An Introduction to Univariate Financial Time Series Analysis

Contents

1. Introduction

We have seen that, although the CER model has been widely adopted as an approximation to the unknown Data Generating Process for financial returns, it shows some undesirable properties when applied to design strategies of portfolio allocation. As a matter of fact, the CER model is the simplest possible econometric specification for the variables to our interest and a natural question arises: can we do better than using the constant as the only predictor for financial returns? Selecting time-varying predictors requires using the properties of observed data to predict future observations of the relevant variables. Time-series analysis is the branch of econometrics that deals with this question. We shall consider in turn univariate and multivariate time series models. In univariate models the relevant information set to predict one variable is restricted to the past history of that variable, for which patterns are discovered and exploited. Restricting the information set to the past history of one variable is giving up a wide source of information for prediction, however the analysis of univariate time-series is interesting because it is possible to investigate the relevant problems for time-series analysis in a simplified environment. After having introduced the basic concepts of time-series analysis in the univariate context, we shall quickly move to the more interesting and relevant multivariate time-series analysis.

2. **Time Series Analysis**

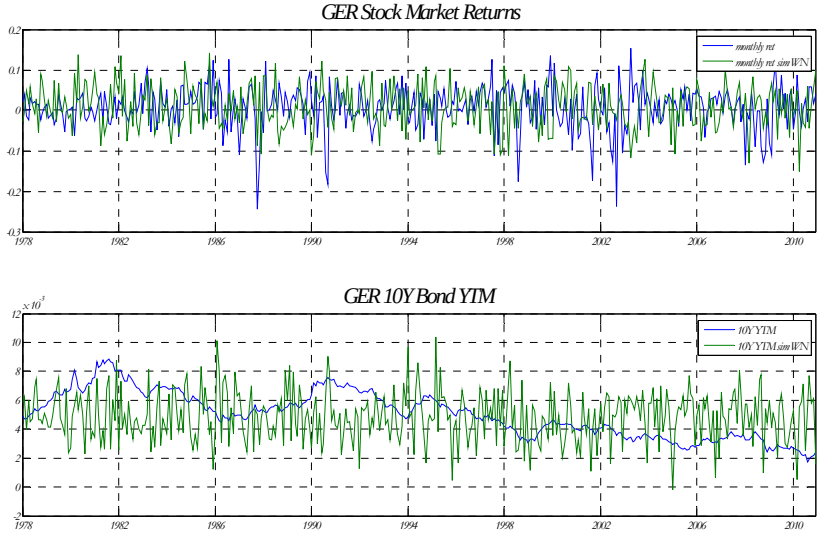
A time series is a sequence

$$\{x_1, x_2, ..., x_T\} \text{ or } \{x_t\}, \ t = 1, ..., T,$$

where t is an index denoting the period in time in which x occurs. Returns on a financial assets observed over a given sample constitute the typical time-series of our interest. x_t is a random variable; hence, a time-series is a sequence of random variables ordered in time. Such a sequence is known as a stochastic process. The probability structure of a sequence of random variables is determined by the joint distribution of a stochastic process. The CER is the simplest case of a probability model for such a joint distribution :

$$x_t = \alpha + \sigma_\epsilon \epsilon_t, \ \epsilon_t \sim NID(0, 1), \tag{1}$$

i.e., x_t is normally independently distributed over time with constant variance and mean equal to α. In other words, x_t is the sum of a constant and a *white noise* process. If a white-noise process were a proper model for financial time-series, than forecasting would not be very interesting as the best forecast for the moments of the relevant time series would be their unconditional moments. However, the longer is the horizon at which a return is defined the more likely that it will include a "persistent" component that cannot be modelled via a simple white-noise. Figure 1 illustrates the point by reporting the actual German 1-month stock market returns and the yield to maturity of German 10-year government bonds along with two artificial series generated by drawing data for process (1) calibrated to have means and variances that match those of the actual data.

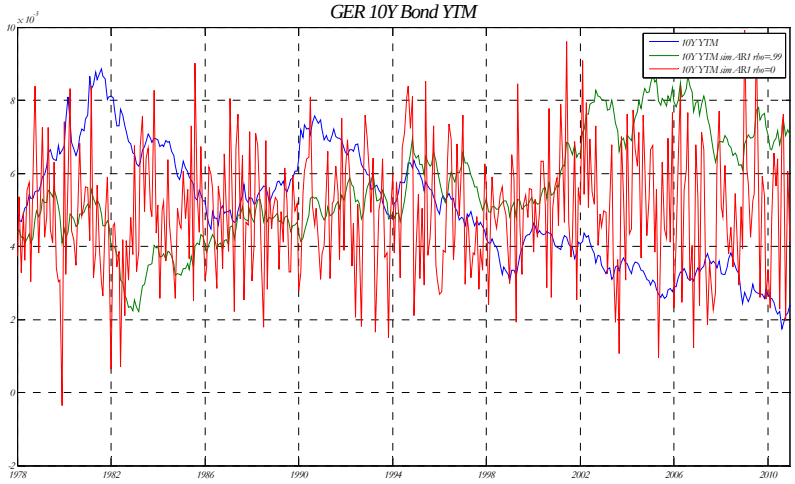


While the CER gives a plausible representation for the 1-month returns, the behavior over time of the YTM of the 10-Year bond does not resemble at all that of the simulated data.

To construct more realistic models, we concentrate on univariate models first, to consider then multivariate models. In univariate models the basic idea is to use combinations of ϵ_t to generate more flexible models capable of replicating the relevant features of the data. In particular, we concentrate on a class of models created by taking linear combinations of the white noise process, the autoregressive moving average (ARMA) models:

$$\begin{aligned}
AR(1) &: \quad x_t = \rho x_{t-1} + \epsilon_t, \\
MA(1) &: \quad x_t = \epsilon_t + \theta \epsilon_{t-1}, \\
AR(p) &: \quad x_t = \rho_1 x_{t-1} + \rho_2 x_{t-2} + ... + \rho_p x_{t-p} + \epsilon_t, \\
MA(q) &: \quad x_t = \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q}, \\
ARMA(p,q) &: \quad x_t = \rho_1 x_{t-1} + ... + \rho_p x_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q}.
\end{aligned}$$

Figure 2 illustrates the potential of an AR(1) of replicating different patterns of persistence in time-series by varying the ρ_1 parameter. Note that a parameter very close to one is needed to replicate the persistence observed in the YTM of 10-Y German Bunds.



3. **Analyzing Time Series: Fundamental Notions**

To illustrate empirically all the basics we consider a specific member of the ARMA family, the AR(1) model with drift,

$$x_t = \rho_0 + \rho_1 x_{t-1} + \epsilon_t \qquad \epsilon_t \sim n.i.d.\left(0, \sigma_\epsilon^2\right) \tag{2}$$

Given that each realization of our stochastic process is a random variable, the first relevant fundamental is the density of each observation. In particular, we distinguish between conditional and unconditional densities. Having introduced these two concepts, we define and discuss stationarity, generalize the form of our specific member to the whole family of ARMA models, and conclude this section with a discussion of deterministic and stochastic trends and de-trending methods.

3.1. *Conditional and Unconditional Densities*

We distinguish between conditional and unconditional densities of a time-series. The unconditional density is obtained under the hypothesis that no observation on the time-series is available, while conditional densities are based on the observation of some realization of random variables. In the case of time series processes, we derive unconditional the density by ideally placing the observer at time zero, i.e., the moment preceding the observation of any realization of the time-series. At that moment the information set contains only the knowledge of the process generating the observations. As observations become available, we can compute conditional densities by combining the information on the process with that of the observed data. To illustrate the difference between conditional and unconditional densities consider first our AR(1) model. Because Gaussian distributions are summarized by their moments, we start by looking at conditional and unconditional moments.

The moments of the density of x_t conditional upon x_{t-1} are immediately obtained from (2):

$$\begin{aligned} E\left(x_t|x_{t-1}\right) &= E\left(\rho_0+\rho_1 x_{t-1}+\epsilon_t|x_{t-1}\right)=\rho_0+\rho_1 x_{t-1}, \\ Var\left(x_t|x_{t-1}\right) &= Var\left(\rho_0+\rho_1 x_{t-1}+\epsilon_t|x_{t-1}\right)=\sigma_\epsilon^2, \\ Cov\left[\left(x_t|x_{t-1}\right),\left(x_{t-j}|x_{t-1}\right)\right] &= Cov\left[\epsilon_t,\epsilon_{t-j}\right]=0 \quad \text{for each } j. \end{aligned}$$

To derive the moments of the density of x_t conditional upon x_{t-2}, we need to substitute x_{t-2} from (2) for x_{t-1}:

$$\begin{aligned} E\left(x_t|x_{t-2}\right) &= E\left(\rho_0+\rho_1(\rho_0+\rho_1 x_{t-2}+\epsilon_{t-1})+\epsilon_t|x_{t-2}\right) \\ &= \rho_0+\rho_0\rho_1+\rho_1^2 x_{t-2}, \\ Var\left(x_t|x_{t-2}\right) &= Var\left(\rho_0+\rho_1(\rho_0+\rho_1 x_{t-2}+\epsilon_{t-1})+\epsilon_t|x_{t-2}\right) \\ &= Var\left(\rho_1\epsilon_{t-1}+\epsilon_t|x_{t-2}\right)=\sigma_\epsilon^2\left(1+\rho_1^2\right), \\ Cov\left[\left(x_t|x_{t-2}\right),\left(x_{t-j}|x_{t-j-2}\right)\right] &= Cov\left[\rho_0+\rho_1(\rho_0+\rho_1 x_{t-2}+\epsilon_{t-1})+\epsilon_t,\rho_0+\rho_1 x_{t-2}+\epsilon_{t-1}\right] \\ &= Cov\left[\rho_1\epsilon_{t-1},\epsilon_{t-1}\right]+Cov\left[\epsilon_t,\epsilon_{t-1}\right]=\rho_1\sigma_\epsilon^2, \quad \text{for } j=1, \\ Cov\left[\left(x_t|x_{t-2}\right),\left(x_{t-j}|x_{t-j-2}\right)\right] &= 0, \quad \text{for } j>1. \end{aligned}$$

Finally, unconditional moments are derived by substituting recursively from (2) to express x_t as a function of information available at time 0, the moment before we start observing realizations of our process.

$$\begin{aligned} E(x_t) &= \rho_0\left(1+\rho_1+\rho_1^2+...+\rho_1^{t-1}\right)+\rho_1^t x_0, \\ Var(x_t) &= \sigma_\epsilon^2\left(1+\rho_1^2+\rho_1^4+...+\rho_1^{2t-2}\right), \\ \gamma(j) &= Cov(x_t, x_{t-j}) = \rho_1^j Var(x_t), \\ \rho(j) &= \frac{Cov(x_t, x_{t-j})}{\sqrt{Var(x_t)\,Var(x_{t-j})}} = \frac{\rho_1^j Var(x_t)}{\sqrt{Var(x_t)\,Var(x_{t-j})}}. \end{aligned}$$

Note that $\gamma(j)$ and $\rho(j)$ are functions of j, known respectively as the autocovariance function and the autocorrelation function.

3.2. *Stationarity*

A stochastic process is strictly stationary if its joint density function does not depend on time. More formally, a stochastic process is stationary if, for each $j_1, j_2, ..., j_n$, the joint distribution,

$$f(x_t, x_{t+j_1}, x_{t+j_2}, x_{t+j_n}),$$

does not depend on t.

A stochastic process is covariance stationary if its two first unconditional moments do not depend on time, i.e. if the following relations are satisfied for each h, i, j:

$$\begin{aligned} E(x_t) &= E(x_{t+h}) = \mu, \\ E\left(x_t^2\right) &= E\left(x_{t+h}^2\right) = \mu_2, \\ E(x_{t+i}x_{t+j}) &= \mu_{ij}. \end{aligned}$$

In the case of our AR(1) process, the condition for stationarity is $|\rho_1| < 1$. When such a condition is satisfied, we have:

$$\begin{aligned} E(x_t) &= E(x_{t+h}) = \frac{\rho_0}{1-\rho_1}, \\ Var(x_t) &= Var(x_{t+h}) = \frac{\sigma_\epsilon^2}{1-\rho_1^2}, \\ Cov(x_t, x_{t-j}) &= \rho_1^j Var(x_t). \end{aligned}$$

On the one hand, these results derive from the basic fact that for a geometric series with term $|\rho_1| < 1$, we know that

$$\sum_{t=0}^{\infty} \rho_1^t = \frac{1}{1-\rho_1},$$

so that

$$\begin{aligned}
\lim_{t\to\infty} E(x_t) &= \rho_0 \sum_{t=0}^{\infty} \rho_1^t + \lim_{t\to\infty} \rho_1^t x_0 = \frac{\rho_0}{1-\rho_1} \\
\lim_{t\to\infty} Var(x_t) &= \sigma_\epsilon^2 \sum_{t=0}^{\infty} \rho_1^{2t} = \sigma_\epsilon^2 \sum_{t=0}^{\infty} (\rho_1^2)^t = \frac{\sigma_\epsilon^2}{1-\rho_1^2} \\
\lim_{t\to\infty} \gamma(j) &= \lim_{t\to\infty} Cov(x_t, x_{t-j}) = \rho_1^j Var(x_t).
\end{aligned}$$

On the other hand, when $|\rho_1| = 1$, the process is obviously non-stationary:

$$\begin{aligned}
E(x_t) &= \rho_0 t + x_0, \\
Var(x_t) &= \sigma_\epsilon^2 t, \\
Cov(x_t, x_{t-j}) &= \sigma_\epsilon^2 (t-j).
\end{aligned}$$

Stationarity is a very important property for forecasting: under stationarity one can legitimately learn from the past to predict the future realizations of a time series. If stationarity is not satisfied the density of the observations estimated from past data is not going to be helpful to predict future observations.

