

Lecture 6, October 6, 2025

Estimation of VAR models

Marta Boczon

Department of Economics

Copenhagen Business School

mbo.eco@cbs.dk

```
In [1]: #install.packages("quantmod")
#install.packages("fredr")
#install.packages("ggfortify")
#install.packages('urca')
#install.packages("tseries")
#install.packages("forecast")
#install.packages("dynlm")
#install.packages("stargazer")
#install.packages("pracma")
#install.packages("dLagM")
#install.packages("gets")
#install.packages("car")
#install.packages("lmtest")
#install.packages("vars")
options(warn=-1)
```

Today we will discuss **Vector Autoregression (VAR) models**, focusing in particular on the role of **real wage / labor compensation in Canada**. We will build on the Canadian GDP and hours worked in the manufacturing sector data introduced earlier, and, in a bit, we will introduce a new variable: wages. However, before we get there, let us circle back to last week's material and practice one more time pseudo-out-of-sample forecasting.

```
In [21]: # Import the dataset from a CSV file downloaded from FRED
gdp = read.csv("NGDPRSADEXCAQ.csv")
# Convert the second column of the dataset into a time series object
tsgdp = ts(gdp[, 2], frequency = 4, start = c(1961, 1))
# Transform the series into (approximate) percentage changes
dgdp = diff(log(tsgdp)) * 100
```

```
In [22]: # Import the dataset from a CSV file downloaded from FRED
hours = read.csv("HOHWMN02CAM065N.csv")
# Convert the second column of the dataset into a time series object
tshours = ts(hours[, 2], frequency = 4, start = c(1960, 1))
# Transform the series into a change from the previous period
dhours = diff(tshours)
```

Exercise 1 (15')

The code below can be found in Lecture 5. We used it to compute recursive out of sample forecasts for AR(1). Modify the code below to find **the Mean Absolute Error and the Root Mean Square Forecast Error** for **1 and 2-step ahead** forecast of Canadian GDP growth rate using AR(1).

```
In [23]: # https://www.rdocumentation.org/packages/forecast VERSIONS/8.16/topics/forecast
library(forecast)

# Length of sample
T = length(dgdp)

# Container for results
results = data.frame()

# Initial estimation window
Tstar = 200

# Number of iterations
Tmax = T - Tstar

cat("Our data has", T, "observations.\n")
cat("We will keep the first", Tstar, "observations fixed for stability (initial estimation window).\n")
cat("From there, we add one observation at a time and forecast one step ahead at each iteration.\n")
cat("We will produce", Tmax, "out-of-sample 1-step-ahead forecasts, ending at observation", T, ".\n\n")

for (i in 1:Tmax) {
  # Expand window estimation: use data up to time Tstar + i - 1
  model = arima(dgdp[1:(Tstar + i - 1)], order = c(1, 0, 0))
```

```

# One-step-ahead forecast with a single 95% PI
f = forecast::forecast(model, h = 1, level = 95)

# Extract components (coerce to numeric to avoid ts attributes in arithmetic)
f.mean = as.numeric(f$mean) # point forecast  $y_{T+1|T}$ 
f.lower95 = as.numeric(f$lower) # 95% Lower bound
f.upper95 = as.numeric(f$upper) # 95% upper bound
obs.val = as.numeric(dgdp[Tstar + i]) # actual  $y_{T+1}$ 

# Absolute forecast error
f.error = abs(f.mean - obs.val)

# Row to append
temp = c(f.mean, f.lower95, f.upper95, obs.val, f.error)
results = rbind(results, temp)
}

# Name the columns
names(results) = c("Mean forecast", "Lower bound 95%", "Upper bound 95%", "Actual value", "Absolute error")

# Quick peek at the first 6 iterations
head(results)

# Quick peek at the last 6 iterations
tail(results)

# Save AR(1) rolling results
results.AR1 = results

# Mean Absolute Error (MAE)
mae.AR1 = mean(results.AR1[, "Absolute error"])

# Root Mean Squared Forecast Error (RMSFE)
rmsfe.AR1 = sqrt(mean((results.AR1[, "Mean forecast"] - results.AR1[, "Actual value"])^2))

cat("The MAE is", round(mae.AR1, 5), "and the RMSFE is", round(rmsfe.AR1, 5))

```

Our data has 256 observations.

We will keep the first 200 observations fixed for stability (initial estimation window). From there, we add one observation at a time and forecast one step ahead at each iteration. We will produce 56 out-of-sample 1-step-ahead forecasts, ending at observation 256 .

A data.frame: 6 × 5

	Mean forecast	Lower bound 95%	Upper bound 95%	Actual value	Absolute error
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	0.8011480	-0.7919774	2.394273	0.18136394	0.6197841
2	0.5725205	-1.0189328	2.163974	1.35896277	0.7864423
3	1.0346830	-0.5565105	2.625876	0.79159116	0.2430918
4	0.8136804	-0.7739418	2.401303	0.06311038	0.7505700
5	0.5283029	-1.0587520	2.115358	0.32454382	0.2037591
6	0.6279366	-0.9554827	2.211356	0.13675060	0.4911860

A data.frame: 6 × 5

	Mean forecast	Lower bound 95%	Upper bound 95%	Actual value	Absolute error
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
51	0.7309076	-1.745116	3.206931	-0.03829651	0.7692042
52	0.7308181	-1.742088	3.203724	0.17659246	0.5542256
53	0.7335084	-1.735429	3.202446	0.62945465	0.1040538
54	0.7451791	-1.718907	3.209266	0.59438355	0.1507956
55	0.7436371	-1.715664	3.202938	0.50857420	0.2350629
56	0.7403866	-1.714256	3.195029	0.54794362	0.1924430

The MAE is 0.98796 and the RMSFE is 2.60835

Solution to Exercise 1

```

In [24]: # https://www.rdocumentation.org/packages/forecast VERSIONS 8.16/topics/forecast
library(forecast)

# Length of sample
T = length(dgdp)

# Initial estimation window
Tstar = 200

# Number of iterations
Tmax = T - Tstar

# Forecast horizon

```

```

h = 2

# Container for results
results = replicate(h, data.frame(), simplify = FALSE)

cat("Our data has", T, "observations.\n")
cat("We will keep the first", Tstar, "observations fixed for stability (initial estimation window).\n")
cat("From there, we add one observation at a time and forecast one step ahead at each iteration.\n")
cat("We will produce", Tmax, "out-of-sample 1-step-ahead forecasts, ending at observation", T, ".\n\n")

for (i in 1:Tmax) {
  # Expand window estimation: use data up to time Tstar + i - h
  model = arima(dgdp[1:(Tstar + i - h)], order = c(1, 0, 0))

  # One-step-ahead forecast with a single 95% PI
  f = forecast::forecast(model, h = h, level = 95)
  # Extract components (coerce to numeric to avoid ts attributes in arithmetic)
  f.mean = as.numeric(f$mean)          # point forecasts
  f.lower95 = as.numeric(f$lower)       # 95% Lower bound
  f.upper95 = as.numeric(f$upper)       # 95% upper bound
  obs.val = as.numeric(dgdp[(Tstar + i + 1 - h) : (Tstar + i)])  # actual values
  # Absolute forecast error
  f.error = abs(f.mean - obs.val)
  # Row to append
  for (j in 1:h) {
    temp = c(f.mean[j], f.lower95[j], f.upper95[j], obs.val[j], f.error[j])
    results[[j]] = rbind(results[[j]], temp)
  }
}

for (j in 1:h) {
  colnames(results[[j]])= c("Mean forecast", "Lower bound 95%", "Upper bound 95%", "Actual value", "Absolute error")
  rownames(results[[j]]) = NULL

  # Quick peek at the first 6 iterations
  # print(head(results[[j]]))

  # Quick peek at the last 6 iterations
  # print(tail(results[[j]]))

  # Mean Absolute Error (MAE)
  temp = mean(results[[j]][, "Absolute error"])
  cat("h=", j, "MAE=", round(temp, 5), "\n")

  # Root Mean Squared Forecast Error (RMSFE)
  temp = sqrt(mean((results[[j]][, "Absolute error"])^2))
  cat("h=", j, "RMSFE=", round(temp ,5), "\n")
}

```

Our data has 256 observations.

