University of Duisburg-Essen
Faculty of Business Administration and Economics
Chair of Econometrics

Winter Term 2019/2020

Dr. Yannick Hoga Thilo Reinschlüssel

Multivariate Time Series Analysis

Solution Exercise Sheet 2

1 Exercise 1: Moments and Simulation of a VAR(1) Process

Take the model from Example 2.4 on Slide 2-6:

a) Derive a formula to obtain the population cross-covariance matrices for the lags 1 to 10 and compute them using R.

Hint: A glance at the slides might save you some time.

Solution:

$$z_t = \phi_1 z_{t-1} + a_t$$

$$\begin{aligned}
\Gamma_0 &= E\left(z_t - \mu\right)\left(z_t - \mu\right)' \\
&= E\left[\phi_1\left(z_{t-1} - \mu\right)\left(z_{t-1} - \mu\right)' \phi_1'\right] + E\left[a_t a_t'\right] \\
&= \phi_1 \ \Gamma_0 \ \phi_1' + \Sigma_a \\
\Leftrightarrow \operatorname{vec}(\Gamma_0) &= (\phi_1 \otimes \phi_1) \cdot \operatorname{vec}(\Gamma_0) + \operatorname{vec}(\Sigma_a) \\
\Leftrightarrow \operatorname{vec}(\Gamma_0) &= (I_{K^2} - \phi_1 \otimes \phi_1) \cdot \operatorname{vec}(\Sigma_a) \\
\Rrightarrow \Gamma_1 &= \phi_1 \Gamma_0 \\
\Rrightarrow \Gamma_l &= \phi_{l-1} \Gamma_{l-1} = \phi_1^l \Gamma_0
\end{aligned}$$

b) Based on your results, compute the cross-correlation matrices.

Hint: A loop might save you some time.

Solution:

```r
# First define parameters/coefficients:
Phi <- matrix(data = c(0.8, -0.3, 0.4, 0.6),
              byrow = FALSE, nrow = 2) # VAR coefficients

Sigma_a <- matrix(data = c(2, 0.5, 0.5, 1.0),
                  byrow = FALSE, nrow = 2) # innovations' covariances

Phi_kron <- kronecker(X = Phi, Y = Phi) # kronecker product
 # alternative command: Phi %x% Phi

Ident <- diag(ncol(Phi_kron))
# identity matrix with the same dimensions as Phi_kron

Gamma0.vec <- solve(Ident - Phi_kron) %*% c(Sigma_a)
# c() works like the "vec" operator

Gamma0.mat <- matrix(data = Gamma0.vec, nrow = 2, byrow = FALSE)

mat_names <- c('z_1', 'z_2') # setting names of the series
colnames(Gamma0.mat) <- mat_names # passing column names
rownames(Gamma0.mat) <- mat_names # assig row names
```

The Γ_0 matrix is then:

```
##            z_1   z_2
## z_1  5.933333 -0.45
## z_2 -0.450000  2.65
```

To derive the Γ_1 matrix, we can simply apply the following formula.

$$\Gamma_1 = \phi_1 \Gamma_0$$

```r
Gamma1.mat <- Phi %*% Gamma0.mat
```

Γ_1 is then:

```
##            z_1   z_2
```

```
## z_1  4.566667 0.700
## z_2 -2.050000 1.725
```

To derive all l^{th} lagged covariance matrices we can use the more general equation and program a loop over all desired lags.

$$\Rrightarrow \Gamma_l = \phi_{l-1}\Gamma_{l-1} = \phi_1^l \Gamma_0$$

To derive the correlation matrices ρ_l we divide the covariances by the standard deviations.

$$\rho_l = (\rho_{ij}(l))_{i,j=1}^{K} = D^{-1}\Gamma_l D^{-1}$$

$$= \begin{pmatrix} \frac{\mathrm{Cov}(x_t, x_{t-l})}{\sqrt{\mathrm{Var}(x_t)\cdot \mathrm{Var}(x_{t-l})}} & \frac{\mathrm{Cov}(x_t, y_{t-l})}{\sqrt{\mathrm{Var}(x_t)\cdot \mathrm{Var}(y_{t-l})}} \\ \frac{\mathrm{Cov}(y_t, x_{t-l})}{\sqrt{\mathrm{Var}(y_t)\cdot \mathrm{Var}(x_{t-l})}} & \frac{\mathrm{Cov}(y_t, y_{t-l})}{\sqrt{\mathrm{Var}(y_t)\cdot \mathrm{Var}(y_{t-l})}} \end{pmatrix}$$