3.3. *ARMA Processes*

Before introducing the basics of time series analysis, we have claimed that white noise processes are too simplistic to describe economic time series and that one may hope to obtain a closer fit to the data by considering combinations of white noises. We have then introduced ARMA models and discussed the fundamental notions useful to understand their properties. However, we have not yet shown that ARMA models represent combinations of white-noise processes. We show this by considering a time series as a polynomial distributed lag of a white-noise process:

$$\begin{aligned}
x_t &= \epsilon_t + b_1\epsilon_{t-1} + b_2\epsilon_{t-2} + ... + b_n\epsilon_{t-n} \\
&= \left(1 + b_1 L + b_2 L^2 + ... + b_n L^n\right)\epsilon_t \\
&= b(L)\epsilon_t,
\end{aligned}$$

where L is the lag operator. The Wold decomposition theorem, which states that any stationary stochastic process can be expressed as the sum of a deterministic component and a stochastic moving-average component, warrants generality of our representation. However, in order to describe successfully a time-series, a very high order in the polynomial $b(L)$ is required. This feature can be problematic for estimation, given the usual limitations for sample sizes. This potential problem is resolved, if the polynomial $b(L)$ can be represented

as the ratio of two polynomials of lower order:

$$\begin{aligned} x_t &= b(L)\,\epsilon_t = \frac{a(L)}{c(L)}\epsilon_t, \\ c(L)\,x_t &= a(L)\,\epsilon_t. \end{aligned} \tag{3}$$

Equation (3) is an ARMA process. The process is stationary when the roots of $c(L)$ lie outside the unit circle. The MA component is invertible when the roots of $a(L)$ lie outside the unit circle. Invertibility of the MA component allows it to be represented as an autoregressive process.

To illustrate how the autocovariance and the autocorrelation functions of an ARMA model are derived, we consider the simplest case, the ARMA(1,1) process:

$$\begin{aligned} x_t &= c_1 x_{t-1} + \epsilon_t + a_1\epsilon_{t-1}, \\ (1 - c_1 L)\,x_t &= (1 + a_1 L)\,\epsilon_t. \end{aligned} \tag{4}$$

Equation (4) is equivalent to:

$$\begin{aligned} x_t &= \frac{1 + a_1 L}{1 - c_1 L}\epsilon_t \\ &= (1 + a_1 L)\left(1 + c_1 L + (c_1 L)^2 + ...\right)\epsilon_t \\ &= \left[1 + (a_1 + c_1)\,L + c_1\,(a_1 + c_1)\,L^2 + c_1^2\,(a_1 + c_1)\,L^3 + ...\right]\epsilon_t. \end{aligned}$$

Because one can see that

$$\sum_{j=0}^{\infty}(c_1 L)^j = \frac{1}{1 - c_1 L},$$

we have that:

$$\begin{aligned} Var(x_t) &= \left[1 + (a_1 + c_1)^2 + c_1^2\,(a_1 + c_1)^2 + ...\right]\sigma_\epsilon^2 \\ &= \left[1 + \frac{(a_1 + c_1)^2}{1 - c_1^2}\right]\sigma_\epsilon^2, \\ Cov(x_t, x_{t-1}) &= \left[(a_1 + c_1) + c_1\,(a_1 + c_1) + c_1^2\,(a_1 + c_1) + ...\right]\sigma_\epsilon^2 \\ &= \left[(a_1 + c_1) + \frac{c_1\,(a_1 + c_1)^2}{1 - c_1^2}\right]\sigma_\epsilon^2. \end{aligned}$$

Hence the first-order autocorrelation is,

$$\rho(1) = \frac{Cov(x_t, x_{t-1})}{Var(x_t)} = \frac{(1 + a_1 c_1)\,(a_1 + c_1)}{1 + a_1^2 + 2a_1 c_1}.$$

Successive values for $\rho(j)$ are obtained from the recurrent relation $\rho(j) = c_1\rho(j-1)$ for $j \geq 2$.

In general it is useful to know how an infinite moving average representation of an ARMA model can be derived. The handiest way to invert finite order polynomials is by matching their representations. For example, suppose $c(L)\,x_t = a(L)\,\epsilon_t$ and you want to find $x_t = d(L)\,\epsilon_t$. Parameters in $d(L)$ are most easily found by writing $c(L)\,d(L) = a(L)$ and by matching terms in $L^{j.}$

For instance, consider $a(L) = 1 + a_1 L$, $c(L) = 1 + c_1 L$. Multiplying out $d(L)$ we have:

$$(1 + c_1 L)\left(1 + d_1 L + d_2 L^2 + ...d_n L^n\right) = 1 + a_1 L$$

Matching the powers of L, we obtain:

$$\begin{aligned} d_1 &= a_1 - c_1 \\ c_1 d_1 + d_2 &= 0 \\ c_1 d_2 + d_3 &= 0 \\ c_1 d_{n-1} + d_n &= 0 \end{aligned}$$

or

$$x_t = \epsilon_t + (a_1 - c_1)\,\epsilon_{t-1} - c_1\,(a_1 - c_1)\,\epsilon_{t-2} + ...\,(-c_1)^{n-1}\,(a_1 - c_1)\,\epsilon_{t-n}.$$

4. **Persistence: Monte-Carlo Experiments**

We have seen hat the longer the horizon at which returns are defined, the higher their persistence. Persistence of time-series threatens (complicates) one of the crucial properties for implementing valid estimation and inference in the linear model. In the following, we use Monte Carlo experiments to discuss these issues. Consider the framework defined by the linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

As it is well known, the following property is required to implement valid estimation and inference

$$E(\boldsymbol{\epsilon}|\mathbf{X}) = \mathbf{0}. \tag{5}$$

Hypothesis (5) implies that

$$E(\boldsymbol{\epsilon}_i|\mathbf{x}_1, ...\mathbf{x}_i, ..., \mathbf{x}_n) = 0, \qquad i = 1, ..., n.$$

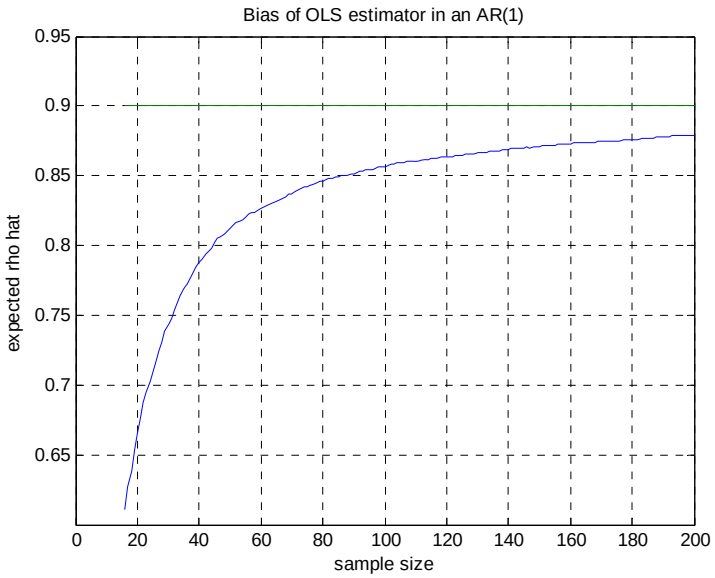
Think now of an AR(1) model for a generic time series process, y_t:

$$y_t = a_0 + a_1 y_{t-1} + \epsilon_t.$$

Clearly, if $a_1 \neq 0$, then, although it is still the case that $E(\epsilon_t|y_{t-1}) = 0$, $E(\epsilon_{t-1}|y_{t-1}) \neq 0$ and (5) breaks down. How serious is the problem? To assess intuitively the consequences of

persistence, we construct a small Monte-Carlo simulation on the small-sample properties of the OLS estimator of the parameters in an AR(1) process.

A set of random numbers is generated from a given distribution (here a normally independent white-noise disturbance) for a sample size of interest (200 observations). Starting from an arbitrary initial value (given by the unconditional mean of the process), we simulate data from an AR(1) process with a persistence parameter $a_1 = 0.9$. On the generated sample of data we estimate the unknown parameter, a_1. If we repeat this process only once, then the set of random numbers drawn is just one possible outcome and the estimates are dependent on one exact sequence of simulated white noise residuals. To overcome this problem, the simulation procedure is replicated many times to construct the process of interest (in our case, an AR(1) process). For each replication we obtain a set of estimates, and compute averages across replications of the estimated parameters, to assess these averages against the known, true values. For each simulated sample estimates can be computed on the sample or on different sub-samples. Figure 3 reports average estimates across 1,000 replications for a initial sample size of 15 that is increased progressively to the full sample of 200 observations by adding one observation at the time.



From this figure, we note that the estimate of a_1 is heavily biased in small samples, but the bias decreases as the sample gets larger, and disappears eventually. One can show analytically that the average of the OLS estimate of a_1 is $a_1\,(1 - 2/T)$. This is an interesting result, which can be generalized. For stationary time series, any high serial correlation, which destroys the orthogonality between residuals and regressors in the linear regression model, tends to disappear as the distance between observations increases. Therefore, as we shall show in the next section, the standard finite sample results can be extended to time series

only by considering large samples. In practice, this is a very useful result: all the standard econometric tools can be applied to time series, provided that large samples are available. The more persistent are the time-series of interest the larger is the sample size needed. The relevant results on the properties of the estimators are obtained by introducing asymptotic theory, which we now proceed to do.

5. **Asymptotic Theory**

Strictly stationary time series processes feature time invariant distributions. As a consequence, the effect of any specific innovation disappears as time elapses. We show in this section that the intuition of the simple Monte Carlo simulation performed above can be extended and formal asymptotic theory used to perform valid estimation and inference when modelling *stationary* time series.[1]

5.1. *Basic elements of asymptotic theory*

Consider a sequence $\{X_T\}$ of random variables with the associated sequence of distribution functions $\{F_T\} = F_1$, ..., F_T, we give the following alternative definitions of convergence for X_T.

5.1.1. **Convergence in distribution**

Given a random variable X with distribution function F, X_T converges in distribution to X if the following equality is satisfied:

$$\lim_{T\to\infty} P\{X_T < x_0\} = P\{X < x_0\},$$

for all x_0, where the function $F(x)$ is continuous.

5.1.2. **Convergence in probability**

Given a random variable X with distribution function F, X_T converges in probability to X if, for each $\epsilon > 0$, the following relation holds:

$$\lim_{T\to\infty} P\{|X_T - X| < \epsilon\} = 1.$$

Note that convergence in probability implies convergence in distribution. We also write that $p\lim X_T = X$.

[1] For a formal treatment of these topics, see White (1984).

5.1.3. **Central limit theorem (Lindberg−Levy)**

This is an important result that relies on convergence in distribution. Given a sequence $\{X_T\}$ of identically and independently distributed random variables with mean μ and finite variance σ^2, define

$$\begin{aligned} \bar{X} &= \frac{1}{T}\sum_{i=1}^{T} X_i, \\ Z &= \sqrt{T}\frac{\left(\bar{X}-\mu\right)}{\sigma}. \end{aligned}$$

Then Z converges in distribution to a standard normal, $N(0,1)$, $Z \xrightarrow{D} N(0,1)$.

5.1.4. **Slutsky's Theorem**

For any random variable X_T, such that plim $X_T = a$, where a is a constant, given a function $g(\cdot)$ continuous at a, $p\lim\, g\left(X_T\right) = g\left(a\right)$.

5.1.5. **Cramer's Theorem**

Given two random variables X_T and Y_T, such that Y_T converges in distribution to Y and X_T converges in probability to a constant a, the following relationships hold:

$X_T + Y_T$ converges in distribution to $(a + Y)$;

Y_T/a_T converges in distribution to (Y/a);

$Y_T \cdot a_T$ converges in distribution to $(Y \cdot a)$.

Note that all theorems introduced so far extend to vectors of random variables, random vectors.