We will keep the first 200 observations fixed for stability (initial estimation window).

From there, we add one observation at a time and forecast one step ahead at each iteration.

We will produce 56 out-of-sample 1-step-ahead forecasts, ending at observation 256 .

```

h= 1 MAE= 0.98779
h= 1 RMSFE= 2.60834
h= 2 MAE= 0.9684
h= 2 RMSFE= 2.21779

```

Wonderful!

Now we can refocus on building a **VAR model**. We are going to construct a three-variable VAR model. Two of the three variables—GDP and hours worked—are already imported and transformed into stationary series. Now, we will import data ob wages and defalte it by Canadian CPI index.

1. Data: Hourly Earnings: Earnings: Economic Activity: Manufacturing: Total Economy for Canada (CANHOUREAMISMEI)

2. Source: Organization for Economic Co-operation and Development via FRED®

3. Frequency: Quarterly (aggregation method: Average)

4. Units: Index 2015 = 100

5. Seasonal Adjustment: Seasonally Adjusted

[View on FRED](#)

1. Data: Consumer Price Indices (CPIs, HICPs), COICOP 1999: Consumer Price Index: Total for Canada (CANCPIALLMINMEI)

2. Source: Organization for Economic Co-operation and Development via FRED®

3. Frequency: Quarterly (aggregation method: Average)

4. Units: Index 2015=100

5. Seasonal adjustment: Not Seasonally Adjusted

[View on FRED](#)

```
In [25]: # Import the dataset from a CSV file downloaded from FRED
wages = read.csv("CANHOUREAMISMEI.csv")
# Convert the second column of the dataset into a time series object
tswages = ts(wages[, 2], frequency = 4, start = c(1961, 1))
```

```
In [26]: # Import the dataset from a CSV file downloaded from FRED
cpi = read.csv("CANCPIALLMINMEI.csv")
# Convert the second column of the dataset into a time series object
tscpi = ts(cpi[, 2], frequency = 4, start = c(1914, 1))
```

```
In [27]: # Align the two time series on their common time span and timestamps.
temp = ts.intersect(tswages, tscpi)
# Compute a REAL WAGE INDEX:
tswages = temp[,1] / temp[,2] * 100
```

Exercise 2 (15')

Transform real wages to a stationary series.

Solution to Exercise 2

```
In [28]: # Calculate the number of observations in the time series object
nd = length(tswages)

# Print a message along with the number of observations
cat("There are", nd, "observations in our data set.")
```

There are 257 observations in our data set.

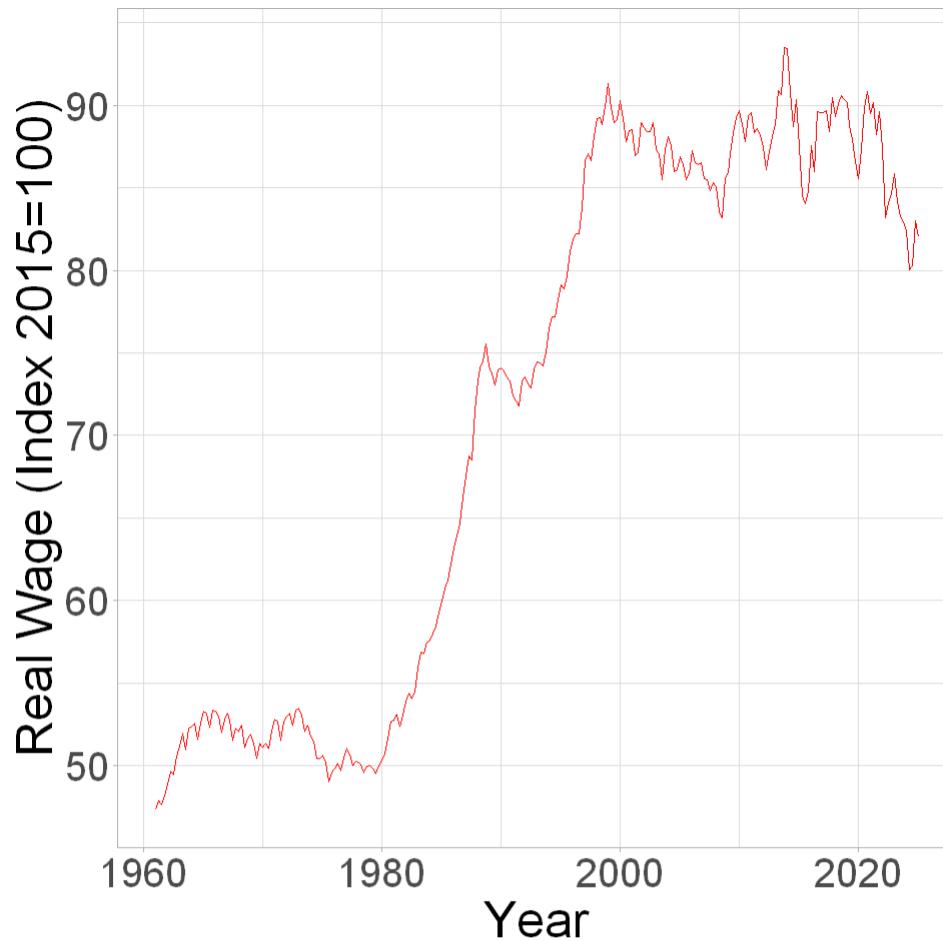
```
In [29]: # Load required libraries
library(ggplot2)    # for plotting
library(ggfortify) # for autoplot of time series objects

# Set default figure size (width = 8, height = 8)
options(repr.plot.width = 8, repr.plot.height = 8)

# Create an initial time series plot of tspce in red
fig = autoplot(tswages, colour = 'red')

# Customize the plot
fig = fig +
  theme(aspect.ratio = 1) +                      # make plot square-shaped
  theme_light() +                                # use a light theme
  theme(aspect.ratio = 1) +                      # reinforce square aspect ratio
  theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) + # adjust margins
  theme(text = element_text(size = 30)) +          # set large font size for readability
  labs(x = "Year") +                            # Label x-axis
  labs(y = "Real Wage (Index 2015=100)")        # Label y-axis

# Display the plot
fig
```



```
In [30]: library(urca)
summary(ur.df(tswages, type='trend', lags=8, selectlags="AIC"))

#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####

Test regression trend

Call:
lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.4380 -0.5102 -0.0191  0.4879  3.2507 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.5632472  0.4781869   1.178  0.24003  
z.lag.1     -0.0084397  0.0105710  -0.798  0.42545  
tt          0.0009547  0.0024227   0.394  0.69388  
z.diff.lag1  0.0683905  0.0643389   1.063  0.28888  
z.diff.lag2  0.0620587  0.0653942   0.949  0.34359  
z.diff.lag3  0.0368339  0.0629825   0.585  0.55922  
z.diff.lag4  0.1989248  0.0632956   3.143  0.00189 ** 
z.diff.lag5 -0.0138480  0.0636163  -0.218  0.82787  
z.diff.lag6 -0.2630739  0.0636725  -4.132  5e-05 ***  
z.diff.lag7  0.0243001  0.0659069   0.369  0.71268  
z.diff.lag8  0.2078281  0.0658597   3.156  0.00181 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.009 on 237 degrees of freedom
Multiple R-squared:  0.1741,    Adjusted R-squared:  0.1393 
F-statistic: 4.997 on 10 and 237 DF,  p-value: 1.382e-06

Value of test-statistic is: -0.7984 1.0079 0.7382

Critical values for test statistics:
  1pct  5pct 10pct
tau3 -3.98 -3.42 -3.13
phi2  6.15  4.71  4.05
phi3  8.34  6.30  5.36
```

ADF (with constant + trend): Interpretation

From the output:

- Test statistics: **tau₃ = -0.7984, phi₂ = 1.0079, phi₃ = 0.7382**
- Critical values:
 - **tau₃**: -3.98 (1%), -3.42 (5%), -3.13 (10%)
 - **phi₂**: 6.15 (1%), 4.71 (5%), 4.05 (10%)
 - **phi₃**: 8.34 (1%), 6.30 (5%), 5.36 (10%)

Deterministic components: Both **phi₂** and **phi₃** are below their 10% critical values \Rightarrow we **cannot reject** that the constant and deterministic trend are zero.

