Macroeconomics

Jay Wei

2023-07-13

Contents

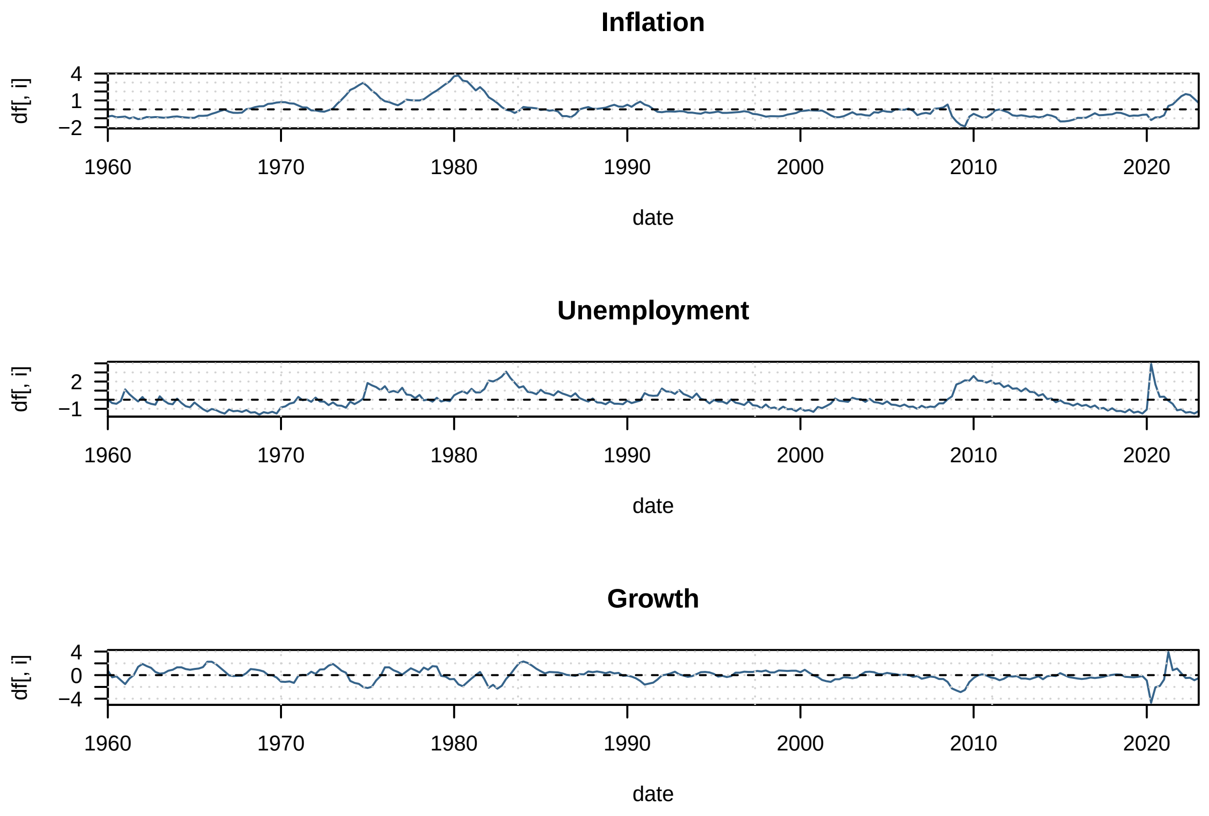
1 Output, Unemployment, and the Inflation Rate

```r
suppressMessages(library("quantmod"))
suppressMessages(library(stargazer))
getSymbols(c("CPALTT01USQ659N","LRUN64TTUSQ156N","A191RO1Q156NBEA","A191RL1Q225SBEA")
           ,src="FRED")
```

```
## [1] "CPALTT01USQ659N" "LRUN64TTUSQ156N" "A191RO1Q156NBEA" "A191RL1Q225SBEA"
```

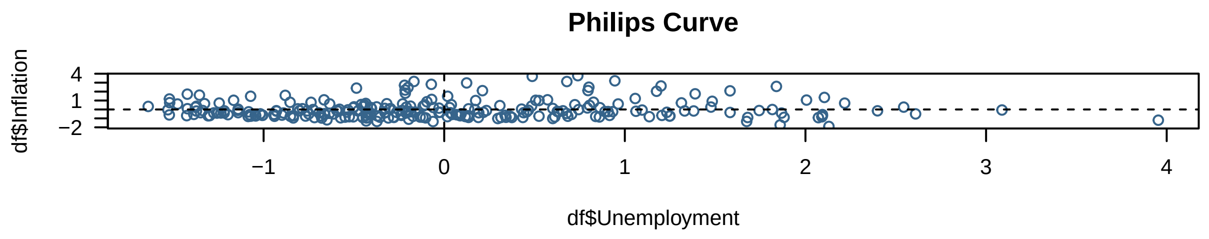
```r
X = cbind(CPALTT01USQ659N,LRUN64TTUSQ156N,A191RO1Q156NBEA)
X = na.omit(X)
df = data.frame(X)
colnames(df) = c("Inflation","Unemployment","Growth")
date = as.Date(index(X))
k = ncol(df)
# Standardization
for (i in 1:k) {
  df[,i] = scale(df[,i],TRUE,TRUE)
}
```

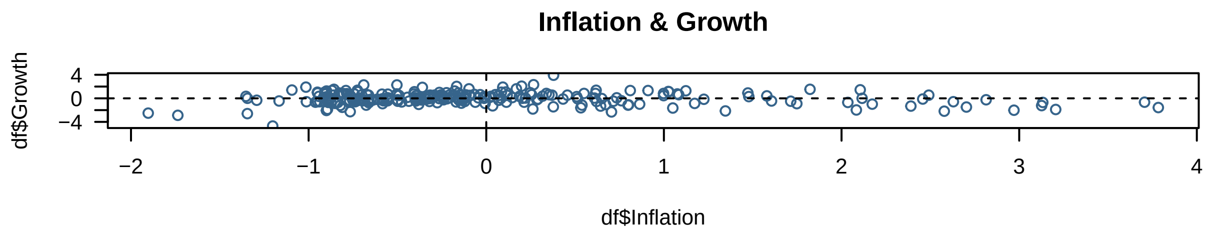
```r
par(mfrow=c(k,1))
for (i in 1:k) {
  plot(date,df[,i],type="l",xaxs="i",las=1,main=colnames(df)[i],col="steelblue4")
  grid()
  abline(h=0,lty=2)
}
```

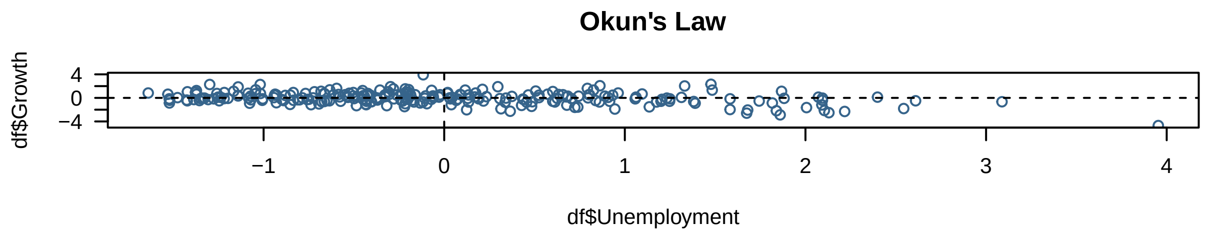


```r
par(mfrow=c(k,1))
plot(df$Unemployment,df$Inflation,las=1,col="steelblue4",main="Philips Curve")
abline(h=0,v=0,lty=2)
```

```
plot(df$Inflation,df$Growth,las=1,col="steelblue4",main="Inflation & Growth")
abline(h=0,v=0,lty=2)
plot(df$Unemployment,df$Growth,las=1,col="steelblue4",main="Okun's Law")
abline(h=0,v=0,lty=2)
```







1.1 Okun's Law

Intuition suggests that if output growth is high, unemployment will decrease, and this is indeed true. This relation was first examined by American economist Arthur Okun and for this reason has become known as Okun's law.

Remark 1.

1. Okun's law is, of course, not a law, but an empirical regularity.

2. The key to decreasing unemployment is a high enough rate of growth.

```
### OKUNS LAW
par(mfrow=c(1,1))
plot(df$Unemployment,df$Growth,las=1,col="steelblue4")
abline(h=0,v=0,lty=2)
lm1 = lm(df$Growth~df$Unemployment,data=df)
stargazer(lm1, title = "Okun's Law Regression Table", header = FALSE)
```

```
# PREDICTED CONFIDENCE INTERVAL
pred1 = predict(lm1, interval = "prediction", level=0.9) # prediction interval
```

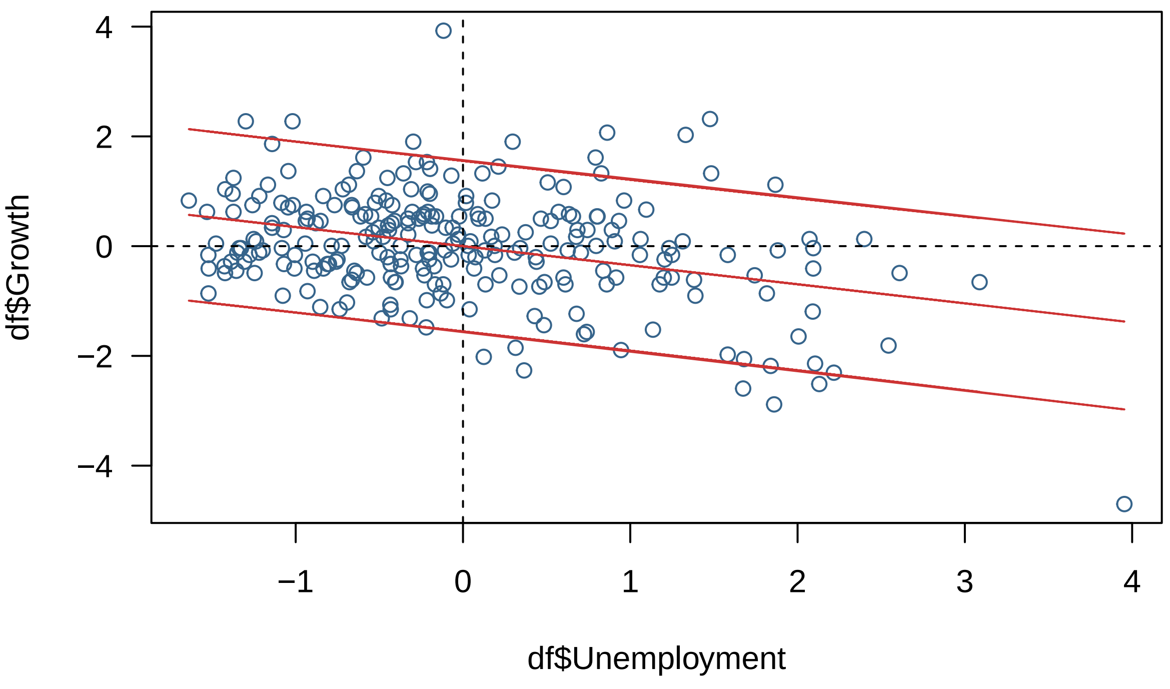
```
## Warning in predict.lm(lm1, interval = "prediction", level = 0.9): predictions on curren
```

```
lines(df$Unemployment,pred1[,1],col="brown3")
lines(df$Unemployment,pred1[,2],col="brown3")
```

Table 1: Okun's Law Regression Table

	Dependent variable:
	Growth
Unemployment	-0.347^{***}
	(0.059)
Constant	0.000
	(0.059)
Observations	253
R^2	0.121
Adjusted R^2	0.117
Residual Std. Error	0.940 (df = 251)
F Statistic	34.451^{***} (df = 1; 251)
Note:	*p<0.1; **p<0.05; ***p<0.01

```
lines(df$Unemployment,pred1[,3],col="brown3")
```



1.2 Inflation and Growth

```
### INFLATION & GROWTH
Df = df[-1,] # remove the first row
inf = embed(df$Inflation,2) # embed the time series into 2 dim'l Euclidean space
Df$dINFL = inf[,1]-inf[,2] # first difference of inflation
```

```r
par(mfrow=c(1,1))
plot(Df$Growth,Df$dINFL,las=1,col="steelblue4")
abline(h=0)
lm1 = lm(Df$dINFL~Df$Growth)
stargazer(lm1, title = "Inflation-Growth Regression Table", header = FALSE)
```

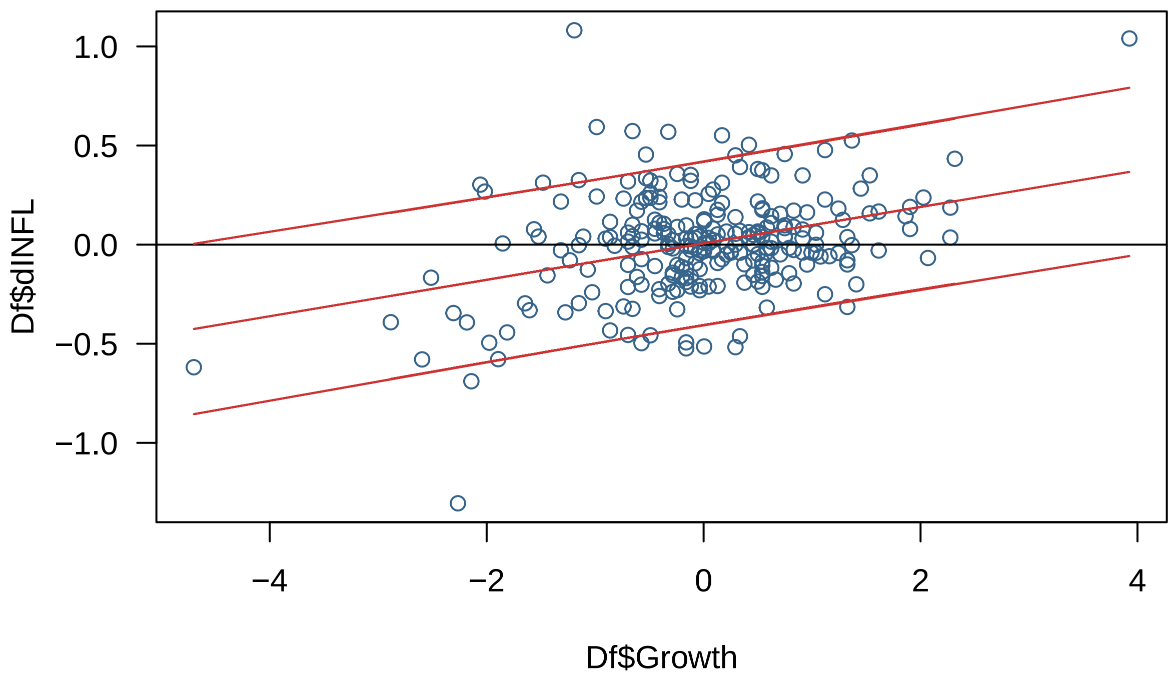
Table 2: Inflation-Growth Regression Table

	Dependent variable:
	dINFL
Growth	0.092***
	(0.016)
Constant	0.006
	(0.016)
Observations	252
R^2	0.120
Adjusted R^2	0.117
Residual Std. Error	0.249 (df = 250)
F Statistic	34.211*** (df = 1; 250)
Note:	*p<0.1; **p<0.05; ***p<0.01

```r
# PREDICTED CONFIDENCE INTERVAL
pred1 = predict(lm1, interval = "prediction", level=0.9) # prediction interval
```

```
## Warning in predict.lm(lm1, interval = "prediction", level = 0.9): predictions on curren
```

```r
lines(Df$Growth,pred1[,1],col="brown3")
lines(Df$Growth,pred1[,2],col="brown3")
lines(Df$Growth,pred1[,3],col="brown3")
```



1.3 The Philips Curve

When unemployment becomes very low, the economy is likely to overheat, and that this will lead to upward pressure on inflation. And, to a large extent, this is true.

This relation was first explored in 1958 by a New Zealand economist, A. W. Phillips, and has become known as the Phillips curve. Phillips plotted the rate of inflation against the unemployment rate.

Since then, the Phillips curve has been redefined as a relation between the *change in the rate of inflation* and the unemployment rate.

```
### PHILIPS CURVE
par(mfrow=c(1,1))
plot(Df$Unemployment,Df$dINFL,las=1,col="steelblue4")
abline(h=0)
lm1 = lm(Df$dINFL~Df$Unemployment)
stargazer(lm1, title = "Philips Curve Regression Table", header = FALSE)
```

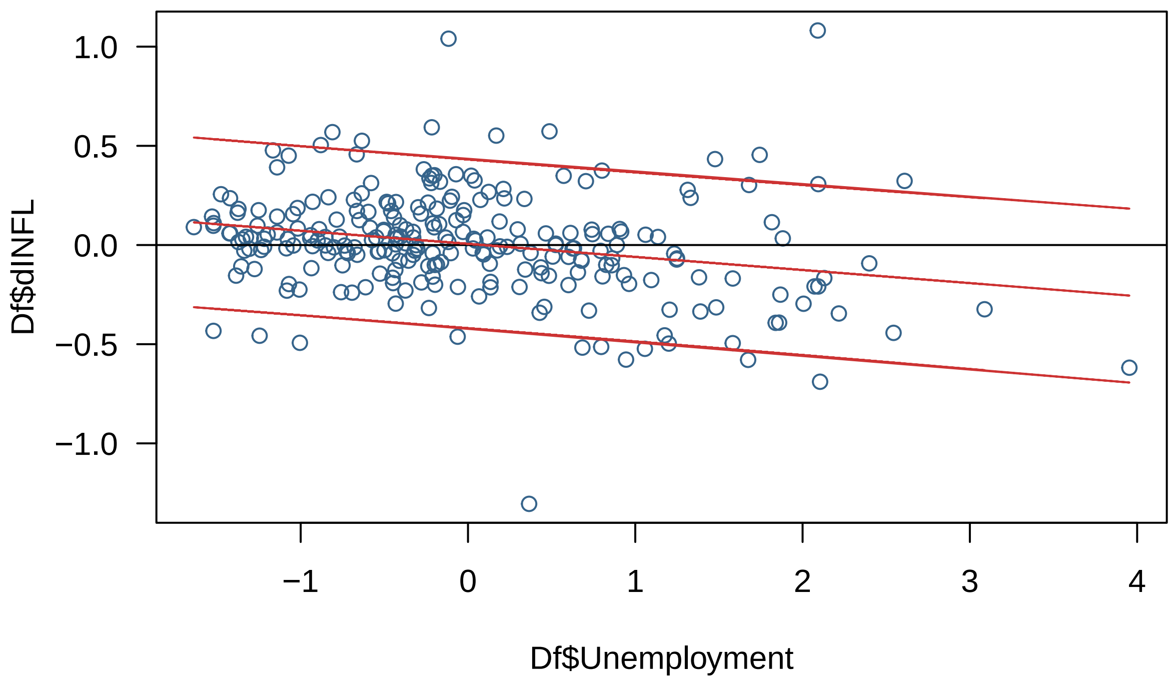
```
# PREDICTED CONFIDENCE INTERVAL
pred1 = predict(lm1, interval = "prediction",level=0.9)
```

```
## Warning in predict.lm(lm1, interval = "prediction", level = 0.9): predictions on curren
```

```
lines(Df$Unemployment,c(pred1[,1]),col="brown3")
lines(Df$Unemployment,c(pred1[,2]),col="brown3")
lines(Df$Unemployment,c(pred1[,3]),col="brown3")
```

Table 3: Philips Curve Regression Table

	Dependent variable:
	dINFL
Unemployment	-0.066^{***}
	(0.016)
Constant	0.006
	(0.016)
Observations	252
R^2	0.062
Adjusted R^2	0.059
Residual Std. Error	0.257 (df = 250)
F Statistic	16.599^{***} (df = 1; 250)
Note:	*p<0.1; **p<0.05; ***p<0.01



1.4 VAR(p)

A Vector autoregressive (VAR) model is useful when one is interested in predicting multiple time series variables using a single model. At its core, the VAR model is an extension of the univariate autoregressive model.