5.1.6. **Mann-Wald Theorem**

Consider a $k \times 1$ vector $\mathbf{z}_t$ of random variables which satisfies the following property:

$$p\lim\; T^{-1}\sum_{t=1}^{T}\mathbf{z}_t\mathbf{z}_t' = \mathbf{Q},$$

where $\mathbf{Q}$ is a positive definite matrix. Consider also a sequence ϵ_t of random variables, identically and independently distributed with zero mean and finite variance, σ^2, for which finite moments of each order are defined. If $E\left(\mathbf{z}_t\epsilon_t\right) = \mathbf{0}$, then

$$p\lim\; T^{-1}\sum_{t=1}^{T}\mathbf{z}_t\epsilon_t = \mathbf{0}, \qquad \sqrt{\frac{1}{T}}\sum_{t=1}^{T}\mathbf{z}_t\epsilon_t \xrightarrow{D} N\left(0, \sigma^2\mathbf{Q}\right).$$

5.2. *Application to models of stationary time series*

Consider the following time series AR(1) model:

$$y_t = \gamma y_{t-1} + \epsilon_t,$$

where y_t is a stationary variable and $|\gamma| < 1$. As already shown, $E(y_t \epsilon_{t-i}) \neq 0$ and the OLS estimator of γ is biased. By applying the Mann-Wald theorem, we can derive the asymptotic distribution of the OLS estimator of γ, $\widehat{\gamma}$:

$$\widehat{\gamma} \xrightarrow{D} N\left[\gamma, \sigma^2 \mathbf{Q}^{-1}\right],$$

where $\mathbf{Q}$ is the asymptotic variance of the regressor that (given stationarity) coincides with the asymptotic variance of the dependent variable. Therefore all the finite sample results available for independent cross sectional observations can be extended to stationary time series just by considering large-sample theory.

6. **Estimation of ARMA models. The Maximum Likelihood Method**

Given the results derived in the previous section the problem of estimation of stationary AR models is easily solved by applying standard methods to linear specification where the regressors are lags of the dependent variables. However, the linear model is no longer applicable when moving average of white noise errors are introduced in the specification to be estimated. A more general method, capable of dealing with non-linearities, needs to be introduced to this end : the Maximum Likelihood method.

Within this framework estimates of the parameters of interests are obtained by maximizing the likelihood function. The likelihood function is the joint probability distribution of the data, that depends on the observations on the time series of interest and an the unknown parameters. The likelihood function can be expressed as $\mathcal{L}(\mathbf{X}_T^1 | \mathbf{X}_0, \boldsymbol{\theta})$. It is defined on the parameter space Θ**,** given the observation of the observed sample $\mathbf{X}_T^1$ and of a set of initial conditions $\mathbf{X}_0$. One can interpret such initial conditions as the pre-sample observations on the relevant variables (which are usually unavailable). Once a sample of observations is available for the time series of interest the likelihood can be treated as a function of the unknown coefficients. The maximum likelihood estimator (MLE) is then obtained by choosing the of value of the unknown parameters that maximize the likelihood function. In practice, the MLE selects the value of parameters to maximize the probability of drawing the data that have been effectively observed.

6.1. *MLE of an MA process*

To illustrate the practice of MLE estimator consider the case of a financial returns that r_{t+1} follows an MA(1) process:

$$r_{t+1} = \theta_0 + \varepsilon_{t+1} + \theta_1 \varepsilon_t \qquad \varepsilon_t \sim NID\left(0, \sigma^2\right).$$

In this case the unknown parameters to be estimated are θ_0, θ_1, and σ^2. To derive MLE estimates first define the time series of residuals:

$$\begin{aligned} \varepsilon_{t+1} &= r_{t+1} - \theta_0 - \theta_1 \varepsilon_t \\ \varepsilon_0 &= 0. \end{aligned}$$

Given the distributional assumption on ε_{t+1}, we have:

$$f\left(\varepsilon_{t+1}\right) = \frac{1}{\left(2\pi\sigma_\varepsilon^2\right)^{1/2}} \exp\left(-\frac{\varepsilon_{t+1}^2}{2\sigma_\varepsilon^2}\right).$$

This expression is the distribution of a single observation, while the likelihood function is the joint distribution of the entire sample. If the ε_{t+1} are independent over time, then the likelihood function can be written as follows:

$$f\left(\varepsilon_1, \varepsilon_2, ...\varepsilon_{t+1}\right) = \prod_{i=1}^{T} f\left(\varepsilon_i\right) = \prod_{i=1}^{T} \frac{1}{\left(2\pi\sigma_\varepsilon^2\right)^{1/2}} \exp\left(-\frac{\varepsilon_i^2}{2\sigma_\varepsilon^2}\right).$$

At this point, the MLE chooses $\theta_0, \theta_1, \sigma_\varepsilon^2$ to maximize the probability that the estimated model has generated the observed data. The optimum is not always found analytically, iterative search is the standard method.

6.2. *MLE of an AR process*

The derivation of the likelihood of an autoregressive process offer the opportunity of thinking further on persistence. Consider a vector $\mathbf{x}_t$ containing observations on time-series variables at time t. A sample of T time series observations on all the variables is represented as:

$$\mathbf{X}_T^1 = \begin{bmatrix} \mathbf{x}_1 \\ . \\ . \\ . \\ \mathbf{x}_T \end{bmatrix}.$$

In case of independent observations the likelihood function can be written as the product of the density functions for each observation. However, this is not the relevant case for time

series, as time series observations are in general sequentially correlated. In the case of time series, the sample density is constructed using the concept of sequential conditioning. The likelihood function, conditioned with respect to initial conditions, can always be written as the product of a marginal density and a conditional density:

$$D\left(\mathbf{X}_T^1|\mathbf{X}_0,\boldsymbol{\theta}\right) = D\left(\mathbf{x}_1|\mathbf{X}_0,\boldsymbol{\theta}\right) D\left(\mathbf{X}_T^2|\mathbf{X}_1,\boldsymbol{\theta}\right),$$

where $D\left(\mathbf{x}_t|\mathbf{X}_0,\boldsymbol{\theta}\right)$ is the density of $\mathbf{x}_t$. Obviously,

$$D\left(\mathbf{X}_T^2|\mathbf{X}_0,\boldsymbol{\theta}\right) = D\left(\mathbf{x}_2|\mathbf{X}_1,\boldsymbol{\theta}\right) D\left(\mathbf{X}_T^3|\mathbf{X}_2,\boldsymbol{\theta}\right),$$

and, by recursive substitution:

$$\mathcal{L}\left(\mathbf{X}_T^1|\mathbf{X}_0,\boldsymbol{\theta}\right) = \prod_{t=1}^{T} D\left(\mathbf{x}_t|\mathbf{X}_{t-1},\boldsymbol{\theta}\right).$$

Having obtained $\mathcal{L}\left(\mathbf{X}_T^1|\mathbf{X}_0,\boldsymbol{\theta}\right)$, we can in theory derive $\mathcal{L}\left(\mathbf{X}_T^1,\boldsymbol{\theta}\right)$ by integrating with respect to X_0 the density conditional on pre-sample observations. In practice this could be intractable analytically, as $D\left(X_0\right)$ is not known. The hypothesis of stationarity becomes crucial at this stage, as stationarity restricts the memory of time-series and limits the effects of pre-sample observations to the first observations in the sample. This is why, in the case of stationary processes, one can simply ignore initial conditions. Clearly, the larger the sample, the better, as the weight of lost information becomes smaller. Moreover, note that even by omitting initial conditions, we have:

$$\mathcal{L}\left(\mathbf{X}_T^1|\mathbf{X}_0,\boldsymbol{\theta}\right) = D\left(\mathbf{x}_1|\mathbf{X}_0,\boldsymbol{\theta}\right) \prod_{t=2}^{T} D\left(\mathbf{x}_t|\mathbf{X}_{t-1},\boldsymbol{\theta}\right).$$

Therefore, the likelihood function is separated in the product on $T-1$ conditional distributions and one unconditional distribution. In the case of non-stationarity, the unconditional distribution is undefined. On the other hand, in the case of stationarity, the data generating process (DGP) is completely described by the conditional density function $D\left(\mathbf{x}_t|\mathbf{X}_{t-1},\boldsymbol{\theta}\right)$. For a sufficiently large-sample, the density of the first observation can be dropped and the approximated likelihood can be maximized rather than the exact likelihood, in the so called quasi-ML method:

$$\mathcal{L}\left(\mathbf{X}_T^1 \mid \mathbf{X}_0,\boldsymbol{\theta}\right) \simeq \prod_{t=2}^{T} D\left(\mathbf{x}_t \mid \mathbf{X}_{t-1},\boldsymbol{\theta}\right).$$

Consider now the case of the univariate first-order autoregressive process,

$$X_t|\mathbf{X}_{t-1} \sim N\left(\lambda x_{t-1}, \sigma^2\right). \tag{6}$$

Hence

$$\mathcal{L}\left(\mathbf{X}_T^1|\lambda,\sigma^2\right) = D\left(X_1|\lambda,\sigma^2\right)\prod_{t=2}^{T} D\left(X_t|\mathbf{X}_{t-1},\lambda,\sigma^2\right). \tag{7}$$

From (7), the likelihood function clearly involves $T-1$ conditional densities and one unconditional density. The conditional densities are given by (6). The unconditional density can be derived only in the case of stationarity:

$$x_t = \lambda x_{t-1} + u_t \qquad u_t \sim NID\left(0,\sigma^2\right).$$

We can obtain by recursive substitution:

$$x_t = u_t + \lambda u_{t-1} + ... + \lambda^{n-1}u_1 + \lambda^n x_0.$$

Only if $|\lambda| < 1$, the effect of the initial condition disappears and we can write the unconditional density of x_t as:

$$D\left(x_t \mid \lambda,\sigma^2\right) = N\left(0, \frac{\sigma^2}{1-\lambda^2}\right).$$

Under stationarity we can derive the exact likelihood function:

$$\begin{aligned}\mathcal{L}\left(\mathbf{X}_T^1 \mid \lambda,\sigma^2\right) &= (2\pi)^{-\frac{T}{2}}\sigma^{-T}\left(1-\lambda^2\right)^{\frac{1}{2}} \\ &\quad \exp\left[-\frac{1}{2\sigma^2}\left(\left(1-\lambda^2\right)x_1^2 + \sum_{t=2}^{T}\left(x_t-\lambda x_{t-1}\right)^2\right)\right],\end{aligned} \tag{8}$$

and estimates of the parameters of interest are derived by maximizing this function. Note that $\hat{\lambda}$ cannot be derived analytically, using the exact likelihood function. Optimization is achieved in practice by conditioning the likelihood and operating a grid search. Note also that, when the first observation is dropped and the approximate likelihood function is considered, one can show that the MLE of λ coincides with its OLS estimate.

7. **Putting ARMA models at work**

Putting ARMA models at work is a multi-step process that can be summarized in the following stages (the so-called *Box-Jenkins approach*):

- PRE-WHITENING: make sure that the time series is stationary, in other words make sure that the model at hand is ARMA and not ARIMA, where the I in the acronym would otherwise given an indication in favor of non-stationarity. In practice, this is commonly achieved for simple univariate time series via differencing, i.e., by considering $\Delta X_t = (1-L)X_t = X_t - X_{t-1}$ instead of X_t. As we shall see later on, there are alternative, and more interesting, ways of achieving stationarity in multivariate time series analysis.

- MODEL SELECTION: the aim of a model selection process is to look for the best ARMA specification. This is often not a easy task because different specification may perform very closely. *Information criteria* are a useful tool to this end. They are model selection criteria based on penalized versions of the maximized log-likelihood function. They may be used to select p and q in an ARMA(p,q) model. The Akaike's information criteria (AIC) and the Schwarz Bayesian Criterion (SBIC, sometimes simply referred to as BIC) are the most commonly used criteria and are defined as follows:

$$\begin{aligned} AIC &\equiv -2\ln(\mathcal{L}) + 2(p+q) \\ BIC &\equiv -2\ln(\mathcal{L}) + \ln(n)(p+q), \end{aligned}$$

  where p and q are the length of the AR and MA polynomials and n is the total number of observations. The models are ranked from the best to the worst in such a way that best model is the one that *minimizes* any chosen criterion. Note that the criterion is minimized when the likelihood function penalized for the lack of parsimony (i.e., when over-parameterization occurs) is maximized, thanks to the presence of a minus sign in front of $\ln(\mathcal{L})$.

- MODEL CHECKING: residual tests. Make sure that residuals are not autocorrelated and check whether their distribution is normal, also ex-post evaluation technique based on the root mean squared forecast error (RMSE) and mean absolute forecast error (MAE) are often employed.

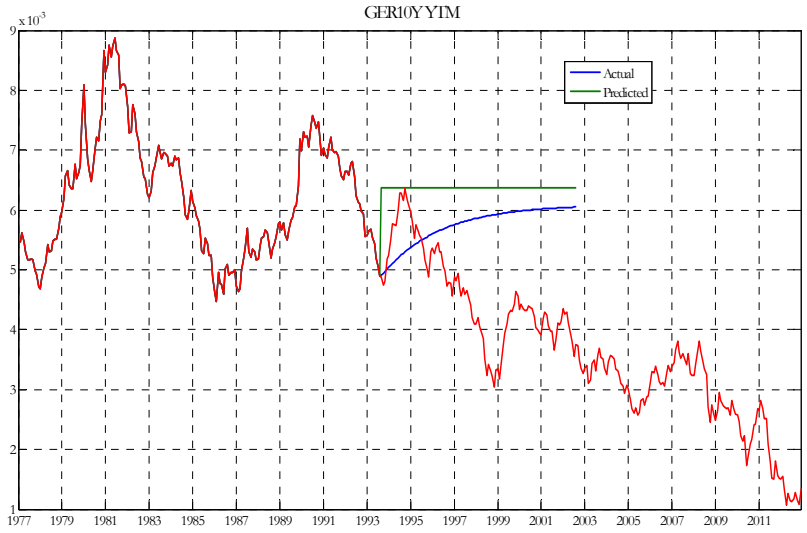
- FORECASTING, the selected model is typically simulated forward after estimation of the estimation of parameters to produce forecasts for the variable of interests at the relevant horizon.