Conclusion: Re-specify the ADF without both constant and trend (no deterministic terms).

```
In [31]: library(urca)
summary(ur.df(tswages, type='none', lags=8, selectlags="AIC"))

#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####

Test regression none

Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.4549 -0.5138  0.0706  0.5528  3.2117 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
z.lag.1    0.0008630  0.0009099   0.949  0.34381  
z.diff.lag1 0.0678321  0.0636449   1.066  0.28759  
z.diff.lag2 0.0608743  0.0643775   0.946  0.34532  
z.diff.lag3 0.0355123  0.0623128   0.570  0.56928  
z.diff.lag4 0.1978536  0.0627936   3.151  0.00184 ** 
z.diff.lag5 -0.0169598 0.0628237  -0.270  0.78743  
z.diff.lag6 -0.2658356 0.0629464  -4.223 3.43e-05 *** 
z.diff.lag7 0.0238540  0.0652583   0.366  0.71504  
z.diff.lag8 0.2073279  0.0651406   3.183  0.00165 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.009 on 239 degrees of freedom
Multiple R-squared:  0.1775,    Adjusted R-squared:  0.1466 
F-statistic: 5.733 on 9 and 239 DF,  p-value: 3.326e-07

Value of test-statistic is: 0.9486

Critical values for test statistics:
  1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

ADF (no constant, no trend): Interpretation

From the output:

- Test statistic: **tau₁ = 0.9486**
- Critical values (1%, 5%, 10%): **-2.58, -1.95, -1.62**

Because **0.9486 > -1.62** (i.e., it is far less negative than any critical value), we **fail to reject** the null hypothesis of a **unit root** at all conventional levels.

Conclusion: With **no deterministic terms**, the series is **non-stationary in levels**.

Next step: Use **first differences** (or log-differences) and re-run the ADF.

```
In [32]: # Interpretation: diff(log(.)) ~ growth rate (in fractions; ×100 = %).
# Plain diff is a change in index points/units.
# It depends on the scale/base year and is harder to compare.
dwages = diff(log(tswages)) * 100
```

```
In [33]: library(urca)
summary(ur.df(dwages, type='none', lags=8, selectlags="AIC"))
```

```

#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####

Test regression none

Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.4202 -0.6380  0.1144  0.8199  4.0168 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
z.lag.1     -0.51530   0.13145 -3.920 0.000116 ***  
z.diff.lag1 -0.41613   0.12546 -3.317 0.001052 **   
z.diff.lag2 -0.32571   0.11888 -2.740 0.006613 **   
z.diff.lag3 -0.29581   0.11614 -2.547 0.011496 *    
z.diff.lag4 -0.03036   0.11290 -0.269 0.788259    
z.diff.lag5 -0.07070   0.10074 -0.702 0.483503    
z.diff.lag6 -0.28062   0.08577 -3.272 0.001226 **   
z.diff.lag7 -0.28121   0.06311 -4.456 1.28e-05 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.269 on 239 degrees of freedom
Multiple R-squared:  0.5883, Adjusted R-squared:  0.5745 
F-statistic: 42.69 on 8 and 239 DF,  p-value: < 2.2e-16

```

Value of test-statistic is: -3.9203

Critical values for test statistics:
1pct 5pct 10pct
tau1 -2.58 -1.95 -1.62

ADF (no constant, no trend): Interpretation

From the output:

- Test statistic: **tau1 = -3.9203**
- Critical values (1%, 5%, 10%): **-2.58, -1.95, -1.62**

Decision: Because $-3.9203 < -2.58$, we **reject** the null hypothesis of a **second unit root** at the **1%** level.

Vector Autoregressive (VAR) Models

The vector autoregression (VAR) model extends the idea of univariate autoregression to k time series regressions, where the lagged values of all k series appear as regressors. Put differently, in a VAR model we regress a vector of time series variables on lagged vectors of these variables.

As for univariate distributed lag models, one should think carefully about variables to include in a VAR, as adding unrelated variables reduces the forecast accuracy by increasing the estimation error. This is particularly important because the number of parameters to be estimated grows **quadratically** to the number of variables modeled by the VAR.

Structural Shocks in SVAR Models

We write the structural VAR as

$$A y_t = \sum_{i=1}^p A_i y_{t-i} + B \varepsilon_t, \quad E[\varepsilon_t \varepsilon_t'] = I_k,$$

where $y_t \in \mathbb{R}^k$, and $A, A_i, B \in \mathbb{R}^{k \times k}$.

Raw parameter count (before identification restrictions)

A , each A_i , and B are $k \times k$ matrices $\rightarrow k^2$ free entries each.

- **A:** k^2
- $\{A_i\}_{i=1}^p: p \times k^2$
- **B:** k^2

Total (raw) parameters:

$$(p+2)k^2 = k^2 + p \times k^2 + k^2.$$

Example for $k = 3$ and $p = 1$:

Plugging $k = 3$ into $(p + 2)k^2$:

- **A:** $3^2 = 9$
- $\{A_i\}_{i=1}^p : 1 \times 3^2 = 9$
- **B:** $3^2 = 9$

Total (raw) parameters:

[27.]

VAR Models in *Reduced Form*

Start with the structural VAR:

$$A y_t = \sum_{i=1}^p A_i y_{t-i} + B \varepsilon_t, \quad E[\varepsilon_t \varepsilon_t'] = I_k.$$

Multiply both sides by A^{-1} to obtain the **reduced form**:

$$y_t = \underbrace{\sum_{i=1}^p \Phi_i y_{t-i}}_{\text{autoregressive part}} + u_t, \quad \Phi_i \equiv A^{-1} A_i, \quad u_t \equiv A^{-1} B \varepsilon_t.$$

The reduced-form innovations have covariance

$$\Sigma_u \equiv E[u_t u_t'] = A^{-1} B B' A^{-\top},$$

which is a **symmetric** $k \times k$ matrix.

Parameter count in the *reduced form* (general k)

- **Autoregressive coefficients:** $\{\Phi_i\}_{i=1}^p$, each $k \times k \rightarrow p k^2$.
- **Innovation covariance:** Σ_u symmetric $k \times k \rightarrow \frac{k(k+1)}{2}$.

Total (reduced-form) parameters:

$$p k^2 + \frac{k(k+1)}{2}.$$

Example for $k = 3$ and $p = 1$

- **Autoregressive coefficients:** $1 \times 3^2 = 9$.
- **Innovation covariance:** $\frac{3(3+1)}{2} = 6$.

Total (reduced-form) parameters:

[15]

Identification of SVAR

How many restrictions are needed to identify A and B ?

Raw structural parameters: $(p + 2)k^2$

Reduced-form parameters: $p k^2 + \frac{k(k+1)}{2}$

Restrictions needed (beyond reduced form) to identify A and B (general k):

$$(p + 2)k^2 - \left(p k^2 + \frac{k(k+1)}{2}\right) = \boxed{\frac{k(3k-1)}{2}}.$$

Example for $k = 3$ and $p = 1$:

$27 - 15 = \boxed{12}.$

Identification Schemes in SVARs: A, B, and AB models

A model (short-run restrictions on contemporaneous relations)

$$A y_t = \varepsilon_t.$$

- Identification via **zero restrictions in A** .
- **Cholesky decomposition** corresponds to choosing A **lower-triangular**, implying a **recursive ordering**:
 - Variable 1 can contemporaneously affect all others,
 - Variable 2 can affect all except variable 1,
 - ...
 - Variable n affects none contemporaneously.
- This is the classic **recursive identification**.

Connection to reduced form: If the reduced-form residuals are u_t , then with this setup $u_t = A^{-1}\varepsilon_t$ and

$$\Sigma_u \equiv \mathbb{E}[u_t u_t'] = A^{-1}(A^{-1})'.$$

B model (restrictions on contemporaneous shocks)

$$y_t = B\varepsilon_t.$$

- Identification via **zero (or other) restrictions in B** that map **structural shocks** directly into variables.
- No recursive/Cholesky structure is assumed; it's **more general** and often **theory-driven**.

Here the reduced-form covariance is

$$\Sigma_u = \mathbb{E}[y_t y_t'] = BB'.$$

AB model (mixed restrictions)

$$Ay_t = B\varepsilon_t.$$

- Most general short-run scheme: impose identifying restrictions **on both A and B** .

The reduced-form residuals $u_t = A^{-1}B\varepsilon_t$ have covariance

$$\Sigma_u = A^{-1}BB'(A^{-1})'.$$

Cholesky decomposition

- **Normalization:** Set $B = I_k$. This imposes k^2 restrictions.

This means each structural shock directly affects only one equation — a **one-to-one mapping** between shocks and equations.

Identification: impose a lower-triangular structure on A

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{k1} & \cdots & a_{k,k-1} & a_{kk} \end{bmatrix},$$

$a_{ii} > 0$, $a_{ij} = 0$ for $i < j$.

This imposes $\frac{k^2-k}{2}$ restrictions.

This recursive structure encodes an **ordering** among variables:

1. Variable 1 is contemporaneously exogenous,
2. Variable 2 can react instantaneously to variable 1,
3. Variable 3 can react instantaneously to variables 1 and 2, etc.

Total number of restrictions

The total equals the difference between the number of parameters in the structural and reduced-form VAR:

$$k^2 + \frac{k^2 - k}{2} = \boxed{\frac{k(3k-1)}{2}}.$$

Interpretation

Now the **contemporaneous comovement is captured by A** — current values of some variables can **instantaneously** affect others via the off-diagonal entries of A . Since $B = I_k$, each structural shock hits only its own equation **on impact**.

Example $k = 3$

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow 9 \text{ restrictions.}$$

$$A = \begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \Rightarrow 3 \text{ restrictions.}$$

Additional Information

New Introduction to Multiple Time Series Analysis

Author: Helmut Lütkepohl

A profound introduction to the main steps of analyzing multiple time series — model specification, estimation, model checking, and using the models for economic analysis and forecasting.

Estimation of a Reduced-form VAR