- where $D = \mathrm{diag}\left\{\sqrt{\Gamma_{11}(0)}, \ldots, \sqrt{\Gamma_{kk}(0)}\right\}$

```
# compute further (lagged) cross-covariance-functions
# preparing the variables to store the matrices into
#covariance
ccovf.list <- list()
#correlation
ccorf.list <- list()

# obtaining standard deviations for the single variables (parts of z)
# and putting them in a diagonal matrix
D.inv <- solve(sqrt(Gamma0.mat * diag(ncol(Phi))))

for (i in 1:10){

    # covariance
    ccovf <- Phi%^%i %*% Gamma0.mat
    rownames(ccovf) <- mat_names
```

```r
  ccovf.list[[i]] <- ccovf

    # correlation
    ccorf <-  D.inv %*% ccovf.list[[i]] %*% D.inv
    rownames(ccorf) <- mat_names
  ccorf.list[[i]] <- ccorf
}
ccovf.list # cross lagged covariances
```

```
## [[1]]
##            z_1   z_2
## z_1  4.566667 0.700
## z_2 -2.050000 1.725
##
## [[2]]
##            z_1   z_2
## z_1  2.833333 1.250
## z_2 -2.600000 0.825
##
## [[3]]
##            z_1  z_2
## z_1  1.226667 1.33
## z_2 -2.410000 0.12
##
## [[4]]
##              z_1    z_2
## z_1  0.01733333  1.112
## z_2 -1.81400000 -0.327
##
## [[5]]
##             z_1     z_2
## z_1 -0.7117333  0.7588
## z_2 -1.0936000 -0.5298
##
## [[6]]
##            z_1      z_2
## z_1 -1.006827  0.39512
## z_2 -0.442640 -0.54552
##
## [[7]]
```

```
##              z_1       z_2
## z_1 -0.9825173  0.097888
## z_2  0.0364640 -0.445848
##
## [[8]]
##              z_1        z_2
## z_1 -0.7714283 -0.1000288
## z_2  0.3166336 -0.2968752
##
## [[9]]
##              z_1        z_2
## z_1 -0.4904892 -0.1987731
## z_2  0.4214086 -0.1481165
##
## [[10]]
##              z_1         z_2
## z_1 -0.2238279 -0.21826509
## z_2  0.3999919 -0.02923795
```

```
ccorf.list # cross lagged correlations
```

```
## [[1]]
##              z_1       z_2
## z_1  0.7696629 0.1765329
## z_2 -0.5169893 0.6509434
##
## [[2]]
##              z_1       z_2
## z_1  0.4775281 0.3152374
## z_2 -0.6556937 0.3113208
##
## [[3]]
##              z_1        z_2
## z_1  0.2067416 0.33541257
## z_2 -0.6077777 0.04528302
##
## [[4]]
##                z_1        z_2
## z_1  0.002921348  0.2804352
## z_2 -0.457472478 -0.1233962
```

```
##
## [[5]]
##              z_1        z_2
## z_1 -0.1199551  0.1913617
## z_2 -0.2757949 -0.1999245
##
## [[6]]
##              z_1         z_2
## z_1 -0.1696899  0.09964527
## z_2 -0.1116293 -0.20585660
##
## [[7]]
##                z_1         z_2
## z_1 -0.165592809  0.02468636
## z_2  0.009195853 -0.16824453
##
## [[8]]
##              z_1         z_2
## z_1 -0.1300160 -0.02522625
## z_2  0.0798518 -0.11202838
##
## [[9]]
##               z_1         z_2
## z_1 -0.08266671 -0.05012857
## z_2  0.10627500 -0.05589301
##
## [[10]]
##              z_1         z_2
## z_1 -0.0377238 -0.05504425
## z_2  0.1008739 -0.01103319
```

c) Draw a corresponding innovation sequence a_t for 300 periods from a (multivariate) Gaussian distribution and simulate the given VAR(1) process without any further built-in functions.

Hint: 'mvrnorm' and 'for' are still allowed

Solution:

- Steps:

1. Set T (in code N)
2. Draw $\{a_1, \ldots, a_T\} \sim [\mu, \Sigma_a]$
3. Set $z_0 = \mathbb{E}(z_t)$
4. $z_1 = \phi_1 z_0 + a_1$
5. Repeat Step 4 $(T-1)$ times
6. (Discard the first few observations to minimize effects of z_0 on the results)

   $\Rightarrow$ *Note* that it is generally advised to discard the first few observations to eliminate the influence of the arbitrary starting point! In our case we skipped this step to keep things short and clear.

