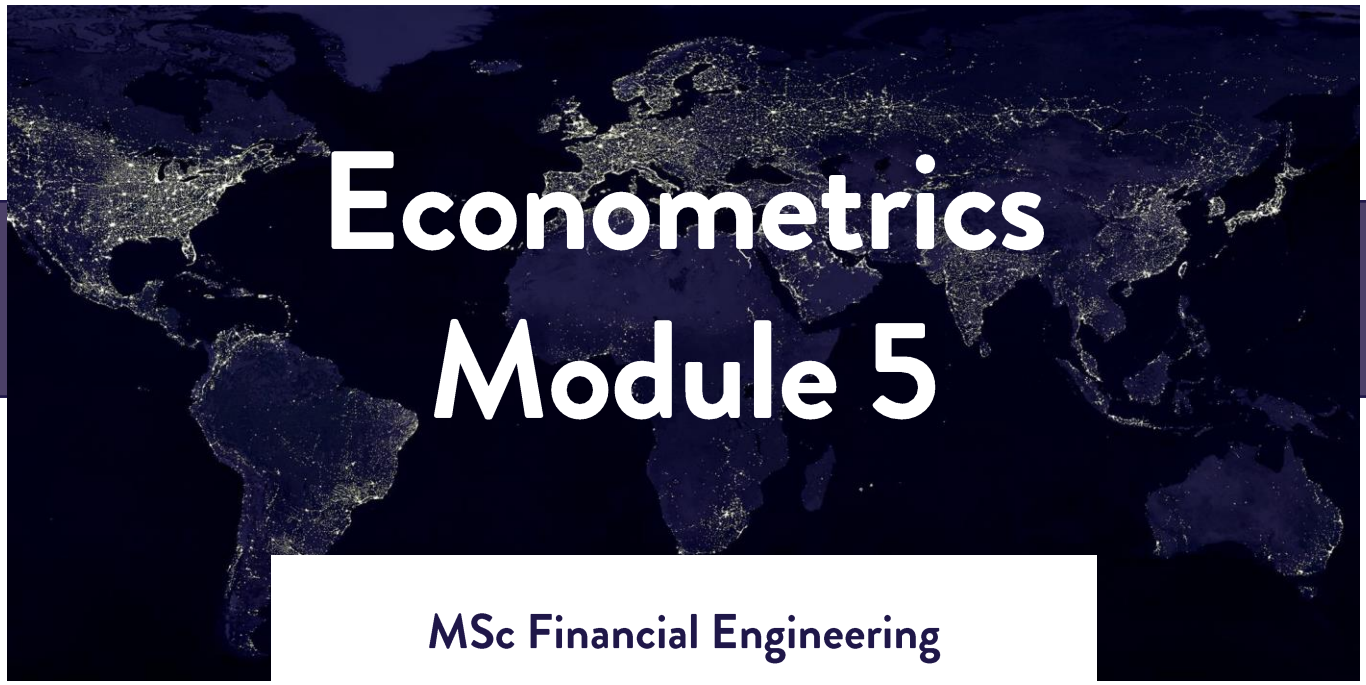
WORLDQUANT UNIVERSITY



Econometrics Module 5

MSc Financial Engineering

Table of Contents

1. Brief 2

2. Course Context 2

2.1 Course-level Learning Outcomes 3

2.2 Module Breakdown 4

3. Module 5: Multivariate Time Series Analysis 5

3.1 Module-level Learning Outcomes 5

3.2 Transcripts and Notes 6

3.2.1 Notes: Introduction 6

3.2.2 Notes: The Stationary Vector Auto-Regressive Model 8

3.2.3 Transcript: A Practical Application to Interest Rates (Part 1) 11

3.2.4 Transcript: A Practical Application to Interest Rates (Part 2) 19

3.2.5 Notes: The Cointegration / Vector Error Correction Model 23

3.2.6 Transcript: VECM Derivations and the Johansen Test 28

3.2.7 Notes: Principal Component Analysis 32

3.2.8 Notes: Multivariate GARCH Analysis 37

1. Brief

This document contains the core content for Module 5 of Econometrics, entitled Multivariate Time Series Analysis. It consists of five sets of notes, three video lecture scripts.

2. Course Context

Econometrics is the second course presented in the WorldQuant University (WQU) Master of Science in Financial Engineering (MScFE) program. In this course, you will apply statistical techniques to the analysis of econometric data. The course starts with an introduction to the R statistical programming languages that you will use to build econometric models, including multiple linear regression models, time series models, and stochastic volatility models. You will learn to develop programs using the R language, solve statistical problems, and understand value distributions in modeling extreme portfolio and basic algorithmic trading strategies. The course concludes with a review on applied econometrics in finance and algorithmic trading.

2.1 Course-level Learning Outcomes

Upon completion of the Econometrics course, you will be able to:

1. Write programs using the R language.
2. Use R packages to solve common statistical problems.
3. Formulate a generalized linear model and fit the model to data.
4. Use graphic techniques to visualize multidimensional data.
5. Apply multivariate statistical techniques (PCA, factor analysis, etc.) to analyze multidimensional data.
6. Fit a time series model to data.
7. Fit discrete-time volatility models.
8. Understand and apply filtering techniques to volatility modeling.
9. Understand the use of extreme value distributions in modeling extreme portfolio returns.
10. Define some common risk measures like VaR and Expected Shortfall.
11. Define and use copulas in risk management.
12. Implement basic algorithmic trading strategies.

2.2 Module Breakdown

The Econometrics course consists of the following one-week modules:

1. Basic Statistics
2. Linear Models
3. Univariate Time Series Models
4. Univariate Volatility Modeling
5. Multivariate Time Series Analysis
6. Introduction to Risk Management
7. Algorithmic Trading

3. Module 5:

Multivariate Time Series Analysis

In this module we will extend the aspects of univariate time series modeling that we covered in Module 3 and 4 to a multivariate context – that is, using more than one time series to describe the time series properties of a multivariate data-generating process.

3.1 Module-level Learning Outcomes

After completing this module, you will be able to:

1. Implement VAR model.
2. Understand cointegration and implement VEC model.
3. Use Principal Components Analysis (PCA) for dimensionality reduction.
4. Understand different versions of multivariate GARCH.

3.2 Transcripts and Notes

3.2.1 Notes: Introduction

In this module we will extend the aspects of univariate time series modeling that we covered in Modules 3 and 4 to a multivariate context – that is, using more than one time series to describe the time series properties of a multivariate data generating process. This is necessary to lay the groundwork for one of the main econometric issues of interest to the financial engineer: building a portfolio of assets to invest in that has the lowest risk with highest return. We will turn to those issues in more detail in Module 6. The reason we don't jump to this immediately is a technical one. As you may recall from previous modules, a well-trained financial engineer must understand essential, formal issues in univariate contexts to develop proper intuition. This is also true for multivariate contexts; there are practical, econometric complications and limitations in the multivariate setting that require attention before we can motivate the main features of optimal portfolio construction and risk management.

For this purpose, we will build the following modeling approaches: the next section will focus on the direct extension of the univariate ARMA modeling approach to multivariate modeling in the stationary time series context. This approach is called Vector Auto-Regressive Modeling (VAR). Section 3 will extend this (in brief) to a situation where each individual time series is non-stationary, but where some linear combination of the individual non-stationary series is stationary, which we will call "co-integration". We will refer to these models as Vector Error Correction Models (VECM).

Section 4 will address the main problem of joint multivariate modeling called the “curse of dimensionality” by examining the standard methods of reducing the number of time series to consider by appropriately reducing the amount of joint processes with Principal Component Analysis or the closely related approach of Factor Analysis. Section 5 will conclude by briefly considering the many series analog of the work we covered in Module 4.

The main practical constraint we will encounter in this module is the “curse of dimensionality”. While it seems sensible to jointly use a large universe of time series processes to find the main features of interest to the financial engineer, it turns out that the practical complications of estimating such processes are quite restrictive. When we add more and more series to jointly model, unless we are willing to make strong assumptions on the joint data generating process, we quickly run out of “degrees of freedom”. In lay terms, the numbers of coefficients we have to estimate (or in near equivalent terms “number of unknowns we need to uncover”) grows at a quadratic rate in the dimensionality (number of simultaneously modeled series) of our model.