```
In [34]: # Align the three (differenced) series on their common time span & timestamps.
data = ts.intersect(dgdp, dhours, dwages)
# Quick peek at the first 6 observations
head(data)
# Quick peek at the last 6 observations
tail(data)
```

A Time Series: 6 × 3

	dgdp	dhours	dwages
1961 Q2	2.5005468	0.4	1.1534143
1961 Q3	2.4042802	0.3	-0.4597739
1961 Q4	1.3513087	-0.6	1.1454925
1962 Q1	2.6262130	0.5	1.3574846
1962 Q2	0.7915743	0.1	1.5961812
1962 Q3	1.2708511	0.2	-0.4105380

A Time Series: 6 × 3

	dgdp	dhours	dwages
2022 Q2	1.0229149	-0.1	-4.9060355
2022 Q3	0.5138201	0.2	1.0363056
2022 Q4	-0.1370987	-0.1	0.6562696
2023 Q1	1.3454326	0.4	1.4410964
2023 Q2	0.1425423	0.2	-1.9717776
2023 Q3	-0.2258095	0.1	-1.1374485

```
In [35]: #https://www.rdocumentation.org/packages/vars/versions/1.5-6/topics/VARselect
library(vars)
# Choose VAR Lag order (up to 8) including an intercept ("const").
out = VARselect(data, lag.max = 8, type = "const")

# Shows the recommended Lag for each information criterion:
# AIC(n), HQ(n), SC(n), and FPE(n). They may disagree.
out$selection
```

AIC(n): 8 HQ(n): 4 SC(n): 1 FPE(n): 8

AIC(n)=4, HQ(n)=4, SC(n)=1, FPE(n)=4

Information criteria recommend these lag orders:

- **AIC** → $p = 4$
- **HQ (Hannan–Quinn)** → $p = 4$
- **SC (Schwarz/BIC)** → $p = 1$
- **FPE (Final Prediction Error)** → $p = 4$

How to read this: AIC/HQ/FPE favor a richer model ($p = 4$), while SC/BIC—being more conservative—prefers a parsimonious model ($p = 1$). This is common in small-to-moderate samples because BIC penalizes extra lags more heavily.

```
In [36]: library(vars)
```

```
# Fit a VAR(1) with an intercept on the aligned (stationary) series.
model.VAR1 = VAR(data, p = 1, type = "const")
summary(model.VAR1)
```

VAR Estimation Results:
=====

Endogenous variables: dgdp, dhours, dwages
Deterministic variables: const
Sample size: 249
Log Likelihood: -912.634
Roots of the characteristic polynomial:
0.2489 0.1905 0.1905
Call:
VAR(y = data, p = 1, type = "const")

Estimation results for equation dgdp:
=====

dgdp = dgdp.l1 + dhours.l1 + dwages.l1 + const

	Estimate	Std. Error	t value	Pr(> t)
dgdp.l1	-0.02591	0.06484	-0.400	0.6897
dhours.l1	0.53520	0.22311	2.399	0.0172 *
dwages.l1	0.11928	0.05675	2.102	0.0366 *
const	0.74515	0.09381	7.943	7.15e-14 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.246 on 245 degrees of freedom
Multiple R-Squared: 0.03979, Adjusted R-squared: 0.02803
F-statistic: 3.384 on 3 and 245 DF, p-value: 0.01884

Estimation results for equation dhours:
=====

dhours = dgdp.l1 + dhours.l1 + dwages.l1 + const

	Estimate	Std. Error	t value	Pr(> t)
dgdp.l1	0.008555	0.017917	0.477	0.633440
dhours.l1	-0.243314	0.061655	-3.946	0.000104 ***
dwages.l1	0.069679	0.015683	4.443	1.34e-05 ***
const	-0.037795	0.025924	-1.458	0.146144

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3443 on 245 degrees of freedom
Multiple R-Squared: 0.128, Adjusted R-squared: 0.1173
F-statistic: 11.99 on 3 and 245 DF, p-value: 2.378e-07

Estimation results for equation dwages:
=====

dwages = dgdp.l1 + dhours.l1 + dwages.l1 + const

	Estimate	Std. Error	t value	Pr(> t)
dgdp.l1	-0.12181	0.07206	-1.690	0.09224 .
dhours.l1	-0.38118	0.24798	-1.537	0.12555
dwages.l1	0.07329	0.06308	1.162	0.24643
const	0.29329	0.10427	2.813	0.00531 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.385 on 245 degrees of freedom
Multiple R-Squared: 0.03214, Adjusted R-squared: 0.02029
F-statistic: 2.712 on 3 and 245 DF, p-value: 0.04561

Covariance matrix of residuals:
dgdp dhours dwages
dgdp 1.5526 0.11983 0.13817
dhours 0.1198 0.11856 -0.02421
dwages 0.1382 -0.02421 1.91796

Correlation matrix of residuals:
dgdp dhours dwages
dgdp 1.00000 0.27930 0.08007
dhours 0.27930 1.00000 -0.05078
dwages 0.08007 -0.05078 1.00000

VAR(1) estimates (SEs in parentheses)

Stars: *** p < 0.01, ** p < 0.05, * p < 0.10

$$\widehat{\text{dgdp}}_t = \frac{0.74515^{***}}{(0.09381)} + \frac{-0.02591}{(0.06484)} \text{dgdp}_{t-1} + \frac{0.53520^*}{(0.22311)} \text{dhours}_{t-1} + \frac{0.11928^*}{(0.05675)} \text{dwages}_{t-1},$$

$$\widehat{\text{dhours}_t} = \frac{-0.037795}{(0.025924)} + \frac{0.008555}{(0.017917)} \text{dgdp}_{t-1} + \frac{-0.243314^{***}}{(0.061655)} \text{dhours}_{t-1} + \frac{0.069679^{***}}{(0.015683)} \text{dwages}_{t-1},$$

$$\widehat{\text{dwages}}_t = \frac{0.29329^{**}}{(0.10427)} + \frac{-0.12181}{(0.07206)} \text{dgdp}_{t-1} + \frac{-0.38118}{(0.24798)} \text{dhours}_{t-1} + \frac{0.07329}{(0.06308)} \text{dwages}_{t-1}.$$

Reduced-form Impulse Response

An **impulse response function** traces out the **dynamic effect over time** of a one-time, unexpected **shock** to one variable on **all variables** in a VAR **relative to its steady state**.

In [37]: `#https://www.rdocumentation.org/packages/vars/versions/1.5-9/topics/irf
library(vars)`