A VAR model comprises one equation per variable in the system. The right hand side of each equation includes a constant and lags of all of the variables in the system.

If the series are stationary, we forecast them by fitting a VAR to the data directly (known as a "VAR in levels”). If the series are non-stationary, we take differences of the data in order to make them stationary, then fit a VAR model (known as a “VAR in differences”). In both cases, the models are estimated equation by equation using the principle of least squares.

The other possibility is that the series may be non-stationary but cointegrated, which means that there exists a linear combination of them that is stationary. In this case, a VAR specification that includes an error correction mechanism (usually referred to as a vector error correction model, VECM) should be included, and alternative estimation methods to least squares estimation should be used.

Question 1. *Why bother with VAR?*

When there is no good reason to assume a one-way causal relationship between two time series variables we may think of their relationship as one of mutual interaction.

Such feedback relationships are allowed for in the vector autoregressive (VAR) framework. In this framework, all variables are treated symmetrically. They are all modelled as if they all influence each other equally. In more formal terminology, all variables are now treated as “endogenous”.

Below, we simulate VAR(1) and SVAR(1) models.

Notice that we’re specifying the model of the form:

$$A(L)y(t) = B(L)w(t) + C(L)u(t) + \text{TREND}(t)$$

```r
## simulate a bivariate VAR(1) model

suppressMessages(library(dse))
suppressMessages(library(vars))

k <- 2 # k equations
p <- 1 # VAR(p=1)
```

For model 1, we have

```r
## lag-polynomial A(L) of the structural VAR model
##   B0*y = B*y(-1) + eps
##   where B0=[1 0;
##            -0.8 1] and
##         B1=[0.5 0.2;
##            0.2 0.34]
## so that the reduced form VAR is
##   y = A1*y(-1) + e
##   where A1 = B0^{-1}*B1 = [0.5 0.2;
##                            0.6 0.5]
B0 <- matrix(c(1, -0.8, 0, 1), 2, 2); B0
```

```
##      [,1] [,2]
## [1,]  1.0    0
## [2,] -0.8    1
```

```r
B1 <- matrix(c(0.5, 0.2, 0.2, 0.34), 2,2); B1
```

```
##      [,1] [,2]
## [1,]  0.5 0.20
## [2,]  0.2 0.34
```

```
A1 <- solve(B0)%*%B1; A1
```

```
##      [,1] [,2]
## [1,]  0.5  0.2
## [2,]  0.6  0.5
```

```
diag(2)-A1
```

```
##      [,1] [,2]
## [1,]  0.5 -0.2
## [2,] -0.6  0.5
```

```
# Specifying the auto-regression polynomial array
mat.A.1 <- array(c(1.0, -0.5,
                   0.0, -0.2,
                   0.0, -0.6,
                   1.0, -0.5),
                 c(p+1, k, k))
```

This is specifying:

$$A(L) = \begin{bmatrix} 1.0 - 0.5L & 0 - 0.2L \\ 0 - 0.6L & 1 - 0.5L \end{bmatrix}$$

Notice that $\Sigma_\epsilon = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

and $A_1\epsilon_t = e_t$; thus $\Sigma_e = A_1\Sigma_\epsilon A_1'$

```
# Sigma_e
A1%*%diag(2)%*%t(A1)
```

```
##      [,1] [,2]
## [1,] 0.29 0.40
## [2,] 0.40 0.61
```

For model 2, we have

```
## lag-polynomial A(L) of the structural VAR model
##   B0*y = B1*y(-1) + eps
##   where B0=[1 0;
##             0.5 1] and
##   B1=[0.6 -0.2;
##         0.1 0.5]
## so that the reduced form VAR is
##   y = A1*y(-1) + e
##   where A1 = B0^{-1}*B1 = [0.6 -0.2;
##                            -0.2 0.6]
B0 <- matrix(c(1, 0.5, 0, 1), 2, 2); B0
```

```
##      [,1] [,2]
## [1,]  1.0    0
## [2,]  0.5    1
```

```
B1 <- matrix(c(0.6, 0.1, -0.2, 0.5), 2,2); B1
```

```
##      [,1] [,2]
## [1,]  0.6 -0.2
## [2,]  0.1  0.5
```

```
A1 <- solve(B0)%*%B1; A1
```

```
##      [,1] [,2]
## [1,]  0.6 -0.2
## [2,] -0.2  0.6
```

```
diag(2) -A1
```

```
##      [,1] [,2]
## [1,]  0.4  0.2
## [2,]  0.2  0.4
```

```
# Specifying the auto-regression polynomial array
mat.A.2 <- array(c(1.0, 0.4,
                   0.0, 0.2,
                   0.0, 0.2,
                   1.0, 0.4),
                 c(p+1, k, k))
```

This is specifying:

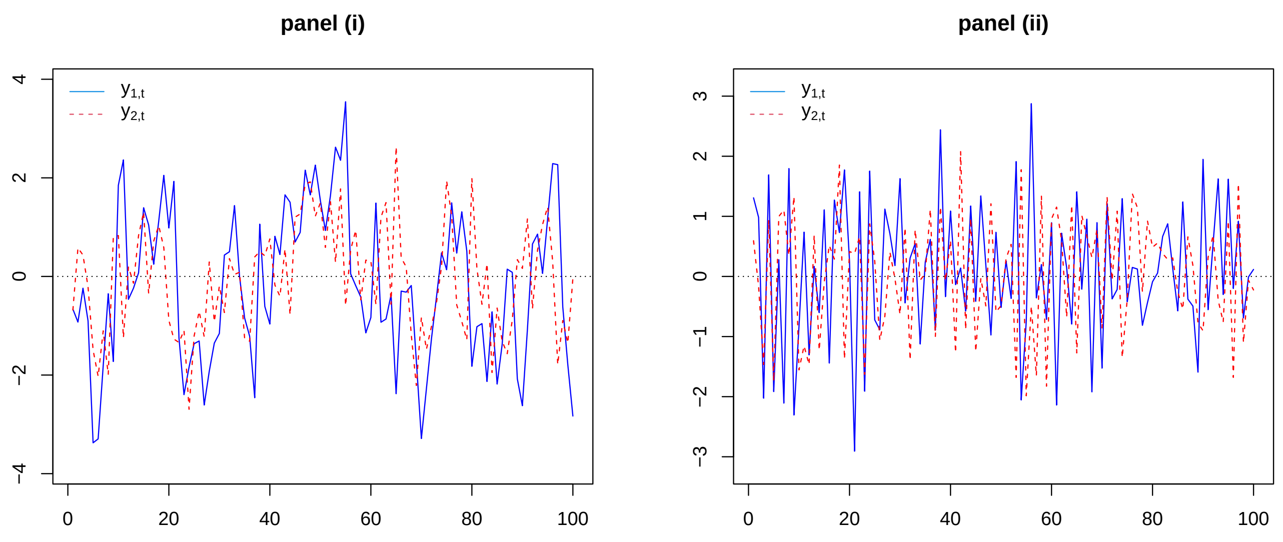
$$A(L) = \begin{bmatrix} 1.0 + 0.4L & 0 + 0.2L \\ 0 + 0.2L & 1 + 0.4L \end{bmatrix}$$

```
## variance-covariance matrix for shocks
# Specifing the moving-average polynomial array
mat.B <- diag(2); mat.B
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
```

This is specifying:

$$B(L) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



A simulated path of $\mathbf{y}_t$ with 100 observations (model 1) is in panel (i) - note that $\{y_{1,t}\}$ and $\{y_{2,t}\}$ tend to move in the same direction (Why? Check $A1$ matrix).

A simulated path of $\mathbf{y}_t$ with 100 observations (model 2) is in panel (ii) - note that $\{y_{1,t}\}$ and $\{y_{2,t}\}$ tend to move in the opposite direction (Why? Check $A1$ matrix).

1.4.1 Lag Selection

- VAR(p) has $k + pk^2$ parameters
- each additional lag introduces additional k^2 parameters, model thus become overparameterized quickly, and a lot of parameters will be insignificant
- information criteria - choose p to minimize AIC, SIC

1.5 Forecasting VAR

```
suppressMessages(library("vars"))
df = df[,c(3,2,1)] # Reorder the columns, this matters!!
```

```
# estimate VAR(p) using AIC to select p
VARselect(df, lag.max = 8, type = "const")|> print(digits=4)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      6      6      5      6
##
## $criteria
##                1         2         3         4         5          6         7
## AIC(n) -5.449501 -5.64926 -5.688414 -5.931043 -6.848560 -6.9354832 -6.894414
## HQ(n)  -5.380442 -5.52841 -5.515766 -5.706601 -6.572324 -6.6074531 -6.514590
## SC(n)  -5.278011 -5.34915 -5.259688 -5.373699 -6.162599 -6.1209047 -5.951218
## FPE(n)  0.004298  0.00352  0.003385  0.002657  0.001062  0.0009736  0.001015
##                8
## AIC(n) -6.852814
## HQ(n)  -6.421196
## SC(n)  -5.781001
## FPE(n)  0.001059
```

We use p=1 to make our life easier though.

```
var1 = VAR(df,p=1) # Fitting VAR(1) model
summary(var1)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: Growth, Unemployment, Inflation
## Deterministic variables: const
## Sample size: 252
## Log Likelihood: -375.64
## Roots of the characteristic polynomial:
## 0.951 0.8476 0.8476
## Call:
## VAR(y = df, p = 1)
##
##
## Estimation results for equation Growth:
## =======================================
## Growth = Growth.l1 + Unemployment.l1 + Inflation.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## Growth.l1        0.820453   0.040354  20.331  < 2e-16 ***
## Unemployment.l1  0.139361   0.040266   3.461 0.000633 ***
## Inflation.l1    -0.120815   0.037886  -3.189 0.001612 **
## const           -0.005759   0.037574  -0.153 0.878307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.5965 on 248 degrees of freedom
## Multiple R-Squared: 0.649,   Adjusted R-squared: 0.6448
## F-statistic: 152.9 on 3 and 248 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation Unemployment:
## =============================================
## Unemployment = Growth.l1 + Unemployment.l1 + Inflation.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## Growth.l1       -0.104390   0.032905  -3.172   0.0017 **
## Unemployment.l1  0.829536   0.032833  25.266   <2e-16 ***
## Inflation.l1     0.061434   0.030892   1.989   0.0478 *
## const           -0.003793   0.030638  -0.124   0.9016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.4864 on 248 degrees of freedom
## Multiple R-Squared: 0.7672,  Adjusted R-squared: 0.7644
## F-statistic: 272.5 on 3 and 248 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation Inflation:
## ==========================================
## Inflation = Growth.l1 + Unemployment.l1 + Inflation.l1 + const
##
##                 Estimate Std. Error t value Pr(>|t|)
## Growth.l1        0.06029    0.01730   3.485 0.000582 ***
## Unemployment.l1 -0.01404    0.01726  -0.813 0.416876
## Inflation.l1     0.97267    0.01624  59.881  < 2e-16 ***
## const            0.00590    0.01611   0.366 0.714496
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.2557 on 248 degrees of freedom
## Multiple R-Squared: 0.9355,  Adjusted R-squared: 0.9347
## F-statistic:  1198 on 3 and 248 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##               Growth Unemployment Inflation
## Growth        0.35577     -0.12505   0.03704
## Unemployment -0.12505      0.23654  -0.02658
## Inflation     0.03704     -0.02658   0.06540
##
## Correlation matrix of residuals:
##              Growth Unemployment Inflation
## Growth        1.0000      -0.4311    0.2428
## Unemployment -0.4311       1.0000   -0.2137
## Inflation     0.2428      -0.2137    1.0000
```

The model equations are:

$$\text{Growth} = \beta_{10} + \beta_{11}\text{Growth}_{t-1} + \beta_{12}\text{Unemployment}_{t-1} + \beta_{13}\text{Inflation}_{t-1} + u_{1t} \quad (1)$$
$$\text{Unemployment} = \beta_{20} + \beta_{21}\text{Growth}_{t-1} + \beta_{22}\text{Unemployment}_{t-1} + \beta_{23}\text{Inflation}_{t-1} + u_{2t} \quad (2)$$
$$\text{Inflation} = \beta_{30} + \beta_{31}\text{Growth}_{t-1} + \beta_{32}\text{Unemployment}_{t-1} + \beta_{33}\text{Inflation}_{t-1} + u_{3t} \quad (3)$$

Notice that u_{1t}, u_{2t}, and u_{3t} can be correlated. We can NOT make causal statements in this case (more on this later).

We end up with the following estimation results:

$$\widehat{\text{Growth}} = -0.0058_{(0.04)} + 0.82_{(0.04)}\text{Growth}_{t-1} + 0.14_{(0.04)}\text{Unemployment}_{t-1} - 0.12_{(0.04)}\text{Inflation}_{t-1} \tag{4}$$

$$\widehat{\text{Unemployment}} = -0.0038_{(0.03)} - 0.10_{(0.03)}\text{Growth}_{t-1} + 0.83_{(0.03)}\text{Unemployment}_{t-1} + 0.06_{(0.03)}\text{Inflation}_{t-1} \tag{5}$$

$$\widehat{\text{Inflation}} = 0.0059_{(0.016)} + 0.06_{(0.02)}\text{Growth}_{t-1} - 0.01_{(0.02)}\text{Unemployment}_{t-1} + 0.97_{(0.02)}\text{Inflation}_{t-1} \tag{6}$$

```r
plot(var1, lag.acf=16, lag.pacf=16)
```

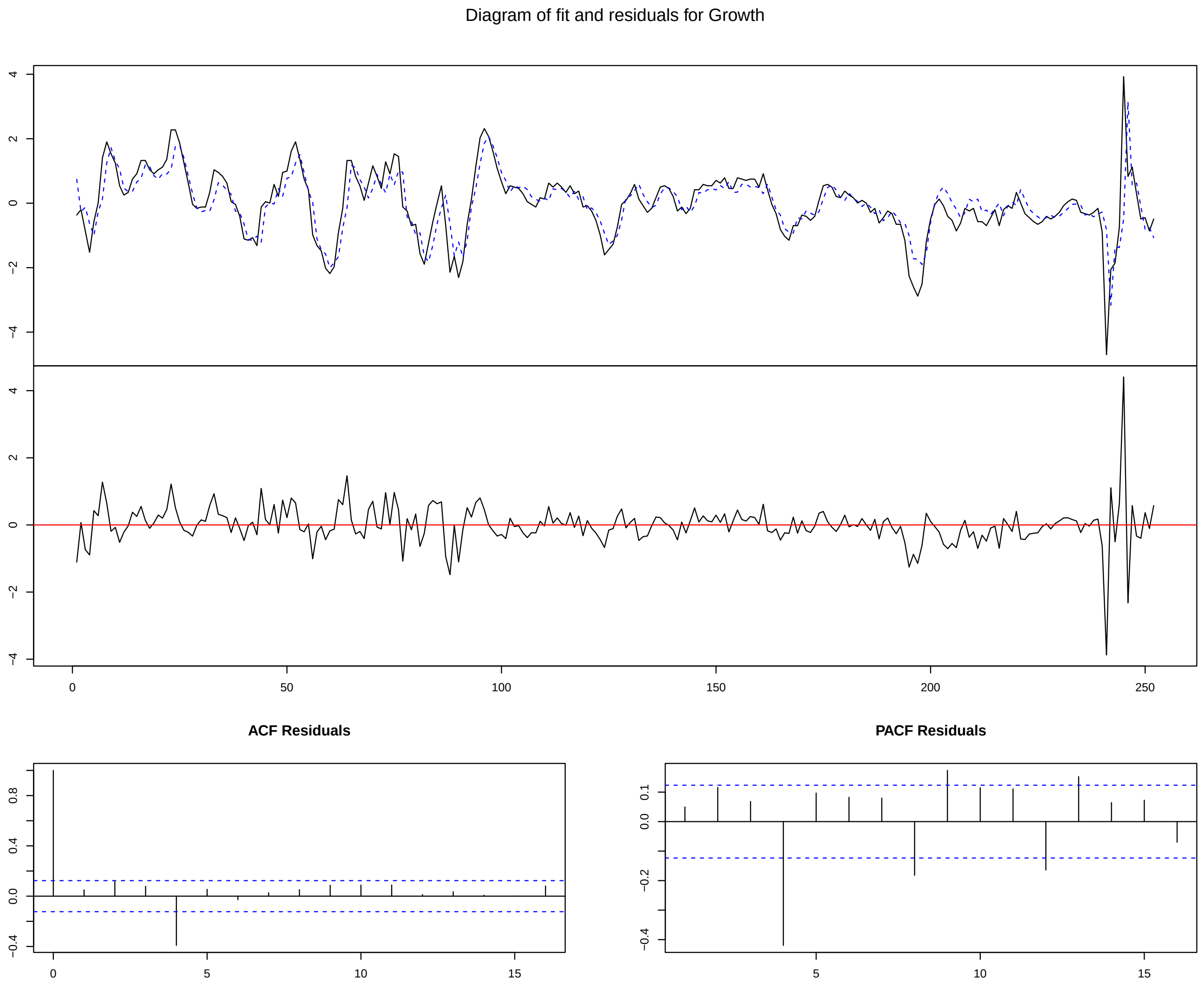
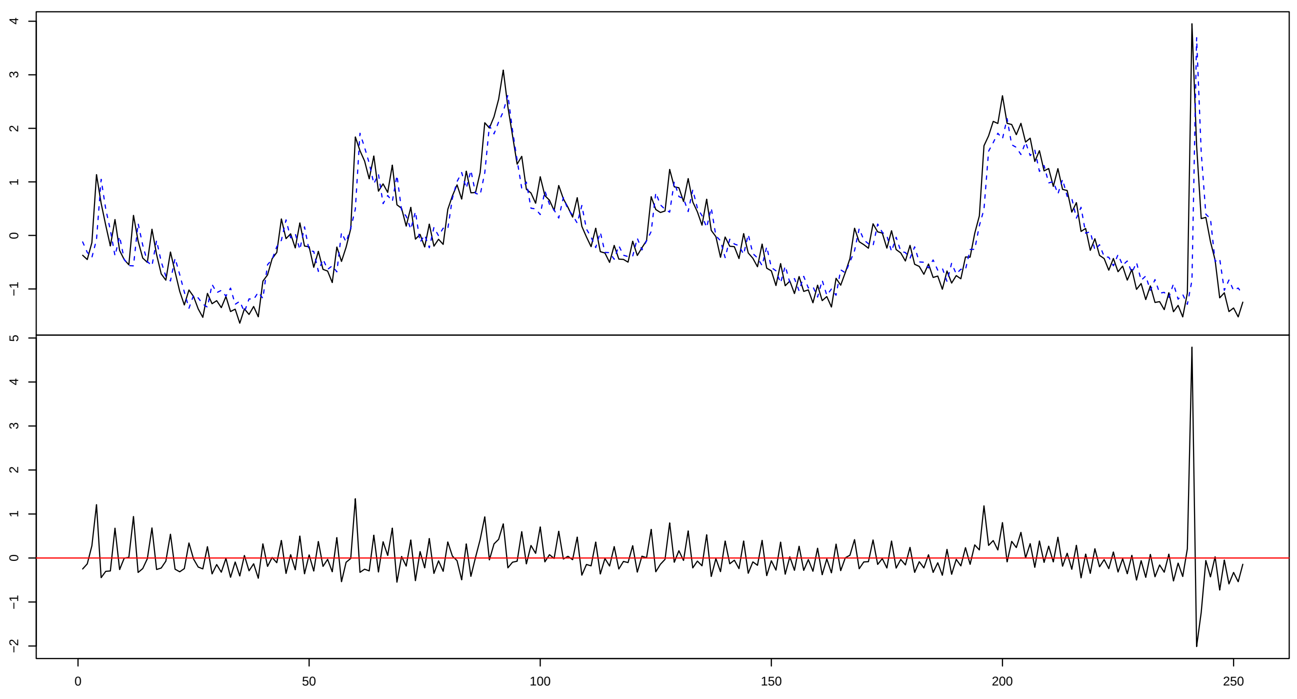
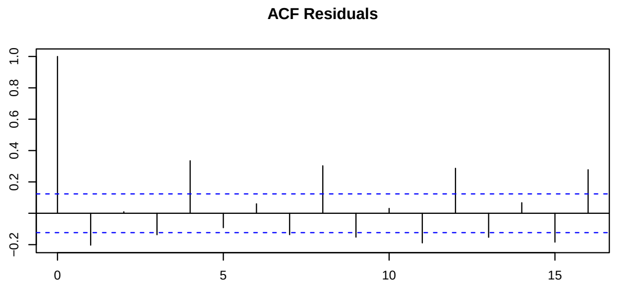
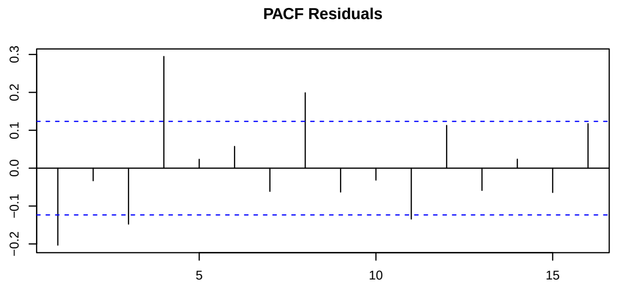
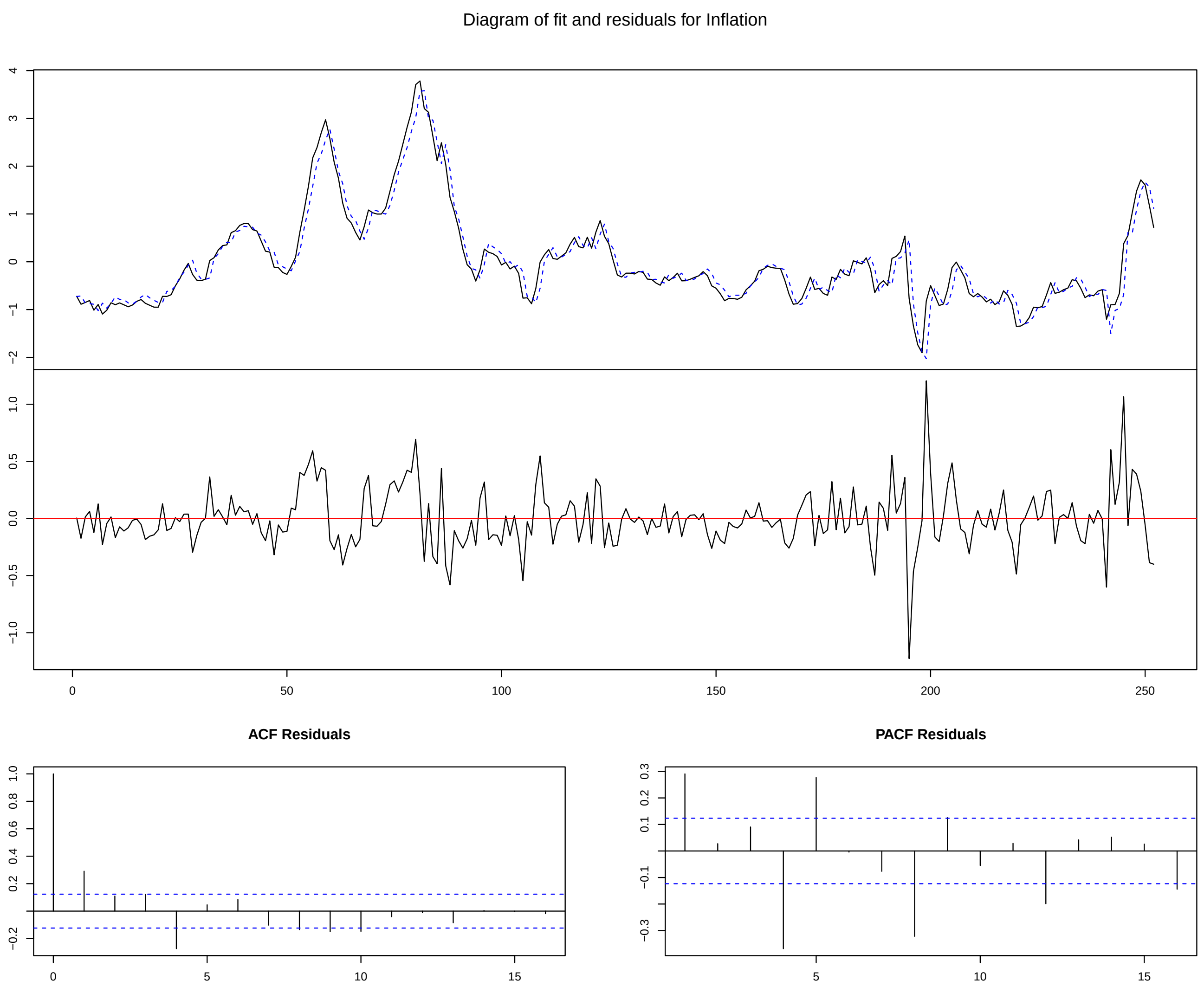


