HUL715: Paper replication

Arundhati Dixit

21/06/2020

Overview

I have chosen to replicate James Stock and Mark Watson's **Understanding Changes in International Business Cycle Dynamics** [(2003) Journal of the European Economic Association. 3. 10.1162/1542476054729446.] [Replication material: http://www.princeton.edu/~mwatson/publi.html, replication files include data, GAUSS and RTS files].

Abstract of the paper

The volatility of economic activity in most G7 economies has moderated over the past 40 years. Also, despite large increases in trade and openness, G7 business cycles have not become more synchronized. After documenting these facts, we interpret G7 output data using a structural VAR that separately identifies common international shocks, the domestic effects of spillovers from foreign idiosyncratic shocks, and the effects of domestic idiosyncratic shocks. This analysis suggests that, with the exceptionof Japan, a significant portion of the widespread reduction in volatility is associated with a reduction in the magnitude of the common international shocks. Had the common international shocks in the 1980s and1990s been as large as they were in the 1960s and 1970s, G7 business cycles would have been substantially more volatile and more highly synchronized than they actually were.

Replication Exercise

Since the replication material is has originally been created for import to gss and rts files, I have converted the files into csv for using in R, and the relevant files, codes and the R Markdown can all be found on https://github.com/therundhati/Stock_Watson_Business_Cycle_Dynamics. The parts that I have chosen to replicate are listed in order below:
1. Data description filtering
2. Volatility and Persistence
3. Synchronisation
4. Factor-Structural VAR model

Data Description and Filtering

We begin with Table 1 in paper, wherein standard deviations of $100ln(GDP_t/GDP_{t-4})$ for each country is calculated. The code chunk below demonstrates one such value for Canada, and subsequently, the exercise is repeated by changing index and for each country to get results in Table 1.

```r
data1 <- as.data.frame(read.csv("~/Desktop/Course/TSF/Paper/Mine/cngdppc.csv",
    sep = ","))
library(pastecs)
# stat.desc(data1[[1]])
sub1 <- data1[171:211, ]  # to retrieve standard deviation of 1990-2002
stat.desc(sub1)
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
## 4.100000e+01 0.000000e+00 0.000000e+00 2.662871e+04 3.395699e+04 7.328280e+03
##          sum       median         mean      SE.mean CI.mean.0.95          var
## 1.228203e+06 2.943693e+04 2.995618e+04 3.487190e+02 7.047873e+02 4.985802e+06
##      std.dev     coef.var
## 2.232891e+03 7.453858e-02
```

Country	1960-69	1970-79	1980-89	1990-2002
Canada	1.83	1.82	2.67	2.24
France	1.24	1.66	1.27	1.43
Germany	2.56	2.13	1.67	1.53
Italy	2.34	3.14	1.33	1.30
Japan	2.19	3.16	1.57	2.08
UK	1.84	2.48	2.51	1.60
US	2.09	2.74	2.66	1.47

The focus of paper is on economic fluctuations over the horizons relevant for medium term macroeconomic policy and over business cycle horizons. Accordingly, they consider transformations of data that filter out the highest frequency, quarter-to-quarter fluctuations. The methods used in the paper are four-quarter growth rates, band-pass-filtered log GDP, and forecast errors at different forecasts horizons.
In order to get the cyclical component of the data, I use the log transformed per capita GDP data with Hodrick-Prescott filter to extract the cyclical component and plot it. I have demonstrated for UK, and the rest have been done similarly. The same goes for annual four-quarter growth rates following these.

```
## [1] "Hodrik Prescott filtered GDP per capita"
```

```r
data1 <- as.data.frame(read.csv("~/Desktop/Course/TSF/Paper/Mine/ukgdppc.csv",
    sep = ","))
data1.ts <- ts(data1, start = c(1950, 1), end = c(2002, 4), frequency = 4)
sub1 <- window(data1.ts, start = c(1954, 4), end = c(2002, 3))

# install.packages('mFilter')
library(mFilter)
```

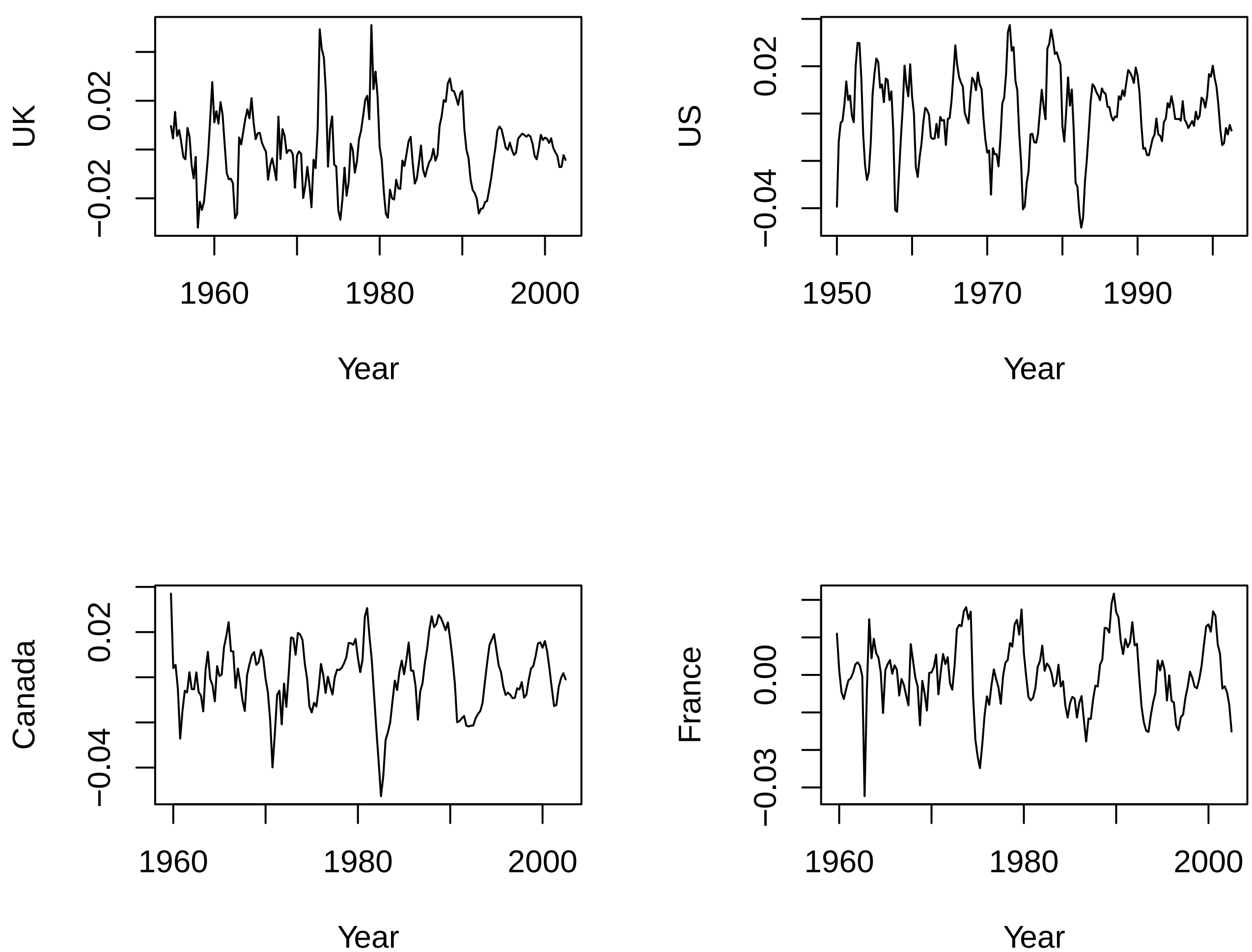
```
## Warning: package 'mFilter' was built under R version 3.5.2
```