```
irf.VAR1 = irf(  
  model.VAR1, # fitted VAR object  
  impulse = c("dgdp", "dhours", "dwages"), # variable(s) to shock  
  response = c("dgdp", "dhours", "dwages"), # variables whose responses you trace  
  n.ahead = 8, # horizons ahead to compute IRFs  
  ortho = FALSE, # FALSE ⇒ generalized (reduced-form) shocks; ordering irrelevant  
  boot = TRUE, # compute bootstrap confidence bands  
  runs = 1000, # number of bootstrap replications (used if boot=TRUE)  
  ci = 0.95, # confidence Level for the bands  
  cumulative = FALSE # FALSE: non-cumulative IRFs; TRUE: cumulated responses  
)  
  
# irf.VAR1$irf # All responses  
  
# irf.VAR1$irf$dgdp # Responses to a shock to dgdp  
# irf.VAR1$irf$dhours # Responses to a shock to dhours  
# irf.VAR1$irf$dwages # Responses to a shock to dwages  
  
# irf.VAR1$Lower # All Lower confidence bands  
# irf.VAR1$Upper # All upper confidence bands
```

In [38]: `# Responses to a shock to dgdp
irf.VAR1irfdgdp`

A matrix: 9 × 3 of type dbl

dgdp	dhours	dwages
1.000000e+00	0.000000e+00	0.000000e+00
-2.591497e-02	8.555295e-03	-1.218129e-01
-9.278925e-03	-1.079117e-02	-9.031774e-03
-6.612220e-03	1.916927e-03	4.581741e-03
1.743781e-03	-2.037315e-04	4.105480e-04
-1.052584e-04	9.309592e-05	-1.046683e-04
4.006817e-05	-3.084525e-05	-3.033538e-05
-2.116495e-05	5.734118e-06	4.653530e-06
4.172431e-06	-1.252007e-06	7.334849e-07

Layout of the table:

- **Rows = horizons.**
 - **Row 1 (t = 0):** the **shock vector** you injected into the system.
 - **Row 2 (t = 1):** the **first-period** responses.
 - **Row 3 (t = 2):** the **second-period** responses.
 - ...and so on (t = 3, t = 4, ...).
- **Columns = variables.** In your case: **Col 1 = dgdp, Col 2 = dhours, Col 3 = dwages**. Each column shows how that variable responds over time to the **same initial shock**.

What each number means:

- Values are **deviations from the baseline/steady state** (intercepts are excluded in IRFs).
- **Sign:** Positive → the variable rises relative to baseline. Negative → it falls relative to baseline.
- **Magnitude:** measured in the **units of the variable** (e.g., growth rates, differences, logs).
 - If variables are in **log-differences**, entries are **approx. % changes**.
 - If variables are levels/differences, they are in those units.

Special notes for reduced-form IRFs (what you computed):

- The initial shock in Row 1 is a **1-unit shock to the chosen residual** (not 1 s.d.).
- **t = 1 (Row 2)** equals the appropriate **column of the VAR lag matrix** Φ_1 when you shock that variable. (That's why your t = 1 numbers matched the coefficients.)
- Σ_u (residual covariance) is ignored here, so **no contemporaneous spillovers** are shown at $t = 0$ beyond the shocked residual.

Dynamic behavior to watch for:

- **Decay to 0** → the effect dies out (stable system).
- **Persistence** → slow decay indicates lasting effects.
- **Overshooting / sign flips** → responses switch sign across horizons.
- **Cross-effects** → look across columns to see how one variable's shock affects the others.

Manual computations:

$t = 1$

$$\begin{aligned} dgdp^{(1)} &= (-0.02591) \cdot dgdp^{(0)} + (0.53520) \cdot dhours^{(0)} + (0.11928) \cdot dwages^{(0)} \\ &= (-0.02591) \cdot 1 + 0 + 0 \\ &= -0.02591, \end{aligned}$$