Diagram of fit and residuals for Unemployment

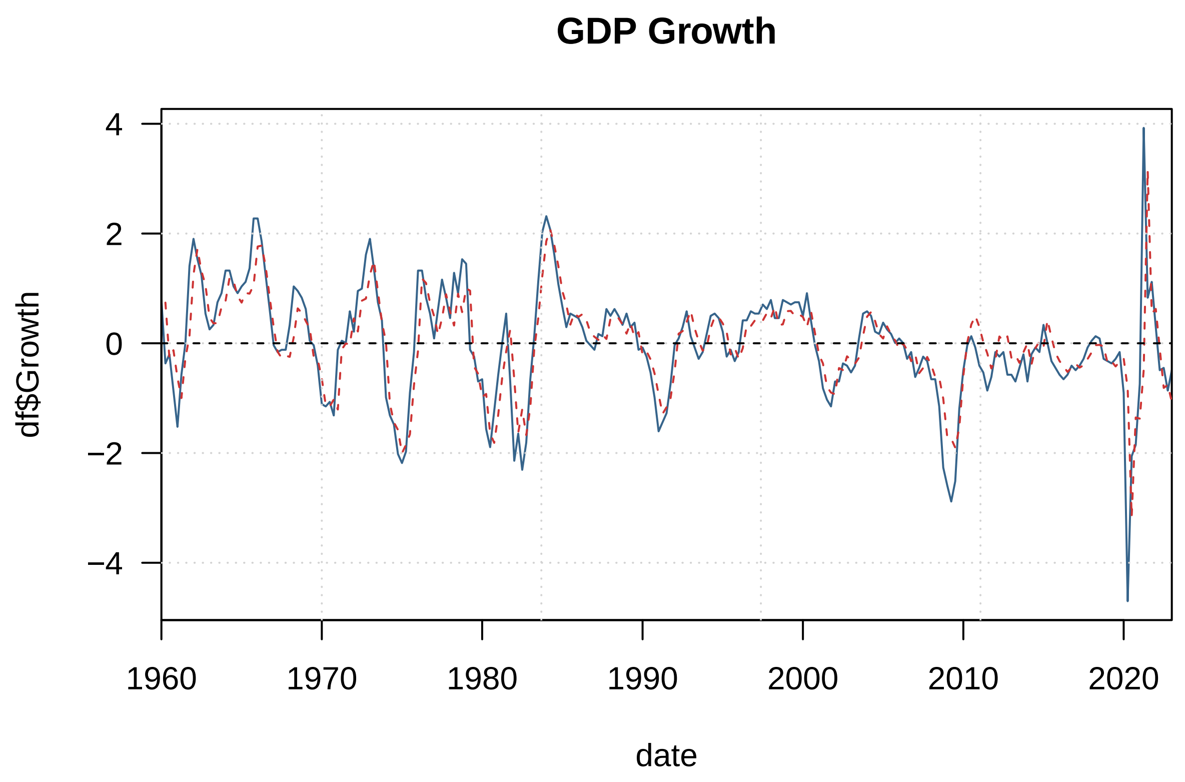




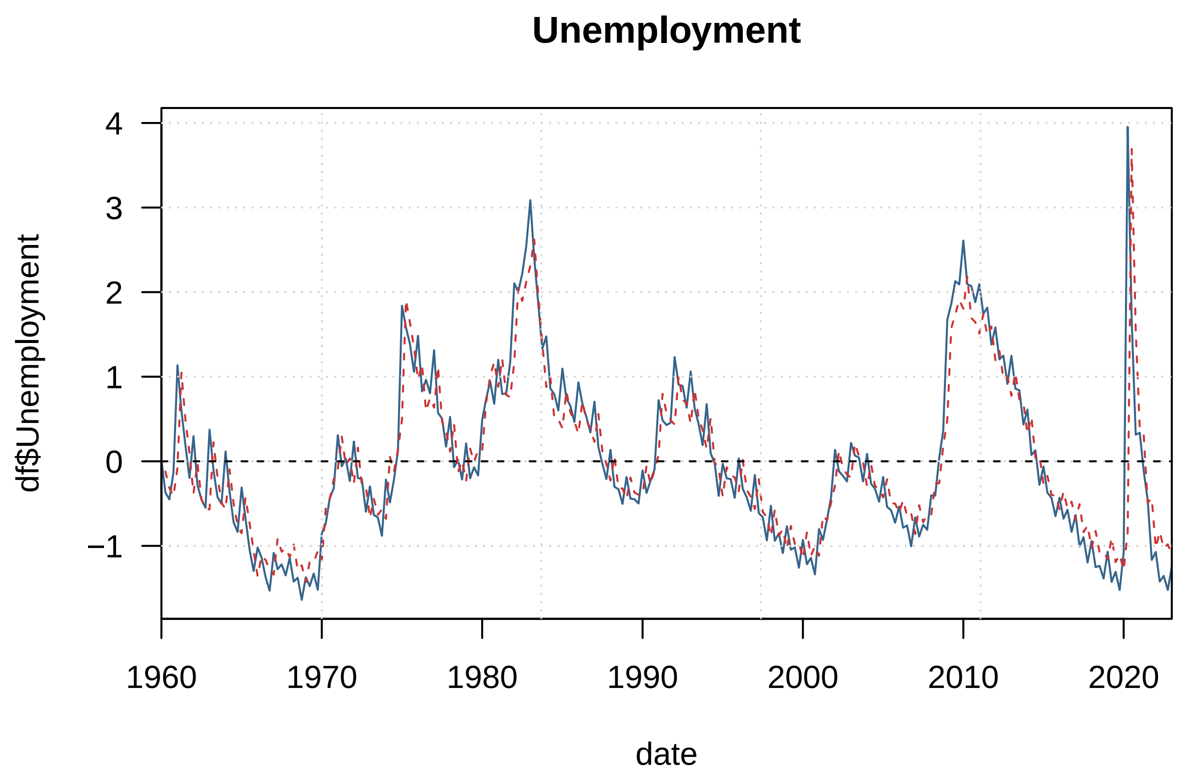




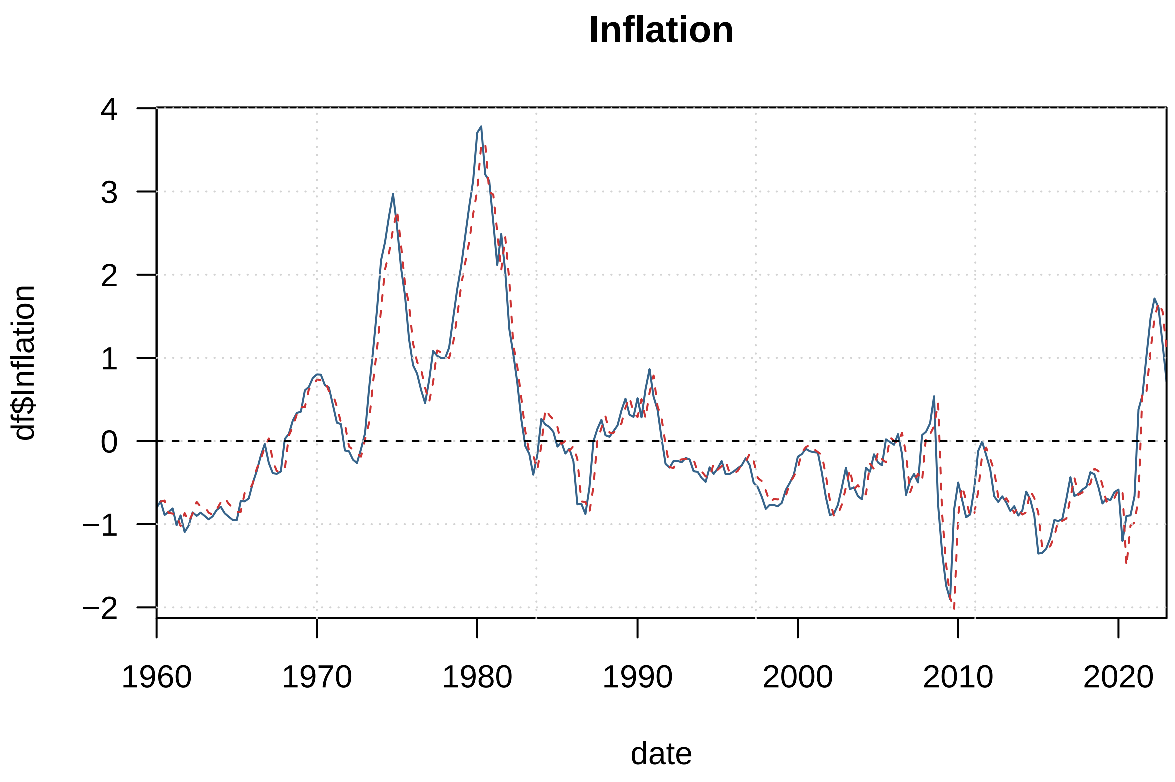
```r
fit = fitted(var1)
plot(date,df$Growth,type="l",xaxs="i",las=1,col="steelblue4",main="GDP Growth")
lines(date[-1],fit[,1],col="brown3",lty=2)
grid()
abline(h=0,lty=2)
```



```r
plot(date,df$Unemployment,type="l",xaxs="i",las=1,col="steelblue4",main="Unemployment")
lines(date[-1],fit[,2],col="brown3",lty=2)
grid()
abline(h=0,lty=2)
```



```r
plot(date,df$Inflation,type="l",xaxs="i",las=1,col="steelblue4",main="Inflation")
lines(date[-1],fit[,3],col="brown3",lty=2)
grid()
abline(h=0,lty=2)
```



1.6 Granger Causality

Test whether lags of one variable enter the equation for another variable

Variable j does not **Granger cause** variable i if all coefficients on lags of variable j in the equation for variable i are zero, that is

$$a_{1,ij} = a_{2,ij} = \dots = a_{p,ij} = 0$$

We thus test $H_0 : a_{1,ij} = \dots = a_{p,ij} = 0$ against $H_A : \exists \ell \in \{1, \dots, p\}$ such that $a_{\ell,ij} \neq 0$ using F-statistic and reject null if the statistic exceeds the critical value at the chosen level

In addition: if the innovation to $y_{i,t}$ and the innovation to $y_{j,t}$ are correlated we say there is **instantaneous causality**

```r
causality(var1, cause = "Growth")
```

```
## $Granger
##
##  Granger causality H0: Growth do not Granger-cause Unemployment
##  Inflation
##
## data:  VAR object var1
## F-Test = 9.1593, df1 = 2, df2 = 744, p-value = 0.0001176
##
##
```

```
## $Instant
##
##  H0: No instantaneous causality between: Growth and Unemployment
##  Inflation
##
## data:  VAR object var1
## Chi-squared = 43.668, df = 2, p-value = 3.293e-10
```

```r
causality(var1, cause = "Inflation")
```

```
## $Granger
##
##  Granger causality H0: Inflation do not Granger-cause Growth
##  Unemployment
##
## data:  VAR object var1
## F-Test = 5.3162, df1 = 2, df2 = 744, p-value = 0.0051
##
##
## $Instant
##
##  H0: No instantaneous causality between: Inflation and Growth
##  Unemployment
##
## data:  VAR object var1
## Chi-squared = 17.272, df = 2, p-value = 0.0001776
```

1.7 IRF and Variance Decompositions

Two important tools used to examine relationships among economic variables

- **impulse-response analysis**: the goal is to track the response of a variable y_i to a one time shock $\varepsilon_{j,t}$
- **forecast error variance decomposition**: the goal is to find the fraction of the overall fluctuations in y_i that is due to shock $\varepsilon_{j,t}$

Impulse response functions (IRF) shows the responses of a VAR endogenous variable in time.

Remark 2.

1. IRFs are constructed using vector moving average (VMA) representation

2. When constructing the IRF the size of the one time shock in ε_j is taken to be one standard deviation σ_{ϵ_j}

3. IRFs trace out the response of $\mathbf{y}_t$ to structural shocks ε_t, not reduced form errors e_t

4. The plot of the IRF for y_i shows the effect ε_j of as deviations of y_i from its long run equilibrium μ_i

5. Because all variables in VAR are weakly stationary, deviations eventually converge to 0 as the system converges back to the long run equilibrium given by μ

There is, however, more subtle things to consider (Identification of Structural Shocks).

```r
# obtain variance-covariance matrix
summary(var1)$covres
```

```
##                  Growth Unemployment   Inflation
## Growth        0.35576688  -0.12504585  0.03704145
## Unemployment -0.12504585   0.23654033 -0.02658451
## Inflation     0.03704145  -0.02658451  0.06539863
```

```
# obtain correlation matrix
summary(var1)$corres
```

```
##                  Growth Unemployment  Inflation
## Growth        1.0000000   -0.4310562  0.2428404
## Unemployment -0.4310562    1.0000000 -0.2137430
## Inflation     0.2428404   -0.2137430  1.0000000
```

Notice the off-diagonal elements of the estimated variance-covariance matrix are not zero, we can assume that there is contemporaneous correlation between the variables in the VAR model.

However, these matrices only describe the correlation between the errors, but it remains unclear in which direction the causalities go. Identifying these causal relationships is one of the main challenges of any VAR analysis.