7.1. *An Illustration*

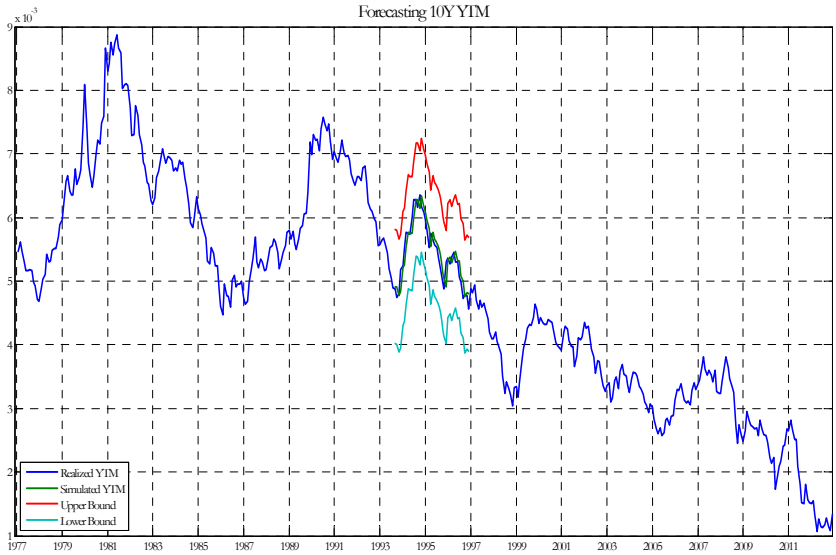
To illustrate how ARMA models may be put at work, consider the case of forecasting the yield to maturity (henceforth, YTM) of 10-year German bonds (y_t^{10}) from January 1994 (i.e., 1994:01) onward, given the availability of data for the period 1978-1993. Estimation is performed in MATLAB by using the appropriate specification in the GARCH procedure (see sample codes below). ML estimation delivers the following result:

$$y_t^{10} = \underset{(0.0004137)}{0.0001423} + \underset{(0.0610)}{0.9764} y_{t-1}^{10} + \underset{(3.2616e-08)}{\sqrt{0.00000007}}\hat{u}_t.$$

Figure 4 illustrates the results of using an AR(1) model to forecast at a horizon up to nine years form 1994:01 onward and compares them with forecasts based on the CER model.



Note that a very different picture is obtained in Figure 5 to illustrate the one-step ahead forecasts recursively derived over the same forecasting horizon:



8. **Trends**

Time series of long-horizon returns—i.e., returns computed as sums of higher frequency returns—besides being persistent, often feature trends. The presence of trends is potentially problematic as it induces non-stationarity. We have seen in the previous section that the

estimated autoregressive parameter for the AR(1) process fitted on German 10-year yields to maturity is close to one (0.9764). Knowing that persistence introduces a small sample downward bias in these estimates, it is worth considering the case in which the autoregressive parameter tends to one and we face instead a *random walk process with drift*:

$$x_t = a_0 + x_{t-1} + \epsilon_t \qquad \epsilon_t \sim NID\left(0, \sigma_\epsilon^2\right).$$

Recursive substitution yields

$$x_t = x_0 + a_0 t + \sum_{i=0}^{t-1} \epsilon_{t-i}, \tag{9}$$

which shows immediately that the series is non-stationary in that the unconditional mean ($E\left(x_t\right) = x_0 + a_0 t$) is a function of time and that a non-stationary series contains both a deterministic ($a_0 t$) and a stochastic trend, here

$$\sum_{i=0}^{t-1} \epsilon_{t-i}.$$

An immediate way to make a non-stationary series stationary is by differencing it:

$$\Delta x_t \equiv x_t - x_{t-1} = (1 - L)\, x_t = a_0 + \epsilon_t.$$

In general, if a time-series needs to be differenced d times to become stationary, then it is said to be *integrated of order* d or $I(d)$. Our random walk is I(1). When the dth difference of a time-series $\{x_t\}$, $\Delta^d x_t$,

$$\Delta^d x_t \equiv (1 - L)^d\, x_t$$

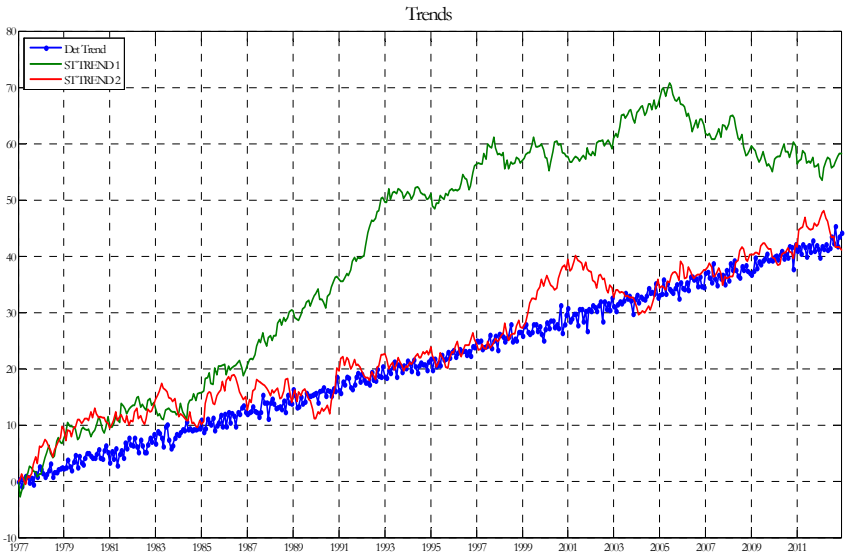
can be represented by an ARMA(p, q) model, we say that x_t is an integrated moving-average process of order p, d, q and denote this as an ARIMA(p, d, q).

It is useful to compare the behavior of an integrated process with that of a trend stationary process. Trend stationary processes simply feature a deterministic trend:

$$z_t = \alpha + \beta t + \epsilon_t. \tag{10}$$

The process for z_t is non-stationary, but non-stationarity is removed simply by regressing z_t on the deterministic trend. Unlike the case of (9), for integrated processes like (9) the removal of the deterministic trend does not deliver a stationary time-series. Deterministic trends have no memory while integrated variables have an infinite memory. Both integrated variables and deterministically trending ones exhibit systematic variations, but in the latter case the variation is predictable, whereas in the former case, it is not. This point is easily illustrated in the Figure that follows (6), where we report three different series for a sample of 200 observations each: one deterministic trend with a slope of 0.1; two random walks,

with a drift of 0.1 The only difference between the two stochastic trend series STREND1 and STREND2 is in the realizations of the error terms, which are different drawings from the same serially independent standard normal distribution. Equivalently, STREND1 and STREND2 are two different, alternative simulated paths. Visibly, while STREND2 shows a behavior that is not remarkably different from a deterministically trending series, STREND1 shows a radically different dynamics.



8.1. *Univariate time series decompositions*

The general solution proposed to the problem introduced in the previous section is the search for a stationary representation of non-stationary time series. This can be done both in univariate and multivariate frameworks. Here, we discuss briefly the available methodologies in a univariate framework, to move swiftly to decompositions in a multivariate framework, which are more interesting to our purposes, in the next chapter.

Beveridge and Nelson (1981) provide an elegant way of decomposing a non-stationary time series into a permanent and a temporary (cyclical) component by applying ARIMA methods. For any non-stationary time-series, x_t, integrated of the first order, the Wold decomposition theorem can be applied to its first difference (to make it stationary), to deliver the following representation:

$$\Delta x_t = \mu + C(L)\,\epsilon_t \qquad \epsilon_t \sim NID\left(0, \sigma_\epsilon^2\right),$$

where $C(L)$ is a polynomial of order q in the lag operator. Consider now the polynomial $D(L)$, defined as:

$$D(L) = C(L) - C(1). \tag{11}$$

Given that $C(1)$ is just a constant, also $D(L)$ will be of order q. Clearly,

$$D(1) = 0,$$

therefore, 1 is a root of $D(L)$, and

$$D(L) = C^*(L)(1-L), \tag{12}$$

where $C^*(L)$ is a polynomial of order $q-1$. At this point, by equating (11) to (12), we have:

$$C(L) = C^*(L)(1-L) + C(1),$$

so that

$$\begin{aligned} \Delta x_t &= \mu + [C^*(L)(1-L) + C(1)]\epsilon_t \\ &= \mu + C^*(L)\Delta\epsilon_t + C(1)\epsilon_t. \end{aligned} \tag{13}$$

By integrating (13), we finally have:

$$\begin{aligned} x_t &= C^*(L)\epsilon_t + \mu t + C(1) z_t \\ &= C_t + TR_t, \end{aligned}$$

where z_t is a process for which $\Delta z_t = \epsilon_t$. C_t is the cyclical component and $TR_t \equiv \mu t + C(1) z_t$ is a trend component made of a deterministic and of a stochastic trend. Note that the trend component can be represented as:

$$TR_t = TR_{t-1} + \mu + C(1)\epsilon_t.$$

8.1.1. **Beveridge-Nelson decomposition of an IMA(1,1) process**

Consider instead the process:

$$\Delta x_t = \epsilon_t + \theta\epsilon_{t-1}, \quad 0 < \theta < 1.$$

In this case:

$$\begin{aligned} C(L) &= 1 + \theta L, \\ C(1) &= 1 + \theta, \\ C^*(L) &= \frac{C(L) - C(1)}{1-L} = -\theta. \end{aligned}$$

The Beveridge and Nelson decomposition gives the following result:

$$\begin{aligned} x_t &= C_t + TR_t \\ &= -\theta\epsilon_t + (1+\theta) z_t. \end{aligned}$$

8.1.2. **Beveridge-Nelson decomposition of an ARIMA(1,1) process**

Consider the process:

$$\Delta x_t = \rho \Delta x_{t-1} + \epsilon_t + \theta \epsilon_{t-1}.$$

Here:

$$\begin{aligned} C(L) &= \frac{1+\theta L}{1-\rho L}, \\ C(1) &= \frac{1+\theta}{1-\rho}, \\ C^*(L) &= \frac{C(L)-C(1)}{1-L} = -\frac{\theta+\rho}{(1-\rho)(1-\rho L)}, \end{aligned}$$

and the Beveridge and Nelson decomposition yields:

$$x_t = C_t + TR_t = -\frac{\theta+\rho}{(1-\rho)(1-\rho L)}\epsilon_t + \frac{1+\theta}{1-\rho} z_t.$$

8.1.3. **Deriving the Beveridge-Nelson decomposition in practice**

The practical derivation of a Beveridge and Nelson decomposition for any ARIMA process is easily implemented through a methodology suggested by Cuddington and Winters (1987). For any I(1) process, the stochastic trend can be represented as:

$$TR_t = TR_{t-1} + \mu + C(1)\,\epsilon_t. \tag{14}$$

The decomposition can then be applied in the following steps:

1. identify the appropriate ARIMA model and estimate ϵ_t and all the parameters in μ and $C(1)$;

2. given an initial value for TR_0, use (14) to generate the permanent component of the time-series;

3. generate the cyclical component as the difference between the observed value in each period and the permanent component.

The above procedure gives the permanent component up to a constant. If the precision of this procedure is unsatisfactory, one can use further conditions to identify the decomposition more precisely. For example, one can impose the condition that the sample mean of the cyclical component is zero, to pin down the constant in the permanent component.

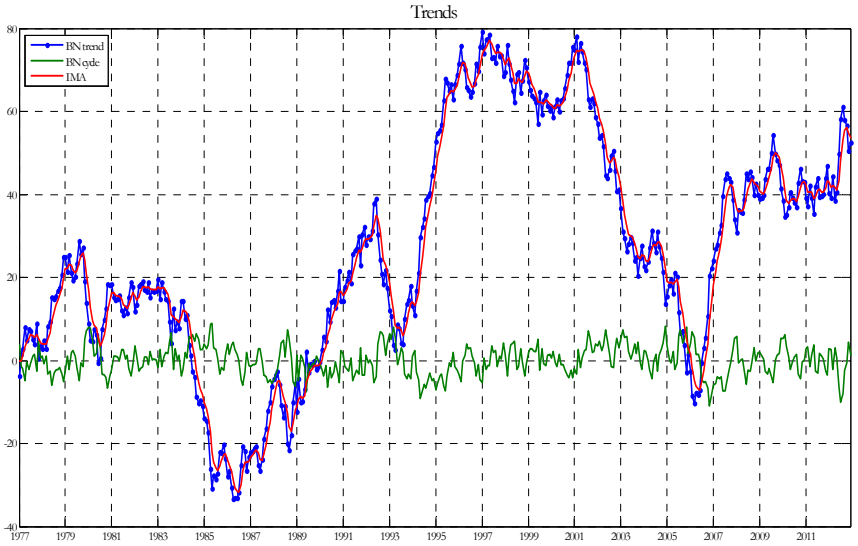
To illustrate how the procedure works in practice, we have simulated the following ARIMA(1,1,1) model for a sample of 200 observations:

$$\Delta x_t = 0.6 \Delta x_{t-1} + \epsilon_t + 0.5 \epsilon_{t-1}.$$

From the previous subsection, we know the exact Beveridge-Nelson decomposition of the series for x_t:

$$\begin{aligned} x_t &= C_t + TR_t = -\frac{1.1}{(1-0.6)(1-0.6L)}\epsilon_t + \frac{1.5}{0.4}z_t, \\ TR_t &= TR_{t-1} + \frac{1.5}{0.4}\epsilon_t. \end{aligned}$$

Given the estimate of ϵ_t, we can therefore generate the permanent and transitory components of x_t. The following Figure (7) reports the series IMA, TR and CYCLE.



This procedure can be followed in practice, after the identification of the relevant ARIMA model and the estimation of the relevant parameters that cannot be imputed from a known Data Generating Process (DGP).

8.1.4. **Assessing the Beveridge-Nelson decomposition**

The properties of the permanent and temporary components of an integrated time series delivered by the Beveridge-Nelson decomposition are worth some comments. Note first that the innovations in the permanent and the transitory components are perfectly negatively correlated; moreover, the trend component is more volatile than the actual time series as the negative correlation between the permanent and the transitory components acts to smooth the original time-series. These results are easily seen for the simplest case we have already discussed. For example, in the case of the IMA(1,1) process, the correlation between the innovations in the permanent and transitory components is minus one and the variance of the innovation in the trend component is $(1.5/0.4)^2 \sigma_\epsilon^2 > \sigma_\epsilon^2$. In general, the variance of the innovations may have an economic interpretation and theory may suggest different

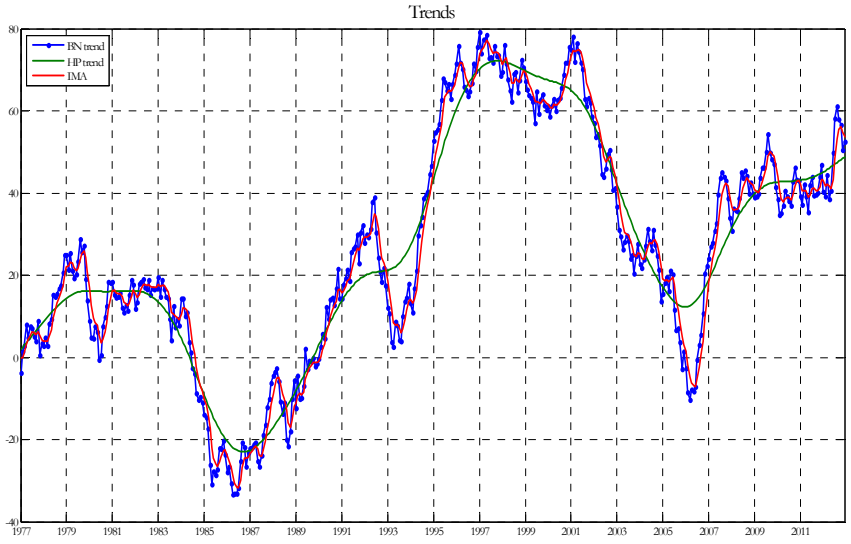
patterns of correlations between innovations in the permanent and transitory components that do differ from a perfectly negative correlation. In fact, alternative restrictions imposed on the correlation between the trend and the cycle components, lead to the identification of different stochastic trends for integrated time series. As a consequence, the Beveridge-Nelson decomposition is not unique. In fact there many different stochastic trends that can be extracted by applying univariate decompositions. To see this point more explicitly we can compare the Beveridge-Nelson trend with the trend extracted using an alternative technique which has been recently very successful in time series analysis, the Hodrick-Prescott filter.

Hodrick and Prescott (1997) proposed their method to analyze post-WWII U.S. business cycles in a working paper circulated in the early 1980s but published only in 1997. The Hodrick-Prescott (HP) filter computes the permanent component TR_t of a series x_t by minimizing the variance of x_t around TR_t, subject to a penalty that constrains the second difference of TR_t to enforce sufficient "smoothness". That is, the HP filter is derived by minimizing the following expression:

$$\sum_{t=1}^{T}(x_t - TR_t)^2 + \lambda \sum_{t=2}^{T-1}\left[(TR_{t+1} - TR_t)^2 - (TR_t - TR_{t-1})^2\right].$$

The penalty parameter λ controls the smoothness of the series, by controlling the ratio of the variance of the cyclical component and the variance of the series. The larger the λ, the smoother the TR_t component is, as in fact it approaches a linear trend. In practical applications, λ is set to 100 for annual data, 1600 for quarterly data, and to 14400 for monthly data.

The following Figure (8) shows the Beveridge-Nelson and the Hodrick-Prescott trends (with $\lambda = 14400$) for the data generated in the previous Section.



Note that the Beveridge-Nelson trend is more volatile than the Hodrick-Prescott trend. It is possible to increase the volatility of the Hodrick-Prescott trend by reducing the parameter λ; however, the Hodrick-Prescott filter reaches at most the volatility of the actual, raw time series, which, as we already know, is smaller than the volatility of the Beveridge-Nelson trend.[2]

The comparison between the Hodrick-Prescott and the Beveridge-Nelson trends reinforces the argument of non-uniqueness of univariate decompositions we have made before. Moreover, we are left with the problem of how to use the filtered series in the empirical analysis and how to relate them to theoretical models. The empirical counterparts of theoretical asset pricing models are multivariate time series. Theoretical models often predict that different time series share the same stochastic trend. The natural question at this point is whether the problem of non-stationarity in time series analysis may be resolved by considering multivariate models. In this context, stationarity is obtained by considering combinations of non-stationary time series sharing the same stochastic trend. If possible, it would justify the identification of trends by relating them to theory. We shall return to these issues in a later chapter.

9. **Asset Allocation with a simple TVER model: the SOP method**

Univariate time series analysis allows us to introduce simple models of time-varying expected returns (TVER). To assess the impact of such models on asset allocation let us consider the sum-of-the-part (SOP) approach to returns predictability discussed in Ferreira and Santa Clara (2012). The idea is to decompose returns in several parts and to implement simple time series analysis to predict each individual component and then generate a time-varying expected (predicted) return by aggregating predictions.

Let's start by noting that total stock market returns in local currency can be re-expressed as follows:

$$1 + R^s_{t+1} \equiv \frac{P_{t+1} + D_{t+1}}{P_t} \tag{15}$$

[2]Unlike the Beveridge and Nelson decomposition, in multivariate applications the HP filter fits and removes a homogeneous trend from all the available time series. This may be desirable only in case the theoretical models indicate that the variables of interest share the same stochastic trend. However, Harvey and Jaeger (1993) showed that the use of such a filter can lead to the identification of spurious cyclical behaviour. Harvey and Jaeger predicate instead a different approach to modelling time series, known as structural time series modelling, which we do not consider in our analysis, as it is less closely related to the analysis of financial returns, but that certainly deserves attention (Harvey and Koopman 1996; Maravall 1995).

$$
\begin{aligned}
&= \frac{P_{t+1}}{P_t} + \frac{D_{t+1}}{P_t} = \frac{P_{t+1}}{P_t} + \frac{D_{t+1}}{P_{t+1}} \frac{P_{t+1}}{P_t} \\
&= \left(1 + \frac{D_{t+1}}{P_{t+1}}\right) \frac{P_{t+1}}{P_t}.
\end{aligned}
$$

Taking logs, we have:

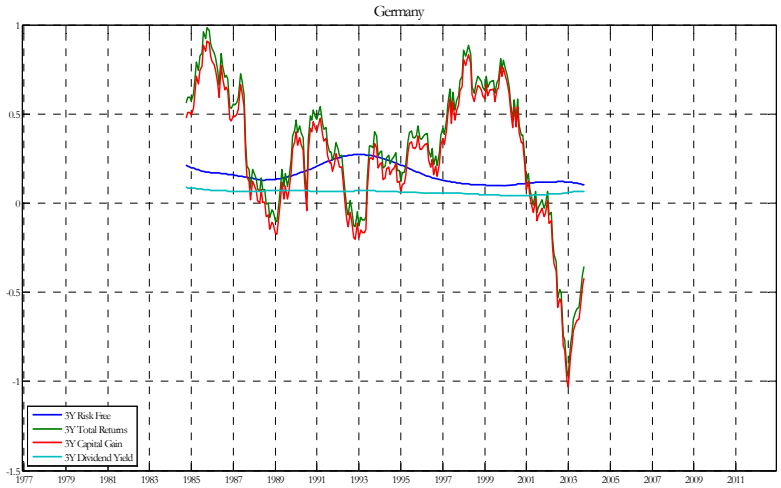
$$
r^s_{t+1} \simeq (p_{t+1} - p_t) + \frac{D_{t+1}}{P_{t+1}},
$$

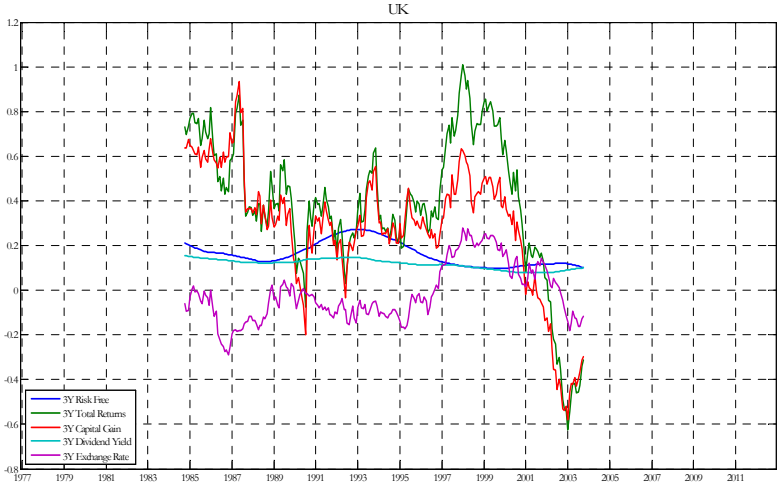
where in general, $\ln(1 + R_{t+1}) \simeq R_{t+1}$ and $\ln R^s_{t+1} = r^s_{t+1}$. Therefore total log returns can be decomposed in two parts: the capital gain and the dividend price ratio. In case of investments in a foreign currency we have a third term capturing exchange rate fluctuations:

$$
r^{s,l}_{t+1} \simeq (p_{t+1} - p_t) + \frac{D_{t+1}}{P_{t+1}} + (e_{t+1} - e_t).
$$

Asset allocation with the simplest TVER model can be applied by specifying univariate time series model to predict each part of the total return, as specified above.

To provide an empirical illustration we re-consider the asset allocation exercise implemented in the previous chapter using the CER model. The investor uses monthly data available over the period 1978-2003 to find the tangency portfolio for an investment over the period 2004-2007. The risky assets available for portfolio allocation are German, US and UK shares (as captured by a national index) and the German 10-Year government bond. The risk-free asset is the German short-term rate. The plots in the Figure below (9) report the decomposition of returns in their components showing that the dividend-yield has very little volatility for all three countries, and that the most relevant components in explaining the time variation in total returns is clearly the capital gain, although the exchange rate fluctuations are also a relevant source of time variation.