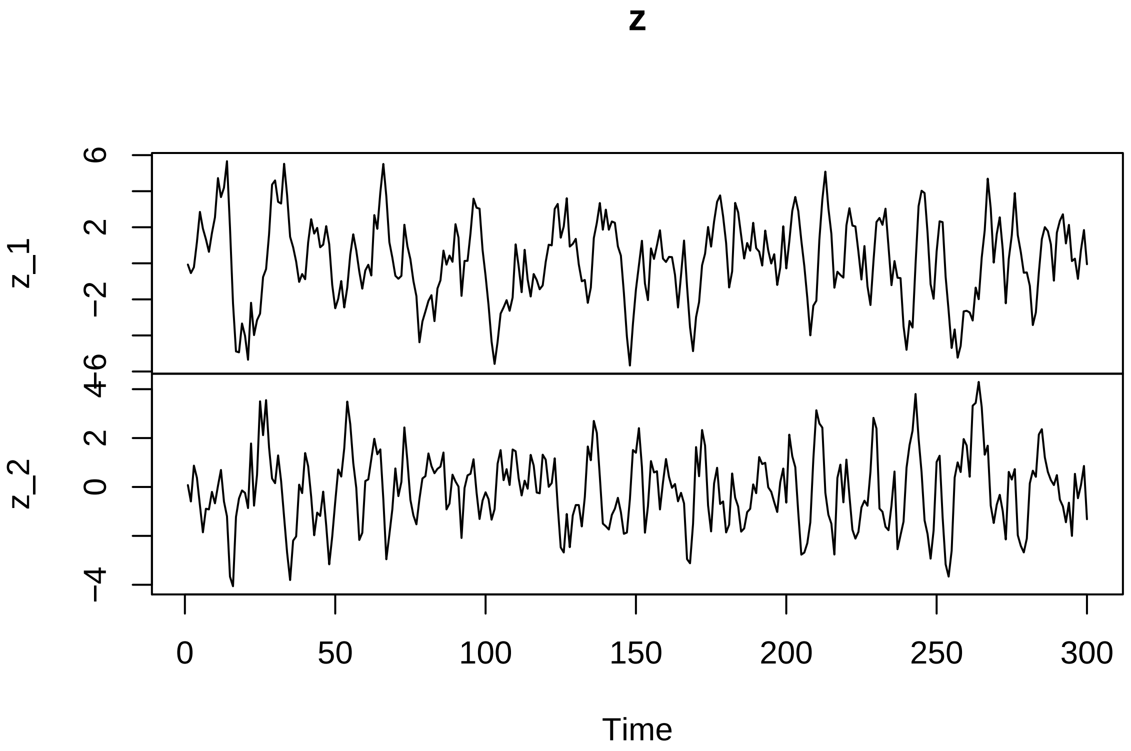
```r
# i) Set sample size and further coefficients
N <- 300
# ii) The Basis: Innovations
set.seed(2^9-1) # for replication: the 'random' numbers drawn
# up from this point will always be the same (given you use
# the same command to draw them...)
a <- mvrnorm(n = N, mu = c(0,0), Sigma = Sigma_a)
# iii) Writing a function to generate 'z' realisations from 'a' using Phi
var1gen <- function(coef.mat, z.lag, innovation){
  z.current <- coef.mat %*% z.lag + innovation
  # z_(t) = phi * z_(t-1) + a_(t)
  return(z.current)
}
# iv) Prepare variable for 'z'
z <- ts(data = matrix(data = NA, nrow = (N), ncol = ncol(Phi))
        ,start = 1, names = mat_names)
# v) Set starting values for z (at mean + innovation)
z[1,] <- c(0,0) + a[1,]
# vi) Generate z repeatedly and store it
for (i in 2:(N)){
  z[i,] <- var1gen(coef.mat = Phi, z.lag = z[(i-1),], innovation = a[i,])
}
```

d) Plot the multivariate time series you have just created. Does it look stationary?

Solution:

The ordinary organic `plot` command works because z is already a *ts*-class object (among other classes) and not a data frame.

```r
plot(z)
```



At least it looks stable hence we cannot rule out stationarity.

e) Estimate the sample cross-covariance and cross-correlation matrices. Compare these with the population moment matrices from task a)

