is to find the matrix A. One idea, initially suggested by Chris Sims, is to allow A to be a lower triangular matrix. This implies

$$A = \begin{pmatrix} a & 0 \\ b & c \end{pmatrix}.$$

To compute the values of A we use

$$AA' = \begin{pmatrix} a & 0 \\ b & c \end{pmatrix} \begin{pmatrix} a & b \\ 0 & c \end{pmatrix} = \begin{pmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{vu} & \sigma_v^2 \end{pmatrix}.$$

Simple algebra can be used to compute the solution for A

$$a = \sqrt{\sigma_u^2} \ , \qquad b = \frac{\sigma_{uv}}{\sqrt{\sigma_u^2}} \ , \qquad c = \sqrt{\sigma_v^2 - \frac{\sigma_{uv}^2}{\sigma_u^2}}.$$

To get an idea of what this means, multiply the dependent variables by the inverse of the matrix A to obtain

$$A^{-1} z_t = \begin{pmatrix} \frac{1}{a} & 0 \\ -\frac{b}{ac} & \frac{1}{c} \end{pmatrix} \begin{pmatrix} y_t \\ x_t \end{pmatrix} = \begin{pmatrix} \frac{1}{a} y_t \\ -\frac{b}{ac} y_t + \frac{1}{c} x_t \end{pmatrix}.$$

The Cholesky decomposition orders the variables from the most exogenous to the less exogenous. Then the shock to the most exogenous variable can be interpreted as an independent shock to that variable, and the shocks to the second variable as an independent shock to the second variable (the first one is exogenous to it) and so on.

An alternative way of estimating a structural VAR is what is known as the Blanchard-Quah decomposition. In this approach we impose long term constraints. Here (and in BQ's original paper) the assumption is that one (we will call it demand) shock does not alter GDP in the long run. Remember that to compute the coefficients of the matrix A we have the following matrix and equations:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix} = \begin{pmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{vu} & \sigma_v^2 \end{pmatrix},$$

where

$$a_{11}^2 + a_{12}^2 = \omega_{11}$$

$$a_{11}\ a_{21} + a_{12}\ a_{22} = \omega_{12}$$

$$a_{21}\ a_{11} + a_{22}\ a_{12} = \omega_{21}$$

$$a_{21}^2 + a_{22}^2 = \omega_{22}.$$

But one equation is redundant here because the second and third equations are identical. We have three equations (though not linear) so, at least, we need one additional equation to solve for the values of the A matrix. We impose our long term restriction to get our remaining equation. The long run effect is computed by summing the effects over time (in our model we sum the effects on the growth rate to impose a restriction on the level of GDP). The cumulative impulse response is the sum of the individual components:

$$\sum_{j=1}^{\infty} A_j = \sum_{j=1}^{\infty} \phi_j A.$$

The restriction that the long run effect is zero is obtained by making the upper left element of the above sequence equal to zero:

$$\sum_{j=1}^{\infty} \phi_{11,t}\ A(0)_{11} + \sum_{j=1}^{\infty} \phi_{12,t}\ A(0)_{21} = 0,$$

which is the result we were looking for by assuming long-term neutrality on the GDP variable of the first shock.

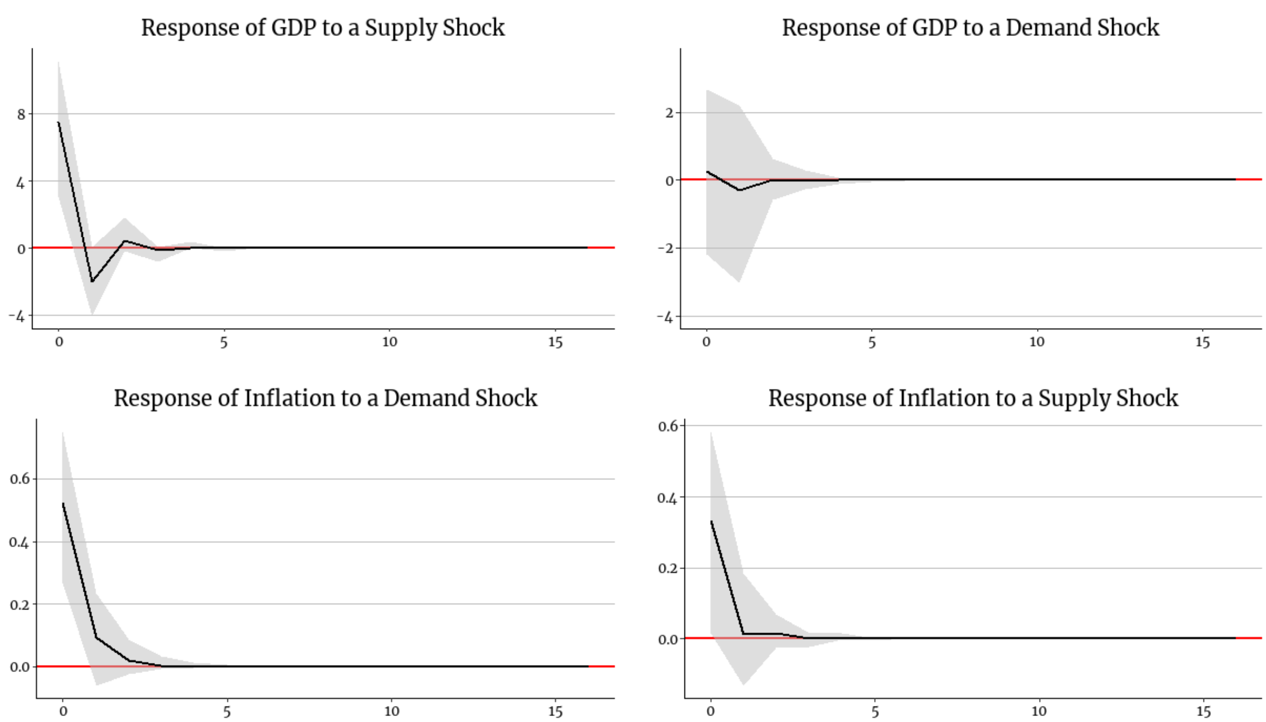
This provides one additional restriction that allows us to solve for the values of A.

Most statistical packages will provide an option to estimate this structural VAR. In the case of RStudio, the "BQ" command allows executing the Blanchard and Quah decomposition on the VAR object that we estimated earlier. Having done this, we can compute the structural impulse response functions as they appear in Figures 6 and 7.

```
##### Blanchard & Quah Decomposition #####
SVAR<-BQ(VAR) #This command generates the decomposition between short and long run of the VAR that we
    estimates on previous steps

#For the Structural Impulse Response Functions, the idea is similar than later. For example, we compute
    the GDP response to a Demand shock, with a thousand of replications by bootstrap and a traditional
    confidence interval (and we do the same for the cumulative structural impulse response function)
svar_irf <- irf(SVAR, impulse = "CPI", response = "GDP", n.ahead = 16,boot = TRUE, runs = 1000, ci=0.95)
svar_cum_irf <- irf(SVAR, impulse = "CPI", response = "GDP", n.ahead = 16,boot = TRUE, runs = 1000,
    ci=0.95, cumulative=TRUE)
```

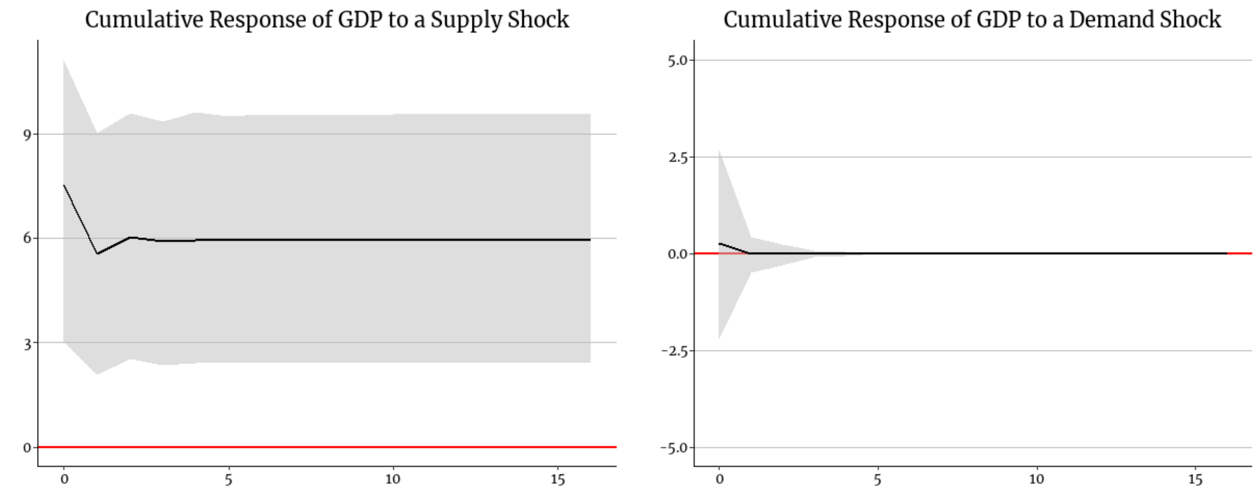
Figure 6: **Structural Impulse Response Functions**



,

And to check long run neutrality example we accumulate the impulse response functions.

Figure 7: **Structural Cumulative Impulse Response Functions for GDP Growth**



Notice that the effect on output of the demand shock is zero!