In Chapter 4 we have derived the tangency portfolio using the CER (and therefore using data over the period 1978-2003 to compute unconditional moments as inputs of the optimization). We consider now a TVER alternative that exploits univariate time series methods. The asset allocation exercise requires the specification of models to predict returns on the four risky assets included in the asset menu of our interest. We adopt the following specification for German stock market returns:

$$
\begin{aligned}
r_{t,t+36}^{GER} &= \left(p_{t+36}^{GER} - p_t^{GER}\right) + \sum_{i=j}^{36} \frac{D_{t+j}^{GER}}{P_{t+j}^{GER}} \\
\left(p_{t+36}^{GER} - p_t^{GER}\right) &= E_t\left(p_{t+36}^{GER} - p_t^{GER}\right) + u_{1,t+36}^{GER} \\
E_t\left(p_{t+36}^{GER} - p_t^{GER}\right) &= \beta_0^{GER} + \beta_1^{GER}\left(p_t^{GER} - \frac{1}{36}\sum_{j=1}^{36} p_{t-j}^{GER}\right) \\
\sum_{j=1}^{36} \frac{D_{t+j}^{GER}}{P_{t+j}^{GER}} &= E_t\left(\sum_{i=1}^{36} \frac{D_{t+j}^{GER}}{P_{t+j}^{GER}}\right) + u_{2,t+36}^{GER}
\end{aligned}
$$

$$E_t\left(\sum_{j=1}^{36}\frac{D_{t+j}^{GER}}{P_{t+j}^{GER}}\right)=\sum_{i=0}^{35}\frac{D_{t-j}^{GER}}{P_{t-j}^{GER}}.$$

In words, log-prices are mean reverting towards a trend that is estimated via the 3 year moving average of past prices, while a random walk model is adopted for the three-year cumulative dividend-price ratio. A similar specification is adopted for US and UK stock market returns where the model adopted for Germany is augmented with a random-walk model for the exchange rate ($i = UK,\, US$):

$$\begin{aligned}
r_{t,t+36}^{i} &= \left(p_{t+36}^{i}-p_{t}^{i}\right)+\sum_{j=1}^{36}\frac{D_{t+j}^{i}}{P_{t+j}^{i}}+\left(e_{t+36}^{i}-e_{t}\right)\\
\left(p_{t+36}^{i}-p_{t}^{i}\right) &= E_t\left(p_{t+36}^{i}-p_{t}^{i}\right)+u_{1,t+36}^{i}\\
E_t\left(p_{t+36}^{i}-p_{t}^{i}\right) &= \beta_0^i+\beta_1^i\left(p_t^i-\frac{1}{36}\frac{\beta_2^i}{\beta_1^i}\sum_{j=1}^{36}p_{t-j}^{i}\right)\\
\sum_{i=1}^{36}\frac{D_{t+i}^{i}}{P_{t+i}^{i}} &= E_t\left(\sum_{i=j}^{36}\frac{D_{t+j}^{i}}{P_{t+j}^{i}}\right)+u_{2,t+36}^{i}\\
E_t\left(\sum_{j=1}^{36}\frac{D_{t+j}^{i}}{P_{t+j}^{i}}\right) &= \sum_{j=0}^{35}\frac{D_{t-j}^{i}}{P_{t-j}^{i}}\\
e_{t+36}^{i} &= E_t\left(e_{t+36}^{i}\right)+u_{3,t+36}^{i}\\
E_t\left(e_{t+36}^{i}\right) &= e_t
\end{aligned}$$

A simple CER model is maintained instead for the German bond returns:

$$r_{t,t+36}^{GER,b}=\mu_4+u_{4,t+36}^{GER,b}$$

The SUR (Seemingly Unrelated Regression) estimation of the system involving the four stochastic equations of interest delivers the following results:

Table 1: SUR Estimation Outputs

	β_0^i	β_1^i	β_2^i	μ_4	R^2
$\left(p_{t+36}^{GER}-p_t^{GER}\right)$	2.29 (12.79)	−0.94 (−18.60)	0.61 (9.88)		0.402
$\left(p_{t+36}^{US}-p_t^{US}\right)$	1.37 (11.16)	−1.20 (−19.18)	1.07 (15.02)		0.687
$\left(p_{t+36}^{UK}-p_t^{UK}\right)$	3.31 (26.39)	−1.10 (−21.57)	0.71 (14.32)		0.510
$r_{t,t+36}^{GER,b}$				0.2362 (30.72)	0.000

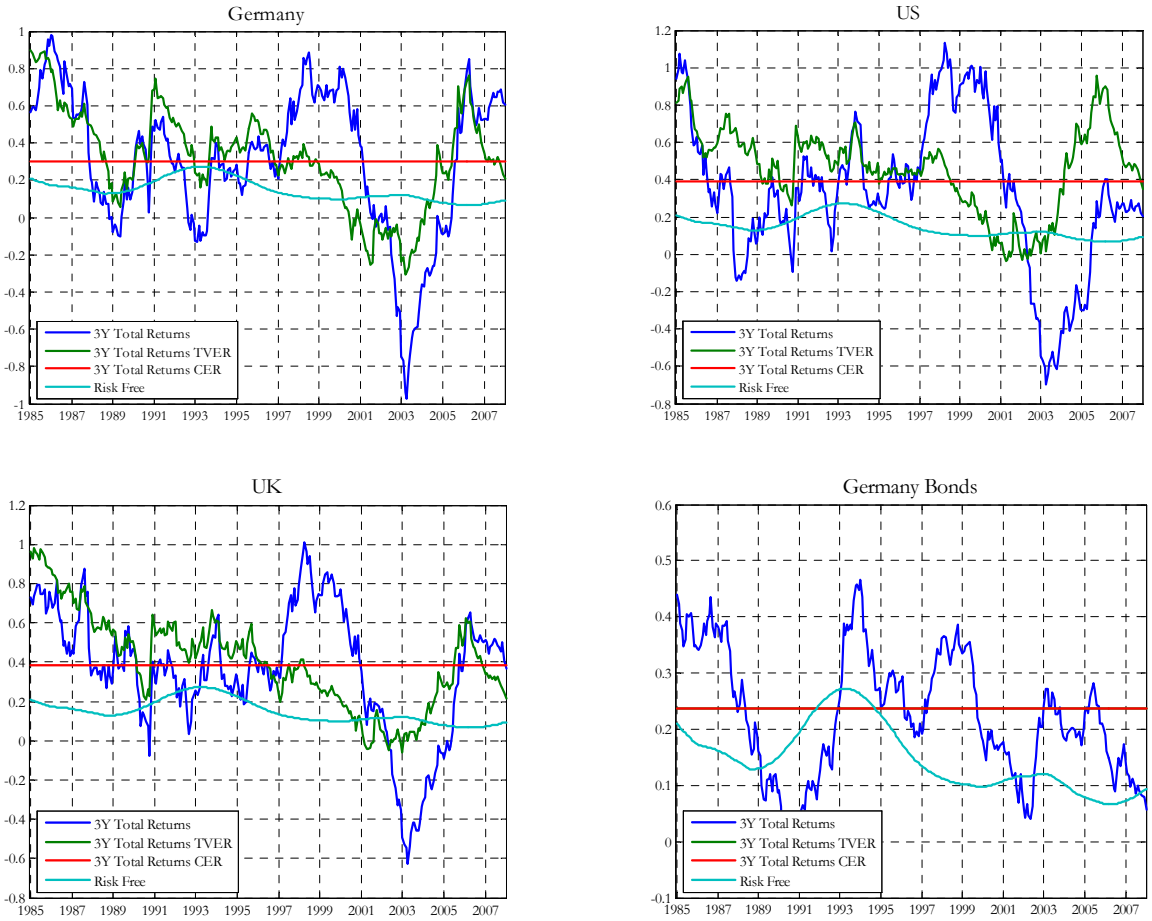
The estimation of the relevant forecasting model leads to the following specification of

the conditional mean and variance-covariance matrix function of returns:

$$\hat{\mu}_t = \begin{bmatrix} \mu_4 \\ \beta_0^{GER} + \beta_1^{GER}\left(p_t^{GER} - \frac{1}{36}\frac{\beta_2^{GER}}{\beta_1^{GER}}\sum_{j=1}^{36} p_{t-j}^{GER}\right) + \sum_{j=0}^{35}\frac{D_{t-j}^{GER}}{P_{t-j}^{GER}} \\ \beta_0^{UK} + \beta_1^{UK}\left(p_t^{UK} - \frac{1}{36}\frac{\beta_2^{UK}}{\beta_1^{UK}}\sum_{j=1}^{36} p_{t-j}^{UK}\right) + \sum_{j=0}^{35}\frac{D_{t-j}^{UK}}{P_{t-j}^{UK}} \\ \beta_0^{US} + \beta_1^{US}\left(p_t^{US} - \frac{1}{36}\frac{\beta_2^{US}}{\beta_1^{US}}\sum_{j=1}^{36} p_{t-j}^{US}\right) + \sum_{j=0}^{35}\frac{D_{t-j}^{US}}{P_{t-j}^{US}} \end{bmatrix}$$

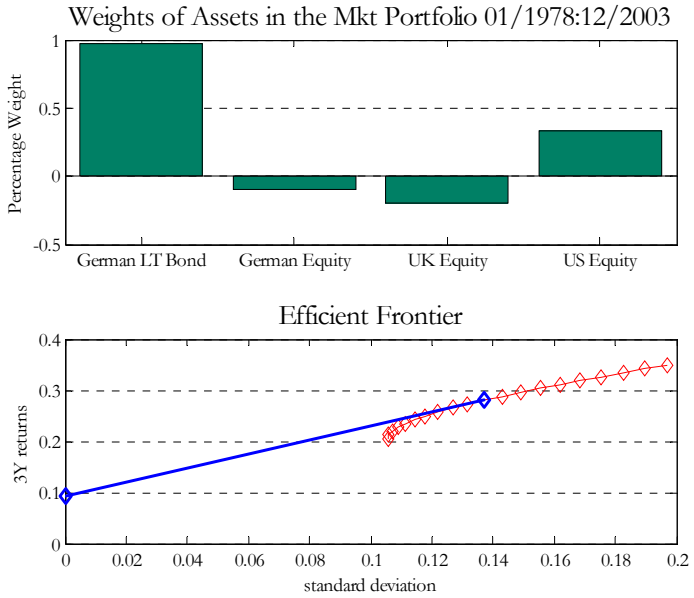
$$\hat{\Omega} = Var\begin{bmatrix} u_{4,t+36}^{GER,b} \\ u_{1,t+36}^{GER} + u_{2,t+36}^{GER} \\ u_{1,t+36}^{UK} + u_{2,t+36}^{UK} + u_{2,t+36}^{UK} \\ u_{1,t+36}^{US} + u_{2,t+36}^{US} + u_{2,t+36}^{US} \end{bmatrix}$$

It is then interesting to compare mean and variance-covariance matrices obtained under the CER and the TVER hypotheses. The following Figure (10) reports the means.



$$\hat{\Omega}^{CER} = \begin{bmatrix} 0.0143 & 0.0168 & 0.0189 & 0.0196 \\ 0.0168 & 0.1524 & 0.1163 & 0.1303 \\ 0.0189 & 0.1163 & 0.1114 & 0.1205 \\ 0.0196 & 0.1303 & 0.1205 & 0.1620 \end{bmatrix}, \quad \hat{\Omega}^{TVER} = \begin{bmatrix} 0.0136 & 0.0028 & 0.0028 & 0.0062 \\ 0.0028 & 0.0845 & 0.0747 & 0.0966 \\ 0.0028 & 0.0747 & 0.0830 & 0.1014 \\ 0.0062 & 0.0966 & 0.1014 & 0.1502 \end{bmatrix}.$$

On the basis of these estimates for μ_t and Ω, the tangency portfolio is then computed. The resulting weights and the efficient frontier are reported in the following Figure (11):



10. **Univariate time-series analysis and portfolio allocation with TVER in MATLAB**

In this section we illustrate a MATLAB programme that allows the practical implementation of all the tools discussed in the chapter. The programme works on the same data used in Chapter 3 and it is called mainCH5.m.

After setting up directories and importing variables the programme calls external m files that perform the analysis commented in the different section of this chapter:

```
%%-------------------------------------------------------------------------
% % -------------- Room saving ----------------------------------------------
close all; clear all; clc; pause(0.01),
randn('seed',sum(clock)), rand('seed',sum(clock)), warning off
TSTART = tic;
% -------------- Load folders ----------------------------------------------
addpath([pwd '\Input\']);
addpath([pwd '\Utilities\']);
addpath([pwd '\Output\']);
% -------------- Import data and transform ---------------------------------
[data,textdata,raw] = xlsread('STOCKINT2013.xls','Monthly');
%[filename,pathname]=uigetfile('*.xls');
%[data,textdata,raw] = xlsread(filename,'Monthly');
%Convert dates into a serial number
date=datenum(textdata(4:end,1),'dd/mm/yyyy');
%-------------------------------------------------------------------------
% Data Transformation
%-------------------------------------------------------------------------
run ch5_datatran
%-------------------------------------------------------------------------
% SECTION 1:  Simulated and Actual Time-Series
%-------------------------------------------------------------------------
run ch5_section1
%-------------------------------------------------------------------------
% SECTION 2:  Persistence:  A Monte Carlo Experiment
%-------------------------------------------------------------------------
run ch5_section2
%-------------------------------------------------------------------------
% SECTION 3:  Putting ARMA models at Work
```

```
%-------------------------------------------------------------------------
run ch5_section3
%-------------------------------------------------------------------------
% SECTION 4:  Stochastic and Deterministic Trends,Beveridge-Nelson Decomposition
%-------------------------------------------------------------------------
run ch5_section4
%-------------------------------------------------------------------------
% SECTION 5:  Portfolio Allocation with univariate TS models
%-------------------------------------------------------------------------
run ch5_section5
tElapsed = toc(TSTART);
display(['The time needed to run the code is:  ',num2str(tElapsed/60),' minutes']);
```