This module will make frequent use of matrix algebra, so you should review this if you are not comfortable with the standard notation and results. You can consult the appendices in Tsay (2010).

Chapter 5 of Enders (2014) gives a very accessible treatment of the main theoretical results and practical estimation concerns relating to VAR, VECM and multivariate GARCH models. Hamilton (1992) gives the full theoretical derivations of these models, as well as for Principal Component and Factor Analysis (1992) and the Kalman filter. Chapter 8 of Tsay (2010) provides applications of these methods and some theoretical analysis focused on the financial engineer.

3.2.2 Notes: The Stationary Vector Auto-Regressive Model

The basic structure of the model

If we are willing to assume that the processes we are interested in are stationary, the simplest extension of the AR(p) model of Module 3 is the standard VAR(p) model.

In this model we consider the joint dynamics of a set of n variables as explained by their mutual history (and nothing else). The formal mathematical description of this model is given by the following set of definitions. Let $\mathbf{x}_t = [x_{1t}, x_{2t}, \dots, x_{nt}]'$ be the column vector that represents the observation at moment t of a set of jointly determined variables. As an example, consider all of the returns on the assets listed on the New York Stock Exchange observed at a specific instant in time.

If we are willing to assume that there is some fundamental, linear relationship that relates the returns on each of these assets relative to each other over time (with no other necessary explanatory variables of interest), we could model this relationship with the following vector equation:

$$\mathbf{x}_t = \mathbf{A}_0 + \mathbf{A}_1\mathbf{x}_{t-1} + \cdots + \mathbf{A}_p\mathbf{x}_{t-p} + \boldsymbol{\varepsilon}_t$$

This is called a Vector Auto-Regressive process of order p (VAR(p)), because $\mathbf{x}_t$ is an $[n \times 1]$ vector of the variables of interest. $\mathbf{A}_0$ is an $[n \times 1]$ vector of regression constants, and each of the $\mathbf{A}_i, i > 0$ matrices are the collection of $[n \times n]$ coefficients that relates each of the n variables' current values to each of the n variables' past values at lag i. Thus, we allow a totally general feedback process from the past of any variable up to lag p to each of the other variables' current values. The VAR model is still linear in parameters, and hence can be simply estimated by OLS.

Moreover, we allow a general correlation structure between the variables by allowing a general variance-covariance matrix in the errors/ residuals:

$$E(\boldsymbol{\varepsilon}_t \boldsymbol{\varepsilon}_t') = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_n^2 \end{bmatrix}$$

A valid variance-covariance matrix (variances on the diagonal and covariances on the off-diagonals) will always be positively definite. In the more complicated models we will consider later, such as the multivariate GARCH model, ensuring this feature is a practical complication to estimation.

While this seems pleasantly general, since no matter what variables we wish to include in our regressions, we can capture *all* linear relationships up to lag p with this approach, it comes with a severe downside: for every extra lag we add to the process, we add n^2 parameters to estimate with every lag. This means that, no matter how large our sample, as we add more lags to the process we are estimating, we are exhausting how much information there can possibly be in a finite sample at a quadratic rate.

In short, as we extend the lag length of the model, we are increasing the number of coefficients we need to estimate at a much faster rate than we can gather information on which to base the estimates of those coefficients by adding additional observations to our sample. So we cannot hope to employ the same approach as we did in a univariate context. We have to *ex ante* constrain how many lags can be used to capture the time series process of interest.

This is called the “curse of dimensionality” that will pervade this entire module.

Spurious regression

The VAR model is appropriate for stationary series and yields valid inference. If series are integrated of order 1 or higher, there is a risk: regressing a unit root process on an independent unit root process can lead to “spurious regression”, or, in other words,

significant results and a well-fitting model even if there is no relationship between the variables of interest.

The one case where it *is* valid to regress one unit root process on another is if they are *co-integrated*, or equivalently, share a common stochastic trend. That is the source of the non-stationarity in each series is the same unit root process, even though they may depend differently on this. For the theoretical details on this issue, see Enders (2014).

3.2.3 Transcript: A Practical Application to Interest Rates (Part 1)

Let's consider a brief practical application to interest rate returns. We will do the analysis with the `vars` package in R.

We will consider the joint dynamics of the weekly observations of interest rates on the one-, three- and six-month Eurodollar deposits. As usual, this data is from the Federal Reserve of the USA (Fred.stlouisfed.org, 2018).

Testing the levels of these series reveal that they are all integrated of order 1, so we work with the stationary first differences.

```
# load packages and data

library(readxl)
library(vars)
library(timeSeries)

eurodollarrates <- read_excel("C:/<your path>/eurodollarrates.xls",
                              sheet = "Weekly,_Ending_Friday")

# find changes in interest rates by differencing:

d_one_month_rate <- diff(eurodollarrates$WED1,trim=TRUE)
d_three_month_rate <- diff(eurodollarrates$WED3,trim=TRUE)
d_six_month_rate <- diff(eurodollarrates$WED6,trim=TRUE)

d_rates = cbind(d_one_month_rate, d_three_month_rate, d_six_month_rate)

# simple correlations:

cor(d_rates)

                  d_one_month_rate d_three_month_rate d_six_month_rate
d_one_month_rate          1.0000000          0.7796145        0.6691881
```

```
d_three_month_rate        0.7796145          1.0000000        0.9073374
d_six_month_rate          0.6691881          0.9073374        1.0000000
```

The simple contemporaneous correlations show that there are strong correlations in the changes in these interest rates, which makes sense from an economic point of view as they are just different maturities on the same interest rate.
Since we do not know what the appropriate lag length of the VAR should be, several tests have been developed for choosing the lag value. These have also been automated.

The command,

```
# estimate a Vector Auto Regression

VAR_model <- VAR(d_rates, lag.max=12, type = "none", ic = "AIC")
summary(VAR_model)
```

uses the Akaike Information Criterion to select the lag length. You should find that it selects a lag length of 9. This would yield too many coefficients for sensible exposition, so we display only the results for the first lag, and reduce the output:

```
VAR Estimation Results:
=========================
Roots of the characteristic polynomial:
0.8873 0.8873 0.8829 0.8641 0.8641 0.8448 0.8448 0.8398 0.8398 0.8252
0.8252 0.8202 0.8202 0.8167 0.8167 0.8066 0.8066 0.7475 0.7475 0.7349
0.7349 0.7228 0.7228 0.6854 0.3642 0.3642 0.2806


Estimation results for equation d_one_month_rate:
=================================================

                     Estimate Std. Error t value Pr(>|t|)
d_one_month_rate.l1   -0.29175    0.05000  -5.835 7.64e-09 ***
d_three_month_rate.l1  0.65731    0.11438   5.747 1.27e-08 ***
d_six_month_rate.l1   -0.15167    0.09298  -1.631  0.10322
```

```
Residual standard error: 0.09574 on 851 degrees of freedom
Multiple R-Squared: 0.306,     Adjusted R-squared: 0.284
F-statistic:  13.9 on 27 and 851 DF,  p-value: < 2.2e-16


Estimation results for equation d_three_month_rate:
===================================================

                      Estimate Std. Error t value Pr(>|t|)
d_one_month_rate.l1   -0.28307    0.03869  -7.316 5.92e-13 ***
d_three_month_rate.l1  0.59244    0.08851   6.693 3.96e-11 ***
d_six_month_rate.l1   -0.06722    0.07196  -0.934  0.35045


Residual standard error: 0.07409 on 851 degrees of freedom
Multiple R-Squared: 0.2483,    Adjusted R-squared: 0.2244
F-statistic: 10.41 on 27 and 851 DF,  p-value: < 2.2e-16


Estimation results for equation d_six_month_rate:
=================================================

                      Estimate Std. Error t value Pr(>|t|)
d_one_month_rate.l1   -0.24709    0.04103  -6.022 2.56e-09 ***
d_three_month_rate.l1  0.42718    0.09386   4.551 6.10e-06 ***
d_six_month_rate.l1    0.06000    0.07630   0.786  0.43183


Residual standard error: 0.07856 on 851 degrees of freedom
Multiple R-Squared: 0.2056,    Adjusted R-squared: 0.1804
F-statistic: 8.156 on 27 and 851 DF,  p-value: < 2.2e-16

Covariance matrix of residuals:
                   d_one_month_rate d_three_month_rate d_six_month_rate
d_one_month_rate           0.009164           0.005351         0.004803
d_three_month_rate         0.005351           0.005487         0.005193
d_six_month_rate           0.004803           0.005193         0.006170

Correlation matrix of residuals:
                   d_one_month_rate d_three_month_rate d_six_month_rate
d_one_month_rate             1.0000             0.7546           0.6388
d_three_month_rate           0.7546             1.0000           0.8925
d_six_month_rate             0.6388             0.8925           1.0000
```

Let's interpret and evaluate this estimation.