Solution:

$$\widehat{\Gamma_0} = \tilde{z}_T' \tilde{z}_{T-1} \cdot (T-1)^{-1}$$

$$\tilde{z}_T' = z_T - \widehat{\mu}_z$$

$$Z \text{ is a } T \times 2 \text{ matrix}$$

$$z_t \text{ is a } 2 \times 1 \text{ vector}$$

$$z_t := \begin{pmatrix} x_t \\ y_t \end{pmatrix} \leftarrow \text{variables}$$

$$Z_t := \begin{pmatrix} X_t & Y_t \\ X_{t-1} & Y_{t-1} \\ X_{t-2} & Y_{t-2} \\ \vdots & \vdots \\ X_{t-t} & Y_{t-t} \end{pmatrix} \leftarrow \text{sample data}$$

$$\widehat{\mathrm{Cov}(x_t, y_{t-1})} = \frac{1}{(T-1)} \sum_{t=1}^{T} \left(\tilde{X}_t \cdot \tilde{Y}_{t-1} \right)$$
$$\Rightarrow \text{is part of:} \frac{1}{T-1} \cdot \tilde{Z}_t' \tilde{Z}_{t-1} = \widehat{\Gamma_1}$$

To estimate moments from simulated data one can simply use the command `cov()` respectively `cor()` , but we do it by "hand" to get a thorough understanding and see the relation to the Yule-Walker equation.

First we need to demean the multivariate time series. Therefore, we calculate the column means.

```
mu <- colMeans(z)
```

To demean the series we now have two coding options to proceed with, one would be to iterate over the rows and subtracte the mean vector every time.

```
z.demeaned <- t(apply(X = z, MARGIN = c(1), FUN = function(x) x - mu))
```

The other option would be to subtracte the column means from each column individually.

```
z.demeaned <- cbind(z[,1]-mu[1], z[,2]-mu[2])
```

Now we are able to compute $\widehat{\Gamma_0}$.

```
Gamma0.hat <- t(z.demeaned) %*% z.demeaned / (N-1)
```

Since we already demeaned the series we need to correct the degrees of freedom $(N-1)$. Please also note that now the first entry is transposed! (*z.demeaned* is a data matrix and not a random vector anymore!)

To compute $\widehat{\rho}_0$ we just need to standardize $\widehat{\Gamma}_0$.

```
standardisation <- matrix(data = c(Gamma0.hat[1,1],
                                   sqrt( Gamma0.hat[1,1] * Gamma0.hat[2,2]),
                                   sqrt(Gamma0.hat[2,2] * Gamma0.hat[1,1]),
                                   Gamma0.hat[2,2]),
                          nrow = 2, byrow = FALSE)
Rho0.hat <- Gamma0.hat / standardisation
```

This is just another way to program it, maybe it makes it more visible what is inside $D.inv \ \%*\% \ D.inv$ (from the lecture slides).

Then $\widehat{\rho_0}$ is:

```
##                z_1         z_2
## z_1  1.00000000 -0.04910251
## z_2 -0.04910251  1.00000000
```

Since we simulated the trajectory with the "true" values we can compare the estimations with these values. The estimated mean values for the two series are slightly negative $(0.1762664, -0.1272957)'$ so the differences are also slightly negative since we simulated the time series without a mean $\mu_{1,2} = 0$.

For the covariance Γ_0 we calculated the analytical solution in part *b* of this exercise. The differences $(\widehat{\Gamma}_0 - \Gamma_0)$ are:

```
##              z_1        z_2
## z_1 -0.5964760  0.2778575
## z_2  0.2778575 -0.3470556
```

The same applies for the comparison of the correlations ρ_0 and the estimated correlations $\widehat{\rho_0}$. The differences $(\widehat{\rho}_0 - \rho_0)$ are:

```
##              z_1          z_2
## z_1 0.00000000 6.438295e-02
## z_2 0.06438295 2.220446e-16
```

2 Exercise 2: Checking VAR(1) Stationarity

Recall the conditions to check if a VAR(1) process is stationary. Now assume the VAR(1) model $z_t = \phi_1 z_{t-1} + a_t$ with a_t as a sequence of i.i.d. innovations:

a) Do you need to make further assumptions on the cross-correlations of a_t to ensure stationarity?

Solution:

$$Z_t = \phi_1 Z_{t-1} + a_t$$
$$a_t \overset{i.i.d.}{\sim} [\mu_a, \Sigma_a]$$
$$i.i.d.: \mathrm{Cov}(a_t, a_{t-1}) = 0$$

Generally not, since i.i.d. errors induce no dynamic structure. Still, finite 1^{st} and 2^{nd} moments are required for weak stationarity! (Gaussian innovations fulfill that condition, of course).