10.1. *Simulated and Actual Time-Series*

Section 1 illustrates the capability of simulated different ARMA time-series to replicate observed financial time-series

```
%-------------------------------------------------------------------------
 % SECTION 1:  Simulated and Actual Time-Series
 %-------------------------------------------------------------------------
 % Defining the parameters

 alpha_1m=mean(GER.Stock.Ret(2:end));
 alpha_3y=mean(GER.Stock.Ret3Ya(38:end));
 beta_1m=var(GER.Stock.Ret(2:end));
 beta_3y=var(GER.Stock.Ret3Ya(38:end));
 alpha_LTB=mean(GER.LTBond.Yield(2:end));
 beta_LTB=var(GER.LTBond.Yield(2:end));
 % Simulate artificial series
 Num=size(GER.Stock.Ret,1);
 Wn_1m=alpha_1m+sqrt(beta_1m)*normrnd(0,1,Num,1);
 Wn_3y=alpha_3y+sqrt(beta_3y)*normrnd(0,1,Num,1);
 Wn_LTB=alpha_LTB+sqrt(beta_LTB)*normrnd(0,1,Num,1);
 % Plot the result
 s_start = '31/01/1978';
s_end = '31/12/2010';
```

```
date_find = datenum([s_start; s_end],'dd/mm/yyyy');
ss = datefind(date_find(1,1),date);
se = datefind(date_find(2,1),date);
T=ss-se+1;
%plot actual and simulated series
run Figure001
 %-------------------------------------------------------------------------
```

10.2. *A Monte Carlo experiment on the impact of persistence on the OLS estimates*

Section 2 illustrates how a Monte-Carlo experiment to assess how the small sample bias of OLS estimators in presence of persitence can be evaluated:

```
%-------------------------------------------------------------------------
% SECTION 2:  Persistence:  A Monte Carlo Experiment
%-------------------------------------------------------------------------

 % Initializing the series:
reps=100;
obs=200;
init=15;
x=NaN(obs,reps);
beta1=NaN(obs-init,reps);
%Defining Parameters
rho=0.9;
alpha=0;
vol=1;
% Initializing the series:
x(1,:)=alpha/(1-rho);
% Artificial series:
u1=vol.*normrnd(0,1,obs,reps);
for j=1:reps
for i=2:obs
x(i,j)=rho*x(i-1,j)+alpha*(1-rho)+u1(i,j)*sqrt((1-rho^2)*vol);
end
for k=init+1:obs
X=NaN(k-1,2);
X(:,1)=ones;
```

```
X(:,2)=x(1:k-1,j);
reg1=ols(x(2:k,j), X(1:k-1,:));
beta1(k-init,j)=reg1.beta(2);
end
end
dim=16:200;
av_beta1_hat(:,1)=mean(beta1,2);
%plot results of the experiment
figure
plot(dim,av_beta1_hat',dim,(ones(size(dim)).*rho))
grid on
%xlim([dim(1) dim(end)])
xlabel('sample size')
ylabel('expected rho hat')
title('Bias of OLS estimator in an AR(1)')
```

10.3. *Putting ARMA models at work*

Section 3 illustrates how to put ARMA models at work through identification, estimation and forecasting

```
%------------------------------------------------------------------------
 % SECTION 3:  Putting ARMA models at Work
 %------------------------------------------------------------------------
 % Select the estimation sample
s_start = '31/01/1979';
s_end = '29/10/1993';
date_find = datenum([s_start; s_end],'dd/mm/yyyy');
first = datefind(date_find(1,1),date);
last = datefind(date_find(2,1),date);
T=first-last+1;
 % Estimate ARMA model
 % ARMA(1,1) estimation

spec=garchset('VarianceModel','Constant','R',1,'M',1);
[coeff,errors,llf,innovation,sigma,summary]=garchfit(spec,
GER.LTBond.Yield(first:last));
garchdisp(coeff,errors);
```

```
%ARMA forecast
horz=108;
[SigmaForecast,MeanForecast] = garchpred(coeff,GER.LTBond.Yield(first:last),horz);
Forecast_arma=MeanForecast;
GER.LTBond.Yield_for_arma=NaN(size(GER.Stock.Ret));
GER.LTBond.Yield_for_arma(1:last,1)=GER.LTBond.Yield(1:last,1);
GER.LTBond.Yield_for_arma(last+1:(last+horz),1)=Forecast_arma(1:horz,1);
% AR(1) estimation
spec=garchset('VarianceModel','Constant','R',1);
[coeff,errors,llf,innovation,sigma,summary]=garchfit(spec,GER.LTBond.Yield(first:las
garchdisp(coeff,errors);
% AR forecast
%spec_pred=garchset('VarianceModel','Constant','C',coeff.C,'R',1,'AR',coeff.AR,'K',c
[SigmaForecast,MeanForecast] = garchpred(coeff,GER.LTBond.Yield(first:last),horz);
Forecast_ar=MeanForecast;
GER.LTBond.Yield_for_ar=NaN(size(GER.Stock.Ret));
GER.LTBond.Yield_for_ar(1:last,1)=GER.LTBond.Yield(1:last,1);
GER.LTBond.Yield_for_ar(last+1:(last+horz),1)=Forecast_arma(1:horz,1);
% Forecast unconditional mean
Forecast_unc=ones(horz,1)*mean(GER.LTBond.Yield(first:last));
GER.LTBond.Yield_for_unc=NaN(size(GER.Stock.Ret));
GER.LTBond.Yield_for_unc(1:last,1)=GER.LTBond.Yield(1:last,1);
GER.LTBond.Yield_for_unc(last+1:(last+horz),1)=Forecast_unc(1:horz,1);

%plot actual and forecast
run Figure003
disp('************************************************************')
% % Recursive One-Step Ahead Forecasts
disp('============================================================')
disp('Recursive One-Step Ahead Forecasts')
s_start = '31/01/1979';
s_end = '29/10/1993';
date_find = datenum([s_start; s_end],'dd/mm/yyyy');
first = datefind(date_find(1,1),date);
last = datefind(date_find(2,1),date);
T=first-last+1;
```

```
    horz=1;
    forsmpl=40;
    LTBond_mean=NaN(size(GER.Stock.Ret));
    LTBond_upp=NaN(size(GER.Stock.Ret));
    LTBond_low=NaN(size(GER.Stock.Ret));

    for j=1:forsmpl
    %AR(1) estimation
    spec=garchset('VarianceModel','Constant','R',1);
    [coeff,errors,llf,innovation,sigma,summary]=garchfit(spec,
GER.LTBond.Yield(first:last));
    garchdisp(coeff,errors);
    %AR(1) forecast
    spec_pred=garchset('VarianceModel','Constant','C',coeff.C,'R',1,'AR',
coeff.AR,'K',coeff.K);
    [SigmaForecast,MeanForecast,SigmaTotal,MeanRMSE] = garchpred(spec_pred,
GER.LTBond.Yield(first:last+j-1),horz);
    forecast_ar=MeanForecast;
    forecast_upp=MeanForecast+2*MeanRMSE;
    forecast_low=MeanForecast-2*MeanRMSE;
    LTBond_mean(last+j:last+j,1)=forecast_ar;
    LTBond_upp(last+j:last+j,1)=forecast_upp;
    LTBond_low(last+j:last+j,1)=forecast_low;
    end
    %plotting recursive forecast performance

    run Figure004

    %computing RMSE
    RMSE=sqrt(sum((LTBond_mean(last+1:last+forsmpl)-
GER.LTBond.Yield(last+1:last+forsmpl)).^2));
    var1=strcat('For_');
    assignin('base',var1,LTBond_mean);
    var2=strcat('UpperBound_');
    assignin('base',var2,LTBond_upp);
```

```
 var3=strcat('LowBound_');
 assignin('base',var3,LTBond_low);
 var4=strcat('RMSE_');
 assignin('base',var4,RMSE);
 clear var1 var2 var3 var4
```

10.4. *Stochastic and deterministic trends*

Section 4 analyzes trends. and trend-cycle decompositions

```
%-------------------------------------------------------------------------
 % SECTION 4:  Stochastic and Deterministic Trends, Decompositions
 %-------------------------------------------------------------------------
 % Select the estimation sample
s_start = '31/01/1979';
s_end = '29/10/1993';
date_find = datenum([s_start; s_end],'dd/mm/yyyy');
first = datefind(date_find(1,1),date);
last = datefind(date_find(2,1),date);
T=first-last+1;
 % Generate artificial trend series
 Trend_det=NaN(size(GER.Stock.Ret));
 Trend=NaN(size(GER.Stock.Ret));
 Trend_st1=NaN(size(GER.Stock.Ret));
 Trend_st2=NaN(size(GER.Stock.Ret));
 Trend(1,1)=0;
 Trend_det(1,1)=0;
 Trend_st1(1,1)=0;
 Trend_st2(1,1)=0;
 ind=rows(Trend_st1);
 for i=2:ind
 Trend(i,:)=Trend(i-1,:)+1;
 Trend_det(i,:)=0.1*Trend(i,:)+normrnd(0,1,1);
 Trend_st1(i,:)=0.15+Trend_st1(i-1,:)+normrnd(0,1,1);
 Trend_st2(i,:)=0.1+Trend_st2(i-1,:)+normrnd(0,1,1);
 end
 % Plot trend series
 t=1:rows(GER.Stock.Ret);
```

```
figure
h3=plot(t',Trend_det,'.-',t',Trend_st1,'--',t',Trend_st2,'-','LineWidth',2);
title('Trends','fontname','garamond','fontsize',16);
set(gca,'fontname','garamond','fontsize',10);
set(gca,'xtick',[1:24:rows(t')]);
set(gca,'xlim',[1 rows(t')]);
set(gca,'xticklabel','1977|1979|1981|1983|1985|1987|1989|1991|1993|
1995|1997|1999|2001|2003|2005|2007|2009|2011');
grid;
set(gcf,'color','w');
h3=legend('Det Trend','ST TREND 1', 'ST TREND 2',2);


%-------------------------------------------------------------------------
%Beveridge-Nelson Decomposition
%-------------------------------------------------------------------------
% Generate artificial IMA series
% Defining the parameters
rho=0.6;
theta=0.5;
% Simulate IMA series
IMA=NaN(size(GER.Stock.Ret));
IMA(1,1)=0;
IMA(2,1)=0;
Num=rows(GER.Stock.Ret);
eps=normrnd(0,1,Num,1);
ind=rows(GER.Stock.Ret);
for i=3:ind
IMA(i,:)=IMA(i-1,:)+rho*(IMA(i-1,:)-IMA(i-2,:))+eps(i,:)+theta*eps(i-1,:);
end
% define BN decomposition

BN_TR=NaN(size(GER.Stock.Ret));
BN_CY=NaN(size(GER.Stock.Ret));
BN_TR(1,1)=0;
for i=2:ind
```

```
    BN_TR(i,:)=BN_TR(i-1,:)+((1+theta)/(1-rho))*eps(i,:);
    end
    BN_CY=IMA-BN_TR;

    % Plot trend series
    t=1:rows(GER.Stock.Ret);
    figure
    h3=plot(t',BN_TR,'.-',t',BN_CY,'--',t',IMA,'-','LineWidth',2);
    title('Trends','fontname','garamond','fontsize',16);
    set(gca,'fontname','garamond','fontsize',10);
    set(gca,'xtick',[1:24:rows(t')]);
    set(gca,'xlim',[1 rows(t')]);
    set(gca,'xticklabel','1977|1979|1981|1983|1985|1987|1989|1991|1993|
1995|1997|1999|2001|2003|2005|2007|2009|2011');
    grid;
    set(gcf,'color','w');
    h3=legend('BN trend','BN cycle', 'IMA',2);
```

10.5. *Asset Allocation with TVER and univariate time-series analysis*

Section 5 reconsiders the problem of finding the tangency portfolio when univariate models for TVER are considered.

```
%-------------------------------------------------------------------------
% SECTION 5:  Portfolio Allocation with univariate TS models
%-------------------------------------------------------------------------
s_start = '31/12/1984';
s_end = '31/12/2003';
date_find = datenum([s_start; s_end],'dd/mm/yyyy');
ss = datefind(date_find(1,1),date);
se = datefind(date_find(2,1),date);
T=se-ss+1;
p_start = '30/01/2004';
p_end = '31/12/2007';
date_find=datenum([p_start; p_end],'dd/mm/yyyy');
ps=datefind(date_find(1,1),date);
pe=datefind(date_find(2,1),date);
n = pe-ps+1;
```

```
   %----------- Historical Excess Returns Matrix ----------------------------
   % % Step 1.  Construct the relevant time series of returns
   Ret3Y = [GER.RF.Ret3Y(ss:se) GER.LTB.Ret3Y(ss:se) GER.Stock.Ret3Y(ss:se)
UK.Stock.Reteu3Y(ss:se) US.Stock.Reteu3Y(ss:se)];
   Ret = [GER.RiskFree(ss:se) GER.LTBond.Ret(ss:se) GER.Stock.Ret(ss:se)
UK.Stock.Reteu(ss:se) US.Stock.Reteu(ss:se)];
   ExRet= [GER.LTBond.ExRet(ss:se) GER.Stock.ExRet(ss:se) UK.Stock.ExRet(ss:se)
US.Stock.ExRet(ss:se)];
   ExRet3Y=[GER.LTB.ExRet3Y(ss:se) GER.Stock.ExRet3Y(ss:se) UK.Stock.ExRet3Y(ss:se)
US.Stock.ExRet3Y(ss:se)];
   % Descriptive data analysis
   Perf = cumsum(Ret);
   run Figure005
   % % ------------------------------------------------------------------------
   % Compute the weights and Efficient Frontier using CER upto 2003
   % ------------------------------------------------------------------------
   % FOrecasting model specification and construction of the relevant measures
   muExR = mean(ExRet3Y);
   SigmaR = (T/(T-1))*cov(ExRet3Y); %compute robust variance covariance matrix
   CorR=corr(ExRet3Y);
   % Optimal weights on the tangency portfolio
   WeightsTP = ((SigmaR^(-1))*muExR')./(ones(size(SigmaR,2),1)'*(SigmaR^(-1)*muExR'));
   %Compute the efficient frontier and plot it
   NumPortf = 20; % number of efficient portfolios to be computed
    % we do not compute all efficient portfolios in the
    % efficient frontier (there is a continuum of them)
    % we compute only a subset of them, equally spaced between
    % the minimum variance portfolio and the portfolio with the
    % highest return.
   mu=muExR+mean(GER.RF.Ret3Y(ss:se)); %generate mean returns
   % generate mean variance efficient frontier
   [WeightsMV, MuMeanVariance, StdMeanVariance] = effront(mu, SigmaR, NumPortf);

   %mean variance for risk fre and tangency portfolio
   Mu_bm = zeros(2, 1);
   SD_bm = zeros(2, 1);
```

```
   Mu_bm(1,1) = mean(GER.RF.Ret3Y(ss:se));
   SD_bm(1,1)=0;
   Mu_bm(2,1)=mu*WeightsTP;
   SD_bm(2,1)=sqrt(WeightsTP'*SigmaR*WeightsTP);
   % Plot weights in the tangency portfolio and the efficient frontier
   run Figure006
   % % ------------------------------------------------------------------------
   % Compute the cumulative performances out-of-sample
   % ------------------------------------------------------------------------
   Ret2 = [GER.LTBond.Ret(ps:pe) GER.Stock.Ret(ps:pe) UK.Stock.Reteu(ps:pe)
US.Stock.Reteu(ps:pe)];
   PerfR2 = cumsum(Ret2);
   Rport = Ret2*WeightsTP;
   PerfPort1 = cumsum(Rport);
   PerfRF=cumsum(GER.RiskFree(ps:pe));
   % Plot the cumulative performances out-of-sample
   run Figure007
   % % ------------------------------------------------------------------------
   % Compute the weights and efficient frontier using a simple
   % TVER
   % ------------------------------------------------------------------------
   % Graphical Inspection of SOP for Germany and US
   run Figure008
   % Econometric specification and estimation of SOP
   %OLS
   XGER=NaN(size(GER.Stock.lp,1),3);
   XGER(:,1)=ones;
   XGER(:,2)=GER.Stock.lp;
   XGER(:,3)=GER.Stock.lpTr;
   resGER=ols(GER.Stock.CG3Y(ss:se,:), XGER(ss-36:se-36,:));
   prt(resGER)
   XUS=NaN(size(GER.Stock.lp,1),3);
   XUS(:,1)=ones;
   XUS(:,2)=US.Stock.lp;
   XUS(:,3)=US.Stock.lpTr;
   resUS=ols(US.Stock.CG3Y(ss:se,:), XUS(ss-36:se-36,:));
```

```
prt(resUS)
XUK=NaN(size(GER.Stock.lp,1),3);
XUK(:,1)=ones;
XUK(:,2)=UK.Stock.lp;
XUK(:,3)=UK.Stock.lpTr;
resUK=ols(UK.Stock.CG3Y(ss:se,:), XUK(ss-36:se-36,:));
prt(resUK)
%SUR
y1 = GER.Stock.CG3Y(ss:se,:); x11 = GER.Stock.lp(ss-36:se-36,:);
x12 = GER.Stock.lpTr(ss-36:se-36,:); % GER
y2 = US.Stock.CG3Y(ss:se,:); x21 = US.Stock.lp(ss-36:se-36,:);
x22 = US.Stock.lpTr(ss-36:se-36,:); % US
y3 = UK.Stock.CG3Y(ss:se,:); x31 = UK.Stock.lp(ss-36:se-36,:);
x32 = UK.Stock.lpTr(ss-36:se-36,:); % 4UK
nobs = length(GER.Stock.CG3Y(ss:se,:));
iota = ones(nobs,1);
vname1 = strvcat('GER','const','lag P','lag TREND');
vname2 = strvcat('US','const','lag P','lag TREND');
vname3 = strvcat('UK','const','lag P','lag TREND');
% set up a structure for y in each eqn
y(1).eq = y1;
y(2).eq = y2;
y(3).eq = y3;
clear X
% set up a structure for X in each eqn
X(1).eq = [iota x11 x12];
X(2).eq = [iota x21 x22];
X(3).eq = [iota x31 x32];
SUR=sur(3,y,X);
prt(SUR);
%BUILD FORECAST
GER.Stock.Ret3YF=NaN(size(GER.Stock.Index));
UK.Stock.Reteu3YF=NaN(size(GER.Stock.Index));
US.Stock.Reteu3YF=NaN(size(GER.Stock.Index));
GER.Stock.Ret3YCER=NaN(size(GER.Stock.Index));
UK.Stock.Reteu3YCER=NaN(size(GER.Stock.Index));
```

```
   US.Stock.Reteu3YCER=NaN(size(GER.Stock.Index));
   GER.LTB.Ret3YCER=NaN(size(GER.Stock.Index));
   for i=ss:pe
   %GER.Stock.Ret3YF(i,:)= resGER.beta(1)+ resGER.beta(2)*GER.Stock.lp(i-36,:)+
   resGER.beta(3)*GER.Stock.lpTr(i-36,:)+GER.Stock.DY3Y(i-36,:);
   %UK.Stock.Reteu3YF(i,:)= resUK.beta(1)+ resUK.beta(2)*UK.Stock.lp(i-36,:)+
   resUK.beta(3)*UK.Stock.lpTr(i-36,:)+UK.Stock.DY3Y(i-36,:);
   %US.Stock.Reteu3YF(i,:)= resUS.beta(1)+ resUS.beta(2)*US.Stock.lp(i-36,:)+
   resUS.beta(3)*US.Stock.lpTr(i-36,:)+US.Stock.DY3Y(i-36,:);
   GER.Stock.Ret3YF(i,:)= SUR(1).beta(1)+ SUR(1).beta(2)*GER.Stock.lp(i-36,:)+
   SUR(1).beta(3)*GER.Stock.lpTr(i-36,:)+GER.Stock.DY3Y(i-36,:);
   UK.Stock.Reteu3YF(i,:)= SUR(3).beta(1)+ SUR(3).beta(2)*UK.Stock.lp(i-36,:)+
   SUR(3).beta(3)*UK.Stock.lpTr(i-36,:)+UK.Stock.DY3Y(i-36,:);
   US.Stock.Reteu3YF(i,:)= SUR(2).beta(1)+ SUR(2).beta(2)*US.Stock.lp(i-36,:)+
   SUR(2).beta(3)*US.Stock.lpTr(i-36,:)+US.Stock.DY3Y(i-36,:);
   end
   %Forecast based on CER
   GER.Stock.Ret3YCER(ss:pe,:)= mean(GER.Stock.Ret3Y(ss:se));
   UK.Stock.Reteu3YCER(ss:pe,:)=mean(UK.Stock.Reteu3Y(ss:se));
   US.Stock.Reteu3YCER(ss:pe,:)=mean(US.Stock.Reteu3Y(ss:se));
   GER.LTB.Ret3YCER(ss:pe,:)= mean(GER.LTB.Ret3Y(ss:se));
   %graphic comparison between TVER and CER
   run Figure009
   %asset allocation with TVER
   %Build forecasting errors
   GER.LTB.Ret3YFE=GER.LTB.Ret3Y-GER.LTB.Ret3YCER;
   GER.Stock.Ret3YFE=GER.Stock.Ret3Y-GER.Stock.Ret3YF;
   UK.Stock.Reteu3YFE=UK.Stock.Reteu3Y-UK.Stock.Reteu3YF;
   US.Stock.Reteu3YFE=US.Stock.Reteu3Y-US.Stock.Reteu3YF;
   Ret3YFE = [GER.LTB.Ret3YFE(ss:se) GER.Stock.Ret3YFE(ss:se) UK.Stock.Reteu3YFE(ss:se)
US.Stock.Reteu3YFE(ss:se)];
   muR_TVER = [GER.LTB.Ret3YCER(pe) GER.Stock.Ret3YF(pe) UK.Stock.Reteu3YF(pe)
US.Stock.Reteu3YF(pe)];
   muExR_TVER = [(GER.LTB.Ret3YCER(pe)-(GER.RF.Ret3Y(pe)))
    (GER.Stock.Ret3YF(pe)-(GER.RF.Ret3Y(pe))) (UK.Stock.Reteu3YF(pe)-(GER.RF.Ret3Y(pe)))
(US.Stock.Reteu3YF(pe)-(GER.RF.Ret3Y(pe)))];
```

```
SigmaR_TVER = cov(Ret3YFE);
% WEIGHTS
WeightsTP_TVER = ((SigmaR_TVER^(-1))*muExR_TVER')./(ones(size(SigmaR_TVER,2),1)'*
(SigmaR_TVER^(-1)*muExR_TVER'));
%Compute the efficient frontier and plot it
NumPortf = 20; % number of efficient portfolios to be computed
 % we do not compute all efficient portfolios in the
 % efficient frontier (there is a continuum of them)
 % we compute only a subset of them, equally spaced between
 % the minimum variance portfolio and the portfolio with the
 % highest return.
% generate mean variance efficient frontier
[WeightsMV, MuMeanVariance, StdMeanVariance] = effront(muR_TVER, SigmaR_TVER,
NumPortf);

%mean variance for risk fre and tangency portfolio
Mu_bm_TVER = zeros(2, 1);
SD_bm_TVER = zeros(2, 1);
Mu_bm_TVER(1,1) = GER.RF.Ret3Y(pe);
SD_bm_TVER(1,1)=0;
Mu_bm_TVER(2,1)=muR_TVER*WeightsTP_TVER;
SD_bm_TVER(2,1)=sqrt(WeightsTP_TVER'*SigmaR_TVER*WeightsTP_TVER);
% Plot weights in the tangency portfolio and the efficient frontier
run Figure010
% Performance
Rport_TVER = Ret2*WeightsTP_TVER;
PerfPort_TVER = cumsum(Rport_TVER);
run Figure011
```

11. **References**

Beveridge, S., and Nelson, C. (1981). 'A New Approach to the Decomposition of Economic Time Series into Permanent and Transitory Components with Particular Attention to the Measurement of Business Cycle'. *Journal of Monetary Economics*, 151-174.

Bhargava, A. (1986). 'On the Theory of Testing for Unit Roots in Observed Time Series'. *Review of Economic Studies*, 53: 369-384.

Cuddington, J. T., and Winters, L. A. (1987). 'The Beveridge-Nelson decomposition of economic time-series. A quick computational method'. *Journal of Monetary Economics*, 19: 125-127.

DeJong, D. N., and Whiteman, C. H. (1991). 'Reconsidering Trends and random walks in macroeconomic time-series'. *Journal of Monetary Economics*, 28: 221-254.

Dickey, D. A., and Fuller, W. A. (1981). 'Likelihood Ratio Statistics for Autoregressive Time Series with a Unit Root'. *Econometrica*, 49:1057-1072.

Enders, W. (1995). *Applied Econometric Time Series*. New York: Wiley Series in Probability and Mathematical Statistics.

Fuller, W. A. (1976). *Introduction to Statistical Time Series*. New York: J. Wiley and Sons.

Granger, C. W. J., and Newbold, P. (1974). 'Spurious Regressions in Econometrics'. *Journal of Econometrics*, 2: 111-120.

Hamilton, J. (1994). *Time-Series Analysis*. Princeton: Princeton University Press.

Harvey, A. C., and Jaeger, A. (1993). 'Detrending, Stylized Facts and the Business Cycles'. *Journal of Applied Econometrics*.

Harvey, A. C., and Koopman, S. J.(1996) 'Multivariate Structural Time Series Models' in C. Heij et al. (eds.), *Systematic Dynamics in Economic and Financial Models.* New York: John Wiley.

Hatanaka, M. (1987). *Time-Series-based-econometrics*. Oxford: Oxford University Press.

Hodrick, R. J., and Prescott, E. C. (1997). 'Postwar U.S. Business Cycles: An Empirical Investigation'. *Journal of Money, Credit, and Banking*, 29: 116.

Maddala, G. S., and Kim, I. M. (1998). *Unit roots, Cointegration and Structural Change*. Cambridge: Cambridge University Press.

Maravall, A. (1995) 'Unobserved Components in Economic Time Series' in M. H. Pesaran and M. Wickens (eds.), *Handbook of Applied Econometrics. Macroeconomics*, Oxford: Basil Blackwell.

Nelson, C. R., and Plosser, C. I. (1982). 'Trends versus Random Walks in economic time-series. Some evidence and implications'. *Journal of Monetary Economics*, 10: 139-162.

White, H. (1984). *Asymptotic Theory for Econometricians*. London: Academic Press.