Interpretation

Given the many coefficients, the simplest method to interpret what this estimation means for the dynamics of these variables is to use two standard tools: the Impulse Response Function and the Forecast Variance Decomposition.

Impulse response function

The Impulse Response Function answers the following question: suppose the system starts in long run equilibrium, i.e. all variables are at their unconditional expected values. Suppose there is now a 1 standard deviation increase in error i. What is the dynamic path of each of the variables in their return to long run equilibrium? Since we have 3 variables with 3 errors, there are 9 of these impulse response functions. To construct these, one must make an untestable assumption about the contemporaneous effects of among the variables. These are called identification assumptions. We will use the Cholesky decomposition (a standard first choice). You should read up on what the strengths and weaknesses are of each of the standard types and select the one most appropriate to your question of interest.

```
# compute and plot the impulse response functions

VAR_irf <- irf(VAR_model, n.ahead = 13,boot = TRUE, ci = 0.95)
plot(VAR_irf)
```

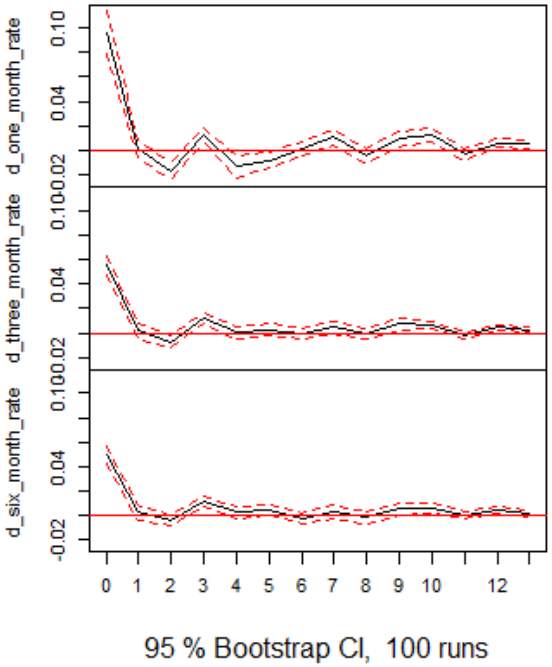


Figure 1: Orthogonal response from d_one_month_rate

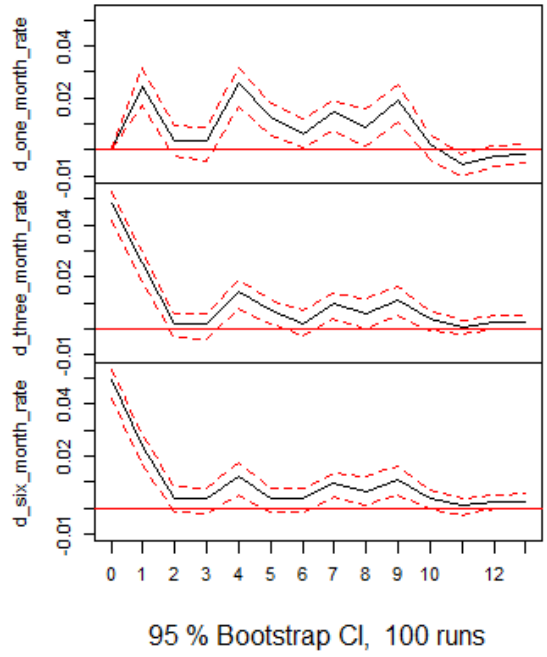


Figure 2: Orthogonal impulse response from d_three_month_rate

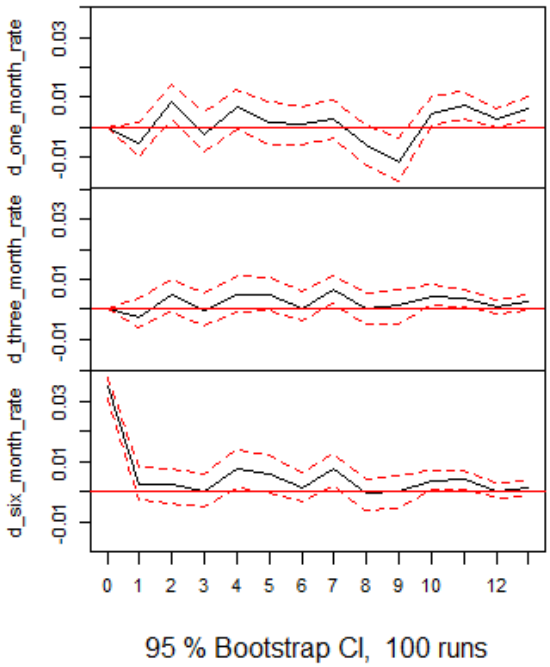


Figure 3: Orthogonal impulse response from d_six_month_rate

We read this as follows:

Each of the combined graphs represents a shock to the specific innovation.
The rows of each graph the responses of each variable to that shock.

Let's focus on the first graph:

By definition, a variable responds immediately to its own innovation. In the top graph, the impact fades out quite quickly, but oscillates around its mean in a dampening pattern for about 13 weeks.

The response of DWED3 and DWED 6 is also large, but slightly smaller in the first period, and follows a similar oscillatory return to mean.

Turning to the second graph:

An innovation to DWED3 initially has no impact on DWED1 – this is due to our assumption of a Cholesky decomposition. In the second to 9th period there is some evidence that the unexpected increase in DWED3 has some positive impact on DWED3.

Forecast variance decomposition

The variance decomposition functions answer the question for each forecast horizon from 1 to h: how much of the forecast error variance is due to the shock in each equation? In this estimate, it will not be very interesting, given the low level of interdependence among the series:

```
# compute and plot the forecast error variance decomposition
VAR_fevd <- fevd(VAR_model,n.ahead = 13)
plot(VAR_fevd)
```