b) Which of the following processes are stationary? $\phi_1 = \ldots$

i) $\begin{pmatrix} 0.2 & 0.3 \\ -0.6 & 1.1 \end{pmatrix}$

ii) $\begin{pmatrix} 0.5 & 0.3 \\ 0 & -0.3 \end{pmatrix}$

iii) $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

iv) $\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$

v) $\begin{pmatrix} 1 & -0.5 \\ -0.5 & 0 \end{pmatrix}$

Solution:

$$\begin{aligned} Z_t &= \phi_1 z_{t-1} + a_t \\ &= \phi_1 \cdot (\phi_1 z_{t-2} + a_{t-1}) + a_t \\ &= \underbrace{\phi_1^p \ z_{t-p}}_{\text{stable ?}} + \underbrace{\sum_{i=0}^{p} \phi_1^i \ a_{t-1}}_{\text{summable ?}} \end{aligned}$$

$$\begin{aligned} &\lim_{p\to\infty} \phi_1^p \longrightarrow 0_{k\times k} \\ &\Rightarrow \text{eigenvalues !} \\ &\Rightarrow \phi_1 x = \lambda x \Rightarrow \phi_1^p x = \lambda^p x \\ &\Rightarrow \text{solve:} \, (\phi_1 - I_K \lambda) \, x = 0 \\ &\text{for } x \neq 0 : |\phi_1 - I_K \lambda| \overset{!}{=} 0 \\ &\text{stability (for stationarity)} \, |\lambda_1| \, , \ldots , |\lambda_k| < 1 \end{aligned}$$

i) $\begin{pmatrix} 0.2 & 0.3 \\ -0.6 & 1.1 \end{pmatrix}$

$$= \begin{pmatrix} 0.2 & 0.3 \\ -0.6 & 1.1 \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$$

$$
\begin{aligned}
&= \begin{vmatrix} (0.2-\lambda) & 0.3 \\ -0.6 & (1.1-\lambda) \end{vmatrix} \\
&= (0.2-\lambda)(1.1-\lambda) - (-0.6)(0.3) \\
&= \lambda^2 - 1.3\lambda + 0.4 \stackrel{!}{=} 0 \\
pq\text{-formula} \Rightarrow \lambda_{1,2} &= -\left(\frac{-1.3}{2}\right) \pm \sqrt{\left(\frac{-1.3}{2}\right)^2 - 0.4} \\
&= 0.65 \pm 0.15 \\
\lambda_1 &= 0.8 \\
\lambda_2 &= 0.5 \\
&|\lambda_1| < 1, |\lambda_2| < 1, \text{ stationary}
\end{aligned}
$$

ii) $\begin{pmatrix} 0.5 & 0.3 \\ 0 & -0.3 \end{pmatrix}$

$$
\begin{aligned}
&\begin{vmatrix} 0.5-\lambda & 0.3 \\ 0 & -0.3-\lambda \end{vmatrix} \\
=&(0.5-\lambda)(-0.3-\lambda) - 0.3 \cdot 0 \stackrel{!}{=} 0 \\
\Rightarrow&\lambda_1 = 0.5,\ \lambda_2 = 0.3 \Rightarrow \text{stationary}
\end{aligned}
$$

iii) $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

$$
\begin{aligned}
&\begin{vmatrix} 1-\lambda & 0 \\ 0 & 1-\lambda \end{vmatrix} \\
=&(1-\lambda)(1-\lambda) - 0 \cdot 0 \stackrel{!}{=} 0 \\
\Rightarrow&\lambda_1 = 1,\ \lambda_2 = 1 \Rightarrow \text{not stationary}
\end{aligned}
$$

iv) $\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$

$$
\begin{aligned}
&\begin{vmatrix} 1-\lambda & -1 \\ 1 & -1-\lambda \end{vmatrix} \\
=&(1-\lambda)(-1-\lambda) - 0 \cdot 0 \stackrel{!}{=} 0 \\
=&\lambda^2 + \lambda - \lambda - 1 + 1 \\
=&\lambda^2 \stackrel{!}{=} 0 \\
\Rightarrow&\lambda_1 = 0,\ \lambda_2 = 0 \Rightarrow \text{ stationary}
\end{aligned}
$$

v) $\begin{pmatrix} 1 & -0.5 \\ -0.5 & 0 \end{pmatrix}$

$$\begin{aligned}
&\begin{vmatrix} 1-\lambda & -0.5 \\ -0.5 & 0-\lambda \end{vmatrix} \\
=&(1-\lambda)(-\lambda) - 0.5 \cdot 0.5 \overset{!}{=} 0 \\
=&\lambda^2 - \lambda - 0.25 \overset{!}{=} 0 \\
\Rightarrow&\lambda_1 = 0.207,\ \lambda_2 = 1.207 \Rightarrow \text{not stationary}
\end{aligned}$$