$$\begin{aligned} dhours^{(1)} &= (0.008555) \cdot dgdp^{(0)} + (-0.243314) \cdot dhours^{(0)} + (0.069679) \cdot dwages^{(0)} \\ &= (0.008555) \cdot 1 + 0 + 0 \\ &= 0.008555, \end{aligned}$$

$$\begin{aligned} dwages^{(1)} &= (-0.12181) \cdot dgdp^{(0)} + (-0.38118) \cdot dhours^{(0)} + (0.07329) \cdot dwages^{(0)} \\ &= (-0.12181) \cdot 1 + 0 + 0 \\ &= -0.12181. \end{aligned}$$

$t = 2$

$$\begin{aligned} dgdp^{(2)} &= (-0.02591) \cdot dgdp^{(1)} + (0.53520) \cdot dhours^{(1)} + (0.11928) \cdot dwages^{(1)} \\ &= (-0.02591)(-0.02591) + (0.53520)(0.008555) + (0.11928)(-0.12181) \\ &\approx -0.00928, \end{aligned}$$

$$\begin{aligned} dhours^{(2)} &= (0.008555) \cdot dgdp^{(1)} + (-0.243314) \cdot dhours^{(1)} + (0.069679) \cdot dwages^{(1)} \\ &= (0.008555)(-0.02591) + (-0.243314)(0.008555) + (0.069679)(-0.12181) \\ &\approx -0.01079, \end{aligned}$$

$$\begin{aligned} dwages^{(2)} &= (-0.12181) \cdot dgdp^{(1)} + (-0.38118) \cdot dhours^{(1)} + (0.07329) \cdot dwages^{(1)} \\ &= (-0.12181)(-0.02591) + (-0.38118)(0.008555) + (0.07329)(-0.12181) \\ &\approx -0.00903. \end{aligned}$$

Understanding Reduced-Form Impulse Response Functions (IRFs) in R

Shock size

→ The shock applied to the system is **1 unit**, not **1 standard deviation**.

Residual covariance matrix

→ It is **ignored** in the reduced-form IRF.

→ This means shocks are **treated as uncorrelated**, even if the residuals from the VAR are actually correlated.

Intercept term

→ The intercept (constant) is **not included** in the IRF recursion.

→ IRFs describe **deviations from steady state**, not absolute level shifts.

What this means

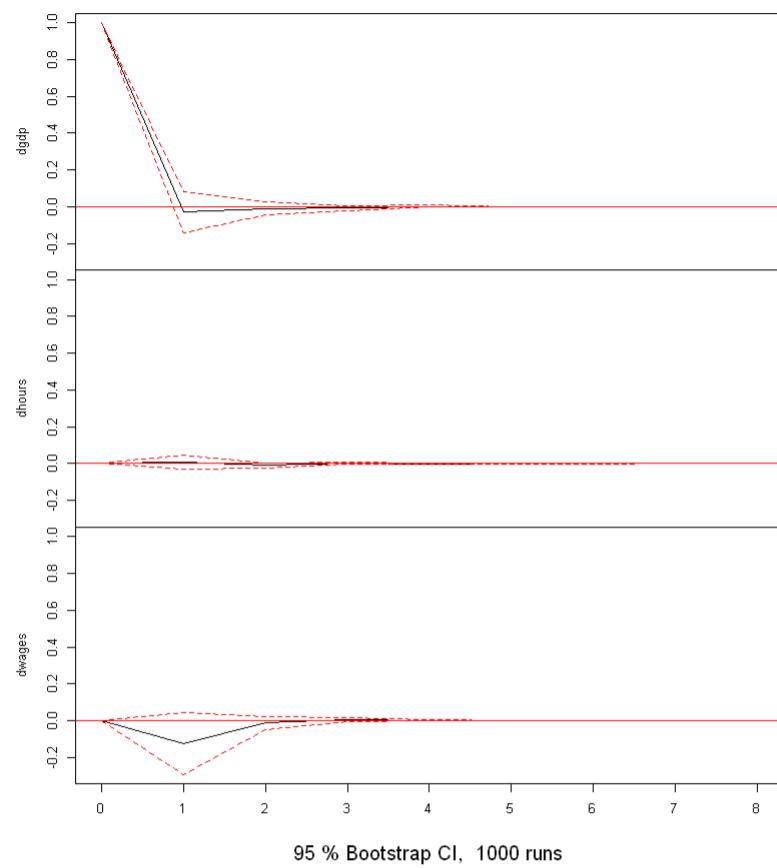
The **reduced-form IRF** does not account for correlated shocks. It represents the response to a **shock in the error term** of variable i , assuming **all other error terms are zero** — not realistic if residuals are correlated.

If the covariance matrix of residuals, Σ_u , is **not diagonal** (i.e., error terms are correlated), then the true response of the system to a shock in, say, GDP growth, would **propagate to other variables instantaneously**.

However, the **reduced-form IRF** you compute with `orth = FALSE` in R does **not** reflect this instantaneous correlation — it's based solely on the **autoregressive propagation** of variables through the lag matrices.

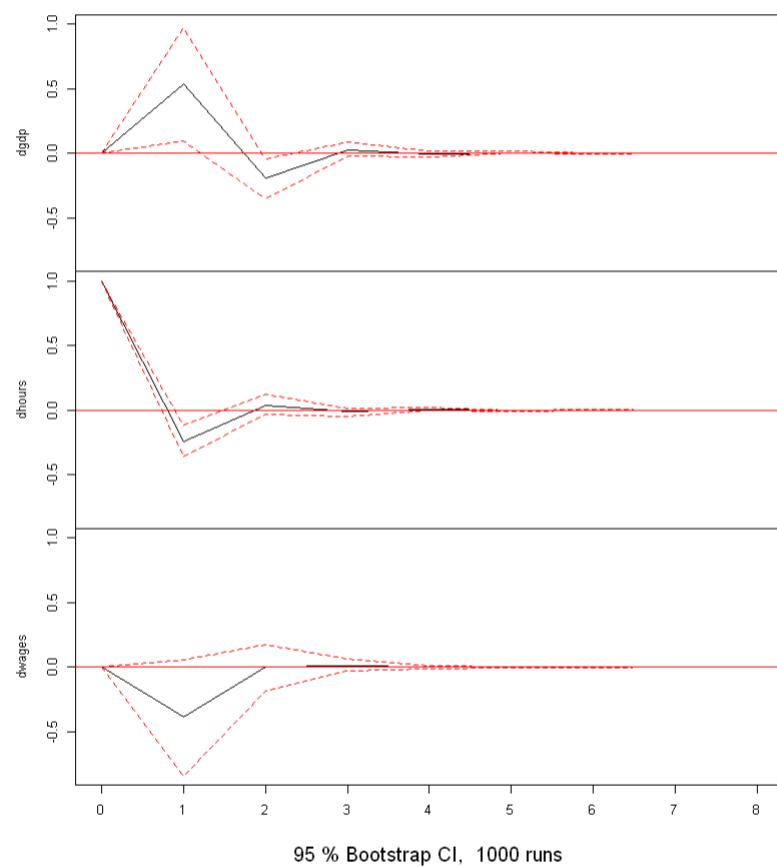
```
In [39]: # Visualize impulse response functions
plot(irf.VAR1)
```

Impulse Response from dgdp



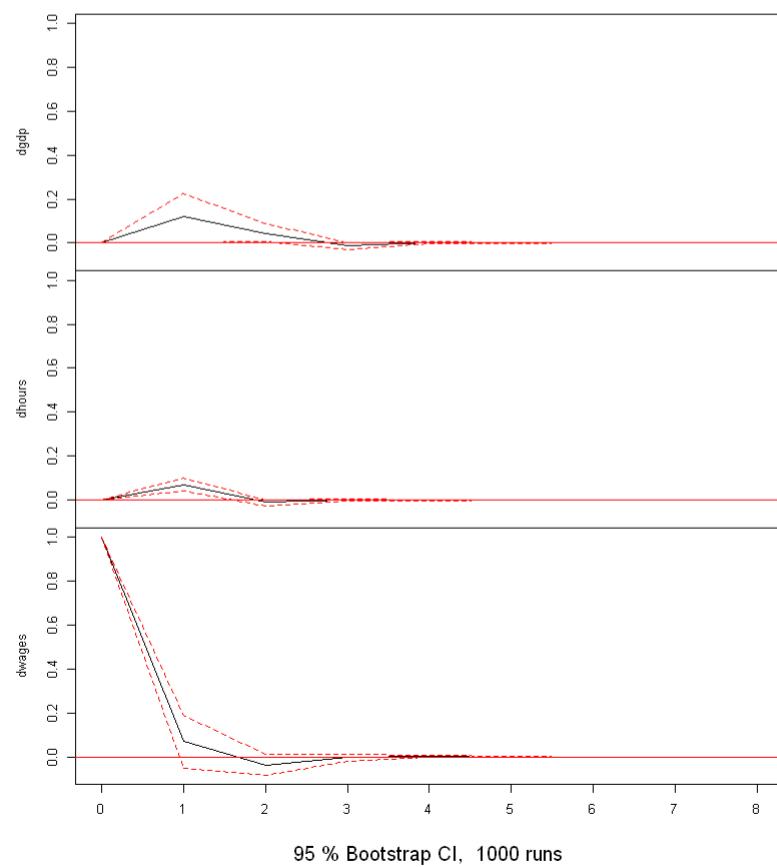
95 % Bootstrap CI, 1000 runs

Impulse Response from dhours



95 % Bootstrap CI, 1000 runs

Impulse Response from dwages



95 % Bootstrap CI, 1000 runs

Estimation of a Structural-form VAR

```
In [40]: # Lower-triangular A-pattern for recursive (Cholesky) SVAR
Amat_pat = matrix(NA, 3, 3) # NA = to be estimated
Amat_pat[1, 2] = 0          # a12 = 0: var2 → var1 (no contemporaneous effect)
Amat_pat[1, 3] = 0          # a13 = 0: var3 → var1 (no contemporaneous effect)
Amat_pat[2, 3] = 0          # a23 = 0: var3 → var2 (no contemporaneous effect)
Amat_pat
```

A matrix: 3 × 3
of type dbl