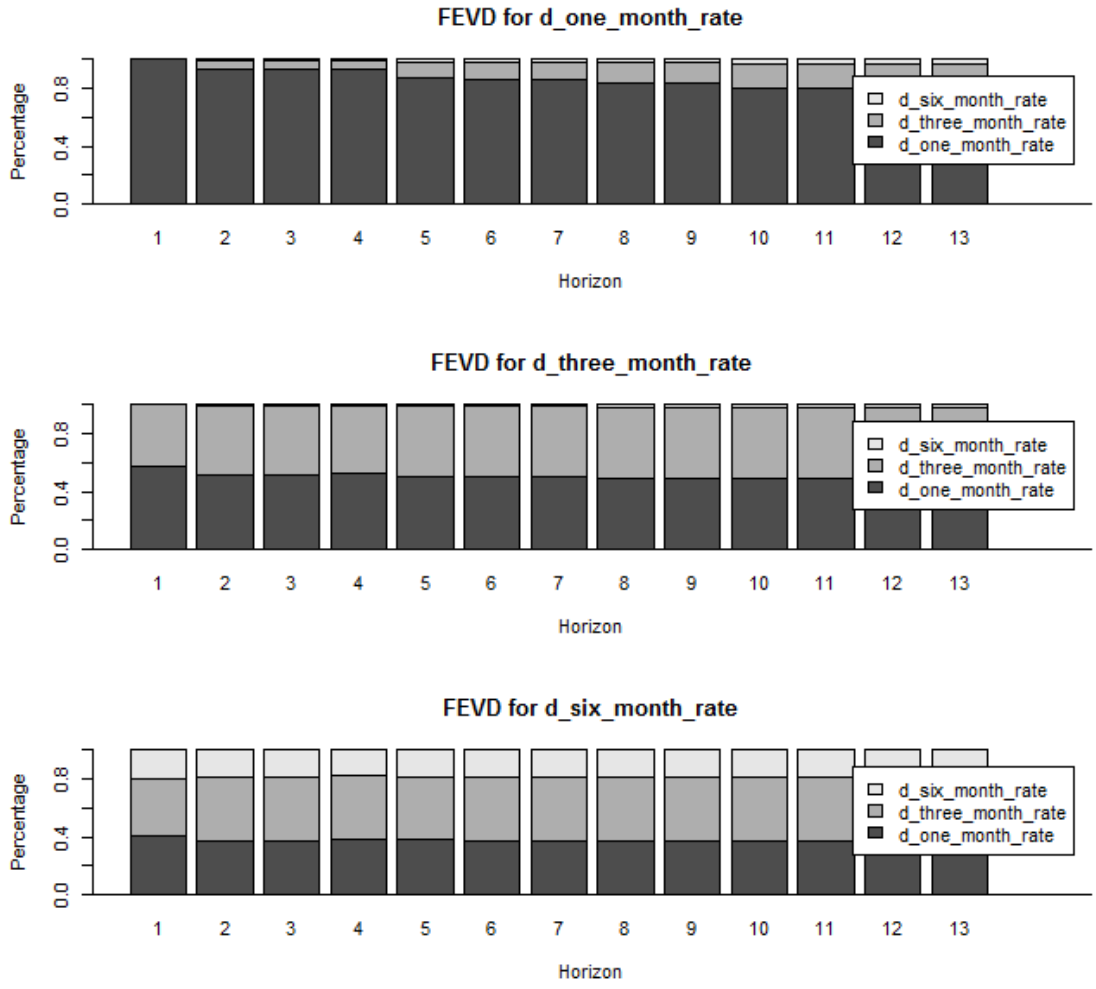


Figure 4: Forecast variance decomposition for multiple rates

The forecast error variance of DWED1 is initially entirely explained by only its own innovations. By 13 weeks, the innovations to DWED3 explain about 20% of this variance.

The forecast error variance of DWED6 looks very different. Even for the first period, the innovations to DWED6 explains only 20% of the forecast variance. It is thus critical to jointly forecast (in this estimated model) all three of these variables to obtain good forecasts of DWED6.

3.2.4 Transcript: A Practical Application to Interest Rates (Part 2)

In this video we will be continuing with the practical application to exchange rate returns.

Evaluation

Stationarity

First, note that all the roots of the characteristic polynomial are inside the unit circle. We concluded a series to be stationary if the roots of the *inverse* characteristic polynomial are *outside* the unit circle. We are thus satisfied that this process is stationary.

Parsimony

If you were to view the full output, you would notice many insignificant coefficients at different horizons and in different equations in the model. One might be tempted to restrict these individually to zero and re-estimate the model. This is not a good strategy, as it implies one is imposing a specific, arbitrary structure on the endogenous joint evolution of the processes without strong a priori economic reasons. Note that in each equation, the F-statistic is large enough in the hypothesis that all of the coefficients in that equation are jointly insignificant.

What is done in VAR models is, at most, to consider removing either a variable from the model (for example, exclude one of the returns series), or reducing the lag structure of the whole model. In other words, if one did an F-test that *all* nine of the eighth lag coefficients were *jointly* insignificant, and did not reject, one could validly make the model more parsimonious by moving to a VAR(7) model. This is called a lag restriction test.

Does the model encompass the data generating process?

A critical aspect of an acceptable model is that it is encompassing – there must be no time series information in the residuals. We could test each of the residual series for auto-correlation but since we have four of these residual series, this is not sufficient. There also must be no cross-correlations between the past of each residual series in the present realization.

A useful visual tool for this purpose is the auto-correlogram and the cross-correlogram. The plot command build for the VAR model displays fitting diagnostics for each individual variable. We consider only the first here:

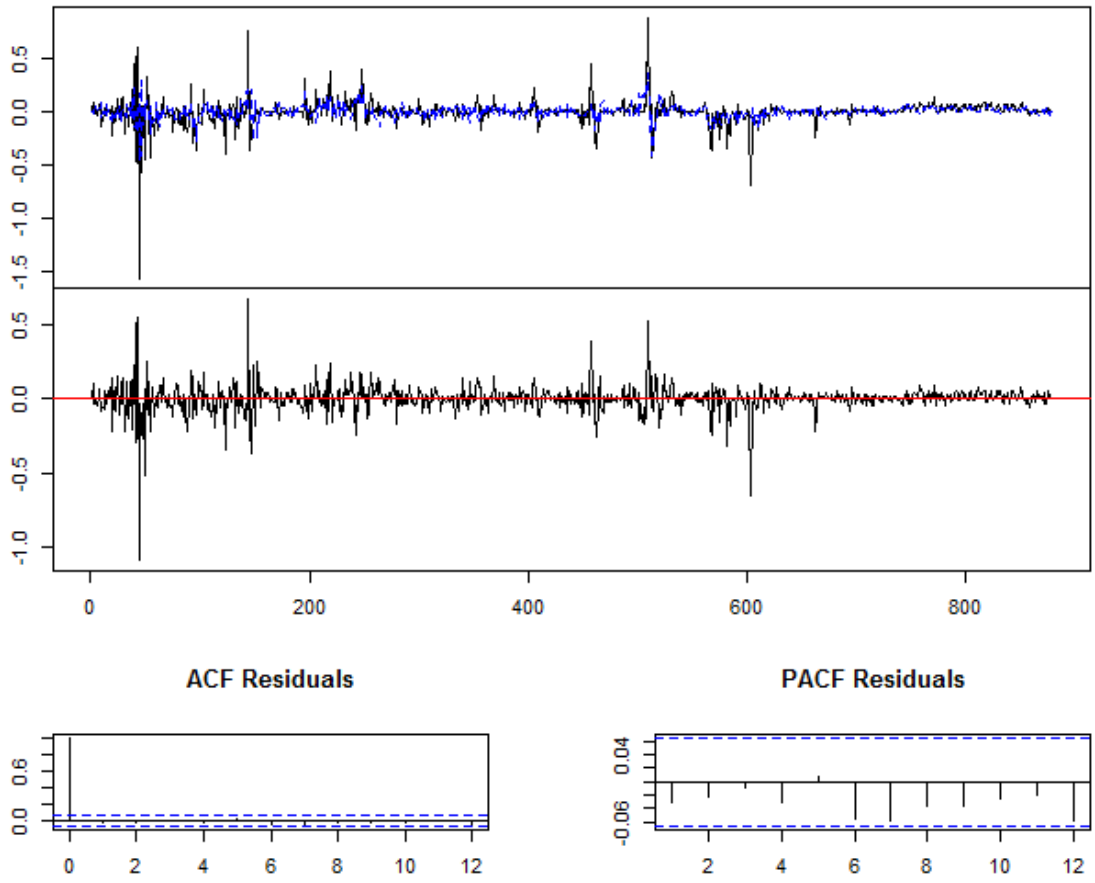


Figure 5: Fit and residuals for d_one_month_rate

The bottom part figure displays auto-correlations and partial autocorrelations of the residuals of the first equation. These are all comfortably insignificant. You should check that there is no systematic pattern in the residuals of the other series as well.