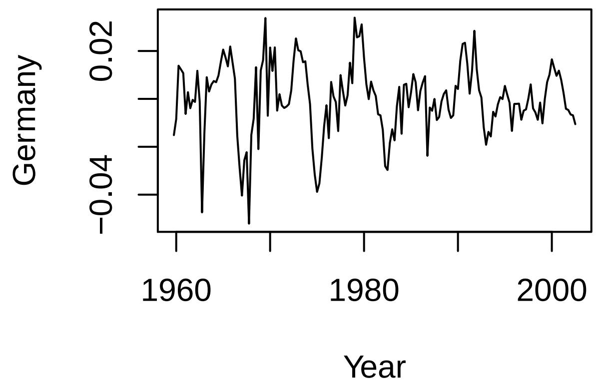
```r
library(dplyr)
```

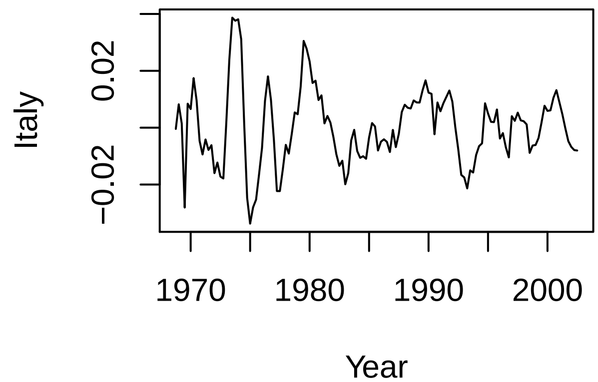
```
## Warning: package 'dplyr' was built under R version 3.5.2

##
## Attaching package: 'dplyr'
```

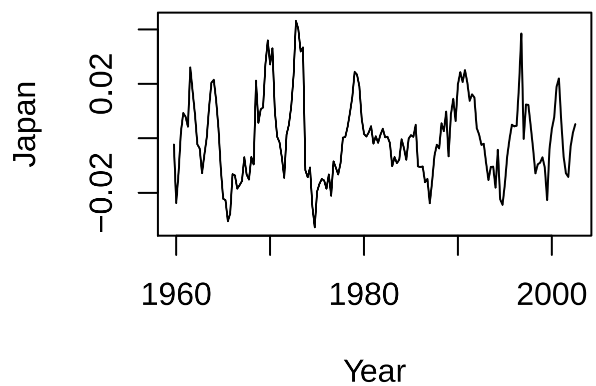
```
## The following objects are masked from 'package:pastecs':
##
##     first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
hp_gdp <- hpfilter(log(sub1), freq = 1600)
# sub1.trans <- mutate(hp = hp_gdp$cycle, lin_cycle =
# log(sub1) - lin_trend)
plot(hp_gdp$cycle, xlab = "Year", ylab = "UK")
```



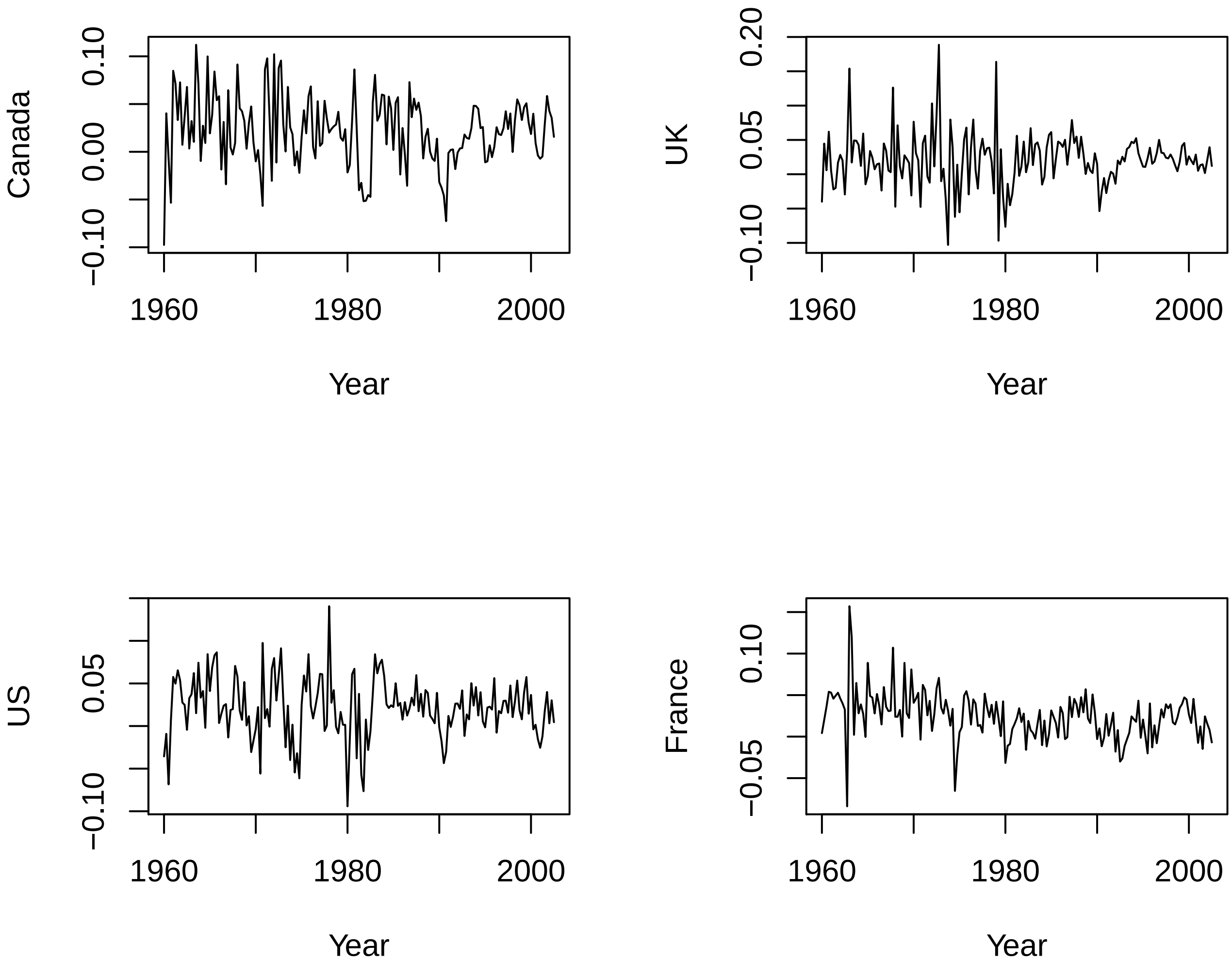


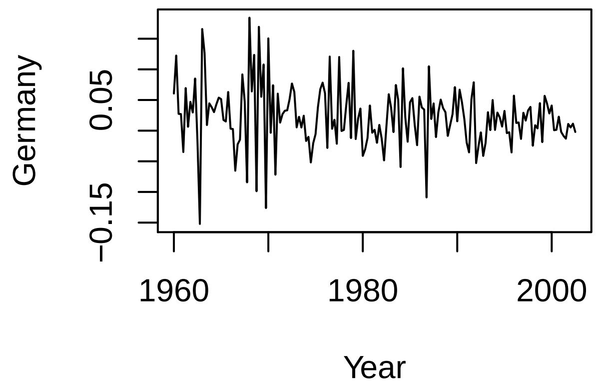


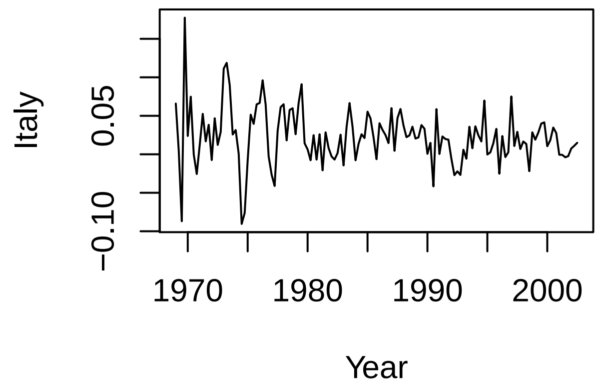


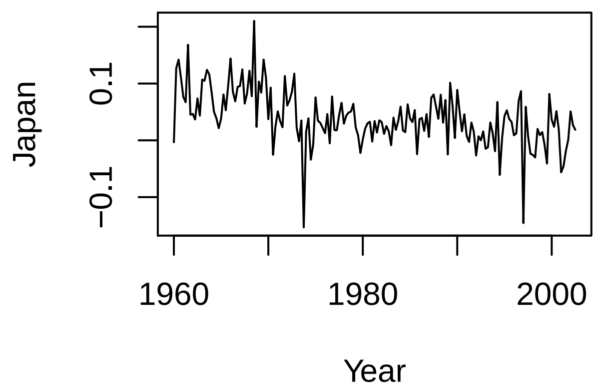

```
## [1] "Annual four-quarter growth rates of GDP per capita"
```

```r
cn <- window(data3.ts, start = c(1959, 4), end = c(2002, 3))
cn_g <- 4 * diff(log(cn))
plot(cn_g, xlab = "Year", ylab = "Canada")
```


Canada
0.10
0.00
−0.10
1960
1980
2000
Year
UK
0.20
0.05
−0.10
1960
1980
2000
Year
US
0.05
−0.10
1960
1980
2000
Year
France
0.10
−0.05
1960
1980
2000
Year







As in the results presented in the paper, there are periods of considerable international synchronization in business cycles. The period of noticeable synchronization appears to be the 1970s, and there is no readily apparent trend towards increased synchronization.

Changes in Volatility and Persistence

Volatility

There has been a substantial moderation in the volatility of economic activity over the past 40 years. For this, we test for breaks in autoregressive parameters. As indicated in the paper, we use an AR(4) process. First we check for a break in AR parameters, as is given in the Conditional Mean column in the paper. Further, we use the Iterative Cumulative Sum of Squares (ICSS) to find the breakpoint in variance (this is in variance of series, not in AR innovations variance). I have demonstrated for one series, the rest are done in a similar manner and the results are tabulated in the form of the succeeding table.

```r
# install.packages('ICSS')
library(ICSS)
library(strucchange)
```

```
## Warning: package 'strucchange' was built under R version 3.5.2
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 3.5.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Warning: package 'sandwich' was built under R version 3.5.2
```

```r
cn <- window(data3.ts, start = c(1959, 4), end = c(2002, 3))
cn_g <- 4 * diff(log(cn))
cn_v <- c(cn_g)
breakpoints(cn_v ~ lag(cn_v, 1) + lag(cn_v, 2) + lag(cn_v, 3) +
    lag(cn_v, 4), breaks = 1)
```

```
##
##   Optimal 2-segment partition:
##
## Call:
## breakpoints.formula(formula = cn_v ~ lag(cn_v, 1) + lag(cn_v,
##     2) + lag(cn_v, 3) + lag(cn_v, 4), breaks = 1)
##
## Breakpoints at observation number:
## 47
##
## Corresponding to breakdates:
## 0.2814371
```

```r
ICSS(cn_v)
```

```
## [1] 56
```

I have refrained from reporting with confidence intervals because of low value of standard error. In the paper, confidence intervals have been reported at 67% significance, which is quite low in itself.

Persistence and Size of Univariate Shocks

To evaluate this, we look at the sum of AR coefficients, and standard error of the regression.

$$y = \alpha(L)\Delta y_t + \epsilon_t$$

From the regression that we run, I report the above said variables in the form of a table. I have demonstrated for one country, and similar analysis is done for other countries.

```
cn <- window(data3.ts, start = c(1959, 4), end = c(1983, 3))
cn_g <- 4 * diff(log(cn))
ar(cn_g, aic = FALSE, order.max = 4)
```

```
##
## Call:
## ar(x = cn_g, aic = FALSE, order.max = 4)
##
## Coefficients:
##       1        2        3        4
##  0.1588  -0.0158   0.0802  -0.0600
##
## Order selected 4  sigma^2 estimated as  0.001782
```

```
cn <- window(data3.ts, start = c(1983, 4), end = c(2002, 3))
cn_g <- 4 * diff(log(cn))
ar(cn_g, aic = FALSE, order.max = 4)
```

```
##
## Call:
## ar(x = cn_g, aic = FALSE, order.max = 4)
##
## Coefficients:
##       1        2        3        4
##  0.3995   0.1441   0.2590  -0.2474
##
## Order selected 4  sigma^2 estimated as  0.0005729
```

Country	Sum of AR coefficients $\hat{\alpha}(1)$, 1960-1983	$\hat{\alpha}(1)$, 1984-2003	Regression standard error $\hat{\sigma}_\epsilon$, 1960-1983	$\hat{\sigma}_\epsilon$, 1984-2003
UK	0.007	0.65	0.05	0.019
US	0.346	0.378	0.04	0.02
Canada	0.007	0.547	0.05	0.024
France	0.105	0.505	0.032	0.018
Germany	0.118	-0.141	0.056	0.035
Italy	-0.123	0.363	0.048	0.028
Japan	0.629	0.339	0.045	0.0406

The standard errors are scaled, but the sum of autoregressive coefficients are nearly consistent with the paper's results. Variance, that it, magnitudes of GDP innovations, declines for countries, but patterns in sum of AR coeff vary greatly accross countries, which represents persistence of innovations. It increases considerably for UK, Canada, France and Italy, is nearly consistent for the US, and declines for Germany and Japan. The trajectory, thus, for all countries is also expected to be different, although I was not able to produce the rolling autoregression plots. The results are consistent with paper's results for this section.

Changes in Synchronization

First we look at correlations of GDP growth across countries as a measure of international output comovements. Then we estimate a reduced form VAR using equation-by-equation seemingly unrelated regression (SUR). If Y_t is the vector of detrended quarterly GDP growth rates, then reduced form VAR is given by

$$Y_t = A(L)Y_{t-1} + \nu_t, E\nu_t\nu_t' = \Sigma$$

where the diagonal elements of the matrix lag polynomial A(L) have degree p_1 and the off-diagonal elements have degree p_2. According to what is given in the paper, we estimate for $p_1 = 4, p_2 = 1$, denoted by VAR(4,1). This is done successively for both 1960-1983, and then 1984-2002.

```
par(mar = rep(2, 4))
print("1959-1983")
```

```
## [1] "1959-1983"
```

```
uk <- window(data1.ts, start = c(1959, 4), end = c(1983, 3))
uk_g <- 4 * diff(log(uk))

us <- window(data2.ts, start = c(1959, 4), end = c(1983, 3))
us_g <- 4 * diff(log(us))

cn <- window(data3.ts, start = c(1959, 4), end = c(1983, 3))
cn_g <- 4 * diff(log(cn))

fr <- window(data4.ts, start = c(1959, 4), end = c(1983, 3))
fr_g <- 4 * diff(log(fr))

bd <- window(data5.ts, start = c(1959, 4), end = c(1983, 3))
bd_g <- 4 * diff(log(bd))

it <- window(data6.ts, start = c(1959, 4), end = c(1983, 3))
it_g <- 4 * diff(log(it))

jp <- window(data7.ts, start = c(1959, 4), end = c(1983, 3))
jp_g <- 4 * diff(log(jp))

main = cbind(uk_g, us_g, cn_g, fr_g, bd_g, it_g, jp_g)
print("Four-quarter growth rate simple correlation coefficients, 1960-1983")
```

```
## [1] "Four-quarter growth rate simple correlation coefficients, 1960-1983"
```

```
cor(main, y = NULL, use = "all.obs", method = c("pearson"))
```

```
##          uk_g      us_g     cn_g     fr_g      bd_g     it_g     jp_g
```

```
## uk_g 1.0000000 0.20846989 0.1533341 0.3766100 0.29977970 0.0307386 0.2774603
## us_g 0.2084699 1.00000000 0.4409412 0.2014419 0.08553932 0.1336788 0.1904469
## cn_g 0.1533341 0.44094123 1.0000000 0.1484954 0.23679024 0.1299871 0.2115093
## fr_g 0.3766100 0.20144187 0.1484954 1.0000000 0.40855439 0.4270677 0.3699540
## bd_g 0.2997797 0.08553932 0.2367902 0.4085544 1.00000000 0.1048604 0.2251494
## it_g 0.0307386 0.13367881 0.1299871 0.4270677 0.10486042 1.0000000 0.2547639
## jp_g 0.2774603 0.19044687 0.2115093 0.3699540 0.22514941 0.2547639 1.0000000
```

```r
# install.packages('vars')
library(vars)
```

```
## Loading required package: MASS

## Warning: package 'MASS' was built under R version 3.5.2

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: urca

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 3.5.2
```

```r
var4 <- VAR(main, p = 4)
restrictm <- matrix(c(1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,
    0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
    1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
    1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0,
    0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1,
    1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
    0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
    1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1,
    1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
    0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
    0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1),
    nrow = 7, ncol = 29, byrow = T)
result <- restrict(var4, method = "manual", resmat = restrictm)
result
```

```
##
## VAR Estimation Results:
## ========================
##
## Estimated coefficients for equation uk_g:
## =========================================
## Call:
```

```
## uk_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
uk_g.l2 + uk_g.l3 + uk_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1 it_g.l1
## -0.10866038 0.27743568 -0.10420422 -0.13980160 -0.04440875 0.10145765
## jp_g.l1 uk_g.l2 uk_g.l3 uk_g.l4 const
## 0.11241625 -0.02663659 0.14205633 -0.10901316 0.01342604
##
##
## Estimated coefficients for equation us_g:
## ==========================================
## Call:
## us_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
us_g.l2 + us_g.l3 + us_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1 it_g.l1
## 0.08808979 0.20747514 0.01625701 -0.02044145 0.03553099 0.04367933
## jp_g.l1 us_g.l2 us_g.l3 us_g.l4 const
## 0.09061278 0.08799897 -0.07976492 -0.01480319 0.01068547
##
##
## Estimated coefficients for equation cn_g:
## ==========================================
## Call:
## cn_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
cn_g.l2 + cn_g.l3 + cn_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1 it_g.l1
## -0.01281851 0.44206049 -0.06719775 0.03260836 0.08865988 0.05118628
## jp_g.l1 cn_g.l2 cn_g.l3 cn_g.l4 const
## 0.08817261 -0.10284066 -0.05518427 -0.04106291 0.01324973
##
##
## Estimated coefficients for equation fr_g:
## ==========================================
## Call:
## fr_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
fr_g.l2 + fr_g.l3 + fr_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1
## 0.1430908591 0.1905424925 -0.0316611511 -0.2057649187 -0.0008018054
## it_g.l1 jp_g.l1 fr_g.l2 fr_g.l3 fr_g.l4
## 0.1974655162 0.0757689103 -0.0628108357 0.1587349860 -0.0944786419
## const
## 0.0197563248
##
##
## Estimated coefficients for equation bd_g:
## ==========================================
## Call:
## bd_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
bd_g.l2 + bd_g.l3 + bd_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1 it_g.l1
```

```
## 0.33678859 0.36962389 0.09925456 -0.14776594 -0.30783538 0.24346557
## jp_g.l1 bd_g.l2 bd_g.l3 bd_g.l4 const
## -0.02452247 -0.07204962 -0.15404914 0.23589809 0.01322589
##
##
## Estimated coefficients for equation it_g:
## =========================================
## Call:
## it_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
it_g.l2 + it_g.l3 + it_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1 it_g.l1
## 0.062203415 0.005470108 0.231645614 -0.148168186 0.005914617 0.239276831
## jp_g.l1 it_g.l2 it_g.l3 it_g.l4 const
## 0.182381415 0.130509538 -0.115559179 -0.115314879 0.017587241
##
##
## Estimated coefficients for equation jp_g:
## =========================================
## Call:
## jp_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
jp_g.l2 + jp_g.l3 + jp_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1 it_g.l1
## 0.07690512 0.01818939 0.03210122 -0.03108392 0.05564832 -0.15040865
## jp_g.l1 jp_g.l2 jp_g.l3 jp_g.l4 const
## 0.16467890 0.28337626 0.20092428 0.02313408 0.01745472
```

```
resultpred <- predict(result)
# fanchart(resultpred)
```

```
## [1] "1983-2002"
```

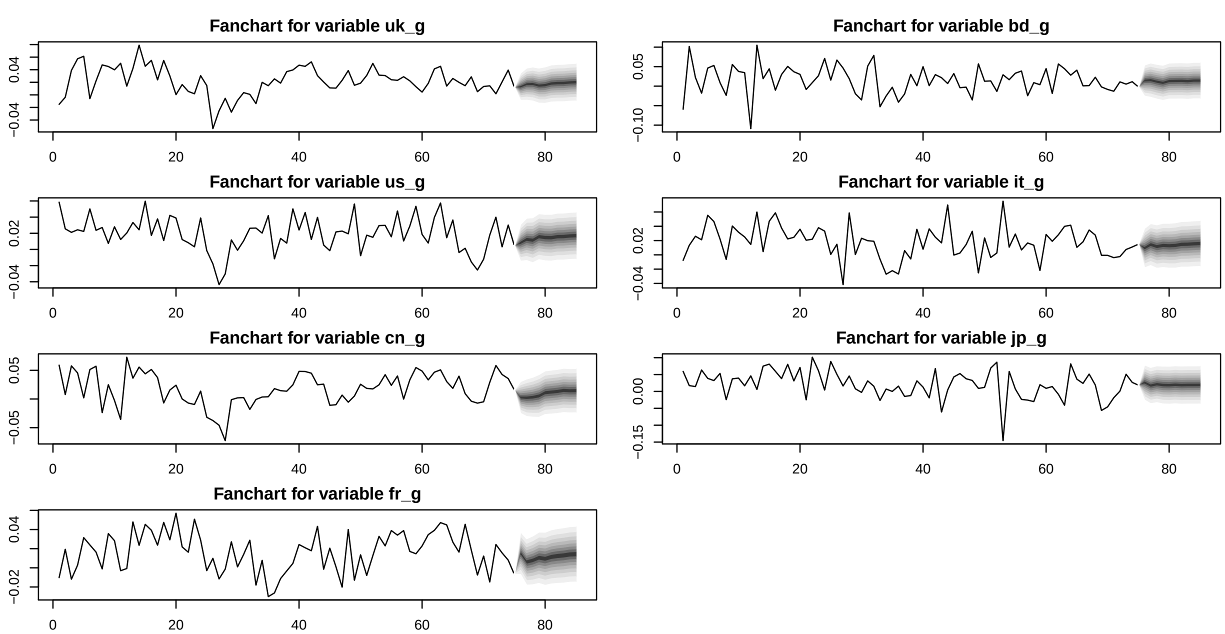
```
## [1] "Four-quarter growth rate simple correlation coefficients, 1984-2002"
```

```
##            uk_g        us_g         cn_g       fr_g        bd_g       it_g
## uk_g 1.00000000  0.29704650  0.348696364 0.23734341  0.07726297 0.27313831
## us_g 0.29704650  1.00000000  0.565791971 0.23714956  0.12104270 0.10564377
## cn_g 0.34869636  0.56579197  1.000000000 0.20875435 -0.11305613 0.14355643
## fr_g 0.23734341  0.23714956  0.208754349 1.00000000  0.41633912 0.56575476
## bd_g 0.07726297  0.12104270 -0.113056134 0.41633912  1.00000000 0.29985937
## it_g 0.27313831  0.10564377  0.143556426 0.56575476  0.29985937 1.00000000
## jp_g 0.05959818 -0.06138004  0.004658491 0.07984053  0.03721004 0.02212214
##              jp_g
## uk_g  0.059598182
## us_g -0.061380040
## cn_g  0.004658491
## fr_g  0.079840528
## bd_g  0.037210043
## it_g  0.022122139
## jp_g  1.000000000
```

```
##
```

```
## VAR Estimation Results:
## =======================
##
## Estimated coefficients for equation uk_g:
## =========================================
## Call:
## uk_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
uk_g.l2 + uk_g.l3 + uk_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1 it_g.l1
## 0.400534341 0.132256086 0.288981467 -0.084986867 -0.030008973 0.061642970
## jp_g.l1 uk_g.l2 uk_g.l3 uk_g.l4 const
## -0.005996417 -0.212325776 0.145945714 0.066747915 0.006747638
##
##
## Estimated coefficients for equation us_g:
## =========================================
## Call:
## us_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
us_g.l2 + us_g.l3 + us_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1 it_g.l1
## 0.467160110 0.007340914 0.080995849 0.295954444 -0.106633098 -0.036230475
## jp_g.l1 us_g.l2 us_g.l3 us_g.l4 const
## 0.019326254 0.052201490 -0.142167816 -0.132240893 0.007420395
##
##
## Estimated coefficients for equation cn_g:
## =========================================
## Call:
## cn_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
cn_g.l2 + cn_g.l3 + cn_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1 it_g.l1
## 0.279529198 0.323153289 0.268577963 0.079556673 -0.100774298 0.091347237
## jp_g.l1 cn_g.l2 cn_g.l3 cn_g.l4 const
## -0.129563360 -0.021369910 0.223464142 -0.267826097 0.001836468
##
##
## Estimated coefficients for equation fr_g:
## =========================================
## Call:
## fr_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
fr_g.l2 + fr_g.l3 + fr_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1
## 0.1001415273 -0.0013613826 0.2329587385 -0.0755942051 -0.0064941701
## it_g.l1 jp_g.l1 fr_g.l2 fr_g.l3 fr_g.l4
## 0.2203801684 0.0696529171 0.1487102480 0.2008031211 0.0122976028
## const
## 0.0007990422
##
##
## Estimated coefficients for equation bd_g:
```

```
## ========================================
## Call:
## bd_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
bd_g.l2 + bd_g.l3 + bd_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1 it_g.l1
## -0.022813733 -0.096269941 0.148464580 0.342432940 -0.285242519 0.272718117
## jp_g.l1 bd_g.l2 bd_g.l3 bd_g.l4 const
## 0.210694852 -0.154664585 -0.169385721 0.149448844 0.007501531
##
##
## Estimated coefficients for equation it_g:
## ========================================
## Call:
## it_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
it_g.l2 + it_g.l3 + it_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1 it_g.l1
## 0.337480497 -0.054791234 0.102645661 0.226477220 -0.040907738 -0.136374104
## jp_g.l1 it_g.l2 it_g.l3 it_g.l4 const
## 0.080874939 0.098639838 0.063432160 -0.011351722 0.004366751
##
##
## Estimated coefficients for equation jp_g:
## ========================================
## Call:
## jp_g = uk_g.l1 + us_g.l1 + cn_g.l1 + fr_g.l1 + bd_g.l1 + it_g.l1 + jp_g.l1 +
jp_g.l2 + jp_g.l3 + jp_g.l4 + const
##
## uk_g.l1 us_g.l1 cn_g.l1 fr_g.l1 bd_g.l1 it_g.l1
## -0.064937500 0.407783034 0.068151882 -0.457777674 0.235942447 0.345858482
## jp_g.l1 jp_g.l2 jp_g.l3 jp_g.l4 const
## -0.019361452 0.038948118 0.233850861 0.094282858 0.003870297
```



This helps us evaluate how VAR coefficients change as a consequence of change in period, and comparison with correlation tells us how much correlation among series is captured by the reduced VAR. Further, the fanchart plots the estimated series (which have again been plotted only for the second case for demonstration purpose, the same can be done fore first case by simply uncommenting the code chunk).

Factor-Structural VAR model

The problem addressed in this section is identifying a world (or G7) shock. In the paper, this is done by defining internationalshocks to be the common components of innovations in the seven-country VAR. A financial crisis that starts in one country but spills over into other G7 financial markets within days would be identified in quarterly FSVAR as a global shock (if it had real effects). Also, an international shock that affects one country first and the others after only a lag of a quarter or more would be misclassified by the FSVAR as an idiosyncratic shock, transmitted via spillovers.
FSVAR is running VAR, where errors have the structure

$$\nu_t = \Gamma f_t + \xi_t$$

$$E(f_t f_t') = diag(\sigma_{f_1}, \sigma_{f_2}, ..., \sigma_{f_k}),$$
$$E(\xi_t \xi_t') = diag(\sigma_{\xi_1}, \sigma_{\xi_2}, ..., \sigma_{\xi_k})$$

where f_t are common international factors, Γ is the $7xk$ matrix of factor loadings, and ξ_t are the country-specific, or idiosyncratic, shocks. Common international shocks are identified as those shocks that affect output in multiple countries contemporaneously. We estimate the FSVAR using Gaussian maximum likelihood. Although I have written the code, I did not understand how specification of the Γ matrix was done in the paper, so I refrained from any further analysis. The code is only representative of the kind of results that we would get should we be able to identify the restriction equivalent to $\frac{k(k-1)}{2}$. Following this, the variance decomposition can be performed by setting horizon time in the function fevd in n.

```
uk <- window(data1.ts, start = c(1969, 1), end = c(2002, 3))
uk_g <- 4 * diff(log(uk))

us <- window(data2.ts, start = c(1969, 1), end = c(2002, 3))
us_g <- 4 * diff(log(us))

cn <- window(data3.ts, start = c(1969, 1), end = c(2002, 3))
cn_g <- 4 * diff(log(cn))

fr <- window(data4.ts, start = c(1969, 1), end = c(2002, 3))
fr_g <- 4 * diff(log(fr))

bd <- window(data5.ts, start = c(1969, 1), end = c(2002, 3))
bd_g <- 4 * diff(log(bd))

it <- window(data6.ts, start = c(1969, 1), end = c(2002, 3))
it_g <- 4 * diff(log(it))

jp <- window(data7.ts, start = c(1969, 1), end = c(2002, 3))
jp_g <- 4 * diff(log(jp))

main <- cbind(uk_g, us_g, cn_g, fr_g, bd_g, it_g, jp_g)
# install.packages('vars')
library(vars)
var4 <- VAR(main, p = 4)
```

```r
restrictm <- matrix(c(1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,
    0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
    1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
    1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0,
    0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1,
    1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
    0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
    1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
    1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
    0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
    0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0),
    nrow = 7, ncol = 29, byrow = T)
result <- restrict(var4, method = "manual", resmat = restrictm)

b = matrix(cbind(1, NA, 0, 0, 0, 0, 0, NA, 1, 0, 0, 0, 0, 0,
    NA, NA, 1, 0, 0, 0, 0, NA, NA, NA, 1, 0, 0, 0, NA, NA, NA,
    NA, 1, 0, 0, NA, NA, NA, NA, NA, 1, 0, NA, NA, NA, NA, NA,
    NA, 1), nrow = 7, ncol = 7)

SVAR(result, Bmat = b)
```

```
## Warning in SVAR(result, Bmat = b): Convergence not achieved after 100
## iterations. Convergence value: 0.00027948572391423 .
```

```
##
## SVAR Estimation Results:
## ========================
##
##
## Estimated B matrix:
##       uk_g   us_g  cn_g   fr_g    bd_g    it_g    jp_g
## uk_g 1.000 0.9847 0.110 0.1118 0.12597 0.09793 0.12901
## us_g 0.949 1.0000 0.129 0.1050 0.10568 0.08981 0.11282
## cn_g 0.000 0.0000 1.000 0.1067 0.09169 0.09539 0.10109
## fr_g 0.000 0.0000 0.000 1.0000 0.11370 0.11071 0.09509
## bd_g 0.000 0.0000 0.000 0.0000 1.00000 0.11066 0.10175
## it_g 0.000 0.0000 0.000 0.0000 0.00000 1.00000 0.08396
## jp_g 0.000 0.0000 0.000 0.0000 0.00000 0.00000 1.00000
```

```r
see <- SVAR(result, Bmat = b)
```

```
## Warning in SVAR(result, Bmat = b): Convergence not achieved after 100
## iterations. Convergence value: 0.00027948572391423 .
```

```r
fevd(see)
```

```
## $uk_g
##           uk_g      us_g       cn_g        fr_g        bd_g        it_g
##  [1,] 0.4910737 0.4761692 0.00594150 0.006139636 0.007792328 0.004709922
##  [2,] 0.4702314 0.4602143 0.01072226 0.030803182 0.007157165 0.006590543
##  [3,] 0.4657270 0.4564402 0.01276131 0.031811344 0.009869002 0.007804381
```

```
##  [4,] 0.4621363 0.4543323 0.01348129 0.032274208 0.010074488 0.009999378
##  [5,] 0.4558235 0.4484858 0.01494941 0.038835991 0.009988877 0.011137163
##  [6,] 0.4542950 0.4471878 0.01573410 0.038725708 0.010308090 0.011931956
##  [7,] 0.4524153 0.4456404 0.01601728 0.039739346 0.010712788 0.012399856
##  [8,] 0.4507505 0.4440836 0.01640748 0.040758441 0.010736805 0.012792448
##  [9,] 0.4501631 0.4435831 0.01663347 0.040680075 0.010895299 0.012982944
## [10,] 0.4494656 0.4429803 0.01673289 0.041065725 0.010961394 0.013123056
##              jp_g
##  [1,] 0.008173638
##  [2,] 0.014281151
##  [3,] 0.015586673
##  [4,] 0.017702076
##  [5,] 0.020779290
##  [6,] 0.021817341
##  [7,] 0.023075090
##  [8,] 0.024470691
##  [9,] 0.025061959
## [10,] 0.025671052
##
## $us_g
##            uk_g      us_g       cn_g        fr_g        bd_g        it_g
##  [1,] 0.4594179 0.5101543 0.00849332 0.005628726 0.005698047 0.004114678
##  [2,] 0.4426161 0.4894578 0.01244804 0.023491810 0.009970802 0.005895509
##  [3,] 0.4374221 0.4832247 0.01554292 0.024747893 0.010408519 0.009053261
##  [4,] 0.4333708 0.4784912 0.01681768 0.027889607 0.010952473 0.010625394
##  [5,] 0.4295718 0.4739709 0.01785769 0.030437837 0.011225491 0.011988400
##  [6,] 0.4274872 0.4715894 0.01862824 0.031094532 0.011678565 0.012687029
##  [7,] 0.4259526 0.4698235 0.01900760 0.031936023 0.011864361 0.013191826
##  [8,] 0.4248435 0.4685271 0.01929140 0.032362051 0.012001706 0.013495875
##  [9,] 0.4241053 0.4676876 0.01949600 0.032593102 0.012098651 0.013681257
## [10,] 0.4235635 0.4670648 0.01961113 0.032785883 0.012207953 0.013798011
##             jp_g
##  [1,] 0.00649308
##  [2,] 0.01611987
##  [3,] 0.01960064
##  [4,] 0.02185286
##  [5,] 0.02494789
##  [6,] 0.02683507
##  [7,] 0.02822415
##  [8,] 0.02947834
##  [9,] 0.03033806
## [10,] 0.03096869
##
## $cn_g
##             uk_g      us_g      cn_g       fr_g        bd_g        it_g
##  [1,] 0.00000000 0.0000000 0.9623611 0.01095754 0.008090624 0.008756476
##  [2,] 0.09979091 0.1108626 0.7204455 0.01392111 0.015513715 0.030052920
##  [3,] 0.12249277 0.1350012 0.6524211 0.02035138 0.015603748 0.038135068
##  [4,] 0.12862509 0.1413985 0.6291484 0.02345916 0.016095558 0.041503743
##  [5,] 0.13738284 0.1508702 0.6051712 0.02597392 0.016588997 0.042995470
##  [6,] 0.14054778 0.1541068 0.5926225 0.02805064 0.016978569 0.043705042
##  [7,] 0.14199673 0.1556083 0.5863165 0.02896910 0.017138177 0.044095496
##  [8,] 0.14321448 0.1568935 0.5816281 0.02974471 0.017308124 0.044313814
##  [9,] 0.14392698 0.1576183 0.5782872 0.03020286 0.017400971 0.044435935
```

```
## [10,] 0.14431675 0.1580225 0.5762482 0.03045553 0.017504412 0.044493411
##             jp_g
##  [1,] 0.009834287
##  [2,] 0.009413183
##  [3,] 0.015994792
##  [4,] 0.019769513
##  [5,] 0.021017336
##  [6,] 0.023988659
##  [7,] 0.025875657
##  [8,] 0.026897306
##  [9,] 0.028127787
## [10,] 0.028959181
##
## $fr_g
##             uk_g       us_g       cn_g      fr_g       bd_g       it_g
##  [1,] 0.00000000 0.00000000 0.00000000 0.9669053 0.01250055 0.01185147
##  [2,] 0.01668462 0.01722698 0.01318500 0.8662976 0.01179654 0.05066629
##  [3,] 0.02703864 0.02851615 0.01752430 0.8286535 0.01476726 0.05572375
##  [4,] 0.03499258 0.03707067 0.01932326 0.7959603 0.01550748 0.06370233
##  [5,] 0.04742643 0.05017148 0.02431252 0.7561311 0.01563089 0.06575129
##  [6,] 0.05270641 0.05595537 0.02541731 0.7404547 0.01655874 0.06588948
##  [7,] 0.05716430 0.06070525 0.02636030 0.7258529 0.01693066 0.06654985
##  [8,] 0.06095873 0.06474138 0.02730546 0.7141465 0.01711984 0.06649275
##  [9,] 0.06271352 0.06665371 0.02761336 0.7084609 0.01741227 0.06642312
## [10,] 0.06428954 0.06832900 0.02791047 0.7032315 0.01754453 0.06641784
##             jp_g
##  [1,] 0.008742702
##  [2,] 0.024143008
##  [3,] 0.027776448
##  [4,] 0.033443362
##  [5,] 0.040576325
##  [6,] 0.043017945
##  [7,] 0.046436739
##  [8,] 0.049235380
##  [9,] 0.050723100
## [10,] 0.052277154
##
## $bd_g
##             uk_g       us_g       cn_g         fr_g      bd_g       it_g
##  [1,] 0.00000000 0.00000000 0.00000000 0.0000000000 0.9779010 0.01197493
##  [2,] 0.04273054 0.04752472 0.01320352 0.0006399896 0.7385525 0.08413423
##  [3,] 0.05421848 0.05941633 0.01888195 0.0063641616 0.7087792 0.08139895
##  [4,] 0.06116449 0.06699259 0.01973721 0.0073299784 0.6889974 0.08300336
##  [5,] 0.06135785 0.06712583 0.02021548 0.0082615609 0.6822220 0.07912425
##  [6,] 0.06925987 0.07560422 0.02170805 0.0092320026 0.6588170 0.08083315
##  [7,] 0.07123753 0.07769364 0.02226508 0.0104046952 0.6530458 0.08013290
##  [8,] 0.07262024 0.07917454 0.02247588 0.0108128052 0.6472769 0.08007083
##  [9,] 0.07358030 0.08017802 0.02269973 0.0111824496 0.6438655 0.07955768
## [10,] 0.07467138 0.08135323 0.02291057 0.0114753368 0.6401144 0.07961400
##             jp_g
##  [1,] 0.01012409
##  [2,] 0.07321451
##  [3,] 0.07094089
##  [4,] 0.07277500
```

```
##  [5,] 0.08169300
##  [6,] 0.08454568
##  [7,] 0.08522039
##  [8,] 0.08756880
##  [9,] 0.08893628
## [10,] 0.08986104
##
## $it_g
##             uk_g       us_g       cn_g       fr_g        bd_g      it_g
##  [1,] 0.00000000 0.00000000 0.00000000 0.00000000 0.000000000 0.9930007
##  [2,] 0.01791925 0.01880503 0.02821958 0.01271737 0.003203420 0.8838012
##  [3,] 0.04182446 0.04463422 0.03891186 0.01672304 0.006864054 0.8084567
##  [4,] 0.05889540 0.06306013 0.04300804 0.02178787 0.008242649 0.7571001
##  [5,] 0.06886093 0.07369168 0.04497427 0.02506991 0.009038695 0.7250158
##  [6,] 0.07484412 0.08012747 0.04579174 0.02675606 0.009840068 0.7061436
##  [7,] 0.07813911 0.08365875 0.04599488 0.02806134 0.010402366 0.6946251
##  [8,] 0.08045681 0.08612817 0.04621899 0.02883571 0.010688892 0.6862622
##  [9,] 0.08191035 0.08768724 0.04635921 0.02932911 0.010922625 0.6808877
## [10,] 0.08289302 0.08873861 0.04642721 0.02967595 0.011107615 0.6770886
##             jp_g
##  [1,] 0.00699927
##  [2,] 0.03533416
##  [3,] 0.04258564
##  [4,] 0.04790584
##  [5,] 0.05334868
##  [6,] 0.05649692
##  [7,] 0.05911841
##  [8,] 0.06140927
##  [9,] 0.06290382
## [10,] 0.06406900
##
## $jp_g
##             uk_g       us_g        cn_g         fr_g       bd_g        it_g
##  [1,] 0.00000000 0.00000000 0.000000000 0.0000000000 0.00000000 0.000000000
##  [2,] 0.02318609 0.02467064 0.004115803 0.0001698827 0.05416937 0.004261575
##  [3,] 0.03204454 0.03449511 0.005228061 0.0010414297 0.05211633 0.005546808
##  [4,] 0.03326249 0.03575386 0.005386283 0.0012821832 0.04793134 0.005047657
##  [5,] 0.04138648 0.04438821 0.006689257 0.0013170296 0.05153719 0.004900199
##  [6,] 0.04407736 0.04735369 0.007224061 0.0021508745 0.05228598 0.005086498
##  [7,] 0.04686847 0.05033323 0.007742709 0.0023612695 0.05100979 0.005320386
##  [8,] 0.04960611 0.05324258 0.008289846 0.0026711458 0.05163650 0.005313544
##  [9,] 0.05086712 0.05462031 0.008562013 0.0030621840 0.05161337 0.005505586
## [10,] 0.05208763 0.05591825 0.008823947 0.0032237494 0.05158865 0.005600060
##            jp_g
##  [1,] 1.0000000
##  [2,] 0.8894266
##  [3,] 0.8695277
##  [4,] 0.8713362
##  [5,] 0.8497816
##  [6,] 0.8418215
##  [7,] 0.8363641
##  [8,] 0.8292403
##  [9,] 0.8257694
## [10,] 0.8227577
```

Results and Discussion

I have tried to capture the key aims of the paper using the following techniques:
1. Extracting the cyclical component of per capita GDP time series data using two mehthods- filtering, and by calculating annual quarterly growth rates.
2. Evaluating volatility of time series by looking at change in AR sum of coefficients and variance and identifying breakpoints which indicate a structural break in the series, and persistence of univariate shocks.
3. Using comovement measures to identify synchronisation using a VAR model for two chunks of time, and by looking at correlation coefficients of time series.
4. Using FSVAR to see how international and idiosyncratic shocks percolate in the time series.
Other results have been presented section wise in the replication exercise itself.

Differences

For some sections, I did not grasp what has been done in the paper, and I think I ended up doing something entirely else.
- The log difference series that I have used in all models is perhaps different from the one used in paper, which used the estimate of Kalman smoother and a country specific unbiased median detrending parameter, which I could not understand from what has been explained in the paper. I have taken the annual logarithmic growth rates of countries, given by $4ln(\frac{GDP_t}{GDP_{t-1}})$.
- I have refrained from plotting repetitive elements so that the methods are more highlighted in the paper, and plots can be retrieved from any code chunk. For example, I have plotted only one fanchart as a representation of what information the plot conveys.
- I have not replicated any part pertaining to spectral densiy or spectrum width calculation, but it was presented to evaluate some key arguments and model extensions in the paper.
- I have skipped all Band Pass filtered analysis and rolling estimates of methods like autoregression. This is primarily because the series generated using Hodrick-Prescott filter varied from the one in the paper, which was using the Band-Pass filter. The same analysis was done for both BP-filter and annual quarterly growth rate series, and I presented the latter one. - Since I was not sure of the FSVAR method, I presented the initial methodology as I felt would be correct, but did not proceed with the same, because I could not find the model specification in the paper.

Challenges

Dealing with data and codes in a format different than the usual was pretty much the only challenge in this exercise. Also, the methods were outdated, which meant less documentation of the same. Lastly, the series has been mislabelled and often is not what it is supposed to be. Sometimes results don't match because of this, so I have manipulated the original gdp series only, which again is hard to cross check with in the methods used in paper. Discussion in paper itself is sometimes ambiguous, so it is tough to understand what is going on when working with data. Soem concepts were out of the scope of what I have studied and I ended up struggling with them.