1.7.1 Orthogonal impulse responses

A common approach to identify the shocks of a VAR model is to use orthogonal impulse responses (OIR). The basic idea is to decompose the variance-covariance matrix so that $\Sigma = PP'$, where P is a lower triangular matrix with positive diagonal elements, which is often obtained by a Cholesky decomposition. Given the estimated variance-covariance matrix P the decomposition can be obtained by

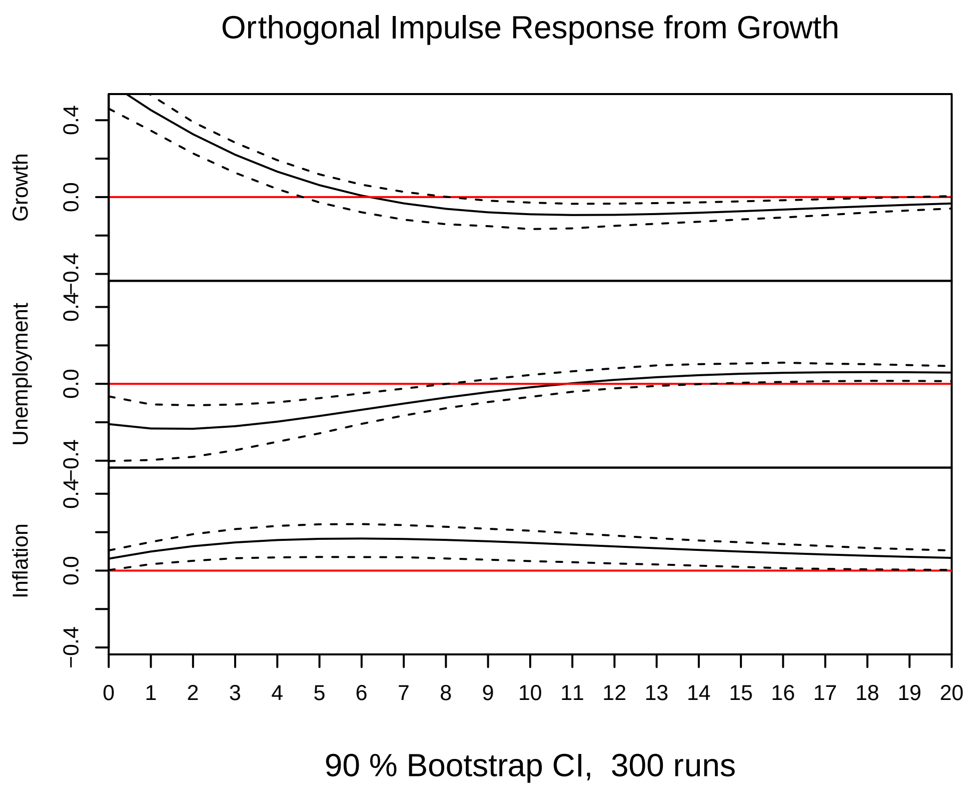
```
t(chol(summary(var1)$covres))
```

```
##                   Growth Unemployment Inflation
## Growth        0.59646197   0.00000000 0.0000000
## Unemployment -0.20964597   0.43884951 0.0000000
## Inflation     0.06210195  -0.03091058 0.2461433
```

From this matrix it can be seen that a shock to Unemployment has a contemporaneous effect on Inflation, but not vice versa.

Note that the output of the Cholesky decomposition is a lower triangular matrix so that the variable in the first row will never be sensitive to a contemporaneous shock of any other variable and the last variable in the system will be sensitive to shocks of all other variables. Therefore, the results of an OIR might be **sensitive to the order of the variables** (This is why we reorder the columns of `df`) and it is advised to estimate the above VAR model with different orders to see how strongly the resulting OIRs are affected by that.

```
plot(irf(var1, impulse = "Growth", ortho=T,n.ahead=20,cumulative=F,runs=300,ci=0.9),col=1,
```

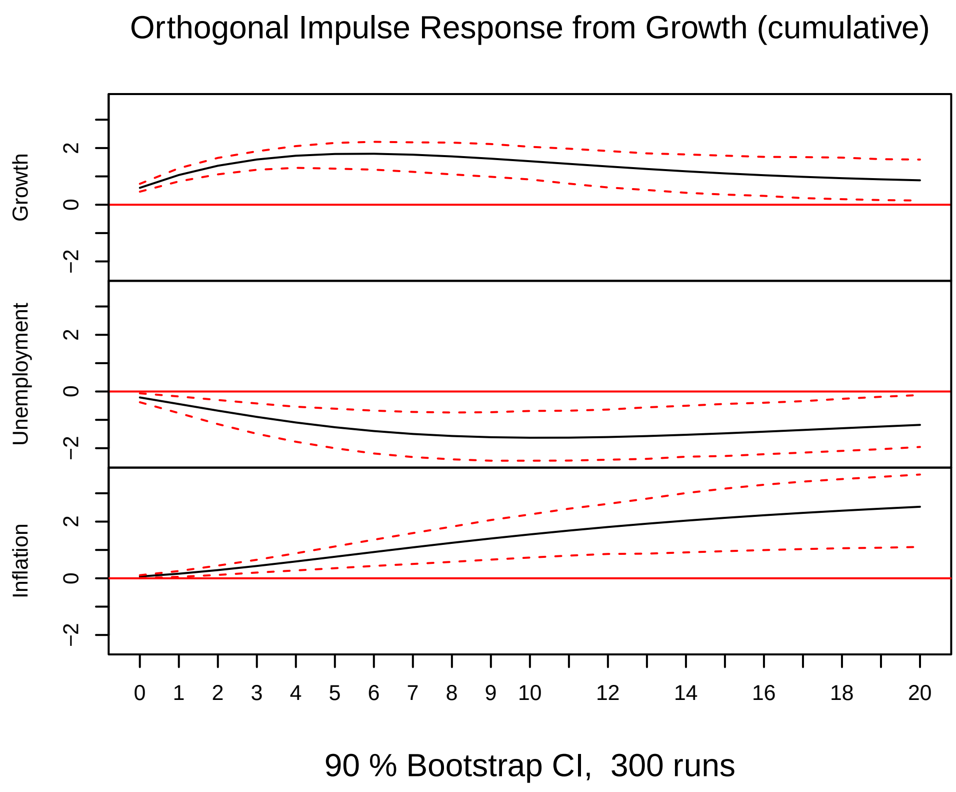


Interpretation: A positive Growth shock has a negative effect on unemployment and a positive effect on inflation, which is in a statistically significant manner as the 90% confidence intervals do not contain 0.

The dotted lines show the 90% interval estimates of these effects.

Sometimes it is interesting to see what the long-run effects of a shock are. To get an idea about that you can also calculate and plot the *cumulative* impulse response function to get an idea of the overall long-run effect of the shock:

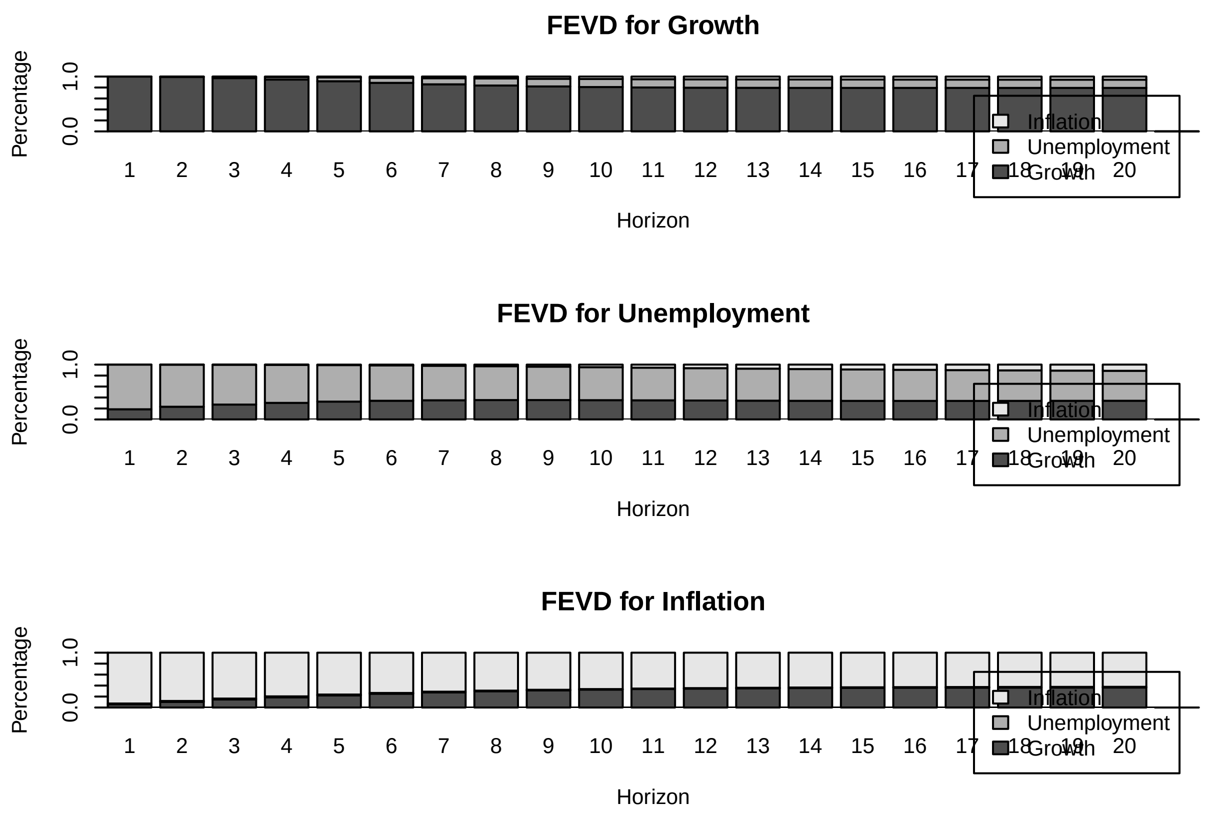
```
plot(irf(var1, impulse = "Growth",ortho=T,n.ahead=20,cumulative=T,runs=300,ci=0.9))
```



1.7.2 Variance Decomposition

Find **fraction of overall fluctuations in y_i that is due to shock $\varepsilon_{j,t}$**

```
plot(fevd(var1,ortho=T,n.ahead=20,cumulative=F,runs=300,ci=0.9),xaxs="i")
```



Forecast variance decomposition estimates the contribution of a shock in each variable to the response in both variables.

The plot shows that almost 100 percent of the variance in Growth is caused by Growth itself, while only about 80 percent in the variance of Unemployment is caused by Unemployment itself.

Remark 3. *Identification of Structural Shocks*

Consider general case: a VAR(p) model with k variables.

1. Reduced form has $k + pk^2 + k(k+1)/2$ parameters

2. Structural form has $k + (p+1)k^2 + k$ parameters

$\implies$ *Identification thus requires $k(k-1)/2$ additional restrictions*

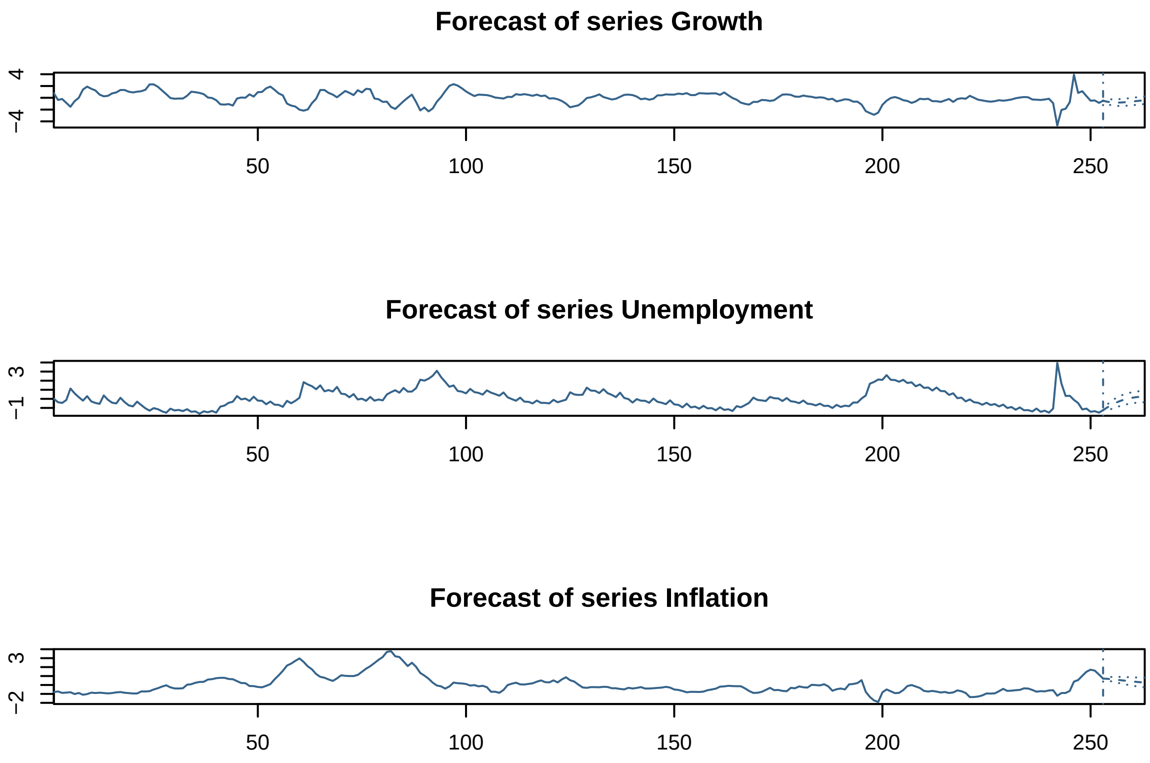
3. Choleski decomposition: set elements of $\mathbf{B}_0$ above main diagonal equal zero - This is how 'vars' package constructs IRFs and FEVD shown on previous pages

4. Ordering of variables in the VAR(p) model thus matters:

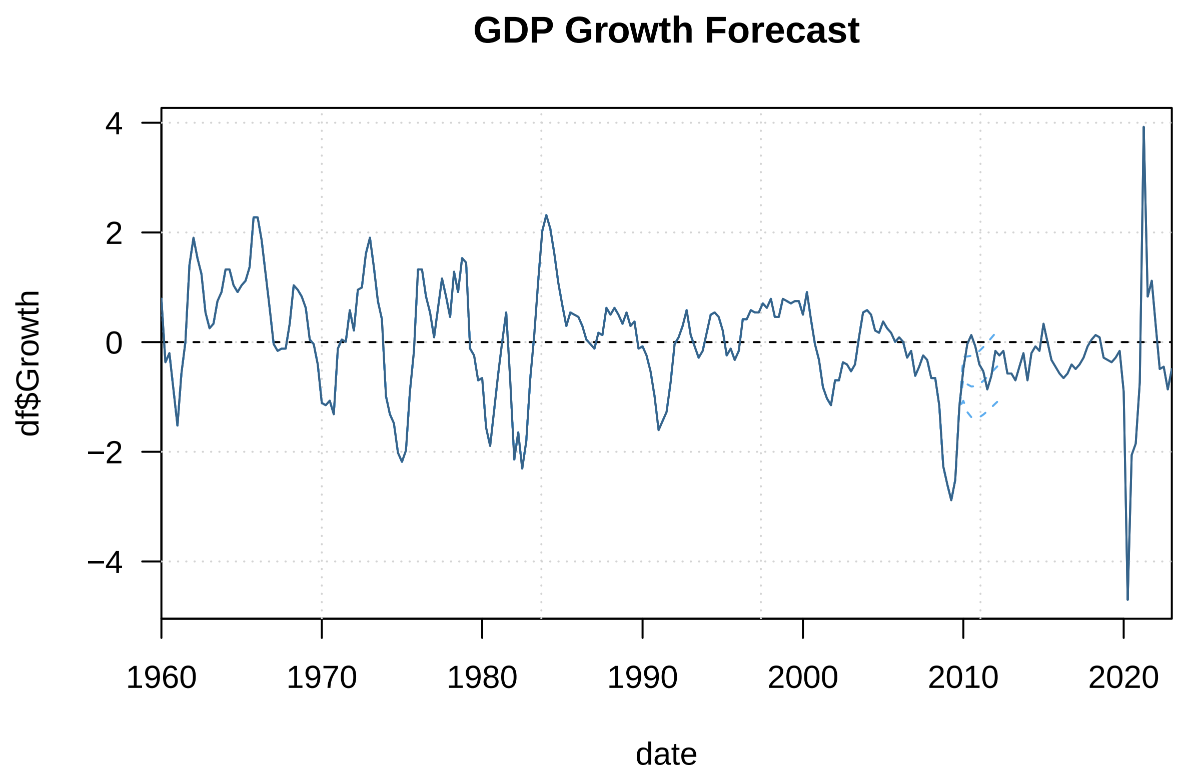
$y_{i,t}$ is only affected by shocks $\varepsilon_{1,t}, \dots, \varepsilon_{i,t}$, remaining shocks $\varepsilon_{i+1,t}, \dots, \varepsilon_{k,t}$ have no contemporaneous effect on $y_{i,t}$ and will only affect $y_{i,t'}$ for $t' > t$ indirectly through their effect on $y_{i+1,t}, \dots, y_{k,t}$

1.8 Forecast

```
### FORECAST
n = 200
nfore = 10
tot = n+nfore
pred1 = predict(var1, n.ahead = nfore, ci = 0.5, newdata=df[1:n,])
plot(pred1,xaxs="i",col="steelblue4")
```



```r
plot(date,df$Growth,type="l",xaxs="i",las=1,col="steelblue4",main="GDP Growth Forecast")
grid()
abline(h=0,lty=2)
lines(date[1:tot],c(df$Growth[1:n],pred1$fcst$Growth[,1]),col="steelblue2",lty=2)
lines(date[1:tot],c(df$Growth[1:n],pred1$fcst$Growth[,2]),col="steelblue2",lty=2)
lines(date[1:tot],c(df$Growth[1:n],pred1$fcst$Growth[,3]),col="steelblue2",lty=2)
lines(date,df$Growth,col="steelblue4")
```



Notice that it's also possible to create rolling forecasts (skip).

```r
rm(list = ls())
```

2 Money, Banking and the Macro-Economy

2.1 VAR: Banking Markup

Federal Funds Rate: In the United States, the federal funds rate is the interest rate at which depository institutions (banks and credit unions) lend reserve balances to other depository institutions overnight, on an uncollateralized basis.

LIBOR: The London Interbank Offered Rate is the average of interest rates estimated by each of the leading banks in London that it would be charged were it to borrow from other banks

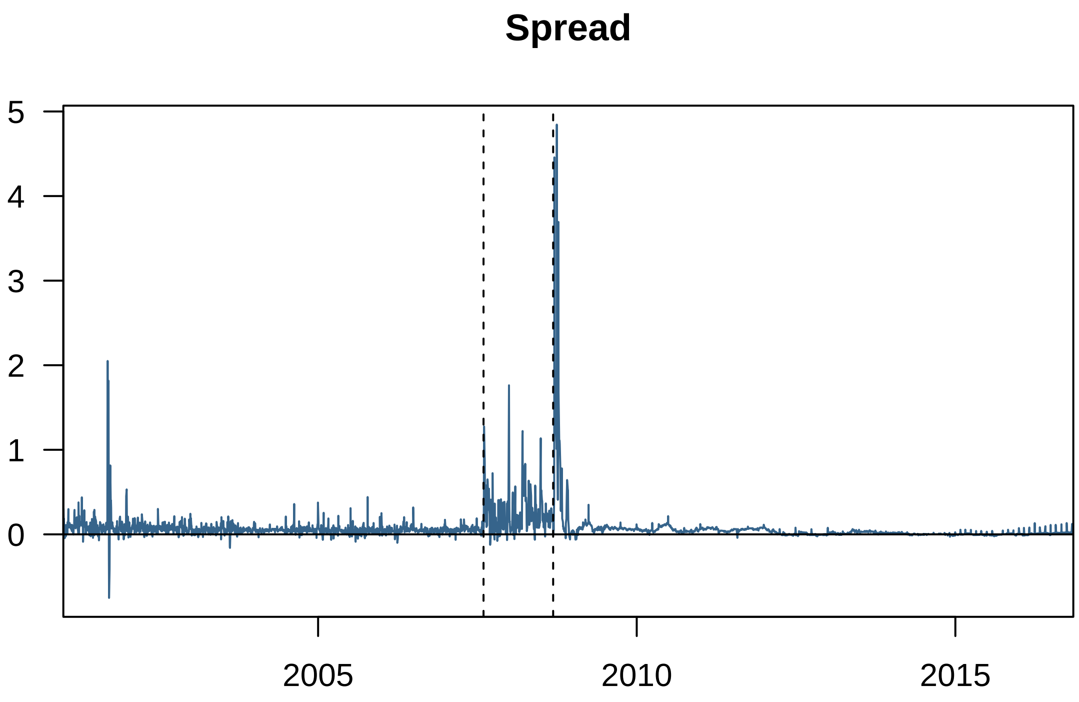
In normal times they are not differing a lot. The spread represents the risk in the financial market.

```r
library(Quandl)
library(quantmod)
USDONTD156N <- Quandl("FRED/USDONTD156N")
getSymbols("DFF",src="FRED")
```

```
## [1] "DFF"
```

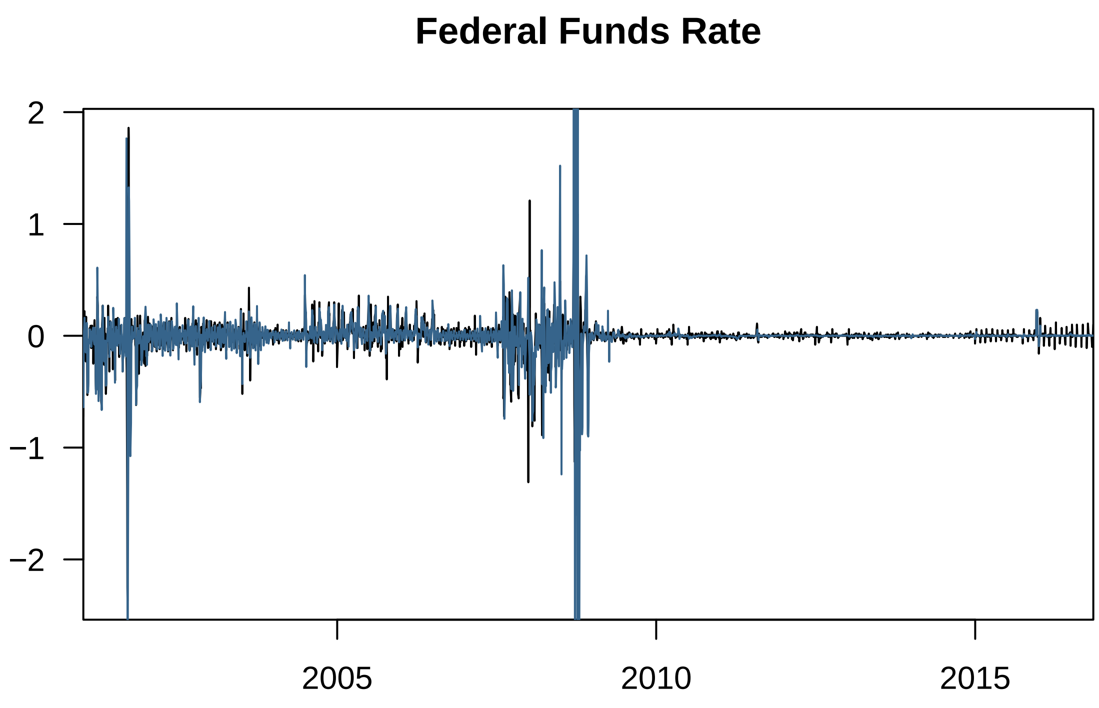
```
X = cbind(DFF,USDONTD156N)
X = na.omit(X)
df = data.frame(X)
date = as.Date(index(X))
```

```
plot(date,df[,2]-df[,1],col="steelblue4",type="l",xaxs="i",xlab="",ylab="",main="Spread",l
abline(h=0)
abline(v=date[which(date=="2007-08-07")],lty=2) # Subprime mortgage crisis
abline(v=date[which(date=="2008-09-09")],lty=2) # Lehman Brother Bankruptcy
```



```
X = diff(X,lag=5)
X = na.omit(X)
df = data.frame(X)
colnames(df) = c("i","LIBOR")
date = as.Date(index(X))
k = ncol(df)
```

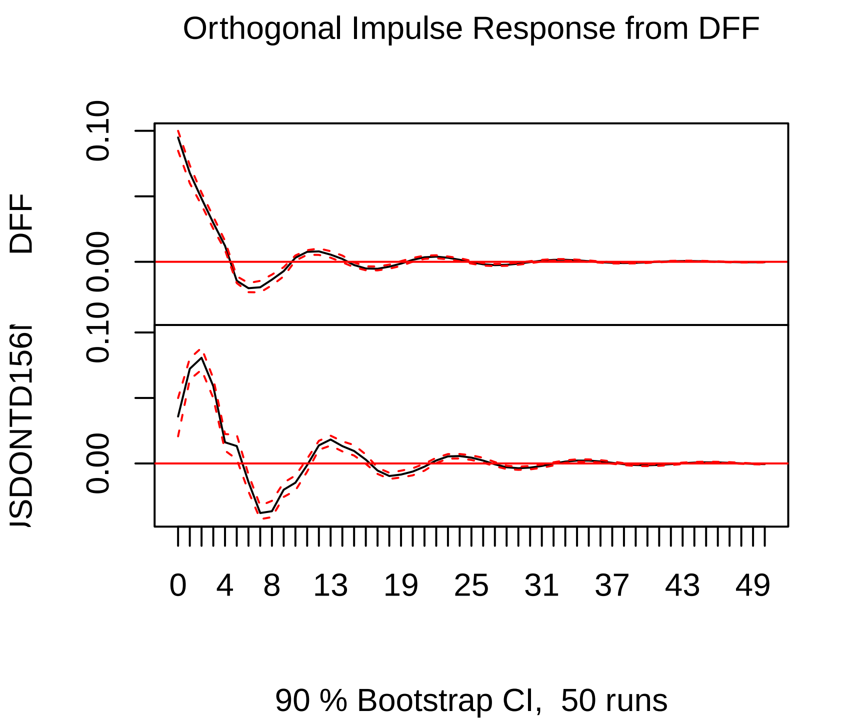
```
par(mfrow=c(1,1))
plot(date,df[,1],type="l",xlab="",ylab="",main="Federal Funds Rate",xaxs="i",las=1)
lines(date,df[,2],col="steelblue4")
```



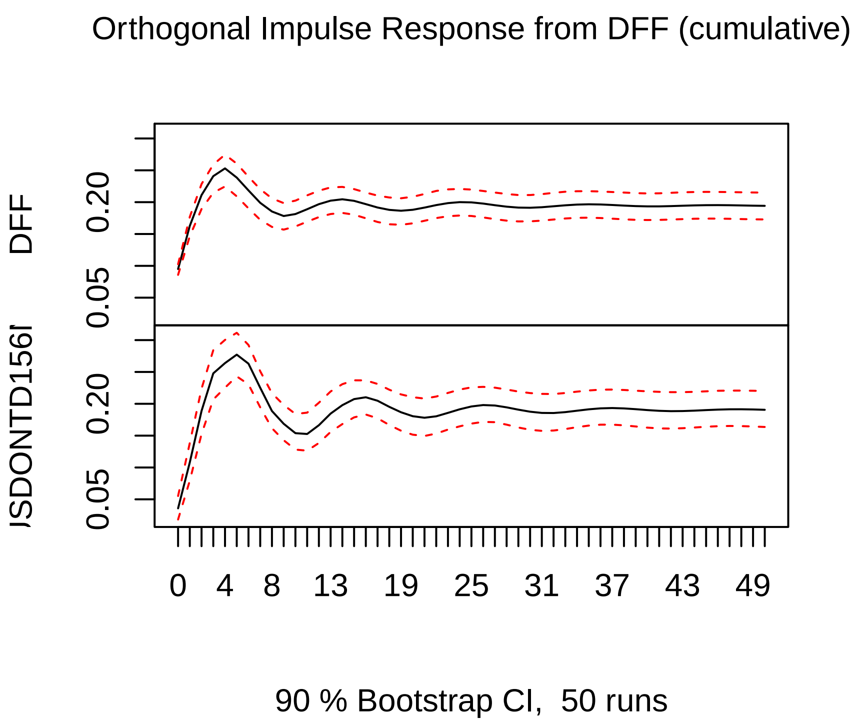
```r
var1 = VAR(X,p=5)
summary(var1)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: DFF, USDONTD156N
## Deterministic variables: const
## Sample size: 3890
## Log Likelihood: 5896.294
## Roots of the characteristic polynomial:
## 0.9134 0.9134 0.8153 0.8153 0.7947 0.7947 0.7567 0.5079 0.5079 0.2977
## Call:
## VAR(y = X, p = 5)
##
##
## Estimation results for equation DFF:
## ====================================
## DFF = DFF.l1 + USDONTD156N.l1 + DFF.l2 + USDONTD156N.l2 + DFF.l3 + USDONTD156N.l3 + DFF
##
##                 Estimate Std. Error t value Pr(>|t|)
## DFF.l1          0.718025   0.015903  45.149   <2e-16 ***
## USDONTD156N.l1 -0.009140   0.010556  -0.866   0.3866
## DFF.l2          0.042214   0.019950   2.116   0.0344 *
## USDONTD156N.l2 -0.105094   0.011677  -9.000   <2e-16 ***
## DFF.l3          0.003278   0.019979   0.164   0.8697
## USDONTD156N.l3  0.005882   0.011760   0.500   0.6170
## DFF.l4         -0.025134   0.019851  -1.266   0.2055
## USDONTD156N.l4 -0.011212   0.011765  -0.953   0.3407
## DFF.l5         -0.210128   0.017524 -11.991   <2e-16 ***
## USDONTD156N.l5  0.103008   0.009718  10.600   <2e-16 ***
## const          -0.003562   0.001527  -2.332   0.0198 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.09494 on 3879 degrees of freedom
## Multiple R-Squared: 0.5806,  Adjusted R-squared: 0.5796
## F-statistic: 537.1 on 10 and 3879 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation USDONTD156N:
## ============================================
## USDONTD156N = DFF.l1 + USDONTD156N.l1 + DFF.l2 + USDONTD156N.l2 + DFF.l3 + USDONTD156N.
##
##                 Estimate Std. Error t value Pr(>|t|)
## DFF.l1          0.582912   0.023532  24.771  < 2e-16 ***
## USDONTD156N.l1  0.473162   0.015620  30.292  < 2e-16 ***
## DFF.l2          0.116188   0.029519   3.936 8.43e-05 ***
## USDONTD156N.l2 -0.114128   0.017278  -6.605 4.51e-11 ***
## DFF.l3         -0.114202   0.029562  -3.863 0.000114 ***
## USDONTD156N.l3  0.107433   0.017401   6.174 7.35e-10 ***
## DFF.l4         -0.241739   0.029372  -8.230 2.53e-16 ***
## USDONTD156N.l4 -0.071507   0.017408  -4.108 4.08e-05 ***
## DFF.l5          0.305758   0.025930  11.792  < 2e-16 ***
## USDONTD156N.l5 -0.246727   0.014379 -17.159  < 2e-16 ***
## const          -0.001509   0.002260  -0.668 0.504463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.1405 on 3879 degrees of freedom
## Multiple R-Squared: 0.624,   Adjusted R-squared: 0.6231
## F-statistic: 643.9 on 10 and 3879 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##                   DFF USDONTD156N
## DFF         0.009014     0.00340
## USDONTD156N 0.003400     0.01973
##
## Correlation matrix of residuals:
##                DFF USDONTD156N
## DFF         1.000       0.255
## USDONTD156N 0.255       1.000
```

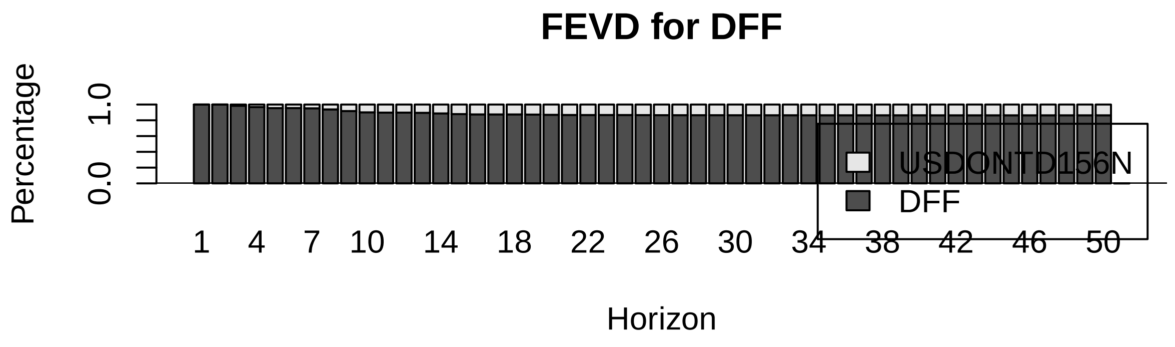
```r
plot(irf(var1, impulse="DFF",ortho=T,n.ahead=50,cumulative=F,runs=50,ci=0.9))
```

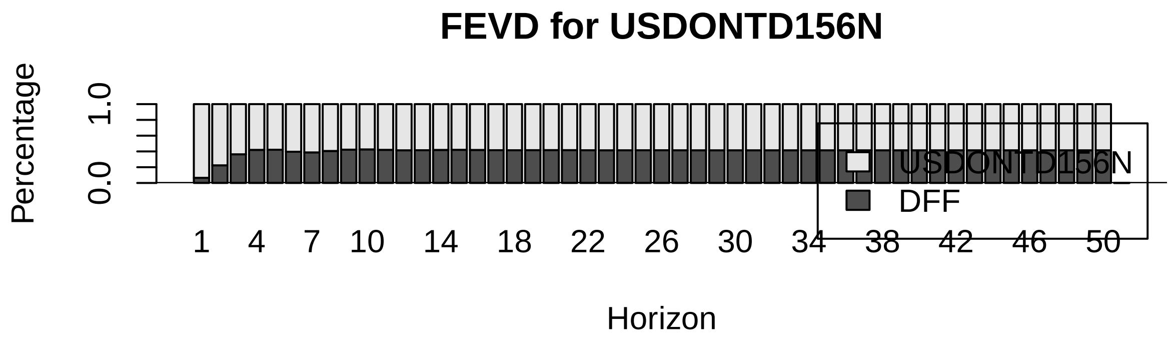


```r
plot(irf(var1, impulse="DFF" ,ortho=T,n.ahead=50,cumulative=T,runs=50,ci=0.9))
```



```r
plot(fevd(var1,ortho=T,n.ahead=50,cumulative=F,runs=50,ci=0.9))
```





```r
rm(list=ls())
```

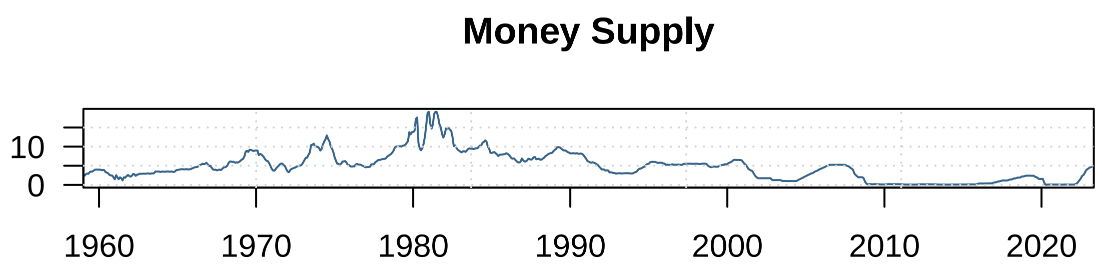
2.2 VAR: Money Market

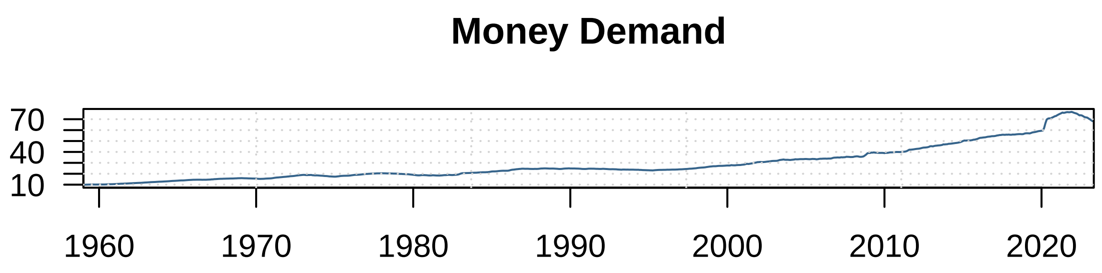
```r
suppressMessages(library(quantmod))
getSymbols(c("FEDFUNDS","M2SL","CPIAUCSL"),src="FRED")
```

```
## [1] "FEDFUNDS" "M2SL"     "CPIAUCSL"
```

```r
X = cbind(FEDFUNDS,M2SL/CPIAUCSL)
X = na.omit(X)
df = data.frame(X)
colnames(df) = c("Interest Rate","Real Money")
date = as.Date(index(X))
k = ncol(df)
```

```r
par(mfrow=c(k,1))
plot(date,df[,1],type="l",xlab="",ylab="",main="Money Supply",xaxs="i",las=1,col="steelblu
grid()
plot(date,df[,2],type="l",xlab="",ylab="",main="Money Demand",xaxs="i",las=1,col="steelblu
grid()
```





```r
lag = 12
df1 = df[-c(1:lag),]
head(df)
```

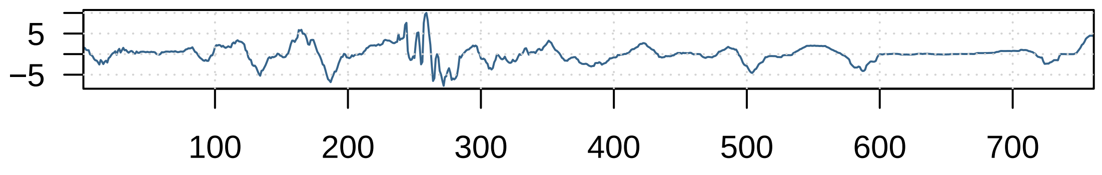
```
##            Interest Rate Real Money
## 1959-01-01          2.48   9.879352
## 1959-02-01          2.43   9.920690
## 1959-03-01          2.80   9.982741
## 1959-04-01          2.96  10.010352
## 1959-05-01          2.90  10.061983
## 1959-06-01          3.39  10.103057
```

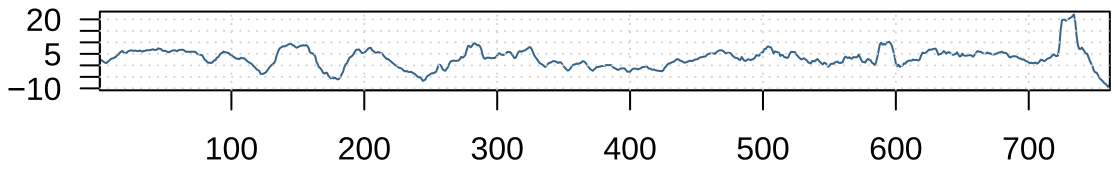
```r
for (i in 1:k) {
   if (i==1) { # first column
      df1[,i] = diff((df[,i]),lag=lag)
   } else {  # second column
      df1[,i] = diff(log(df[,i]),lag=lag)*100
   }
}
head(df1)
```

```
##            Interest Rate Real Money
## 1960-01-01          1.51   2.734374
## 1960-02-01          1.54   2.247768
## 1960-03-01          1.04   1.925398
## 1960-04-01          0.96   1.475072
## 1960-05-01          0.95   1.125334
## 1960-06-01         -0.07   1.046967
```

```r
par(mfrow=c(k,1))
plot(df1[,1],type="l",xlab="",ylab="",main="",xaxs="i",las=1,col="steelblue4")
```

```r
grid()
plot(df1[,2],type="l",xlab="",ylab="",main="",xaxs="i",las=1,col="steelblue4")
grid()
```

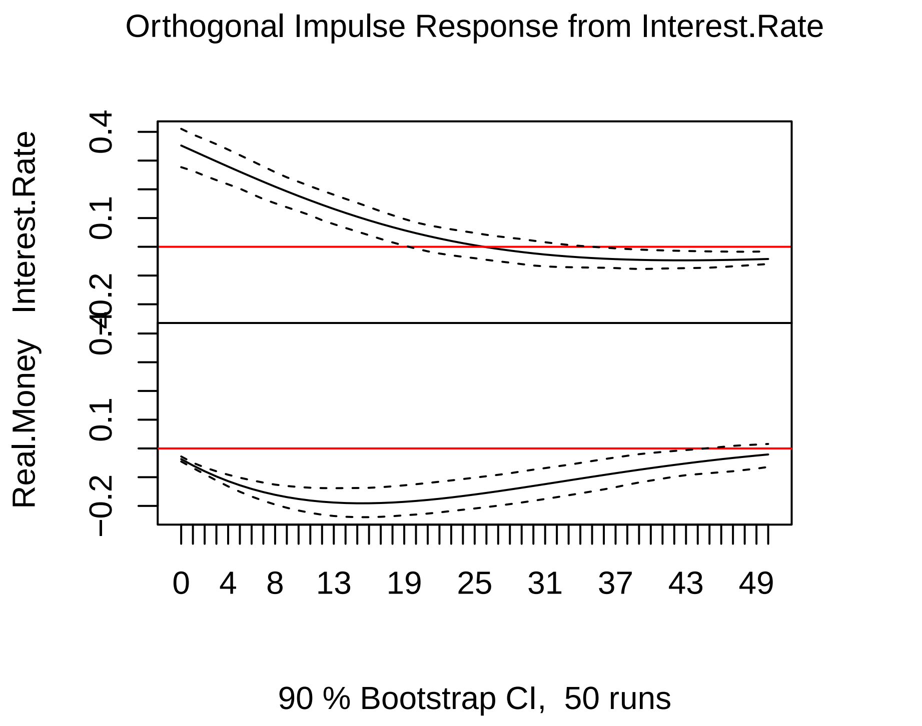




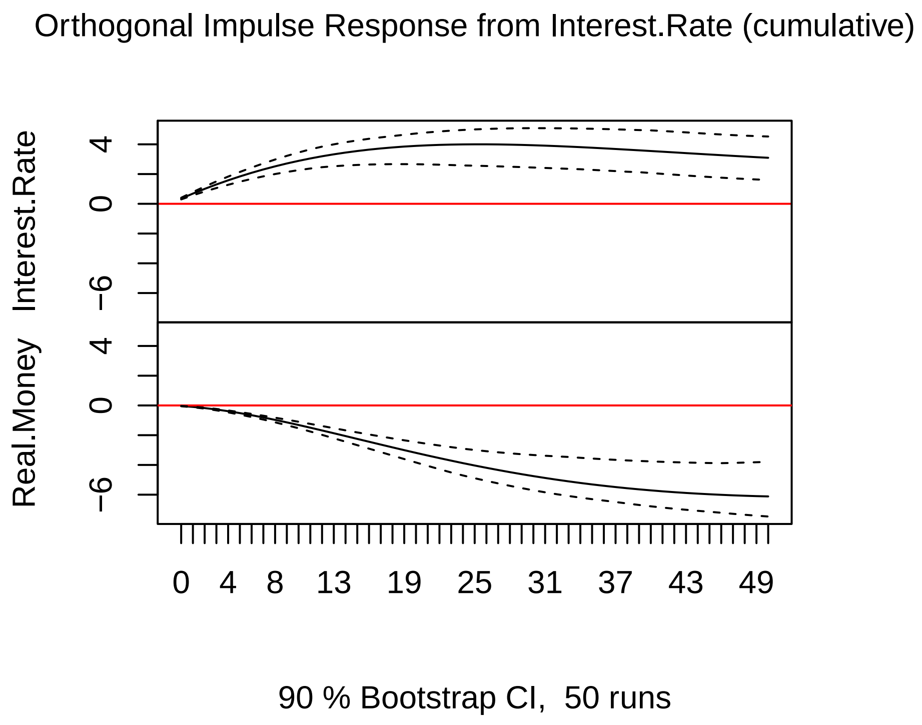
```r
var2 = VAR(scale(df1[,c(1,2)],T,T),p=1)
summary(var2)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: Interest.Rate, Real.Money
## Deterministic variables: const
## Sample size: 760
## Log Likelihood: -22.337
## Roots of the characteristic polynomial:
## 0.9589 0.9589
## Call:
## VAR(y = scale(df1[, c(1, 2)], T, T), p = 1)
##
##
## Estimation results for equation Interest.Rate:
## ==============================================
## Interest.Rate = Interest.Rate.l1 + Real.Money.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## Interest.Rate.l1 0.952048   0.013752  69.228  < 2e-16 ***
## Real.Money.l1    0.040105   0.013781   2.910  0.00372 **
## const            0.001462   0.012784   0.114  0.90899
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.3524 on 757 degrees of freedom
## Multiple R-Squared: 0.8762,  Adjusted R-squared: 0.8759
## F-statistic:  2679 on 2 and 757 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation Real.Money:
## ===========================================
## Real.Money = Interest.Rate.l1 + Real.Money.l1 + const
##
##                   Estimate Std. Error t value Pr(>|t|)
## Interest.Rate.l1 -0.066718   0.006858  -9.729   <2e-16 ***
## Real.Money.l1     0.963034   0.006872 140.141   <2e-16 ***
## const            -0.003473   0.006375  -0.545    0.586
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.1757 on 757 degrees of freedom
## Multiple R-Squared: 0.9692,  Adjusted R-squared: 0.9692
## F-statistic: 1.193e+04 on 2 and 757 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##               Interest.Rate Real.Money
## Interest.Rate       0.12420   -0.01309
## Real.Money         -0.01309    0.03088
##
## Correlation matrix of residuals:
##               Interest.Rate Real.Money
## Interest.Rate        1.0000    -0.2114
## Real.Money          -0.2114     1.0000
```

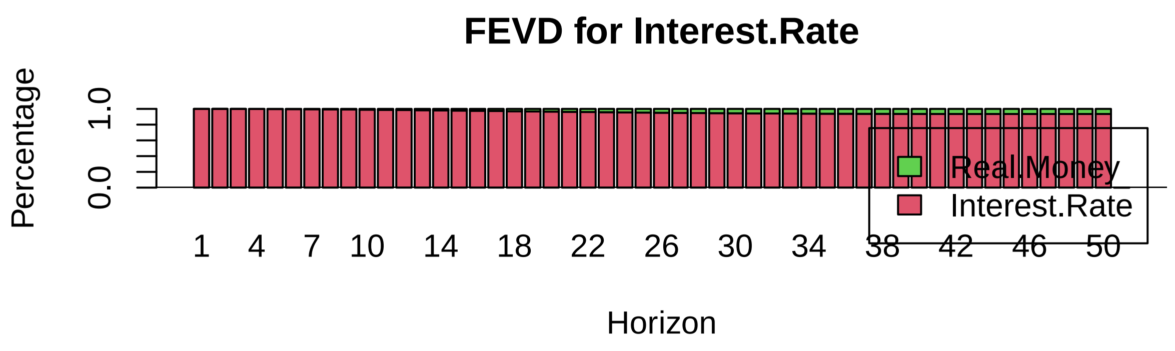
```r
### MONEY MARKET (SUPPLY AND DEMAND SHOCK)
plot(irf(var2, impulse = "Interest.Rate", ortho=T,n.ahead=50,cumulative=F,runs=50,ci=0.9),
```

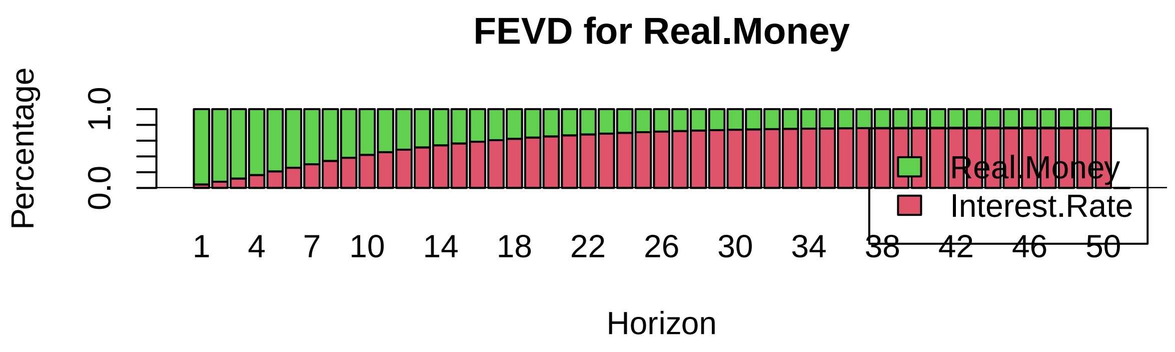


```r
plot(irf(var2, impulse = "Interest.Rate", ortho=T,n.ahead=50,cumulative=T,runs=50,ci=0.9),
```



```r
plot(fevd(var2,ortho=T,n.ahead=50,cumulative=F,runs=50,ci=0.9), col=(2:3))
```





2.3 SVAR: Money Market (Supply and Demand Shock)

A critical drawback of VAR models in their standard form is their missing ability to describe contemporaneous relationships between the analysed variables. This becomes a central issue in the impulse response analysis for such models, where it is important to know the contemporaneous effects of a shock to the economy. Usually, researchers address this by using orthogonal impulse responses, where the correlation between the errors is obtained from the (lower) Cholesky decomposition of the error covariance matrix (aka zero contemporaneous restrictions). This requires them to arrange the variables of the model in a suitable order.

Remark 4.

1. Choleski decomposition is one way to uncover $\mathbf{B}_0$

2. If Choleski decomposition is used, ordering of variables in VAR matters for the IRFs and FEVDs

3. The ordering however often does not have direct economic interpretation and is ad hoc

4. How much the order of variables matters, and how much the IRFs and FEVD change depends on the magnitude of correlation among elements of $\mathbf{e}_t$

- For example, in a bivariate VAR(1)

- if $corr(e_{1,t}, e_{2,t}) = 0$ *then* $\varepsilon_{i,t} = e_{i,t}$ *so structural shocks are identical to reduced form errors, and ordering does not matter at all*

- if $corr(e_{1,t}, e_{2,t}) = 1$ *then there is actually only one structural shock and whichever variable is first determines which structural error its going to be*

5. We will thus look at several alternative ways of introducing restrictions consistent with some economic theory

An alternative to this approach is to use so-called structural vector autoregressive (SVAR) models, where the relationship between *contemporaneous* variables is modeled more directly.

Short run restrictions: restrictions on $\mathbf{B}_0$ which captures the contemporaneous relationships of variables

- Note that Choleski decomposition is essentially a particular way to impose short run restrictions that imposes a recursive structure

Long run restrictions: restrictions on $\mathbf{B}_0$ arise by dividing shocks into two groups - those that have a permanent effect on some variables, and those that have no permanent effects on any variable

Sign restrictions: Skip.

Let's consider a VAR model:

$$\mathbf{y}_t = A_1\mathbf{y}_{t-1} + \mathbf{u}_t, \quad \mathbf{u}_t \sim N(0, \Sigma)$$

where $\mathbf{y}_t$ is a kx1 vector of k variables in period t. A_1 is a kxk coefficient matrix and $\mathbf{u}_t$ is a kx1 vector of errors, which has a multivariate normal distribution with zero mean and a kxk variance-covariance matrix Σ.

Contemporaneous causality or, more precisely, the structural relationships between the variables is analysed in the context of SVAR models, which impose special restrictions on the covariance matrix and – depending on the model – on other coefficient matrices as well.

There are 4 approaches to model structural relationships between the endogeneous variables of a VAR model: The A-model, the B-model, the AB-model and long run restrictions (Blanchard and Quah).

2.3.1 Short-Run Restrictions

2.3.1.1 The A-model The A-model assumes that the covariance matrix is diagonal and contemporaneous relationships between the observable variables are described by an additional matrix A so that:

$$A\mathbf{y}_t = A_1^*\mathbf{y}_{t-1} + \dots + A_p^*\mathbf{y}_{t-p} + \epsilon_t$$

where $A_j^* = AA_j$ and $\epsilon_t = A\mathbf{u}_t \sim (0, \Sigma_\epsilon = A\Sigma_u A' = I_k)$

Notice the matrix A contains $\frac{k(k-1)}{2}$ restrictions.

Note 1.

Matrix A is essentially $\mathbf{B}_0$

2.3.1.2 The B-model The B-model describes the structural relationships of the errors directly by adding a matrix B to the error term and normalises the error variances to unity so that:

$$\begin{aligned}\mathbf{y}_t &= A_1\mathbf{y}_{t-1} + \dots + A_p\mathbf{y}_{t-p} + B\epsilon_t && (7)\\ &= A_1\mathbf{y}_{t-1} + \dots + A_p\mathbf{y}_{t-p} + \mathbf{u}_t && (8)\end{aligned}$$

where $\mathbf{u}_t = B\epsilon_t$ and $\epsilon_t \sim (0, I_k)$

B must contain at least $\frac{k(k-1)}{2}$ restrictions.

2.3.1.3 The AB-model The AB-model is a mixture of the A- and B-model, where the errors of the VAR are modelled as:

$$A\mathbf{u}_t = B\epsilon_t$$

with $\epsilon_t \sim (0, I_k)$

For an 'AB-model' the number of restrictions amounts to: $k^2 + \frac{k(k-1)}{2}$

The reduced form residuals can be obtained from the above equation via the relation: $\mathbf{u}_t = A^{-1}B\epsilon_t$, with variance-covariance matrix $\Sigma_u = A^{-1}B\Sigma_\epsilon B'A^{-1'} = A^{-1}BB'A^{-1'}$

Finally, in case of an overidentified SVAR, a likelihood ratio statistic is computed according to:

$$LR = T(\ln\det(\Sigma_R) - \ln\det(\Sigma_u))$$

with R number of over-identifying restrictions, i.e., number of restrictions exceeding $\frac{k(k-1)}{2}$, Σ_R being the restricted variance-covariance matrix, and Σ_u being the variance covariance matrix of the reduced form residuals. The test statistics has χ^2 distribution with R degrees of freedom.

This can be used to test whether over-identifying restrictions are consistent with data.

Here, we introduce the A- and B-model method.

Personally, I prefer A-model method.

```
amat = diag(k)
diag(amat) = NA
amat[2,1] = NA
amat
```

```
##      [,1] [,2]
## [1,]   NA    0
## [2,]   NA   NA
```

```
bmat = diag(k)
diag(bmat) = NA
bmat[2, 1] = NA
bmat
```

```
##      [,1] [,2]
## [1,]   NA    0
## [2,]   NA   NA
```

A- and B-model must contain at least $\frac{k(k-1)}{2}$ restrictions, which in this case is $\frac{2(2-1)}{2} = 1$ restriction. Hence, we set the upper triangular element to be 0 (assuming that real money has no contemporaneous effect on interest rate). The other elements are to be estimated, so we set them to be `NA`.

```
## Estimation method using a scoring algorithm
svar2_A = SVAR(x = var2, estmethod = "scoring", Amat = amat, Bmat = NULL, max.iter = 100,
```

```
## Warning in SVAR(x = var2, estmethod = "scoring", Amat = amat, Bmat = NULL, :
## The A-model is just identified. No test possible.
```

```
summary(svar2_A)
```

```
##
## SVAR Estimation Results:
```

```
## ========================
##
## Call:
## SVAR(x = var2, estmethod = "scoring", Amat = amat, Bmat = NULL,
##     max.iter = 100, conv.crit = 1e-08, maxls = 1000)
##
## Type: A-model
## Sample size: 760
## Log Likelihood: -25.343
## Method: scoring
## Number of iterations: 15
##
## Estimated A matrix:
##               Interest.Rate Real.Money
## Interest.Rate        2.8375      0.000
## Real.Money           0.6137      5.822
##
## Estimated standard errors for A matrix:
##               Interest.Rate Real.Money
## Interest.Rate       0.07278     0.0000
## Real.Money          0.10412     0.1493
##
## Estimated B matrix:
##               Interest.Rate Real.Money
## Interest.Rate             1          0
## Real.Money                0          1
##
## Covariance matrix of reduced form residuals (*100):
##               Interest.Rate Real.Money
## Interest.Rate        12.420     -1.309
## Real.Money           -1.309      3.088
```

```
#check, should be close to identity matrix
svar2_A$A%*%summary(var2)$covres%*%t(svar2_A$A)
```

```
##               Interest.Rate   Real.Money
## Interest.Rate  1.000000e+00 8.105749e-12
## Real.Money     8.105739e-12 1.000000e+00
```

```
solve(svar2_A$A) # just the B matrix in B-model method
```

```
##               Interest.Rate Real.Money
## Interest.Rate     0.3524225  0.0000000
## Real.Money       -0.0371482  0.1717687
```

```
#Estimation method by using directly minimising the negative log-likelihood with optim()
svar2_B = SVAR(x = var2, estmethod = "direct", Amat = NULL, Bmat = bmat,
            hessian = TRUE, method="BFGS")
```

```
## Warning in SVAR(x = var2, estmethod = "direct", Amat = NULL, Bmat = bmat, : The
## B-model is just identified. No test possible.
```

```
summary(svar2_B) # Interest is only influencing itself in time t whereas Real Money Dema
```

```
##
```

```
## SVAR Estimation Results:
## ========================
##
## Call:
## SVAR(x = var2, estmethod = "direct", Amat = NULL, Bmat = bmat,
##     hessian = TRUE, method = "BFGS")
##
## Type: B-model
## Sample size: 760
## Log Likelihood: -25.366
## Method: direct
## Number of iterations: 182
## Convergence code: 0
##
## Estimated A matrix:
##               Interest.Rate Real.Money
## Interest.Rate             1          0
## Real.Money                0          1
##
## Estimated B matrix:
##               Interest.Rate Real.Money
## Interest.Rate       0.35242     0.0000
## Real.Money         -0.03715     0.1718
##
## Estimated standard errors for B matrix:
##               Interest.Rate Real.Money
## Interest.Rate      0.009039   0.000000
## Real.Money         0.006303   0.004405
##
## Covariance matrix of reduced form residuals (*100):
##               Interest.Rate Real.Money
## Interest.Rate        12.420     -1.309
## Real.Money           -1.309      3.089
```

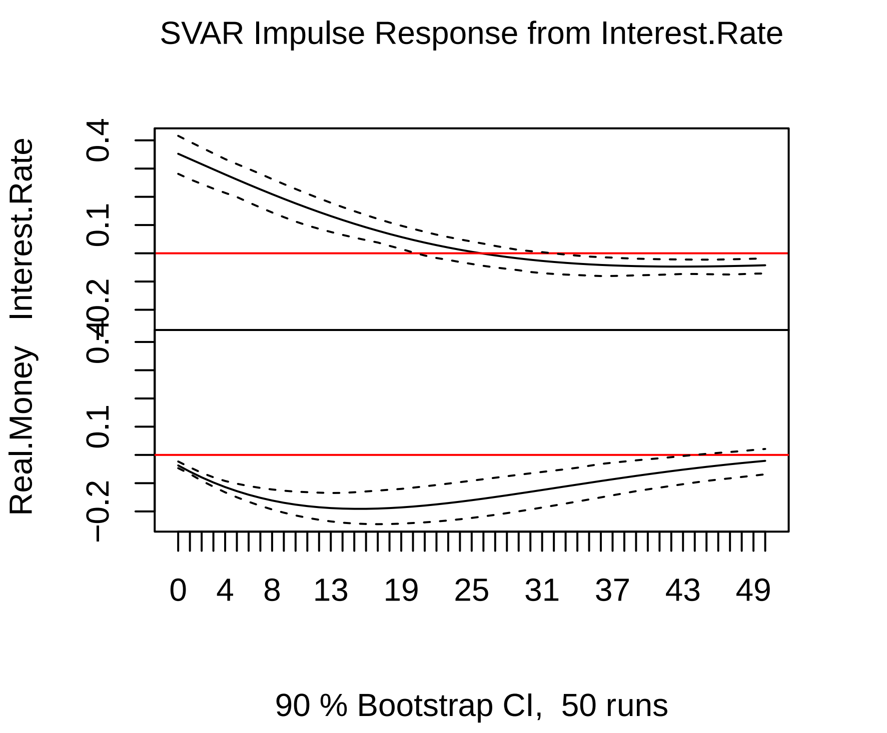
```
#check, should be close to covariance matrix under var model
svar2_B$B%*%t(svar2_B$B)
```

```
##               Interest.Rate  Real.Money
## Interest.Rate    0.12420301 -0.01309183
## Real.Money      -0.01309183  0.03088593
```

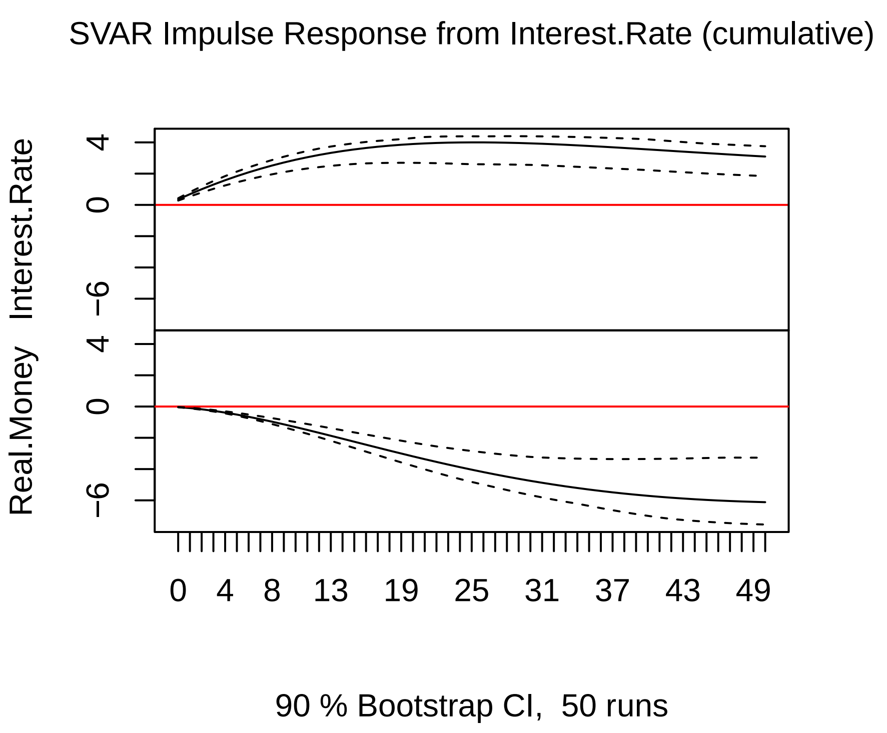
```
solve(svar2_B$B) # just the A matrix in A-model method
```

```
##               Interest.Rate Real.Money
## Interest.Rate     2.8374874   0.000000
## Real.Money        0.6136397   5.821637
```

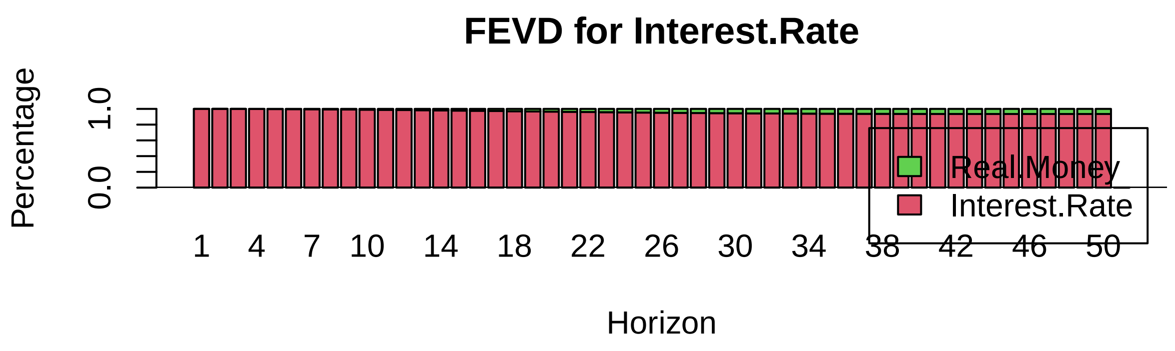
```
plot(irf(svar2_B, impulse = "Interest.Rate", ortho=T,n.ahead=50,cumulative=F,runs=50,ci=0.
```

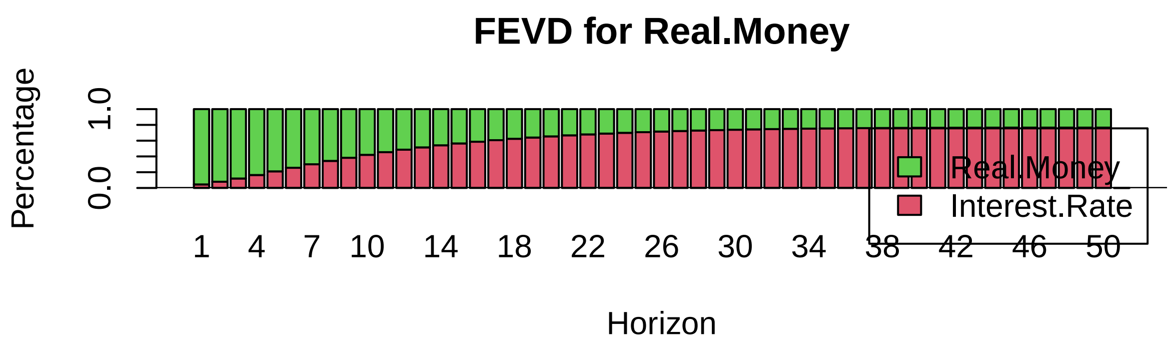


```
plot(irf(svar2_B, impulse = "Interest.Rate" ,ortho=T,n.ahead=50,cumulative=T,runs=50,ci=0.
```



```
plot(fevd(svar2_B,ortho=T,n.ahead=50,cumulative=F,runs=50,ci=0.9), col=(2:3))
```





```
data <- scale(df1[,c(1,2)],T,T)
data <- as.data.frame(data)
```

```
var2 = VAR(data,p=1)
```

```
# specify matrix B0 with contemporaneous restrictions - unrestricted coefficients are le
B0 <- diag(2)

B0[1, 1] <- 1
B0[2, 2] <- 1
B0[2, 1] <- NA # we impose 3 restrictions, but only 1 restriction is required, i.e., R=3
B0
```

2.3.1.4 Over-identifying test

```
##      [,1] [,2]
## [1,]    1    0
## [2,]   NA    1
```

```
svar1.d <- SVAR(var2, estmethod = "scoring", Amat = B0)
svar1.d$LR
```

```
##
##  LR overidentification
##
## data:  data
## Chi^2 = 4262.9, df = 2, p-value < 2.2e-16
```

The p-value is very small, so such restrictions are strongly rejected by data.

```
rm(list = ls())
```

2.3.2 Long-Run Restrictions

Blanchard and Quah (1989) propose an approach, which does not require to directly impose restrictions on the structural matrices A or B. Instead, structural innovations can be identified by looking at the accumulated effects of shocks and placing zero restrictions on those accumulated relationships, which die out and become zero in the long run.

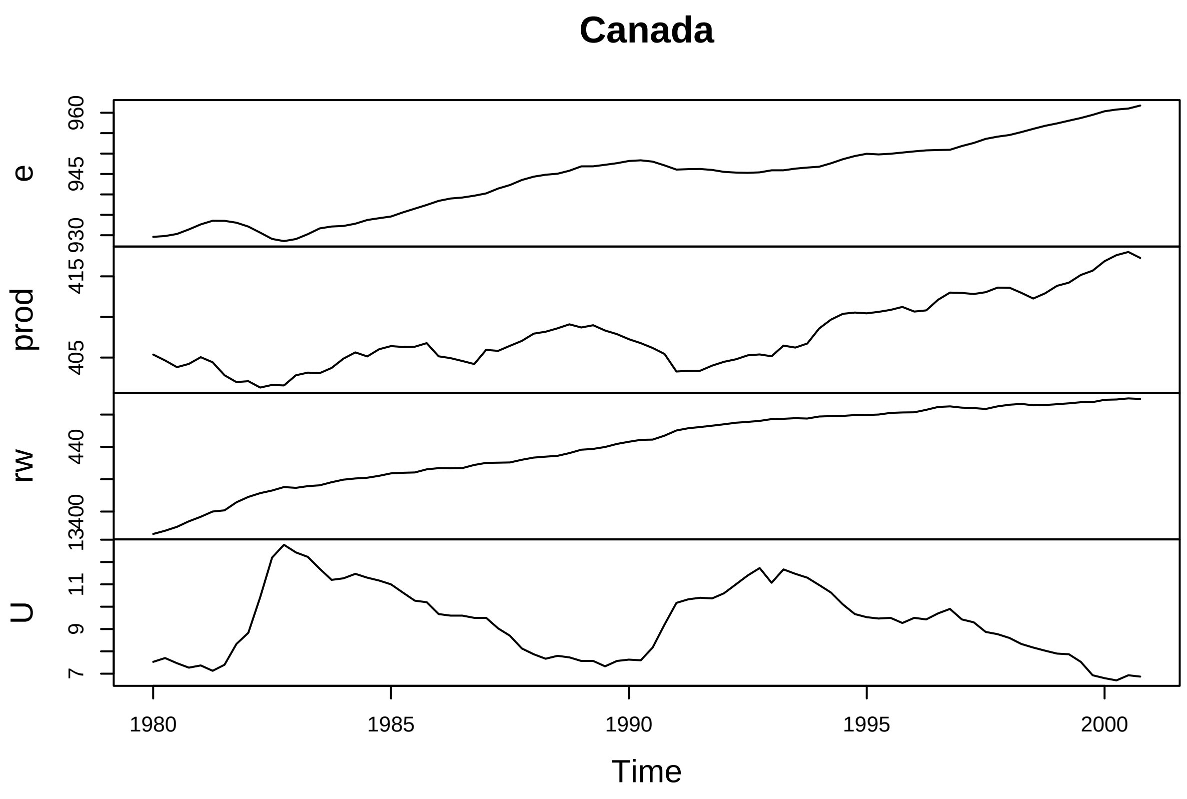
Intuition: Identification is achieved by assuming that some shocks have zero cumulative effect on some of the endogenous variables in the long run.

e.g. assume that monetary policy is neutral in the long-run and has no cumulative effect on the real economy.

Note 2.

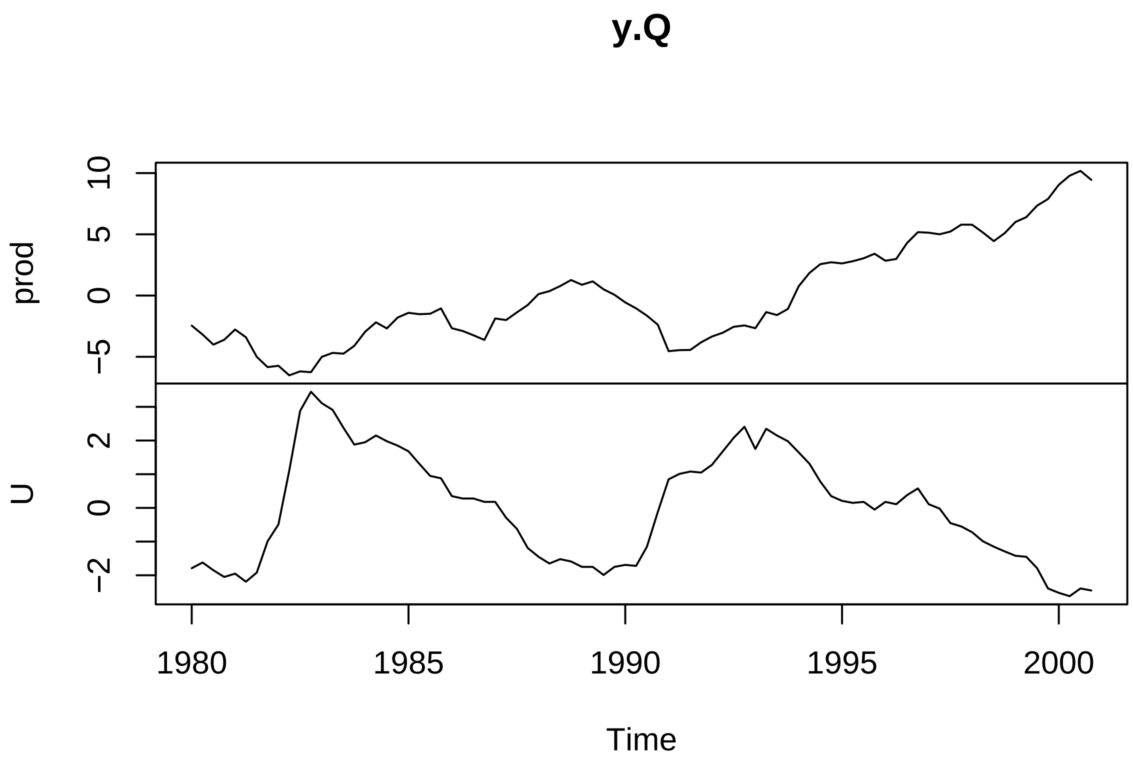
To use Blanchard and Quah technique, at least one variable must be nonstationary, I(0) variables do not have a permanent component.

```
# import macroeconomic time series for Canada
#  e    - employment
#  prod - productivity (real GDP per worker)
#  rw   - real wage in manufacturing
#  U    - unemployment rate
data(Canada)
plot(Canada)
```



```
# SVARs with long run restrictions

y.Q <- cbind(Canada[,"prod"], Canada[,"U"])
y.Q <- sweep(y.Q, 2, apply(y.Q, 2, mean)) # demean
```

```
colnames(y.Q) <- c("prod","U")
plot(y.Q)
```



```
# clearly, both are non-stationary
```

```
VARselect(y.Q, lag.max=8, type="none")
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      2      2
##
## $criteria
##                  1           2           3           4           5           6
## AIC(n) -2.59293371 -2.95868577 -2.87612762 -2.80796781 -2.73883975 -2.65173597
## HQ(n)  -2.54390878 -2.86063590 -2.72905281 -2.61186807 -2.49371507 -2.35758635
## SC(n)  -2.47026354 -2.71334542 -2.50811709 -2.31728711 -2.12548887 -1.91571491
## FPE(n)  0.07480209  0.05189717  0.05638968  0.06042198  0.06484403  0.07090549
##                  7           8
## AIC(n) -2.66412406 -2.63971055
## HQ(n)  -2.32094951 -2.24751107
## SC(n)  -1.80543283 -1.65834915
## FPE(n)  0.07025558  0.07230024
```

```
var2 <- VAR(y.Q, ic="SC", lag.max=8, type="none")

## Blanchard-Quah long run restriction: row 1 column 2 element of the cumulative effect
svar2 <- BQ(var2); summary(svar2)
```

```
##
## SVAR Estimation Results:
```

```
## ========================
##
## Call:
## BQ(x = var2)
##
## Type: Blanchard-Quah
## Sample size: 82
## Log Likelihood: -109.603
##
## Estimated contemporaneous impact matrix:
##        prod       U
## prod 0.5249 -0.4117
## U    0.2275  0.2462
##
## Estimated identified long run impact matrix:
##        prod     U
## prod 27.838 0.000
## U    -6.199 3.195
##
## Covariance matrix of reduced form residuals (*100):
##        prod      U
## prod 44.503  1.807
## U     1.807 11.233
```

The contemporaneous impact matrix reported is $\mathbf{B}_0^{-1}$, it shows the immediate effect of $\varepsilon_{j,t}$ on $y_{i,t}$ upon impact

Rows refer to two variables $(\text{Prod}_t, \text{U}_t)$, and the columns to the two shocks - technology shock $\varepsilon_{1,t}$ and non-technology shock $\varepsilon_{2,t}$

Here on impact a positive one standard deviation technology shock increases Productivity by 0.5249 and increases unemployment rate by 0.2279 (Why? Is it weird? Hint: compare this contemporaneous impact with the long run impact.)

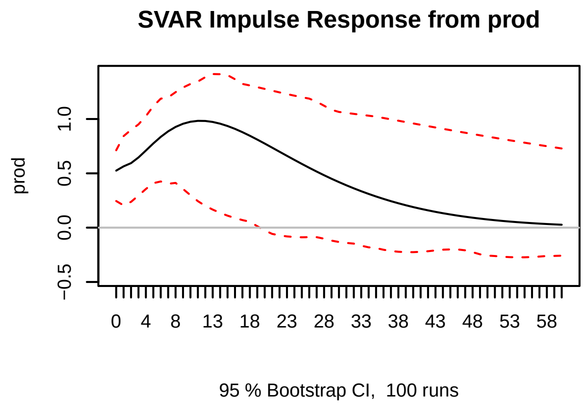
A negative one standard deviation non-technology shock lowers GDP on impact by 0.4117, increases unemployment rate by 0.2462.

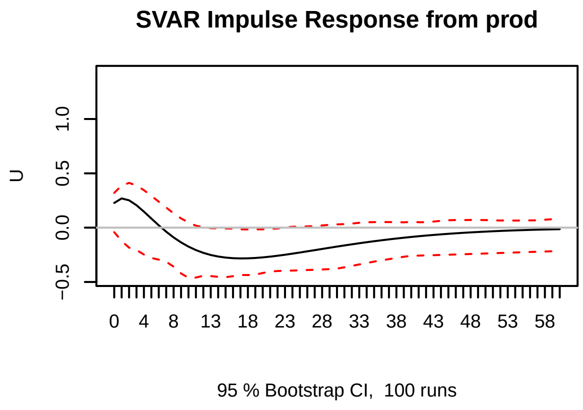
The long run impact matrix reported shows the cumulative long run impact.

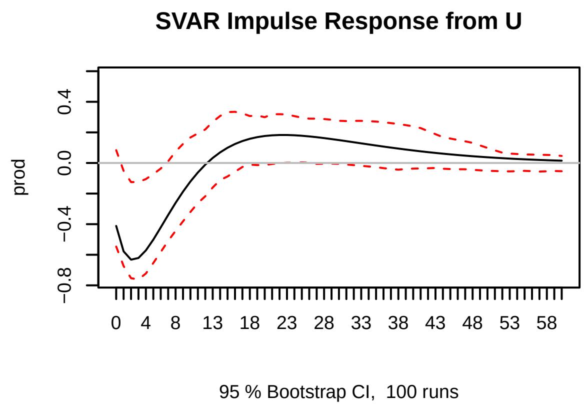
The long run cumulative effect of any non-technology shock on Productivity is 0 (this is the long run constraint we imposed)

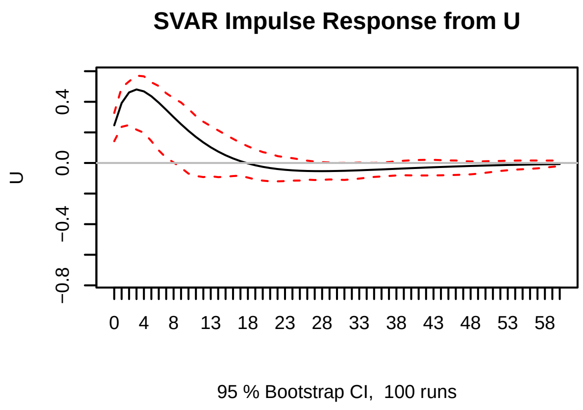
The long run cumulative effect of a single positive one standard deviation technology shocks on Productivity is to increase it by 27.838.

```r
par(mfrow=c(2,2), cex=0.6)
plot( irf(svar2, n.ahead=60), plot.type="single", ask=FALSE )
```









Note that by construction the contemporaneous impact matrix from `summary(svar2)` is identical to the elements of the IRFs for period 0 (impact period)

```
svar_irf_cumulative <- irf(svar2, n.ahead=60, cumulative = TRUE)
svar_irf_cumulative$irf[[1]][1,]
```

```
##      prod         U
## 0.5249342 0.2274585
```

```
svar_irf_cumulative$irf[[2]][1,]
```

```
##       prod          U
## -0.4116726  0.2461508
```

Also note that the long run impact matrix from `summary(svar2)` is essentially the same as the elements of the IRFs for period 100.

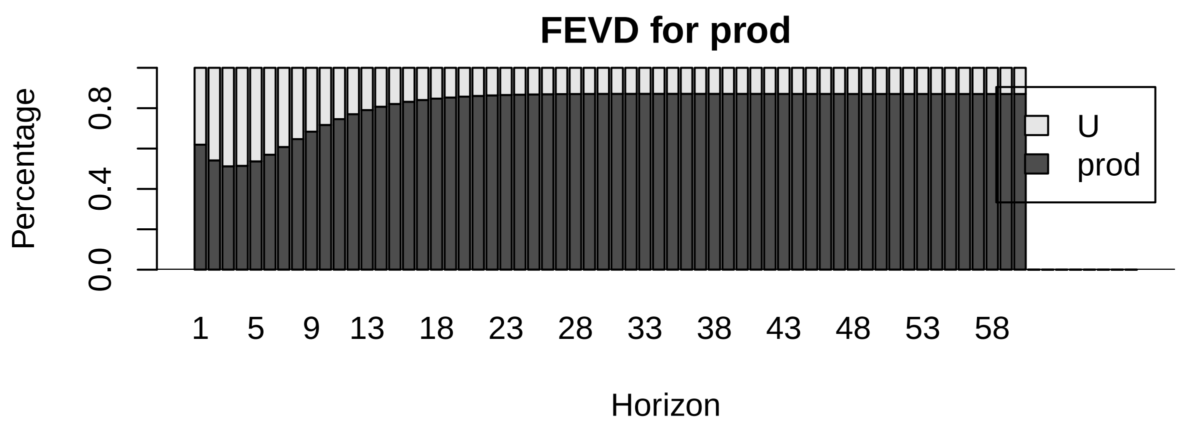
```
svar_irf_cumulative_long <- irf(svar2, n.ahead=100, cumulative = TRUE)
svar_irf_cumulative_long$irf[[1]][101,]
```

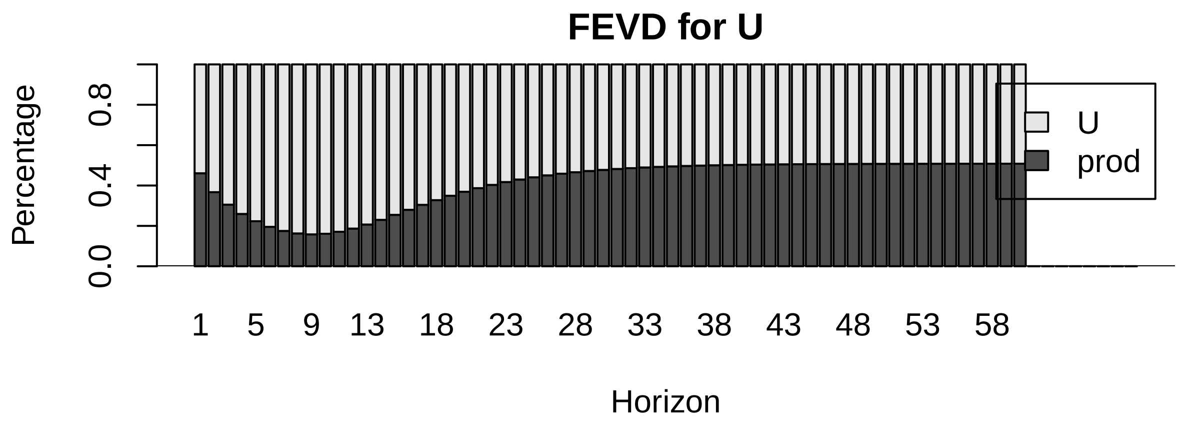
```
##      prod         U
## 27.839856 -6.199261
```

```
svar_irf_cumulative_long$irf[[2]][101,]
```

```
##         prod            U
## 0.0002371425 3.1953376608
```

```
par(mar=c(4,5,2,1))
plot( fevd(svar2, n.ahead=60) ,addbars=8 )
```





```r
rm(list = ls())
```

2.4 Historical Decomposition

Question 2. *What is the historical contribution of each structural shock in driving deviations of the VAR's the endogenous variables away from their equilibrium?*

For example, What was the contribution of oil shocks in driving the fall in GDP growth in 1973:Q4?

Historical decomposition quantifies how important a shock was in driving the behavior of the endogenous variables in a specific time period in the past.

HD allow to track, at each point in time, the role of structural shocks in driving the VAR's endogenous variables away from their steady state.

Consider simple bivariate VAR:

$$\begin{aligned} x_t &= \Phi x_{t-1} + u_t && (9) \\ &= \Phi(\Phi x_{t-2} + u_{t-1}) && (10) \\ &= \ldots && (11) \\ &= \Phi^t x_0 + \sum_{j=0}^{t-1} \Phi^j \mathbf{B}\epsilon_{t-j} && (12) \end{aligned}$$

where $u_t = \mathbf{B}\epsilon_t, \epsilon_t \sim N(\mathbf{0}, I_2)$

In particular consider t=2, we can write x_2 as a function of present (t=2) and past (t=1) structural shocks plus the initial condition (x_0)

$$x_2 = \underbrace{\Phi^2 x_0}_{\text{init}} + \underbrace{\Phi\mathbf{B}}_{\Theta_1}\epsilon_1 + \underbrace{\mathbf{B}}_{\Theta_0}\epsilon_2$$

Rewrite x_2 into matrix form:

$$\begin{bmatrix} y_2 \\ r_2 \end{bmatrix} = \begin{bmatrix} \text{init}_y \\ \text{init}_r \end{bmatrix} + \begin{bmatrix} \theta^1_{11} & \theta^1_{12} \\ \theta^1_{21} & \theta^1_{22} \end{bmatrix} \begin{bmatrix} \epsilon^y_1 \\ \epsilon^r_1 \end{bmatrix} + \begin{bmatrix} \theta^0_{11} & \theta^0_{12} \\ \theta^0_{21} & \theta^0_{22} \end{bmatrix} \begin{bmatrix} \epsilon^y_2 \\ \epsilon^r_2 \end{bmatrix}$$

Then x_2 can be expressed as

$$y_2 = \text{init}_y + \theta^1_{11}\epsilon^y_1 + \theta^1_{12}\epsilon^r_1 + \theta^0_{11}\epsilon^y_2 + \theta^0_{12}\epsilon^r_2 \tag{13}$$
$$r_2 = \text{init}_r + \theta^1_{21}\epsilon^y_1 + \theta^1_{22}\epsilon^r_1 + \theta^0_{21}\epsilon^y_2 + \theta^0_{22}\epsilon^r_2 \tag{14}$$

The historical decomposition is given by

$$\text{HD}^{\epsilon^y}_{y_2} = \theta^1_{11}\epsilon^y_1 + \theta^0_{11}\epsilon^y_2 \tag{15}$$
$$\text{HD}^{\epsilon^r}_{y_2} = \theta^1_{12}\epsilon^r_1 + \theta^0_{12}\epsilon^r_2 \tag{16}$$
$$\text{HD}^{\text{init}}_{y_2} = \text{init}_y \tag{17}$$

This sums up to y_2

$$\text{HD}^{\epsilon^y}_{r_2} = \theta^1_{21}\epsilon^y_1 + \theta^0_{21}\epsilon^y_2 \tag{18}$$
$$\text{HD}^{\epsilon^r}_{r_2} = \theta^1_{22}\epsilon^r_1 + \theta^0_{22}\epsilon^r_2 \tag{19}$$
$$\text{HD}^{\text{init}}_{r_2} = \text{init}_r \tag{20}$$

This sums up to r_2

```r
source("HD.R")
```

Notice that this function is translated from the function `VARhd` from Ambrogio Cesa-Bianchi's Matlab Toolbox.

For more details, please check Historical Decomposition In R

As input argument you have to use the output of the VAR function from the `vars` packages in R. The function returns a 3-dimensional array:

number of observations $\times$ number of shocks $\times$ number of variables

(Note: didn't translate the entire function, e.g. omitted the case of exogenous variables.)

To run it, you need two additional functions which were also translated from Bianchi's Toolbox.

```r
data(Canada)
var2 = VAR(Canada, p = 2, type = "both")
colnames(Canada)
```

```
## [1] "e"    "prod" "rw"   "U"
```

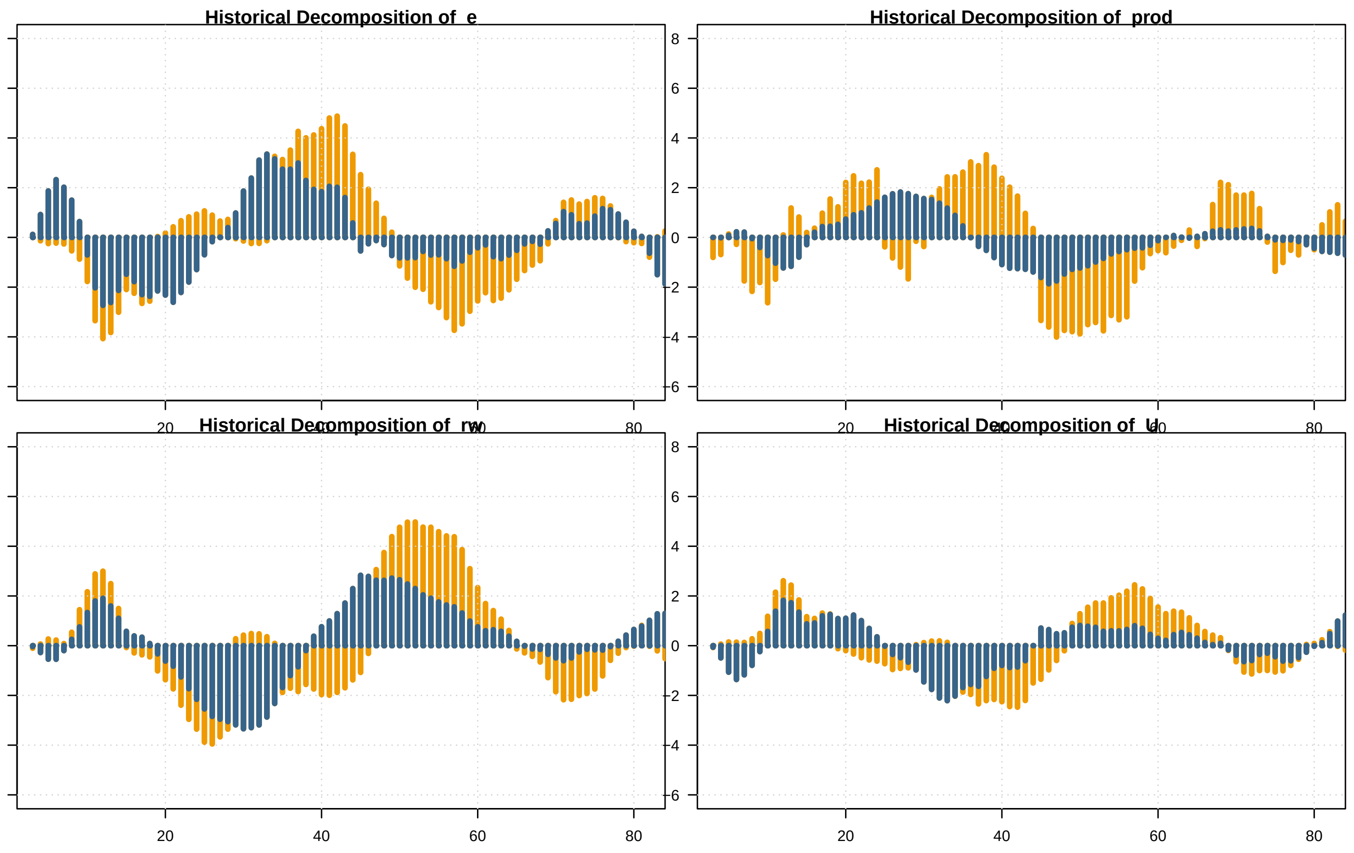
```
df1 = Canada
k <- ncol(df1)
```

```
HD = VARhd(Estimation=var2)
head(HD[,,1]) # historical decomposition of the first variable (employment)
```

```
##            [,1]       [,2]       [,3]       [,4]
## [1,]         NA         NA         NA         NA
## [2,]         NA         NA         NA         NA
## [3,] 0.1207048  0.0000000 0.00000000  0.0000000
## [4,] 0.9198561 -0.1366951 0.03350053 -0.1066752
## [5,] 1.8694315 -0.2490509 0.04521428 -0.2839466
## [6,] 2.3235205 -0.2311285 0.03862405 -0.4084350
```

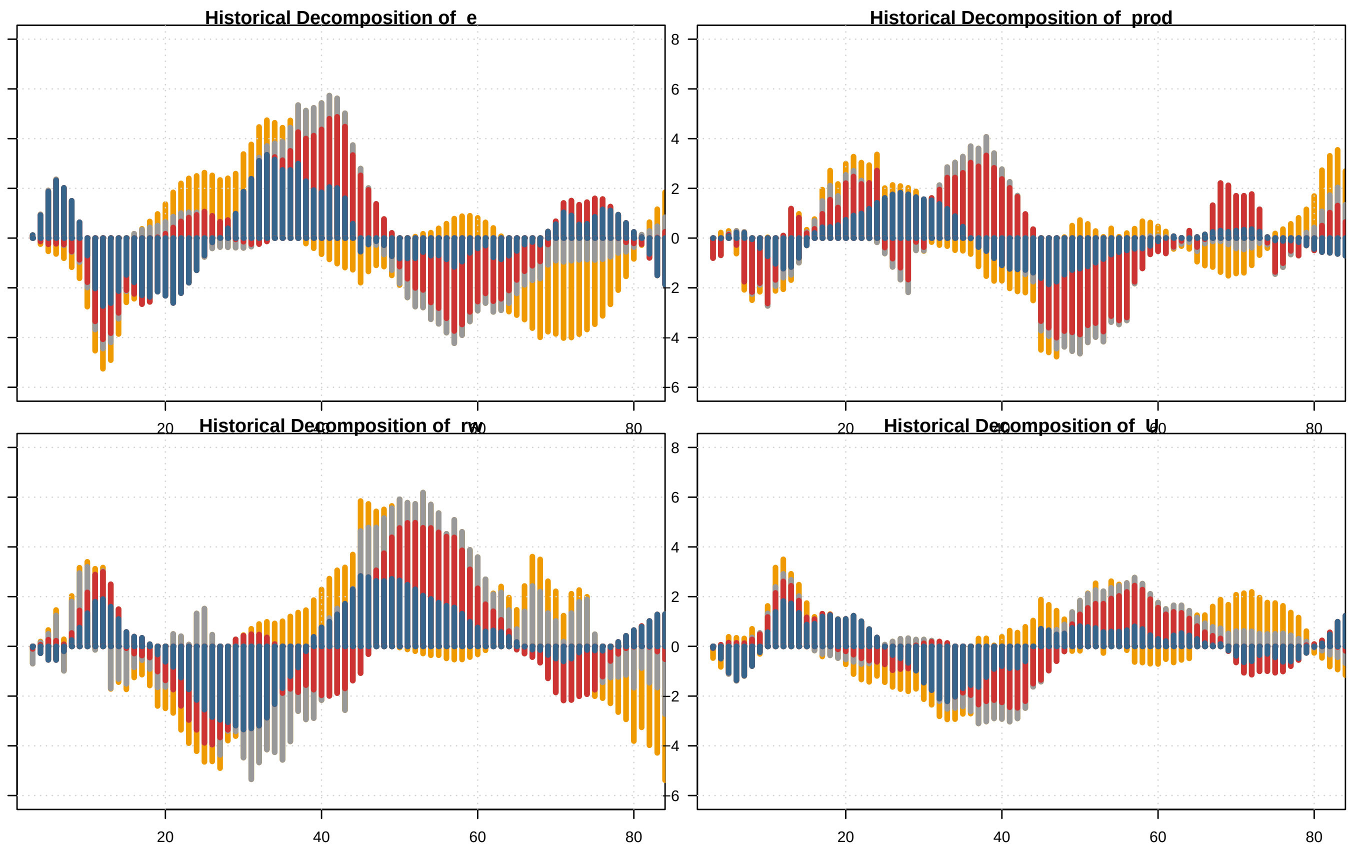
```
HD1 = ifelse(HD>=0,HD,0)
HD2 = ifelse(HD<0,HD,0)
```

```
par(mfrow=c(k,k/2), mar=c(0.9, 0.9, 0.9, 0.9))
for (i in 1:ncol(df1)) {
   plot(apply(HD1[,c(1,2),i],1,sum),type="h",col="orange2",ylim=c(-6,8),lwd=4,xaxs="i",las
   grid()
   lines(HD1[,1,i],type="h",col="steelblue4",ylim=c(-6,8),lwd=4) # historical decompositi

   lines(apply(HD2[,c(1,2),i],1,sum),type="h",col="orange2",ylim=c(-6,8),lwd=4) # histori
   lines(HD2[,1,i],type="h",col="steelblue4",ylim=c(-6,8),lwd=4) # historical decompositi
}
```



```
par(mfrow=c(k,k/2), mar=c(0.9, 0.9, 0.9, 0.9))
for (i in 1:ncol(df1)) {
   plot(apply(HD1[,c(1,2,3,4),i],1,sum),type="h",col="orange2",ylim=c(-6,8),lwd=4,xaxs="i'
   grid()
   lines(apply(HD1[,c(1,2,3),i],1,sum),type="h",col="grey60",ylim=c(-6,8),lwd=4)
   lines(apply(HD1[,c(1,2),i],1,sum),type="h",col="brown3",ylim=c(-6,8),lwd=4) # historic
   lines(HD1[,1,i],type="h",col="steelblue4",ylim=c(-6,8),lwd=4) # historical decompositi

   lines(apply(HD2[,c(1,2,3,4),i],1,sum),type="h",col="orange2",ylim=c(-6,8),lwd=4)
```

```
    lines(apply(HD2[,c(1,2,3),i],1,sum),type="h",col="grey60",ylim=c(-6,8),lwd=4)
    lines(apply(HD2[,c(1,2),i],1,sum),type="h",col="brown3",ylim=c(-6,8),lwd=4) # historica
    lines(HD2[,1,i],type="h",col="steelblue4",ylim=c(-6,8),lwd=4) # historical decompositi
}
```



```
rm(list = ls())
```