Since we also have potential cross variable time series connections, we should also check that the residuals of different equations are also uncorrelated over time. For this we use the following command in the standard stats package:

```
# obtain residuals:

resids  = residuals(VAR_model)
resid1 = resids[,1]
resid2 = resids[,2]

# view cross correlogram:

ccf(resid1, resid2, lag.max = 13, type = "correlation", plot = TRUE)
```

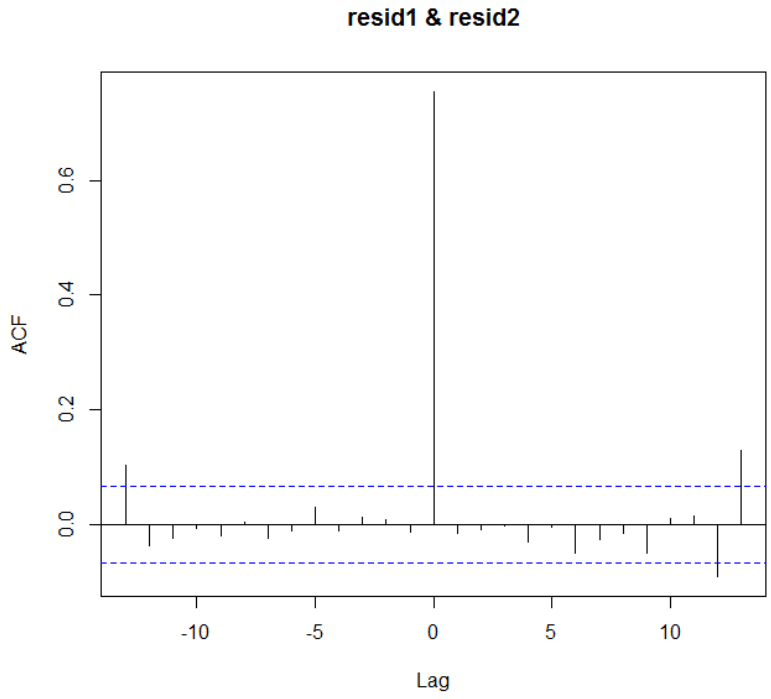


Figure 6: R-generated graphical output

There are strong contemporaneous correlations. This is standard in residuals and what the identification problem is built on disentangling. There is also significant correlation at long lags. It is not immediately obvious if this is systematic. Sampling variation will always yield one or two spuriously significant correlations, so you must use good judgment and statistical tests to conclude whether a VAR model is encompassing the data-generating process of interest.

3.2.5 Notes: The Cointegration / Vector Error Correction Model

Definition and basic results

Definition: the $I(p)$ vector process $\mathbf{x}_t$ (each element is integrated of order p, i.e. the p^{th} difference is stationary), is linearly cointegrated of order q (with $p > q \geq 1$; p and q both integers), which we denote as $CI(p, q)$, if there exists at least one non-zero row-vector $\mathbf{b}$, unique up to a scalar multiple, such that $\mathbf{b}\mathbf{x}_t$ is $I(p - q)$.

In simple terms: a set of variables are cointegrated if they are individually integrated to the same, non-zero order, and there exists a linear combination of them that is integrated of a strictly lower order.

In practice, we almost always work with a set of $I(1)$ variables (i.e. variables where the first difference is stationary) where there exists one (or more) linear relationship that renders their linear combination $I(0)$, i.e. stationary.

There are a number of crucial facts you should be aware of and study the reasons for carefully (most simply illuminated in Enders (2014)):

1. All variables in the vector $\mathbf{x}_t$ must, individually, be integrated of the same order.
2. Among n cointegrated variables there must be at least 1, and *at most* $n - 1$ cointegration relationships.
3. If there are $r \in [1, n - 1]$ cointegration relationships this implies that there are *exactly* $n - r$ common stochastic trends. In simple terms: if there are r different linear ways to combine n non-stationary variables to obtain a stationary result, there *must* be $n - r$ different sources of non-stationarity in the variables.
4. If $\mathbf{b}$ is a cointegration relationship, so is $\lambda\mathbf{b}$ for any $\lambda \neq 0$. Thus, we can rescale any cointegration relationship and it remains a cointegration relationship. For this reason, it is standard to normalize the cointegration vector. We will see how this is done in the practical application.

5 We can consider $\mathbf{b}$ to represent a long run relationship. In any specific period, the values of $\mathbf{x}_t$ will not exactly be in this long run "balance" so there will be a deviation from long-run equilibrium which we can characterize as an "additional error": $\mathbf{b}\mathbf{x}_t = u_t$.

The vector error correction representation

When a set of variables are cointegrated, there exists a very intuitive representation that is extremely useful in time series econometrics. We call this the Vector Error Correction representation, leading to the Vector Error Correction Model, which we label VECM.

Consider a VAR(2) model:

$$\mathbf{x}_t = \mathbf{A}_0 + \mathbf{A}_1\mathbf{x}_{t-1} + \mathbf{A}_2\mathbf{x}_{t-2} + \boldsymbol{\varepsilon}_t.$$

Add and subtract $\mathbf{A}_2\mathbf{x}_{t-1}$ on the right-hand side and rearrange:

$$\mathbf{x}_t = \mathbf{A}_0 + \mathbf{A}_1\mathbf{x}_{t-1} + \mathbf{A}_2\mathbf{x}_{t-1} - \mathbf{A}_2\mathbf{x}_{t-1} + \mathbf{A}_2\mathbf{x}_{t-2} + \boldsymbol{\varepsilon}_t$$

$$= \mathbf{A}_0 + (\mathbf{A}_1 + \mathbf{A}_2)\mathbf{x}_{t-1} - \mathbf{A}_2\Delta\mathbf{x}_{t-1} + \boldsymbol{\varepsilon}_t$$

where we define the first difference operator as: $\Delta\mathbf{x}_{t-1} \equiv \mathbf{x}_{t-1} - \mathbf{x}_{t-2}$.

Next subtract $\mathbf{x}_{t-1}$ from both sides:

$$\mathbf{x}_t - \mathbf{x}_{t-1} = \mathbf{A}_0 + (\mathbf{A}_1 + \mathbf{A}_2)\mathbf{x}_{t-1} - \mathbf{x}_{t-1} - \mathbf{A}_2\Delta\mathbf{x}_{t-1} + \boldsymbol{\varepsilon}_t$$

$$\Delta\mathbf{x}_t = \mathbf{A}_0 + (\mathbf{A}_1 + \mathbf{A}_2 - \boldsymbol{I})\mathbf{x}_{t-1} - \mathbf{A}_2\Delta\mathbf{x}_{t-1} + \boldsymbol{\varepsilon}_t$$

where $\boldsymbol{I}$ is the $[n\ x\ n]$ identity matrix.

Lastly, relabel the coefficient matrices to be in standard additive form:

$$\Delta \mathbf{x}_t = \mathbf{A}_0 + \mathbf{\Pi} \mathbf{x}_{t-1} + \mathbf{C}_1 \Delta \mathbf{x}_{t-1} + \boldsymbol{\varepsilon}_t$$

You should check that this can be generalized to any lag order, so we can transform a VAR(p) process into a VECM(p) representation:

$$\Delta \mathbf{x}_t = \mathbf{A}_0 + \mathbf{\Pi} \mathbf{x}_{t-1} + \mathbf{C}_1 \Delta \mathbf{x}_{t-1} + \cdots + \mathbf{C}_p \Delta \mathbf{x}_{t-p} + \boldsymbol{\varepsilon}_t$$

Why is this rewriting of the process useful?

1. The statistical properties of the $[n \; x \; n]$ matrix $\mathbf{\Pi}$ leads to the popular Johansen test for cointegration (Johansen, 1988).
2. Using an obvious decomposition of the $\mathbf{\Pi}$ matrix yields the most important intuitive interpretation of cointegration.

The Johansen test for cointegration

If there are r cointegration relationships among n variables, the matrix $\mathbf{\Pi}$ has, asymptotically, rank r. This means it has only $r < n$ linearly independent columns/rows. In other words, it is rank-deficient/ non-invertible. A square matrix of dimension n with rank $r < n$ has exactly r non-zero eigenvalues (it is essential that you review the basic

results from matrix algebra if you are uncertain what "rank" or "eigenvalues" refers to).

In any finite sample, the matrix $\mathbf{\Pi}$ will be estimated imperfectly, and so will have full rank considered purely numerically. Johansen (1988) derived the probabilistic characteristics of estimated eigenvalues, developing a test for the number of *statistically significant* eigenvalues of a matrix.

The details of these derivations are beyond the scope of this module, but the Johansen test involves estimating an unrestricted VECM(p), where p is large enough to capture all of the time series properties of the process involved (so that the errors are white noise), and then testing for the number of statistically significant eigenvalues (or equivalently, the rank of the matrix $\mathbf{\Pi}$).

The conclusion of this test yields our best estimate of the number of cointegration relationships in the system.

The final step is then to re-estimate the system imposing exactly the number of cointegration relationships as identified by the Johansen test and interpreting the results. We will illustrate this with an example at the end of this section.

The VECM approach to cointegration

Once we conclude that there are r cointegrating relationships in a system and impose them on our restricted VECM estimation, the matrix $\mathbf{\Pi}$ will have exactly rank r.

An $[n\ x\ n]$ matrix with rank $r < n$ has a unique (up to relative scaling) decomposition:

$$\underset{[n\times n]}{\mathbf{\Pi}} = \underset{[n\times r]}{\mathbf{A}}\underset{[r\times n]}{\mathbf{B}}$$

where $\mathbf{A}$ has r linearly independent columns and $\mathbf{B}$ has r linearly independent rows.

Why this is useful is most easily seen when $r = 1$, i.e. there is only one cointegration relationship. In this case, we can rewrite the system of equations (abstracting from all constants for ease of exposition – it is trivial to allow for them, but the notation is messier without adding to intuition) as:

$$\Delta\mathbf{x}_t = \underset{[n\times 1]}{\mathbf{A}}\underset{[1\times n]}{\mathbf{B}}\mathbf{x}_{t-1} + \mathbf{C}_1\Delta\mathbf{x}_{t-1} + \boldsymbol{\varepsilon}_t$$

$$\Delta \mathbf{x}_t = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} [b_1 \quad b_2 \quad \cdots \quad b_n] \mathbf{x}_{t-1} + \mathbf{C}_1 \Delta \mathbf{x}_{t-1} + \boldsymbol{\varepsilon}_t$$

Focusing on row i of the system (where $\mathbf{c}_{i1}$ refers to the i^{th} row of $\mathbf{C}_1$):

$$\Delta x_{it} = a_i [b_1 \quad b_2 \quad \cdots \quad b_n] \mathbf{x}_{t-1} + \mathbf{c}_{i1} \Delta \mathbf{x}_{t-1} + \varepsilon_{it}$$

If we use the definition of the deviation from long run equilibrium mentioned above: $\mathbf{b}\mathbf{x}_t = u_t$ this becomes:

$$\Delta x_{it} = a_i u_{t-1} + \mathbf{c}_{i1} \Delta \mathbf{x}_{t-1} + \varepsilon_{it}$$

Since u_{t-1} is the deviation from long run equilibrium, the variables in our system must in some way "correct" for this deviation (or error), otherwise the "long run equilibrium" has no meaning. If deviations from long run equilibrium are never corrected, how can it be a sensible definition of a long run equilibrium?

Thus, we can interpret a_i as a "speed-of-adjustment" parameter: it measures how much of the deviation from long run equilibrium in the previous period, u_{t-1}, is corrected *by the short run dynamics of variable* i (which is represented by the process we assign to Δx_{it}).

Lastly, $\mathbf{c}_{i1} \Delta \mathbf{x}_{t-1}$, captures the strictly short run dynamics – how the change in variable i in the current period depends on last periods' change in *all* the variables in the system.

We illustrate all these results in a practical application.

3.2.6 Transcript: VECM Derivations and the Johansen Test

In the first part of the practical application to exchange rate returns we modeled the stationary changes in weekly interest rates with a VAR process. In practice, this differencing "throws away" all level information, but was necessary because we can only apply the VAR approach to stationary series.

When we turn to the VECM approach we can simultaneously model both the level *and* the difference of the series, provided the levels of the series are cointegrated. We will do just the very basics of the cointegration approach here, given space constraints. All issues of parsimony of models remain crucial, but we will not address them here as they are evaluated and tested for in exactly the same way as above.

So, for this application, consider the relationship between the *levels* of the exchange rates we considered above.

The Johansen test involves testing for the number of statistically non-zero eigenvalues of the matrix **Π.** For this, we estimate an unrestricted VECM (which is just a rewriting of an unrestricted VAR) and do two tests: the trace and max eigenvalue tests developed by Johansen (1988).

```
## Cointegration Analysis
library(urca)
library(tsDyn)

# store the levels of interest rates:

one_month_rate <- eurodollarrates$WED1
three_month_rate <- eurodollarrates$WED3
six_month_rate <- eurodollarrates$WED6

rates = cbind(one_month_rate, three_month_rate, six_month_rate)
```

```
# Johansen test

jotest1=ca.jo(rates, type="eigen", K=9, ecdet="none",
spec="longrun")
summary(jotest1)
jotest2=ca.jo(rates, type="trace", K=9, ecdet="none",
spec="longrun")
summary(jotest2)
```

The output is:

```
######################
# Johansen-Procedure #
######################

Test type: maximal eigenvalue statistic (lambda max) , with linear trend

Eigenvalues (lambda):
[1] 0.146156486 0.065201628 0.006132301

Values of teststatistic and critical values of test:

          test 10pct  5pct  1pct
r <= 2 |   5.41  6.50  8.18 11.65
r <= 1 |  59.27 12.91 14.90 19.19
r = 0  | 138.89 18.90 21.07 25.75

Eigenvectors, normalised to first column:
(These are the cointegration relations)

                   one_month_rate.l9 three_month_rate.l9 six_month_rate.l9
one_month_rate.l9          1.0000000            1.000000          1.000000
three_month_rate.l9       -1.6015884            2.904978         -1.257257
six_month_rate.l9          0.6080626           -3.886571          3.767824

Weights W:
(This is the loading matrix)

                  one_month_rate.l9 three_month_rate.l9 six_month_rate.l9
one_month_rate.d         -0.5234407         -0.03698967     -0.0002448355
three_month_rate.d       -0.1128012         -0.03317527     -0.0005654808
six_month_rate.d         -0.1715670         -0.01719750     -0.0008244080
```

```
######################
# Johansen-Procedure #
######################

Test type: trace statistic , with linear trend

Eigenvalues (lambda):
[1] 0.146156486 0.065201628 0.006132301

Values of teststatistic and critical values of test:

           test 10pct  5pct  1pct
r <= 2 |   5.41  6.50  8.18 11.65
r <= 1 |  64.67 15.66 17.95 23.52
r = 0  | 203.56 28.71 31.52 37.22

Eigenvectors, normalised to first column:
(These are the cointegration relations)

                    one_month_rate.l9 three_month_rate.l9 six_month_rate.l9
one_month_rate.l9           1.0000000            1.000000          1.000000
three_month_rate.l9        -1.6015884            2.904978         -1.257257
six_month_rate.l9           0.6080626           -3.886571          3.767824

Weights W:
(This is the loading matrix)

                   one_month_rate.l9 three_month_rate.l9 six_month_rate.l9
one_month_rate.d          -0.5234407         -0.03698967     -0.0002448355
three_month_rate.d        -0.1128012         -0.03317527     -0.0005654808
six_month_rate.d          -0.1715670         -0.01719750     -0.0008244080
```

These tests show strong evidence of cointegration, satisfying the requirements we place on the **Π**.

Test two

If test value > critical value (10 %, 5 % or 1 %) -> reject the null hypothesis

If test value < critical value (10 %, 5 % or 1 %) -> fail to reject the null hypothesis

Null Hypothesis		Alternative Hypothesis	10%	
test 10pct 5pct 1pct r <= 2 | 5.41 6.50 8.18 11.65 r <= 1 | 64.67 15.66 17.95 23.52 r = 0 | 203.56 28.71 31.52 37.22	r > 2	fail to reject the null hypothesis	5.41 < 6.50	so there are maximum 2 cointegration relations
	r > 1	reject the null hypothesis	64.67 > 15.66	so there is more than 1 cointegration relations
	r > 0	reject the null hypothesis	203.56 > 28.71	so there is more than zero cointegration relations

Table 1: Relationship between hypotheses and cointegration

Both tests strongly reject the hypotheses that there are no or at most 1 cointegration relationship but fail to reject that there are at most 2 cointegration relationships between these variables. Thus, we conclude that there are 2 cointegration relations between the variables and estimate a VECM model imposing this. Again, we truncate the output to focus only on the new features.

To keep the output simple, we estimate with only 2 lagged differences:

```
# Fit cointegrated VECM

VECM_fit = VECM(rates,1,r = 2, include = "const",estim = "ML",
LRinclude = "none")
summary(VECM_fit)
```

```
Intercept          one_month_rate -1
```

3.2.7 Notes: Principal Component Analysis

It is clear that there is a large amount of information in, say, the price series of the full set of assets traded on the New York Stock Exchange, and that there is a lot of mutual information. It is therefore infeasible to use a model like the VAR model as there would be too many coefficients to estimate.

Therefore, we need a method to reduce the dimensionality of the dataset. One approach is to generate an index: some linear combination that represents the stock exchange as a whole. This is a rather arbitrary approach and loses all information of the correlations over time between the various asset prices.

The most important approaches to reduce the dimensionality are Principal Component Analysis (PCA) and the closely related Exploratory Factor Analysis (EFA). You should explore the differences to choose which to use. In brief, Factor Analysis places more ex ante structure on the model behind the common variance or correlations, while Principal Component Analysis is a purely statistical decomposition exercise. We will focus on PCA in this section.

PCA is based on standard decomposition results in Linear Algebra: Consider an [T x n] matrix $\mathbf{X}$ where the T observations are recorded for each of the n variables in the columns of the matrix. Let T>n. If $\mathbf{X}$ has full column rank, in other words, all the columns are linearly independent and no column is an exact linear function of another, they will not, in the datasets of interest to us, be *orthogonal*. This means that there will be non-zero correlations between the columns and decomposition will exist:

$$\underset{[T\times n]}{\mathbf{X}} = \underset{[T\times n]}{\mathbf{C}}\underset{[n\times n]}{\mathbf{L}}$$

Where $\mathbf{C}$ is orthogonal (i.e. even the *sample* correlations between the columns are zero) and called the "Components" of the process, and $\mathbf{L}$ is called the "Loading

Matrix". Thus, we can decompose a set of correlated columns / variables into a set of *uncorrelated* components and a loading matrix.

Why is this useful? We typically find that there are only a few columns of $\mathbf{C}$ (usually less than 4 even when $\mathbf{X}$ has more than 100 columns) that explain the majority of the variation (usually more than 95%) in *all* the columns of $\mathbf{X}$. Thus, we can reduce a dataset of 100 variables to 3 and still capture almost all of the time series information in the original dataset, including cross correlations induced by the loading matrix. It is much easier to analyze characteristics of 3 uncorrelated variables than the joint correlation structure of 100 variables.

In matrix algebra terms, we decompose $\underset{[T\times n]}{\mathbf{X}} = \underset{[T\times n][n\times n]}{\mathbf{C}}\ \underset{}{\mathbf{L}}$, with the columns of $\mathbf{C}$ so that the individual variance of each column is decreasing from first to last. Then we assume that the first m columns capture as much of the variation in $\mathbf{X}$ as we deem necessary for the purpose of the empirical analysis. We call these m components the "principal components" of $\mathbf{X}$. Then we assume the rest forms part of the error of this approximation (which is orthogonal to $\mathbf{C}$ by construction):

$$\underset{[T\times n]}{\mathbf{X}} = \underset{[T\times m][m\times n]}{\mathbf{C}}\ \underset{}{\mathbf{L}} + \underset{[T\times(n-m)][(n-m)\times n]}{\bar{\mathbf{C}}}\ \underset{}{\bar{\mathbf{L}}}$$

$$\underset{[T\times n]}{\mathbf{X}} = \underset{[T\times m][m\times n]}{\mathbf{F}}\ \underset{}{\mathbf{\Lambda}} + \underset{[T\times n]}{\boldsymbol{\varepsilon}}$$

The next step is typically to assign some meaning to each of the orthogonal principal components by comparing them to aggregate level variables. For example: the first principal component that explains the variation in NYSE stock prices may be an "aggregate domestic economic condition", the second "aggregate international economic conditions in developed economies" and the third "aggregate international economic conditions in developing economies". It is left up to you to study papers using PCA to see how interpretations of the principal components are explained / motivated.

Lastly, the financial engineer will use these factors to choose a portfolio that should be robustly controlling for all aggregate factors to build the most desirable portfolio for a client who has specific, known and desired exposures to the various factors of interest to the client's investment strategy.

Application

While we motivated PCA as a data reduction technique, we can show how it works (briefly) using our standard example of changes in the three interest rate series. The method princomp is part of the standard stats packages.

```
# Apply PCA to the changes in the interest rates:
pca = princomp(d_rates)

# plot the variance of the three components
plot(pca)
```

This yields a graph of the relative variance of each component (and there are 3 by construction since we use a dataset of three variables):

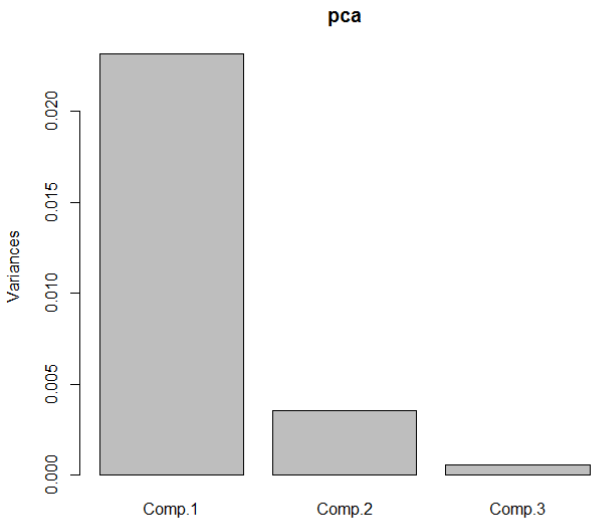


Figure 7: Relative variance of interest rate series

We observe that the first component has an almost 5 times larger variance than the second, and that the third component has almost negligible variance in comparison. This means that we can capture a large part of the variability in all three base series by using only the first component without losing a lot of information.

To see the "orthogonalization" of the method, let's construct the correlation matrix between the different components:

```
# compute the correlations of the components:
cor(pca$scores)

              Comp.1        Comp.2        Comp.3
Comp.1  1.000000e+00 -6.689019e-15 -1.395343e-14
Comp.2 -6.689019e-15  1.000000e+00  5.770984e-15
Comp.3 -1.395343e-14  5.770984e-15  1.000000e+00
```

Note that the off diagonals are zero up to 14 decimal positions – exactly what the procedure wanted to achieve.
The method also yields the loading matrix (which are the eigen vectors corresponding to the covariance matrix of the original data):

```
> pca$loadings

Loadings:
                  Comp.1 Comp.2 Comp.3
d_one_month_rate   -0.689  0.709  0.152
d_three_month_rate -0.521 -0.337 -0.784
d_six_month_rate   -0.505 -0.619  0.601

               Comp.1 Comp.2 Comp.3
SS loadings     1.000  1.000  1.000
Proportion Var  0.333  0.333  0.333
Cumulative Var  0.333  0.667  1.000
```

If we were to take the [3 x 3] loading matrix and pre-multiply it by the [n x 3] score matrix (each column containing a successive component), we would recreate the original [n x 3] data on the changes in the three interest rates exactly.

3.2.8 Notes: Multivariate GARCH Analysis

The most general multivariate GARCH model extends the univariate GARCH model to consider the feedback from the conditional expectations among a set of variables to allow the conditional variance of each element to affect the conditional variance of each other element of the set. Moreover, it is used to allow each conditional variance to be influenced by the conditional *covariance* between *every* two variables; and even *more*, to allow the conditional *covariance* between any pair of variables to affect the conditional covariance between *all* other pairs of variables.

As the simplest possible example, consider a model where the conditional expectations of *only* two variables are perfectly modeled with just a constant (a heroic assumption in its own right), but whose conditional variances have joint ARCH characteristics (Enders, 2014):

$$x_{1t} = \mu_1 + u_{1t}$$
$$x_{2t} = \mu_2 + u_{2t}$$
$$u_{1t} = \varepsilon_{1t}\sqrt{h_{11t}}$$
$$u_{2t} = \varepsilon_{2t}\sqrt{h_{22t}}$$
$$\varepsilon_{it} \sim N(0,1)\ for\ i = 1,2$$

where the full multivariate GARCH(1,1) variance-covariance matrix conditional on the information in period $t-1$ is modeled as:

$$E_{t-1}\begin{bmatrix} u_{1t}^2 & u_{1t}u_{2t} \\ u_{1t}u_{2t} & u_{2t}^2 \end{bmatrix} = \begin{bmatrix} h_{11t} & h_{12t} \\ h_{12t} & h_{22t} \end{bmatrix}$$

We model the full generality of the interactions between the variances and covariances in the following 3 equations which describe the time-variant variance-covariance matrix relationships (relate the content of these equations to the text of the first paragraph of this section and refer to Module 4 to see the

simpler, univariate versions of these equations for comparison – it is critical to understand the univariate relationships before these more complicated relationships will make sense):

$$h_{11t} = \omega_{10} + \alpha_{11}\varepsilon_{1t-1}^2 + \alpha_{12}\varepsilon_{1t-1}\varepsilon_{2t-1} + \alpha_{13}\varepsilon_{2t-1}^2 + \beta_{11}h_{11t-1} + \beta_{12}h_{12t-1} + \beta_{13}h_{22t-1}$$

$$h_{12t} = \omega_{20} + \alpha_{21}\varepsilon_{1t-1}^2 + \alpha_{22}\varepsilon_{1t-1}\varepsilon_{2t-1} + \alpha_{23}\varepsilon_{2t-1}^2 + \beta_{21}h_{11t-1} + \beta_{22}h_{12t-1} + \beta_{23}h_{22t-1}$$

$$h_{22t} = \omega_{30} + \alpha_{31}\varepsilon_{1t-1}^2 + \alpha_{32}\varepsilon_{1t-1}\varepsilon_{2t-1} + \alpha_{32}\varepsilon_{2t-1}^2 + \beta_{31}h_{11t-1} + \beta_{32}h_{12t-1} + \beta_{33}h_{22t-1}$$

As such, the fully general approach suffers to a dramatically greater degree of the *curse of dimensionality*. Even this general specification of *only* the variance-covariance properties (thus ignoring the VAR aspects that may be necessary to yield an encompassing model their conditional expectations) of *only* two variables, at *only* one lag, requires 23 coefficient estimates (two means and 7 coefficients per equation per unique element of the [2x2] variance-covariance matrix) for a simple multivariate GARCH(1,1) model. This clearly makes a fully general estimation of a multivariate GARCH among more than a few variables entirely infeasible. The number of coefficients we would need to estimate per additional lag or additional variable will grow at a much greater than quadratic rate in either of these dimensions of extension.

For this reason, the feasible models in this category have been developed as a compromise to account for the most important effects that we want to allow for between variables that are joint GARCH processes while avoiding the massive accumulation of estimates that is implied by the fully general approach. You should carefully study Enders (2014) to learn about the basic problem of the many coefficients as well as all of the various models developed to side step this problem.

We will only discuss the most important version of these practical compromises between the most general and the feasibly estimable models.

The Constant Conditional Correlation Multivariate GARCH model

This model is a compromise between a fully general covariance process (that depends on past conditional variances and covariances in a general way) and a process that allows the conditional covariance to depend only on the conditional variances of each of the variables under consideration.

In the two-variable case, it implies replacing the equation for h_{12t} given above with the following equation:

$$h_{12t} = \rho_{12}\sqrt{h_{11t}h_{22t}}$$

where ρ_{12} is the sample version of the constant unconditional covariance between x_{1t} and x_{2t} across the full sample (Juselius, 1990).

This means we are not allowing a change in the fundamental correlation between the two (or in the larger model) series in the model that is not reliant on the individual conditional variances in the series.

This is unsatisfactory for a financial engineer as there are strong reasons to be concerned about times where two or more series are more correlated relative to times when they are not. In times where two (or more) series are less correlated, there is more scope to diversify portfolios across them.

If there are times where we can predict that two (or more series) will be more correlated than previous times, there exists less opportunity to use these processes as an effective risk diversification strategy. The CCC-GARCH model cannot help a financial engineer with identifying these periods reliably.

The Dynamic Conditional Correlation Multivariate GARCH model

The DCC-MGARCH model makes use of a feasible generalization that maintains the parsimony of the CCC-MGARCH, extended to a second step where the residuals from the first step are used to construct an estimate of the time-varying estimate of the conditional correlation (Engle, 2002).

The procedure works as follows:

Step 1: Estimate a CCC Garch and obtain the residuals $\hat{u}_{it}$ from the conditional expectation equation of each variable in the system.

Step 2: Construct the series of *standardized residuals* (these can be interpreted as estimates of the fundamental standard normal errors ε_{it} in the equations above)*:*

$$s_{it} = \frac{\hat{u}_{it}}{\sqrt{h_{iit}}}$$

Step 3: Construct an estimate of the time-varying variances and covariances based on these standardized residuals, using an iterative smoother. In the original paper, Engle analyses many options, the simplest being the exponential smoother:

$$q_{ijt} = (1-\lambda)s_{it}s_{jt} + \lambda q_{ijt-1} \text{ for } \lambda < 1, \text{and all pairs of } i, j$$

Note that this is simply a smoothed version of the contemporaneous cross-products of the standardized residuals. The unconditional covariance would be estimated as the *average* of all of these cross-products, which would be used as the first value q_{ijt}.

Step 4: Convert this into dynamic correlations between variables:

$$\rho_{ijt} = \frac{q_{ijt}}{\sqrt{q_{iit}q_{jjt}}}$$

This last process can then be used and forecast to consider situations where correlations get higher for some period (e.g. when exchange rates co-move more than normal times, like a global financial crisis).

References

Berlinger, E. et al. *Mastering R for Quantitative Finance*. Packt Publishing, Chapter 1: Time Series Analysis, pages 7-38.

Blanchard, O. J. and Quah, D. (1989) 'The Dynamic Effects of Aggregate Demand and Supply Disturbances', *National Bureau of Economic Research Working Paper Series*, No. 2737(2737).

Enders, W. (2014) *Applied Econometric Time Series*. 4th edn. Wiley.

Engle, R. (2002) 'Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models', *Journal of Business and Economic Statistics*, 20(3), pp. 339–350. doi: 10.1198/073500102288618487.

Fred.stlouisfed.org (2018) 'Data: weekly interest rates on Eurodollar deposits'.

Granger C. W. J. (1992). 'Forecasting stock market prices: lessons for forecasters.' *International Journal of Forecasting.*

Hamilton, J. D. (1992) *Time Series Analysis*. Princeton: Princeton University Press.

Jeet P. and Vats P. *Learning Quantitative Finance with R. Packt Publishing,* Chapter 5: Algorithmic Trading, pages 125-162.

Johansen, S. (1988) 'Statistical analyis of cointegration vectors', *Journal of Economic Dynamics and Control*, 12, pp. 231–254.

Juselius, K. (1990) *Long-run Relations in a Well Defined Statistical Model for the Data Generating Process. Cointegration Analysis of the PPP and the UIP Relations*.

Lack, C. (2006). 'Forecasting Swiss inflation using VAR models.' *Swiss National Bank Economic Studies.*

Mazzoni, T. (2010). Are Short Term Stock Asset Returns Predictable? An Extended Empirical Analysis.

Stephens, D. (2004). 'The equilibrium exchange rate according to PPP and UIP.' *Reserve Bank of New Zeeland.*

Tsay, R. S. (2010) 'Analysis of Financial Time Series.' Wiley.