```
NA 0 0
NA NA 0
NA NA NA
```

```
In [41]: # https://www.rdocumentation.org/packages/vars/versions/1.5-9/topics/SVAR
library(vars)
model.SVAR1 = SVAR(model.VAR1, Amat = Amat_pat, estmethod = "scoring", lrtest = FALSE)
names(model.SVAR1)
summary(model.SVAR1)
```

'A' · 'Ase' · 'B' · 'Bse' · 'LRIM' · 'Sigma.U' · 'LR' · 'opt' · 'start' · 'type' · 'var' · 'iter' · 'call'

SVAR Estimation Results:
=====

Call:
SVAR(x = model.VAR1, estmethod = "scoring", Amat = Amat_pat,
 lrtest = FALSE)

Type: A-model

Sample size: 249
Log Likelihood: -918.683
Method: scoring
Number of iterations: 14

Estimated A matrix:

```
dgdp dhours dwages
dgdp  0.80255 0.0000 0.0000
dhours -0.23344 3.0246 0.0000
dwages -0.08255 0.2318 0.7265
```

Estimated standard errors for A matrix:

```
dgdp dhours dwages
dgdp  0.03596 0.0000 0.00000
dhours 0.05192 0.1355 0.00000
dwages 0.05310 0.1920 0.03256
```

Estimated B matrix:

```
dgdp dhours dwages
dgdp    1     0     0
dhours   0     1     0
dwages   0     0     1
```

Covariance matrix of reduced form residuals (*100):

```
dgdp dhours dwages
dgdp  155.26 11.983 13.817
dhours 11.98 11.856 -2.421
dwages 13.82 -2.421 191.796
```

Structural-form Impulse Response

An **impulse response function (IRF)** traces out the **dynamic effect over time** of a one-time, unexpected **structural shock** to one equation on **all variables** in a VAR, relative to their steady states.

In other words, it shows how a **structural disturbance** in one variable **propagates through the system**, affecting both the shocked variable itself and the others over subsequent periods.

```
In [42]: #https://www.rdocumentation.org/packages/vars/versions/1.5-9/topics/irf
# irf(x, impulse = NULL, response = NULL, n.ahead = 10,
# ortho = TRUE, cumulative = FALSE, boot = TRUE, ci = 0.95, runs = 100, seed = NULL, ...)
library(vars)
irf.SVAR1 = irf(model.SVAR1, impulse = "dgdp",
                response = c("dgdp", "dhours", "dwages"),
                boot = TRUE, n.ahead = 8, seed=123)
# Responses to a shock to dgdp
irf.SVAR1$irf$dgdp
```

A matrix: 9 × 3 of type dbl

dgdp	dhours	dwages
1.246026e+00	9.616972e-02	1.108905e-01
3.240564e-02	-5.012514e-03	-1.803129e-01
-2.502939e-02	-1.106723e-02	-1.525157e-02
-7.093665e-03	1.415956e-03	6.149742e-03
1.675162e-03	2.330011e-05	7.750702e-04
6.150527e-05	6.266864e-05	-1.561343e-04
1.332324e-05	-2.560128e-05	-4.282294e-05
-1.915476e-05	3.359252e-06	4.997321e-06
2.890318e-06	-6.330166e-07	1.419063e-06

In [43]: # Manual computation of contemporaneous responses
solve(model.SVAR1\$A)

A matrix: 3 × 3 of type dbl

	dgdp	dhours	dwages
dgdp	1.24602599	0.0000000	0.0000000
dhours	0.09616972	0.3306247	0.0000000
dwages	0.11089049	-0.1054888	1.376421

Interpretation of a Structural Shock

A **Structural VAR (SVAR)** goes beyond a standard reduced-form VAR by explicitly modeling the **economic (structural) shocks** that drive the system.

From Reduced Form to Structure

A reduced-form VAR looks like this:

$$y_t = \sum_{i=1}^p \Phi_i y_{t-i} + u_t, \quad u_t \sim (0, \Sigma_u)$$

- u_t are **residuals**, but they are **correlated across equations**.
- Because they move together, we cannot interpret a single element of u_t as a separate "shock" — it may contain mixtures of multiple underlying forces.

Structural Representation

To uncover the **independent economic shocks**, we impose a structure:

$$Ay_t = B\varepsilon_t, \quad \varepsilon_t \sim (0, I)$$

- A describes **how variables affect each other contemporaneously**.
- B maps the **structural shocks** ε_t into the system.
- The shocks ε_t are **independent** and **normalized to have unit variance**.

Why SVAR Has Structural Shocks

The SVAR explicitly enforces:

$$\Sigma_\varepsilon = I,$$

meaning each shock is **standardized** and **independent** of the others.

These are called **structural shocks** because they correspond to **economically meaningful, exogenous forces** — for example:

- a monetary policy shock,
- a technology shock,
- a demand shock.

Contemporaneous Effects via A

If A is not diagonal, variables can affect one another **within the same period**.

For example, a shock to GDP can **instantaneously influence hours and wages** through the nonzero off-diagonal elements of A^{-1} .

Contemporaneous Effects via B

If B is not diagonal, **one structural shock** can immediately affect **multiple variables**. That means the shocks themselves are independent, but their effects are **felt across several equations** at once.

For instance:

- a **technology shock** may affect **GDP**, **hours**, and **wages** simultaneously,
- because all three variables have nonzero loadings on that shock in the corresponding column of B .

In contrast, the reduced-form shocks u_t are **statistical mixtures** of these structural shocks.

Interpretation Caveats

In a Structural VAR, the shocks ε_t are **mathematically independent** — but giving them **economic meaning** (e.g., calling one a *technology shock* or a *demand shock*) is often **nontrivial and assumption-driven**.

Although SVARs recover **independent structural shocks**, labeling them as specific economic forces requires **strong identifying assumptions** and **economic reasoning** — it cannot be inferred from the data alone.

Connection to Reduced-Form IRFs in R

The residuals u_t are **correlated across equations**, but R's reduced-form IRF (with `ortho = FALSE`) **shocks only one equation at a time**. That is, R sets one element of u_t to 1 and the rest to 0, **without using the covariance structure** Σ_u to transmit the effect through correlated shocks.

As a result:

- The reduced-form IRF **traces a shock through the dynamic lag structure**,
- but it does **not model contemporaneous transmission** via the correlated errors.

Structural Interpretation of Orthogonalized Impulse Responses

When estimating a **VAR model**, we can compute **Impulse Response Functions (IRFs)** to see how each variable reacts over time to a one-time shock to the equation of a particular variable.

If we set `ortho = TRUE` in the `irf()` function, the shocks are **orthogonalized** — that is, transformed into **structural shocks** that are independent. This is achieved through the **Cholesky decomposition** of the residual covariance matrix.

As a result, the impulse responses can be interpreted as the **dynamic reactions of each variable to a structural shock**, rather than to a correlated reduced-form disturbance.

```
In [44]: #https://www.rdocumentation.org/packages/vars/versions/1.5-9/topics/irf
# irf(x, impulse = NULL, response = NULL, n.ahead = 10,
# ortho = TRUE, cumulative = FALSE, boot = TRUE, ci = 0.95, runs = 100, seed = NULL, ...)
library(vars)
irf.VAR1orth = irf(model.VAR1, impulse = c("dgdp", "dhours", "dwages"),
                    ortho = TRUE, response = c("dgdp", "dhours", "dwages"),
                    boot = TRUE, n.ahead = 8, seed=123)
irf.VAR1orth$irf$dgdp
```

A matrix: 9 × 3 of type dbl

dgdp	dhours	dwages
1.246026e+00	9.616972e-02	1.108905e-01
3.240564e-02	-5.012514e-03	-1.803129e-01
-2.502939e-02	-1.106723e-02	-1.525157e-02
-7.093665e-03	1.415956e-03	6.149742e-03
1.675162e-03	2.330011e-05	7.750702e-04
6.150527e-05	6.266864e-05	-1.561343e-04
1.332324e-05	-2.560128e-05	-4.282294e-05
-1.915476e-05	3.359252e-06	4.997321e-06
2.890318e-06	-6.330166e-07	1.419063e-06

The code below is **tailor-made** to make it easier for you to plot **Impulse Response Functions (IRFs)** in a more professional and visually appealing way.

Enjoy!

```
In [45]: library(devtools)
source_url("https://raw.githubusercontent.com/anguyen1210/var-tools/master/R/extract_varirf.R");
library(dplyr)
multiple.SVAR1 = extract_varirf(irf.SVAR1)
multiple.VAR1orth = extract_varirf(irf.VAR1orth)

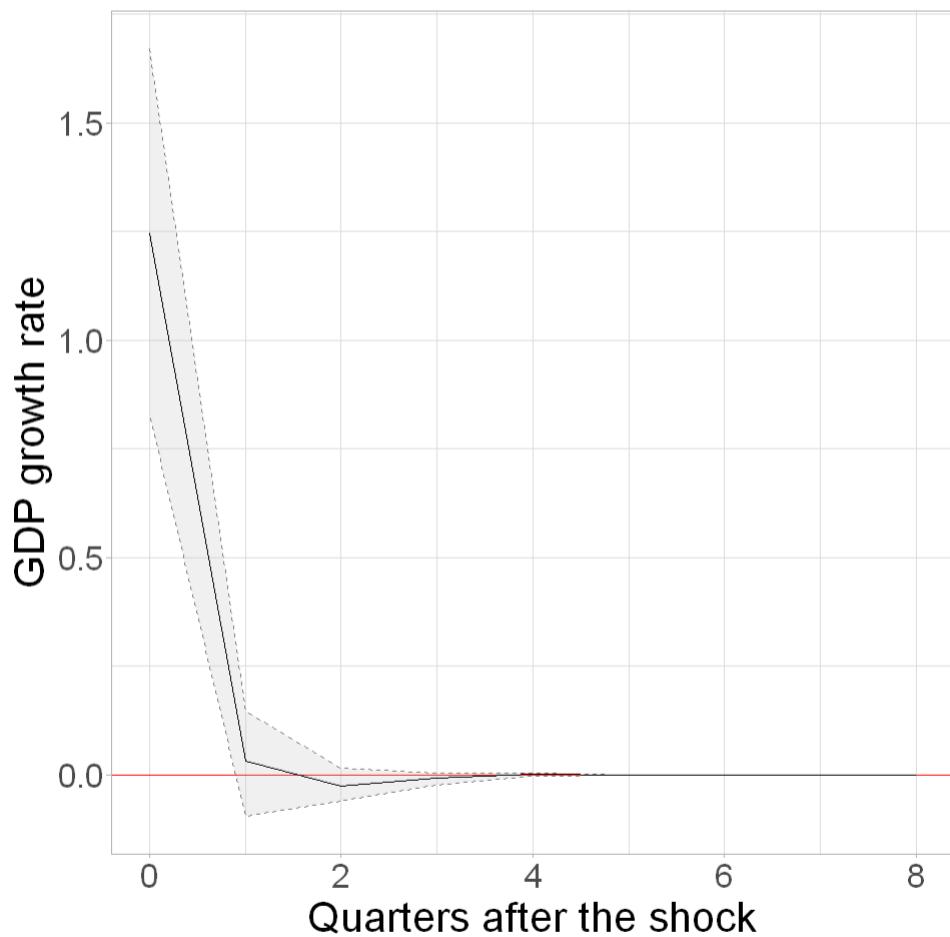
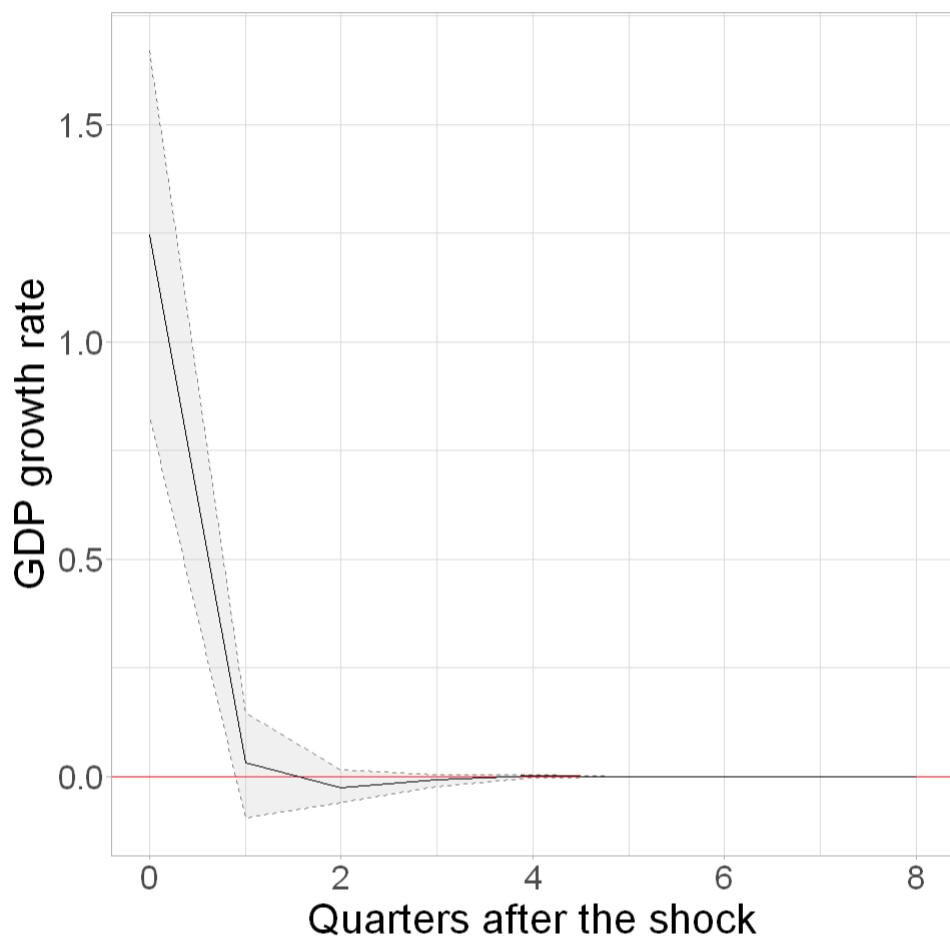
multiple.SVAR1 %>%
  ggplot(aes(x=period, y=irf_dgdp_dgdp, ymin=lower_dgdp_dgdp, ymax=upper_dgdp_dgdp)) +
  geom_hline(yintercept = 0, color="red") +
  geom_ribbon(fill="grey", alpha=.2, color="grey50", linetype="dashed") +
```

```

geom_line() +
theme_light() +
theme(aspect.ratio=1) +
theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) +
theme(text=element_text(size=25)) +
labs(x = "Quarters after the shock") +
labs(y = "GDP growth rate")
multiple.VAR1orth %>%
  ggplot(aes(x=period, y=irf_dgdp_dgdp, ymin=lower_dgdp_dgdp, ymax=upper_dgdp_dgdp)) +
  geom_hline(yintercept = 0, color="red") +
  geom_ribbon(fill="grey", alpha=.2, color="grey50", linetype="dashed") +
  geom_line() +
  theme_light() +
  theme(aspect.ratio=1) +
  theme(plot.margin = ggplot2::margin(0.2, 0.2, 0.2, 0.2, "cm")) +
  theme(text=element_text(size=25)) +
  labs(x = "Quarters after the shock") +
  labs(y = "GDP growth rate")

```

SHA-1 hash of file is "cb23ce6a462d73b67920ae56dc83c828d18fe7e4"



In []: