4 Statistical Analyses of Time-Varying Phenomena

In this chapter, we will utilize the concepts discussed in the earlier chapter on statistics and probability, and describe a toolkit to analyze time-varying behavior of real world systems. Real world systems are messy and predictions may not be exactly trustworthy. Economists know it very well. In fact, the repeated failure of precise forecasting ability led to the adage that an economist is an expert who can figure out in future why the predictions that were made in the past, did not take place in the present! Although this description was originally about economics, it is probably equally applicable for complex systems as well, especially given the fact that one of the defining feature of complex systems is emergence of features that may not be traced back to the behavior of an average constituent element. However, we will see that one can still make a lot of sense of time-varying phenomena from observed data.

Examples of systems that generates time-varying responses are abound. Rainfall, temperature, gross domestic productions in a region, all of these variables fluctuate over time. Some of them fluctuate in a manner more periodical than other variables, e.g. rainfall may peak around July in the eastern India whereas snowfall may peak around January in the east coast of the USA. Over 24 hours in a day, one can keep track of number of cars passing by in a busy street in a city. A likely pattern would be found that the number of cars passing by, would increase between 9 AM-12 noon and again between 5-10 PM. In the other extreme, we have stock price returns which seems to exhibit almost no patterns of periodic oscillation.

Any such observation can be represented by a time series object, which can be simply thought of as a vector of values recorded over time. To fix the notations, we denote a time series by $X_T = \{x_1,\ x_2,\ \ldots,\ x_T\}$. Here x_t denotes an observation (say, rainfall) recorded at time t where the time index t varies between 1 and T, T being a positive integer. For all t, the values would be real i.e. $x_t \in \mathbb{R}$ (we will not consider complex values).

Here, we present two examples of time series from two very different systems. In figure 4.1, we show evolution of the total global production measured via world gross domestic product, from 1960 to 2020 (publicly available from the World Bank website). The top panel shows the increasing trend in the total production in logarithmic scale. In the bottom panel, we have plotted the yearly growth rates of the same series, which exhibits a lot more fluctuations. For example, we see that around the time of the global financial crisis (2008-09) the global growth turned negative. Another example of time series can be seen in the top panel of figure 4.2

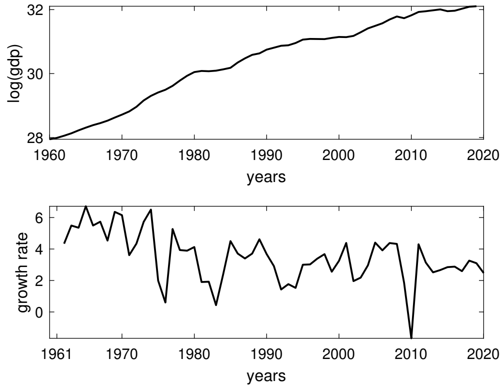

Figure 4.1 Total global production over six decades. Upper panel: Evolution of world gross domestic product (GDP) from 1961-2020. Lower panel: Corresponding growth rates.

which presents evolution of the market capitalization of Bitcoin. The initial period of slow growth with mild fluctuations followed by a period of explosive growth in the market size is obvious from the data. However in the log scale (bottom panel), the growth seems to have a linear component with a clear upward trend and occasional fluctuations.

In this chapter, our task would be two-fold. One, we will create models that will allow us to see if there is any connection in a statistical sense between these values observed over time, for a given system. Two, we will discuss and describe statistical apparatus to estimate the time series models based on a given set of observations. We will be generally agnostic towards the source of the data. The techniques we will see in this chapter are widely applicable and can be directly applied to many different types of time series data. However, for the sake of exposition, we note that many of these techniques were specifically built for economic and financial variables. Therefore, our approach would also be from that angle.

4.1 Some basic definitions and constructions

Let us first start with a time series object: $X_T = \{x_1,\ x_2,\ \ldots,\ x_T\}$. We will interchangeably use X_T or $\{x_t\}$ to denote the time series. For the time being, we will assume that X_T is obtained from a *stationary* process. Intuitively, what we mean is that such processes are *well-behaved* and have nice statistical properties. We will elaborate later on exactly what we mean by stationarity in a more technical sense.

Let us first define the mean of the observations as $\mu(X_T) = 1/T\sum_{t=1}^{T} x_t$ and the corresponding variance as $\sigma^2(X_T) = 1/T\sum_{t=1}^{T}(x_t-\mu)^2$. The population analogues will be denoted below by $E(.)$ and $Var(.)$. Since these observations are recorded

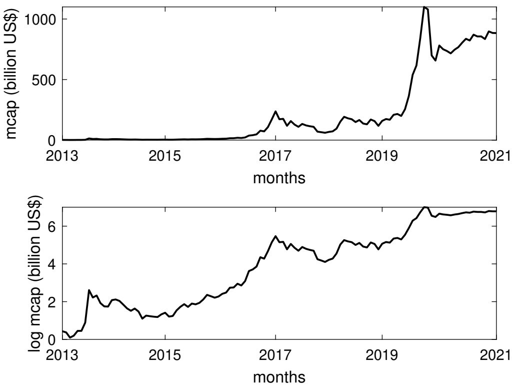

Figure 4.2 Evolution of market capitalization of Bitcoin from April 2013 to August 2021. Upper panel: In level. Lower panel: In logarithm.

over time t, a starting point to see if there is any pattern, would be to check if a high draw of x is followed by a high draw of x or the consecutive draws have no relationship whatsoever. To quantify this idea, we define *autocovariance*, which allows to see if there is any lagged comovements in the recorded observations. For notational purposes, we will call observation x_{t-j} to lag observation x_t by j lags. Formally, autocovariance at the j-th lag is given by

$$\gamma_j = Cov(x_t, x_{t-j}). \tag{4.1}$$

For a class of processes called stationary processes, the time index t doesn't matter as covariance across j time points will be the same for all time points t. We will define stationarity formally later. If the process has a zero mean, then the j-lag autocovariance can be simply written as

$$\gamma_j = E(x_t x_{t-j}). \tag{4.2}$$

Note that 0-lag autocovariance is just the variance:

$$\gamma_0 = Var(x_t). \tag{4.3}$$

Typically it is more useful to scale the autocovariances by the variance so that different series with different degree of variances can be compared. The normalized values represent *autocorrelations.*

Definition 4.1. The autocorrelation function (a.c.f.) is defined as

$$\rho_j = \frac{\gamma_j}{\gamma_0} \tag{4.4}$$

for the j-th lag (j can be positive or negative).

4.1.1 Frequency and time domain

Although we will almost exclusively deal with time domain analysis of the time series objects, it would be useful here to provide a brief introduction to frequency domain representation. For that purpose, we need to first define two terms - *period* and *frequency.*

Definition 4.2. The period of a wave is the time τ that a wave takes to go through a whole cycle (2π rotation).

Definition 4.3. The frequency ω is the number of whole cycles completed in unit time by a wave.

Combining these two definitions we can write

$$\tau = \frac{2\pi}{\omega} \tag{4.5}$$

There exists a very useful tool that allows us to go back and forth between the time domain and the frequency domain. Formally it is called *Fourier transform* named after its inventor Joseph Fourier. Given a time series $\{x_t\}$, its Fourier transformation is:

$$x(w) = \frac{1}{2\pi} \sum_{t=-\infty}^{\infty} e^{-it\omega} x(t), \tag{4.6}$$

and the inverse Fourier transform is:

$$x(t) = \int_{-\pi}^{\pi} e^{itw} x(\omega) d\omega. \tag{4.7}$$

We will see that instead of converting the series itself into frequencies, it would be more useful to deal with the autocovariance function to study its frequency spectrum.

4.1.2 Autocovariance function

Before we get into the derivations in this section, a small point on notations is in order. Here, we will deal with complex numbers as well as lags in time series. We will continue to denote the complex number $\sqrt{-1}$ by i (as we used for describing characteristic functions in section 3.2.3 in chapter 3) and the lags by h. We will not use j here that we have used above to denote the lags. The reason is that often j is also used to denote complex numbers. Unless explicitly stated, this convention of writing the complex number will be contained only within this section (section 4.1.2). Also, the following discussion will assume that the time index of the series goes from negative to positive infinity.

As we described above, the autocovariance function for a zero-mean stationary process $\{x_t\}$ is given by

$$\gamma_h = E(x_t x_{t-h}). \tag{4.8}$$

We define the spectrum of the time series $\{x_t\}$ as

$$S_x(\omega) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} e^{-ih\omega} \gamma_h. \tag{4.9}$$

This is an useful expression. We see that if we take $\omega = 0$ in equation 4.9, we get

$$\sum_{h=-\infty}^{\infty} \gamma_h = 2\pi S_x(0), \tag{4.10}$$

or in words, the strength of the spectrum evaluated at $\omega = 0$ is proportional to the sum of all autocovariances. We can simplify the right hand side of the equation 4.9 by using Euler's identity:

$$e^{i\phi} = \cos\phi + i \sin\phi. \tag{4.11}$$

Simply by substituting the exponential terms $e^{-ih\omega}$ from the Euler's identity, the equation boils down to

$$S_x(\omega) = \frac{1}{2\pi} \left[\gamma_0 + 2 \sum_{h=1}^{\infty} \gamma_h \cos(h\omega) \right]. \tag{4.12}$$

Here we have used the identity that cos $(-\omega)$ = cos (ω) and sin $(-\omega)$ = sin (ω).

We can also use the inverse Fourier transform to generate autocovariance in a reverse operation –

$$\gamma_h = \int_{-\pi}^{\pi} e^{i\omega h} S_x(\omega) d\omega. \tag{4.13}$$

At $h = 0$, we get

$$\gamma_0 = \int_{-\pi}^{\pi} S_x(\omega) d\omega. \tag{4.14}$$

We can have an alternate representation by utilizing the formalism of moment generating functions that we have introduced in chapter 3. The autocovariance generating function can be written as

$$g_x(z) = \sum_{h=-\infty}^{\infty} \gamma_h z^h. \tag{4.15}$$

Therefore, if we carry out a substitution $z = exp(-i\omega)$, then the spectrum is simply given by

$$S_x(\omega) = \frac{g_x(z)}{2\pi}. \tag{4.16}$$

Before ending the present discussion, we note that one can carry out all the above analysis in terms of the autocorrelation function as well by dividing both sides by $\gamma_x(0)$. The analog of equation 4.9 would be

$$f(\omega) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} e^{-ih\omega} \rho_h. \tag{4.17}$$

With inverse Fourier transform, we can recover the autocorrelation function as

$$\rho_h = \int_{-\pi}^{\pi} e^{i\omega h} f_x(\omega) d\omega. \tag{4.18}$$

The above description shows that it is possible to go back and forth between frequency domain representation and the time domain representation for a time series. The usefulness of the frequency domain representation becomes very clear when analyzing signals for their frequency contents. However, in the present context, we will not focus on regular oscillatory phenomena which have clear spikes in frequencies, making the periodicity in the time series prominent. We will instead focus on time series for which the periodic behavior is less clear and although oscillations exist, they are not as regular as those of periodic waves. Often the time-domain analysis for such time series is easier to conduct and interpret.

At this stage, it would be useful consider some examples of variables for which this kind of modeling would be appropriate. An useful example of the type of data that we can analyze with time series modeling, would be *business cycles.* These are defined as fluctuations found in aggregate economic activities in a given country, typically calculated as the deviations of per capita gross domestic product from its long-term growth path. The crests are called *booms* and the *troughs* are called *recessions.* The important point to note here is that these are not regular cycles with given frequencies. Economists and the policy-makers have been busy predicting the timing of these economic booms and recessions for a very long time, with mixed success. The lack of success provides evidence of the fact that such oscillations are clearly not periodic in nature. Other examples would be asset prices (like stocks and bonds) or in the context of physical world, one can apply this toolkit to modeling of temperature, rainfall and so forth.

4.2 Stationary time series

In this section, we will analyze well behaved time series, roughly corresponding to time series that do not exhibit explosive tendency over time and the underlying data generating process does not change over time. However, we need a formal definition of exactly what we mean by *well behaved.* The way to capture the idea is to consider the concept of *stationarity.*

4.2.1 What is stationarity?

There are two types of stationarity. First we consider the strong version of stationarity.

Definition 4.4. A time series is strongly stationary if the joint distribution of $x_{t_1}, \ldots, x_{t_k}$ remains identical with respect to the joint distribution of the variables shifted over τ time-points – $x_{t_1+\tau}, \ldots, x_{t_k+\tau}$ for all k and τ.

The definition of *strong stationarity* is too strong as it requires the joint probability density function to be preserved. For our purpose, we need a weaker definition. The following version is called weakly stationary process. Also, sometimes it is called *covariance stationary* process.

Definition 4.5. A weakly/covariance stationary stochastic process is one for which the mean $E(x_t) = c$ where c is independent of time t, the variance $Var(x_t) = \sigma^2$ where σ is independent of time t, and finally, the covariance $Cov(x_t, x_{t-j}) = \gamma_j$ i.e., it depends only on the lag j.

It is worthwhile to note that a strongly stationary series is not necessarily weakly stationary (e.g. may not have finite second moment) and vice versa (e.g. higher order moments can be time varying).

4.2.2 Building a time series model

So far we have described and characterized different properties of a time series vector. Now, we would like to build a model that can mimic the properties of a given time series $\{x_t\}$. The building block of time series models is the *white noise* process.

Let us denote a white noise process by the variable ε_t. The idea behind a white noise process is that the distribution of ε_t does not change over time and an outcome of ε_t does not affect the probability of any future outcomes. In short, a simple representation of white noise would be an independent and identically distributed variable. In order to work with ε_t, we impose some more desirable properties. Generally, we assume that the variable is normalized to have zero mean and finite variance. Formally, a *white noise* process would exhibit zero serial correlation and homoskedasticity. Before we get into the discussion about white noise, here is a fun trivia. One may ask – why is white noise called 'white'? The answer is that if one constructs the spectrum following the discussion in section 4.1.2 above, then it can be shown that all frequencies would be present equally, a property which resembles white color. That is why the process is called white noise.

A simple and very commonplace example of a white noise is normally distributed independent draws:

$$\varepsilon_t \sim \text{i.i.d. } N(0, \sigma_\varepsilon^2). \tag{4.19}$$

Then, the implications of the above mentioned assumption would be: for all t, $E(\varepsilon_t) = 0$, $Cov(\varepsilon_t, \varepsilon_{t-j}) = 0$ for all j and $Var(\varepsilon_t) = \sigma_\varepsilon^2$.

Given this simple stochastic process, we can build a wide range of time series models simply by linearly combining the building blocks i.e. ε_ts. But before discussing the model building exercise, we need to establish a crucially important result regarding the scope of all the models we can build. Suppose, we can build only a small set of models with our so-called building blocks, the ε_ts. Then if we observe a time series that falls outside this class of models, we will have to look for newer toolkits and modeling methodologies. Fortunately for us, there exists a very useful and influential result called the *Wold decomposition theorem* which basically states that any weakly stationary process would be amenable to modeling via a linear combination of white noise. Thus, as long as we can ensure weak stationarity of the process, we are assured to have a model for it.

Theorem 4.6 (Wold decomposition theorem (Prop. 4.1, Hamilton (1994)))**.** *Any mean zero covariance stationary process $\{x_t\}$ can be represented as the sum of one deterministic and one linearly deterministic components. Mathematically, one can write the decomposition as*

$$x_t = \sum_{j=0}^{\infty} \beta_j \varepsilon_{t-j} + \theta_t \tag{4.20}$$

where ε_t represents white noise, $\beta_0 = 1$, the possibly infinite sequence β_j is square summable (i.e. $\sum_{j=1}^{\infty} \beta_j^2 < \infty$), and θ_t is deterministic in the sense that a linear function defined over the past values of $\{x_t\}$ can predict θ_t arbitrarily well.

The tremendous importance of this theorem arises from the fact that any weakly stationary process has this representation in terms of linear combination of white noise. This is a very useful feature that we will utilize below when building time series models.

4.2.3 Introduction to the ARMA class of models

AutoRegressive Moving Average (ARMA) models are created by literally taking linear combinations of white noise. First, we will set up the model and then describe a succinct representation of the model in terms of lag polynomials, which make analysis of the time series in terms of their correlation structure much easier. Then we extend the system of equations to consider multiple variables at once. Readers can refer to Hamilton (1994) for a very detailed and analytical introduction to the ARMA models.

The simplest model of this class, autoregressive process of order 1, in short AR(1), is given by

$$x_t = \alpha x_{t-1} + \varepsilon_t \tag{4.21}$$

where α is a constant, often called the *AR* coefficient. To get a sense of the process, let us say that time starts from $t = 0$ when the value of the variable x is x_0. Then we can recursively construct the sequence by adding white noise as follows:

$$\begin{aligned} x_1 &= \alpha x_0 + \varepsilon_1 \\ x_2 &= \alpha x_1 + \varepsilon_2 \\ &\vdots \end{aligned} \tag{4.22}$$

and so forth. A different but complementary class of models is produced by moving average processes. A MA(1) process is defined as follows:

$$x_t = \beta\varepsilon_{t-1} + \varepsilon_t. \tag{4.23}$$

These processes can easily be generalized to multiple lags. We can define an $AR(p)$ process as follows:

$$x_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + \varepsilon_t, \tag{4.24}$$

and a $MA(q)$ process as follows:

$$x_t = \sum_{j=0}^{q} \beta_j \varepsilon_{t-j}. \tag{4.25}$$

By combining the AR and the MA terms, we get the general form $ARMA(p, q)$:

$$x_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i x_{t-i} + \sum_{j=1}^{q} \beta_j \varepsilon_{t-j} + \varepsilon_t \tag{4.26}$$

where α_0 is a constant and β_0 is normalized to 1. Without loss of generality, we assume that $E(x) = 0$ i.e. the process has a zero mean.

Note that we have not specified the range of values that the parameter vectors $\{\alpha_i\}$ and $\{\beta_j\}$ can take. Intuitively it would be obvious that they are not unrestricted. For example, consider the AR(1) process described in equation 4.21. If the coefficient $\alpha > 1$, then the sequence of values of x will explode. Therefore, it will not be stationary series any more (recall that for weak stationarity, the series must possess finite second moment along with satisfying other conditions). Thus we need some restrictions on the parameter values to make sure that the processes remain stationary. The standard way to do it is to consider a representation in terms of lag polynomials.

Representation in terms of lag polynomials

First, we have to define lag or backshift operator. This operator simply allows us to go back and forth on a time series by advancing and retreating in time. While either of the names are fine, here we will call it backshift operator and use the notation $\mathcal{B}$ to denote it. There are some textbooks which use the notation L to denote the same (for example Hamilton (1994)). Formally, we write

$$\mathcal{B}(x_t) = x_{t-1}. \tag{4.27}$$

While in principle we can also work with its inverse lead operator

$$\mathcal{B}^{-1}(x_{t-1}) = x_t, \tag{4.28}$$

usually it is easier to describe it in the form of lags. This operator provides a very useful and succinct way to deal with the parameter vectors $\{\alpha_i\}$ and $\{\beta_j\}$ along with the lagged values of x_t and the white noise ε_t. We can define a polynomial over the lag operators in the following way:

$$\alpha(\mathcal{B}) = \sum_{i=0}^{n} \alpha_i \mathcal{B}^i \quad \text{and} \quad \beta(\mathcal{B}) = \sum_{j=0}^{n} \beta_j \mathcal{B}^j. \tag{4.29}$$

In this notation, AR(1) process can be written $(1 - \alpha\mathcal{B})x_t = \varepsilon_t$ and MA(1) process can be written as $x_t = (1 + \beta\mathcal{B})\varepsilon_t$, where we have ignored the subscripts of α and β since there are only single instances of them in each process. The generalized version for an ARMA(p, q) process can be written concisely as

$$\alpha(\mathcal{B})x_t = \beta(\mathcal{B})\varepsilon_t. \tag{4.30}$$

As will be evident in the following text, this representation in terms of lag polynomials is very useful for two purposes. One, it allows us to interchange between the AR and MA representations, and in parallel, the invertibility of lag polynomials give us the conditions for existence of stationary process.

Let us first elaborate on the ease of going from one representation to the other. Let's consider the process from equation 4.21

$$x_t = \alpha x_{t-1} + \varepsilon_t \quad \text{where } |\alpha| < 1. \tag{4.31}$$

The condition $|\alpha| < 1$ is required to ensure stationarity of the process. For real data, this parameter needs to be estimated. For the time being, let us assume that this condition is satisfied. The easiest way to think about mapping x_t into the values of the white noise ε_t is to simply recursively substitute terms:

$$\begin{aligned} x_t &= \alpha x_{t-1} + \varepsilon_t \\ &= \alpha^2 x_{t-2} + \alpha\varepsilon_{t-1} + \varepsilon_t \\ &\vdots \\ &= \alpha^t x_0 + \sum_{j=0}^{t-1} \alpha^j \varepsilon_{t-j}. \end{aligned} \tag{4.32}$$

Note that here we already see why $|\alpha| < 1$. Specifically, under that condition, if we assume that the process started infinite periods ago, we have

$$x_t = \sum_{j=0}^{\infty} \alpha^j \varepsilon_{t-j}. \tag{4.33}$$

There is an alternative method to get the same expression. Using lag operator, we know that the AR(1) process can be written as

$$(1 - \alpha\mathcal{B})x_t = \varepsilon_t. \tag{4.34}$$

We can expand on the expression (assuming $|\alpha| < 1$):

$$\begin{aligned} x_t &= \frac{\varepsilon_t}{(1-\alpha\mathcal{B})} \\ &= (1 + \alpha\mathcal{B} + \alpha^2\mathcal{B}^2 + \alpha^3\mathcal{B}^3 + \ldots)\varepsilon_t \\ &= \varepsilon_t + \alpha\varepsilon_{t-1} + \alpha^2\varepsilon_{t-2} + \ldots \\ &= \sum_{j=0}^{\infty} \alpha^j \varepsilon_{t-j}. \end{aligned} \tag{4.35}$$

Following the same logic, we can expand AR(p) process to MA(∞) as long as the lag polynomials are invertible. To see why invertibility is important, recall from equation 4.26 that a standard ARMA(p, q) model can be written as

$$\alpha(\mathcal{B})x_t = \beta(\mathcal{B})\varepsilon_t. \tag{4.36}$$

Assuming that the lag polynomials $\alpha(\mathcal{B})$ and $\beta(\mathcal{B})$ are invertible, we write the same process in two different fashions:

$$\begin{aligned} x_t &= \alpha(\mathcal{B})^{-1}\beta(\mathcal{B})\varepsilon_t, \\ \varepsilon_t &= \beta(\mathcal{B})^{-1}\alpha(\mathcal{B})x_t. \end{aligned} \tag{4.37}$$

Note that in the above calculations (for example, in equation 4.35), we have treated the lag polynomial as a regular algebraic polynomial and conducted the Taylor expansion in the usual way. One question naturally arises as to whether this is a permissible operation or not. The short answer is yes, it is a permissible operation. We will not elaborate on the reason here. Interested readers can consult Hamilton (1994) for a detailed description covering in particular, the fact that lag operator satisfies commutative, associative and distributive laws for multiplication and addition.

4.2.4 Generalization to vector autoregression model

So far we have dealt with only one time series $\{x_t\}$. In many cases, multiple time series might naturally coexist. For example, one can consider asset returns in a multi-asset market or growth rates of different firms in an economy. The interesting feature of such a scenario is that not only the variables have non-trivial time–dependence, they also affect each other. Thus there may exist mutual interaction across the variables. *Vector autoregression* model (VAR model, in short) is a generalized version of the simple autoregression model that allows us to capture such mutual dependence across variables. It is worthwhile to note that in principle, one can describe a *vector autoregressive moving average* model as well. However, typically a well specified VAR model provides enough flexibility to capture the patterns in the data and it is less complicated than a VAR model augmented with moving average terms. Also, a VAR model can be written as a combination of infinite number of moving average terms. Therefore, in the following we will focus only on the VAR setup for simplicity and tractability.

Let us begin with a two-variables example with self and cross-dependence on only one lag. Let us imagine we have two variables x_{1t} and x_{2t} that are related to each others' lagged values linearly and they themselves also depend on their own past values. In vector notation, the variables and the corresponding white noise terms can be written as

$$x_t = \begin{pmatrix} x_{1t} \\ x_{2t} \end{pmatrix}, \qquad \varepsilon_t = \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix}. \tag{4.38}$$

We follow the same structure for the white noise as before except that now we allow the series to be correlated to each other. Thus the expectation of the error terms is

$$E(\varepsilon_t) = 0. \tag{4.39}$$

and the variance-covariance matrix is

$$E(\varepsilon_t \varepsilon_t') = \begin{pmatrix} \sigma^2_{\varepsilon_1} & \sigma_{\varepsilon_1\varepsilon_2} \\ \sigma_{\varepsilon_2\varepsilon_1} & \sigma^2_{\varepsilon_2} \end{pmatrix}. \tag{4.40}$$

Note that we are allowing them to be correlated with each other as the off-diagonal terms are not necessarily zero. If we assume that there is no correlation between the error terms, then we will have

$$E(\varepsilon_t \varepsilon_t') = \begin{pmatrix} \sigma^2_{\varepsilon_1} & 0 \\ 0 & \sigma^2_{\varepsilon_2} \end{pmatrix}. \tag{4.41}$$

Lack of time-lagged correlation implies

$$E(\varepsilon_t \varepsilon_{t-j}') = 0 \quad \text{for } j = 1, 2, \ldots \tag{4.42}$$

In matrix form, the relationship between variables x_{1t} and x_{2t} can be expressed as

$$\begin{pmatrix} x_{1t} \\ x_{2t} \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix} \begin{pmatrix} x_{1,t-1} \\ x_{2,t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix}. \tag{4.43}$$

In short, we can write the same model as

$$x_t\big|_{2\times 1} = \alpha\big|_{2\times 2}.x_{t-1}\big|_{2\times 1} + \varepsilon_t\big|_{2\times 1}, \tag{4.44}$$

which is exactly the same as equation 4.21 except that x now represents a vector and α represents the matrix of coefficients (the dimensions are mentioned after the | sign for each vector and matrix).

This formulation is known as VAR model of order 1 or simply VAR(1). There is no reason to restrict ourselves with only one lag and only two interacting variables. In general, we can consider a vector of n-variables ($x_t = [x_{1t}\ x_{2t}\ \ldots x_{nt}]'$) with an $n \times n$ coefficient matrix $[\alpha_{ij}]$ and a vector of error terms ($\varepsilon_t = [\varepsilon_{1t}\ \varepsilon_{2t}\ \ldots \varepsilon_{nt}]'$). An n-variable VAR(1) model can be written in the same form

$$x_t\Big|_{n\times 1} = \alpha\Big|_{n\times n}.x_{t-1}\Big|_{n\times 1} + \varepsilon_t\Big|_{n\times 1}. \tag{4.45}$$

Mentioning the dimensions for every vector and matrix would be cumbersome. A easier way is to express the same model would be to consider the vector representation for an n-variable VAR(1) model:

$$\mathbf{x}_t = \alpha \mathbf{x}_{t-1} + \varepsilon_t \tag{4.46}$$

where both $\mathbf{x}$ and ε represent $n \times 1$ vectors. We can generalize it further to consider a VAR(p) model i.e. a VAR model with p lags as follows:

$$\mathbf{x}_t = \sum_{l=1}^{p} \boldsymbol{\alpha}_l \mathbf{x}_{t-l} + \varepsilon_t \tag{4.47}$$

where $\boldsymbol{\alpha}_l$ represents $n \times\ n$ matrix of coefficients for l varying between 1 to p.

4.2.5 Finding moments

In this section, we want to develop some techniques for finding unconditional moments of a given ARMA process. To fix ideas, let us first consider an $AR(1)$ process (as in equation 4.21) with $|\alpha| < 1$. By repeated substitutions, we can express it as an MA(∞) process (equation 4.35)

$$x_t = \sum_{j=0}^{\infty} \alpha^j \varepsilon_{t-j}. \tag{4.48}$$

There are two useful methods to find the moments. Let us first describe the straightforward method by the help of a simple example. Say we need to find the first moment of the variable x. We can directly take expectation of equation4.21 to get

$$E(x_t) = \sum_{j=0}^{\infty} \alpha^j E(\varepsilon_{t-j}). \tag{4.49}$$

Since ε is a white noise and by assumption has zero mean, we can easily derive

$$E(x_t) = 0. \tag{4.50}$$

The second one uses a trick based on the idea of stationarity. Recall that for a stationary process, the first and the second moments are constant. Therefore, by taking expectation on the AR(1) process (equation 4.21), we can write

$$E(x_t) = \alpha E(x_{t-1}) + E(\varepsilon_t), \tag{4.51}$$

which can be rewritten as

$$(1 - \alpha)E(x_t) = E(\varepsilon_t) \tag{4.52}$$

implying that

$$E(x_t) = 0. \tag{4.53}$$

Notice that we have used equated the expectation terms $E(x_t) = E(x_{t-1})$, which is implied by weak stationarity.

This trick becomes more useful for finding the second moment. On the same equation 4.21, we can apply variance operator on both sides to derive

$$var(x_t) = \alpha^2 var(x_{t-1}) + \sigma_\varepsilon^2. \tag{4.54}$$

The covariance term between x_{t-1} and ε_t is zero since ε_t is a white noise. By using the same trick, we see that $var(x_t) = var(x_{t-1})$ due to stationarity. Therefore, we can write

$$var(x_t) = \frac{\sigma_\varepsilon^2}{1-\alpha^2}. \tag{4.55}$$

Analogous calculations will work for a VAR(p) process, except that one needs to account for the number of variables properly and the solution would be in matrix form.

Autocorrelation function of ARMA process

In this section, we will discuss how to find autocorrelation functions (*a.c.f.* henceforth) of ARMA processes. This is an useful tool to get a sense of the underlying data generating process. Note that the AR and the MA representations are clearly not unique as we can convert one into the other (as long as the corresponding lag polynomials are invertible). Thus it would be useful to get an unique representation of the process. The *a.c.f.* serves two other important purposes. One, it will help us to guess the structure of an ARMA process to fit a given set of data. Note that so far we have worked with ARMA(p, q) process by relying on the Wold decomposition theorem that an ARMA process will allows us capture any given weakly stationary stochastic process. But we have not discussed yet, given a dataset how to find the best fit of an ARMA model. We will see that the *a.c.f.* sheds some light on that. Two, it tells us about persistence of a process. A highly persistent process will have high autocorrelation coefficients and a process with very low persistence will have low autocorrelation coefficients.

Let us start with the simplest example of a white noise $\varepsilon_t \sim \text{iid}(0, \sigma_\varepsilon^2)$. We can immediately see that $\gamma_0 = \sigma_\varepsilon^2$ and $\rho_0 = 1$. For all lags j larger than 0, clearly $\gamma_j = 0$ and therefore, $\rho_0 = 0$.

A useful and non-trivial exercise would be to find the *a.c.f.* of an AR(1) process (equation 4.21). First, we have to find out the autocovariances at different lags. Lag-0 is the easiest:

$$\gamma_0 = \text{var}(x_t) = \frac{\sigma_\varepsilon^2}{1-\alpha}. \tag{4.56}$$

Since the process has zero mean,

$$\gamma_1 = E(x_t x_{t-1}) = \frac{(\alpha\sigma_\varepsilon^2)}{(1-\alpha)} = \alpha\gamma_0 \tag{4.57}$$

Continuing in the same way, we get

$$\gamma_2 = E((\alpha x_{t-1} + \varepsilon_t)x_{t-2}) = \alpha^2 E(x_{t-2}^2) = \alpha^2\gamma_0 \tag{4.58}$$

and so on. The pattern is obvious. We can easily derive the autocorrelation function by dividing each autocovariance value by the variance, and generate the *a.c.f.* as follows: $\rho_1 = \alpha$, $\rho_2 = \alpha^2$ and so forth. Generally, we write

$$\rho_j = \alpha^j \tag{4.59}$$

for all j.

There is an alternative method to generate the *a.c.f.* by utilizing the MA(∞) representation. Recall that equation 4.21 can also be written as equation 4.33. One can try to directly find the *a.c.f.* by direct substitution of the expression in the equations for autocovariances. To help thinking about the idea, below we provide the calculations for *a.c.f.* of an MA(1) process. Consider the process given in equation 4.23. At lag $j = 0$, we can easily find

$$\gamma_0 = \text{var}(\beta\varepsilon_{t-1} + \varepsilon_t) = (\beta^2 + 1)\sigma_\varepsilon^2. \tag{4.60}$$

For the second equality, we have utilized the fact that ε_ts are serially uncorrelated. Next, we calculate the autocovariance at lag $j = 1$:

$$\gamma_1 = E[(\beta\varepsilon_{t-1} + \varepsilon_t)(\beta\varepsilon_{t-2} + \varepsilon_{t-1})] = \beta\sigma_\varepsilon^2. \tag{4.61}$$

Next, we calculate the autocovariance at lag $j = 2$:

$$\gamma_2 = E[(\beta\varepsilon_{t-1} + \varepsilon_t)(\beta\varepsilon_{t-3} + \varepsilon_{t-2})] = 0. \tag{4.62}$$

Clearly, the autocorrelation is exactly zero. The intuition is important here to understand the nature of the process. Recall that in the MA(1) process in equation 4.23, the right-hand side consists of only two white noise terms ε_t and ε_{t-1}. If we lag the variable x_t by 2 periods, then the corresponding terms would be ε_{t-2} and ε_{t-3}. Note that neither of these terms have any overlap with ε_t and ε_{t-1}. Therefore, the cross-correlation values have to be zero. Naturally, the same logic can be applied for all lags $j > 2$ as well and it can be easily shown that

$$\gamma_j = 0 \quad \forall\, j > 1. \tag{4.63}$$

Therefore, we can write down the autocorrelation function now as

$$\rho_0 = 1 \tag{4.64}$$

$$\rho_1 = \frac{\beta}{1 + \beta^2} \tag{4.65}$$

$$\rho_j = 0 \;\; \forall\, j > 1. \tag{4.66}$$

There are more general and useful techniques to find a.c.f., e.g. Yule-Walker equations. Here we would not describe the procedures. Interested readers may consult textbooks like Brockwell and Davis (2016) and Hamilton (1994). Most of the softwares and time series toolboxes on programming language environments allow users to generate the empirical *a.c.f.* from the data very easily. Such plots of *a.c.f.* gives us a preliminary understanding of the number of MA lags to incorporate when trying to fit an ARMA model. Also, we get a sense of persistence in the data. It should be noted in this context that the empirical *a.c.f.* alone is often is not precise enough

Table 4.1 Description of the visually identifiable properties of the autocorrelation functions and the partial autocorrelation functions for stationary AR(p), MA(q) and ARMA(p, q) models

Process	*a.c.f.* ($\rho(j)$)	*p.a.c.f.* ($\pi(j)$)
AR(p)	infinite length; dampened exponential or sinusoidal curve	finite length; for lags $j > p$, $\pi(j) = 0$
MA(q)	finite length; for lags $j > q$, $\rho(j) = 0$	infinite length; dampened exponential or sinusoidal curve
ARMA(p, q)	mimics AR(p) for lags $j > q$	mimics MA(q) for lags $j > p$

to clearly differentiate between specifications of ARMA models. For that purpose, we will develop some new ideas in the following sections.

Before wrapping up this discussion, let us also mention that there is a concept of *partial autocorrelation.* This measure $\pi_x(j)$ for $j \geq 2$ is defined as the coefficient β_j from the optimal linear prediction of x_t on the basis of the j-th previous observation, accounting for all intermediate observations:

$$x_t = \beta_1 x_{t-1} + \beta_2 x_{t-2} + \ldots + \beta_t x_{t-j} + u_t. \tag{4.67}$$

The same equation can be easily modified to account for a process with non-zero mean as well.

This measure allows us to check the relationship between pairs of observations at a given lag *controlling for* the intermediate observations. The idea behind this operation can be understood easily from the *a.c.f.* of the baseline AR(1) process as in equation 4.21. Recall from equation 4.59 that the *a.c.f.* decays as α^j for the j-th lag. Therefore, say if $j = 4$, we will have $\rho_4 = \alpha^4$. But given the form of the process in equation 4.21, there is no direct relationship between the observations made at times t and $t - 4$. Partial autocorrelation captures precisely this point. One can show that for AR(p) processes, the partial autocorrelation after p lags would be zero, whereas for MA(q) processes it will continue indefinitely. In this way, it mirrors the properties of the standard *a.c.f.* for which the values are zeros after q lags for MA(q) processes and continues indefinitely for AR(p) processes. It is useful to gather these properties in one place for references. In Table 4.1, we have described the properties of the *a.c.f.* and the *p.a.c.f.*

As an example, we show the sample ACF and PACF of the GDP growth rate shown in figure 4.1, in figure 4.3. We have not plotted the error bars for the point estimates to keep the figure clean. Many of the point estimates may turn out to be statistically insignificant, especially at higher lags.

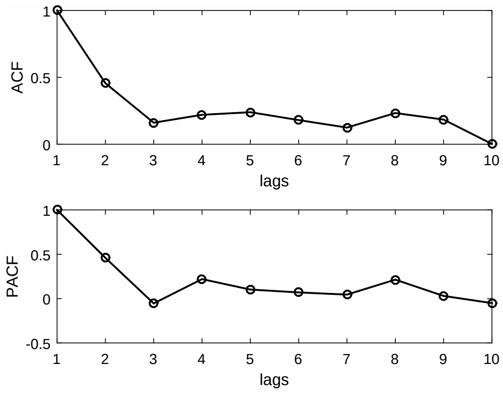

Figure 4.3 Sample ACF and PACF of the GDP growth rate shown in Figure 4.1.

4.2.6 Estimation procedure

So far we have discussed about the ARMA and its multivariate generalization as theoretical objects and analyzed their properties. Now we are in a position to ask the most important question from a data-oriented point of view: Given a series of observations $X_T = \{x_1, x_2, \ldots, x_T\}$, how can we fit an ARMA(p, q) model? Also, if we have a set of multivariate observations, how can we fit a VAR model to it? A multivariate ARMA model can also be estimated in the exact same way. However, it is more commonplace to see VAR models rather than VARMA models since a VARMA model requires a much larger number of parameters to be estimated and often the lagged dependence on the error terms do not have theoretical justifications. In the present context, we will focus only on ARMA models and VAR models, as is standard in the literature.

Let us start from the univariate context. We rely on the fundamental result obtained from Wold decomposition theorem that any weakly stationary data generating process can be approximated by ARMA(p, q) process. There are three main steps in the estimation procedure. First, we need to determine the order (p and q) of the process. Second, we need to find the values of the coefficients and estimate the residuals. Third, we then perform tests to check validity of the estimated model. This procedure goes back to to original formulation by Box and Jenkins in 1970s (a recent textbook exposition is given in Box et al. (2015)) and has evolved into a standard method.

Box-Jenkins methodology

Box-Jenkins method is fundamentally concerned about model estimation as well as model selection. The basic recursive model selection technique works as follows: Given a covariance stationary time series –

(i). Identify the orders of a potentially suitable ARMA(p, q) model.

(ii). Estimate the corresponding coefficients under the assumption that the orders are accurate.

(iii). Perform diagnostics on the fitted model and the residuals.

(iv). If the fitted model fails the tests, go back to the first step.

(v). If the model seems satisfactory in terms of the statistical tests, one can use the model for further analysis, e.g. for forecasting.

Model identification

Below, we describe the steps for model identification.

(i). First, given the set of data $\{x_t\}$, we need to check for stationarity. This can be done in two ways:

1. Visual inspection of the data often tells us immediately whether the data is stationary or not.
2. One can also utilize statistical tests to check for stationarity. We will discuss some well known statistical tests like Dickey-Fuller (or Phillips-Perron) tests later in this chapter.

(ii). If the data series is non-stationary, transform the data to achieve stationarity (see section 4.3 for a discussion on non-stationarity).

1. For example, GDP per capita (G_t) can be non-stationary but the growth rate ($g_t = log(G_t) - log(G_{t-1})$) can be stationary.
2. Sometimes, more than one round of differencing might be required to achieve stationarity. Price indices (like consumer price index) often display this kind of behavior.

(iii). The next step is to select the orders p and q.

1. First, compute empirical *a.c.f.* and *p.a.c.f.*. Most of the standard softwares generate empirical *a.c.f.* and *p.a.c.f.* very easily.
2. These plots will give an idea about the orders of the MA and AR lags, respectively.
3. However, the empirical *a.c.f.* and *p.a.c.f.* are often non-informative. One can utilize more sophisticated statistical criteria as well (like AIC, BIC; described below).

(iv). Once we fix the orders p and q, we know the number of parameters to be estimated.

1. For AR(p) processes, ordinary least square estimation can be applied.
2. For general ARMA(p, q) processes, maximum likelihood estimation can be applied.

(v). Next, we check for autocorrelation in the residuals. The idea is that the residuals of a good model should not be autocorrelated. All autocorrelated components in the data should already be explained by the model itself.

1. One can perform Ljung-Box test. Other possibilities are using tests like Breusch-Godfrey and/or Durbin-Watson tests.
2. If the residuals do not have any autocorrelation, then the model is acceptable.
3. Else, we need to modify the specification by going back to the first step above.

Estimation of the lag orders

Note that the above procedure depends crucially on the fact that we can recursively define the models and check for their suitabilities. Thus if we start with a small number of lags, we may want to go for a model with larger number of lags if the original model is not suitable and vice versa. There are two factors that we have to be careful about while choosing the lags. In one extreme, we can pick very large p and q leading to overfitting the data. In such cases, we will have high in-sample fit but poor out-of sample performance. In the other extreme, we can pick very small p and q. Then the model underfits the data leading to low in-sample fit but better out-of sample performances. Either way it is bad since the maximum likelihood estimator would not be consistent with a mis-specified ARMA(p, q) model. We have already noted that using *a.c.f.* and *p.a.c.f.* gives us a preliminary understanding of the lag orders. But visual inspection may not be very accurate, especially for small samples and visual inspection is certainly not a good substitute of a formal statistical test. Below, we will discuss the usage of information criteria to resolve this issue.

Information criteria

This is a commonly used tool for model selection from a set of competing models that can explain the same set of data to varying degrees. The basic goal of this exercise is to minimize some kind of *information loss* to come up with a better model. A model which loses lesser amount of information can be taken to be a better model (see also the discussion in chapter 3.3.7). If we over-parameterize, then the in-sample fit has to necessarily improve, but that is not good from the perspective of using the model for prediction. Intuitively if we try to fit each and every squiggle on a curve, then the fitted model will perform very badly to capture the average behavior of the process in out-of-sample calculations.

Let us denote the fit of the model by the variance of the residuals ($\hat{\sigma}^2_{p,q}$). Clearly, if the variance of the residuals reduce then the model has better in-sample fit. Thus there would be a tendency to increase p and q to reduce the variance. The objective is to penalize higher values of p and q and correct for $\hat{\sigma}^2_{p,q}$ to attain a balance. The problem statement boils down to minimizing information criteria IC with respect to p and q where the information criteria is defined as the variance of the residuals along with a penalty term for over-parametrization (pp. 101, Neusser (2016)):

$$\text{Akaike IC } (AIC) \;=\; \log(\hat{\sigma}^2_{p,q}) + \frac{2}{T}\Big(p+q\Big), \tag{4.68}$$

Table 4.2 AIC and BIC estimated for ARMA(p, q) estimation on the GDP growth rate data from figure 4.1

ARMA(p,q)	0	1	2	3
0	(225.05,229.2)	(213.03,219.26)	(214.81,223.12)	(216.35,226.74)
1	(213.32,219.56)	(214.68,222.99)	(213.73,224.12)	(214.93,227.4)
2	(215.15,223.46)	(216.21,226.59)	(215.22,227.69)	(216.17,230.72)
3	(214.31,224.69)	(214.71,227.17)	(216.67,231.21)	(218.02,234.64)

$$\text{Bayesian or Schwarz IC } (BIC) = \log(\hat{\sigma}^2_{p,q}) + \frac{\log(T)}{T}\Big(p+q\Big), \tag{4.69}$$

and

$$\text{Hannan-Quinn IC } (HQIC) = \log(\hat{\sigma}^2_{p,q}) + \frac{2(\log(\log(T)))}{T}\Big(p+q\Big). \tag{4.70}$$

While there are many other model selection techniques, these three measures are standard and most of the softwares would produce these values. It might be noted that as a general rule, BIC and HQIC penalizes over-parameterization more than AIC and hence typically economizes on the number of parameters. However, simulation results indicate that AIC might pick a better model (closer to the true data generating process) in small samples. Finally, we note that none of these criteria clearly dominates the rest.

In table 4.2, we provide the estimated AIC and BIC values from ARMA(p, q) models on the GDP growth rate data from figure 4.1. Each cells refers to (AIC,BIC) value of corresponding ARMA(p, q) model. i.e. first cell can be read as ARMA(0,0) model has AIC value of 225.05 and BIC value of 229.2. Row values correspond to lag p (AR(p)) and column values correspond to lag q (MA(q)) in each cell. Both lags vary from 0 to 3. One can create a larger model as well. While a larger model fits the data better, most of the time it does not have any interpretation. Generally, the preference is for choosing smaller models. In terms of AIC and BIC, AR(1) and MA(1) models seem to be good contenders for the best model in the present context. However, in many cases, the model choice due to AIC and BIC do not coincide. An additional criteria could be model interpretation. If a model is more economized than others in terms of parameters or more easily explainable and have comparable AIC/BIC values, then one can go for that model.

Estimation of ARMA(p, q) parameters

So far we have described the procedure of lag selection. Now, we describe the final step of estimating the parameters once the lags have been optimally selected. We need to provide a clarification here. Computer programs often do the estimation of parameters jointly with information criteria among many other variables. In table

4.2 for example, we have provided only the AIC and BIC. But for each estimation we already had the estimated parameter values as well. Here it is only for descriptive purposes that we are describing parameter estimation sequentially after describing information criteria.

We will proceed in two steps. First we will discuss two methods to estimate AR(p) processes and then, we will discuss a general maximum likelihood-based method for estimating ARMA(p, q) process. The reason for describing the first two estimation techniques are two-fold. First, the methods shed light on the working of the AR(p) processes in two different ways and that is often useful to know. Secondly, such a discussion also shows that the AR(p) processes are amenable to multiple types of estimation including the standard least square method. The following discussion has been partially inspired by the discussion on the estimation of ARMA models in Neusser (2016).

Let us say we have a model of the form (same as in equation 4.24):

$$x_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + \varepsilon_t \tag{4.71}$$

The goal is to estimate the parameter vector $\{\alpha_1,\ \alpha_2, ...\alpha_p\}$ from a given set of observations X_t. We will describe two methods to estimate the AR(p) process, viz. Yule-Walker procedure and the ordinary least square procedure.

Yule-Walker method: The easiest way to see the working of this method is to simply multiply equation 4.71 by x_{t-k} for $k = 0, ..., p$ on both sides and applying expectation operator on them. For example, multiplication by x_t and applying expectation would give us

$$\gamma_0 = \sum_{i=1}^{p} \alpha_i \gamma_i + \sigma_\varepsilon^2. \tag{4.72}$$

Multiplication by x_{t-1} and applying expectation would give us

$$\gamma_1 = \sum_{i=1}^{p} \alpha_i \gamma_{i-1}. \tag{4.73}$$

Note that x_{t-1} has no matching term for ε_t and therefore, the covariance value is zero. This way one can keep on multiplying and generate $p+1$ equations involving autocovariances of $\{x_t\}$ and the parameter vector $\{\alpha_1, ..., \alpha_p, \sigma_\varepsilon^2\}$. Solving this set of equations allow us to estimate values of the parameter vector $\{\hat{\alpha}_1, ..., \hat{\alpha}_p, \hat{\sigma}_\varepsilon^2\}$.

Ordinary Least Square (OLS) method: Consider the same process as in equation 4.71. We can treat it directly as a regression and estimate it via OLS method. We can assume that x_t is the endogenous variable, $x_{t-1},\ x_{t-2}, ...x_{t-p}$ are exogenous variables, ε_t is the error term and finally, the coefficient vector $\{\alpha_1, ..., \alpha_p\}$ is unknown and to be estimated. To apply OLS method, we augment the model with a constant α_0.

$$x_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i x_{t-i} + \varepsilon_t \tag{4.74}$$

Let us collect terms in the form of a matrix. Let

$$Y = \begin{bmatrix} x_{p+1} \\ \vdots \\ x_T \end{bmatrix}, \quad \beta = \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_p \end{bmatrix}, \quad \varepsilon = \begin{bmatrix} \varepsilon_{p+1} \\ \vdots \\ \varepsilon_T \end{bmatrix} \tag{4.75}$$

and

$$X = \begin{bmatrix} 1 & x_p & x_{p-1} & x_{p-2} & \dots & x_1 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_{T-1} & x_{T-2} & x_{T-3} & \dots & x_{T-p} \end{bmatrix}. \tag{4.76}$$

Combining these matrices, we can express the equation 4.74 in a more convenient form

$$Y = X\beta + \varepsilon. \tag{4.77}$$

Therefore, the OLS estimator is

$$\hat{\beta} = (X'X)^{-1}X'Y. \tag{4.78}$$

The derivation of the OLS estimator follows from section 3.3.5 in chapter 3. Interested readers may additionally consult Hamilton (1994) for a time series perspective and Greene (2003) for an econometric exposition.

There are two technical issues with this estimator. First, the regressors (i.e. the right hand side variables in equation 4.74) are clearly correlated with error terms. Secondly, the first p observations ($x_1,\ x_2, \dots,\ x_p$) affect the OLS estimates. However, as Theorem 5.2. in Neusser (2016) (see also Brockwell et al. (1991)) shows, $\hat{\beta}_{OLS}$ asymptotically normal, and asymptotically equivalent of the Yule-Walker estimator (pp. 92 in Neusser (2016)). For ARMA(p, q) models, OLS estimation is problematic since the error terms ($\varepsilon_{t-1}, ..., \varepsilon_{t-q}$) are not observable.

Maximum Likelihood Estimator: To estimate general ARMA(p, q) model, we utilize maximum likelihood estimation. Let us say we have a model of the form (same as in equation 4.26 with $\alpha_0 = 0$):

$$x_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + \sum_{j=1}^{q} \beta_j \varepsilon_{t-j} + \varepsilon_t \tag{4.79}$$

with normally distributed white noise ε_t i.e., $\varepsilon_t \sim N(0, \sigma^2)$. Note that maximum likelihood method is paramteric in nature and we have to start from some distributional assumptions. The most standard assumption is normality. Following Neusser (2016) (pp. 95), we can write the exact Gaussian likelihood function as

$$L(\beta|x) = (2\pi)^{-T/2}|\Gamma(\beta)|^{-1/2} exp\left(-\frac{1}{2}x'\Gamma(\beta)^{-1}x\right) \tag{4.80}$$

where $\Gamma(\beta) = E(xx')$ is the $T \times T$ covariance matrix of x which is a function of β. Theorem 5.3 from Neusser (2016) shows that the maximum likelihood estimator is asymptotically normally distributed. Finally, this estimator is also asymptotically efficient.

The actual computation of the maximum likelihood estimation can be quite involved and we have to resort to numerical techniques to maximize the likelihood function. All popular softwares/programming languages (R, matlab, python, Stata etc.) have these kind of programs inbuilt.

4.2.7 Financial networks from multivariate time series

With the concept developed in terms of estimation of stationary processes, here we discuss an application of the concept to construct financial network from multivariate return series. We discuss the concept of networks and the associated statistics in chapter 6. Here we treat networks as simply collections of a set of nodes and linkages between them. Consider a stock market where N number of stocks are being traded. Each of them will have a price at any given point of time t. We assume that t is discrete and goes from 1 to T. Let us denote the price of the i-th asset at time point t as p_{it} – say this denotes the daily closing price of that stock. This price series itself will be non-stationary. Let us consider the log-return series arising out of it

$$r_{it} = \log(p_{it}) - \log(p_{i,t-1}) \tag{4.81}$$

for all $i = 1,\ 2, ..., N$ and $t = 1,\ 2, ..., T$. This kind of operation of differencing is discussed in more details in section 4.3.

Cross-correlation network

A standard way to construct a network out of the multivariate time series is to consider each stock to represent one node and their correlations (actually, a transformation of that as we will discuss below) to represent the linkages. From N stocks, we can construct a correlation matrix ρ of size $N \times N$. The correlation value itself is not a metric as it can be negative. A standard method is to convert it into a metric by taking a simple transformation

$$w_{ij} = \sqrt{2(1 - \rho_{ij})}. \tag{4.82}$$

This metric was proposed by Gower (1966) (see Mantegna and Stanley (2007)).

In figure 4.4, we show one example of financial network constructed from intraday data sampled at ten seconds frequency, of the largest 50 stocks traded in the National Stock Exchange of India which constitute the NIFTY-50 index, sampled from December 2020 listing (see Bhachech et al. (2022) for a statistical analysis of the networks constructed from the return data). Since the full network will have a large number of edges – specifically, $\binom{N}{2}$ number of edges where $N = 50$, we apply a threshold for the purpose of visualization.

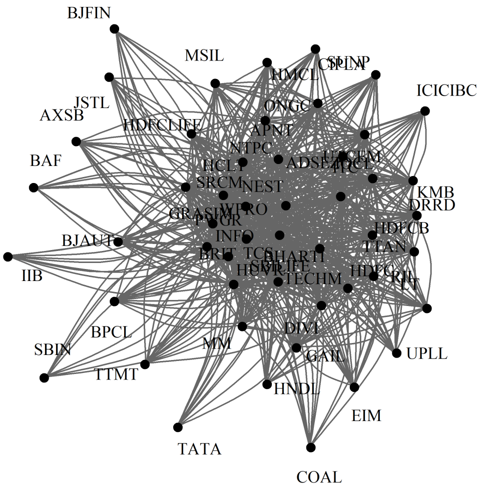

Figure 4.4

Stock return correlation network constructed from intra-day data of Indian firms on a given day. See text for description.

Lagged correlation network

Here we introduce the concept of lagged correlation and Granger causality. Consider the bivariate vector autoregression model we have considered in section 4.2.4 –

$$\begin{pmatrix} x_{1t} \\ x_{2t} \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{pmatrix} \begin{pmatrix} x_{1,t-1} \\ x_{2,t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix}. \tag{4.83}$$

If α_{12} is significant, then we say that x_2 Granger causes (systematically predicts) x_1. If α_{21} is significant, then the direction of causality is reversed. It is possible that neither of the two series cause each other or both do. In general, on ecan allow for a larger number of lags and variables.

This concept allows us to construct lagged correlation network in the form of Granger causal network. Here we show an example. We take fifteen randomly chosen companies featuring in the stock network in figure 4.4 and construct the Granger causal network out of them. We note two things here. One, in general, such a network would be directed. Two, the edges are binary. They either exist or not based on whether the relationships are significant or not. In figure 4.5 we show the Granger causal networks at 5% and 10% levels of significance.

Here we make an observation based on the construction of the Granger causal network. The construction involves comparison of multiple hypothesis at the same time independent of each other (whether the edges exist or not). This leads to the problem of *multiple comparisons* where the false discovery of edges might happen. In such cases, the classical approach is to apply Bonferroni correction or Duncan's

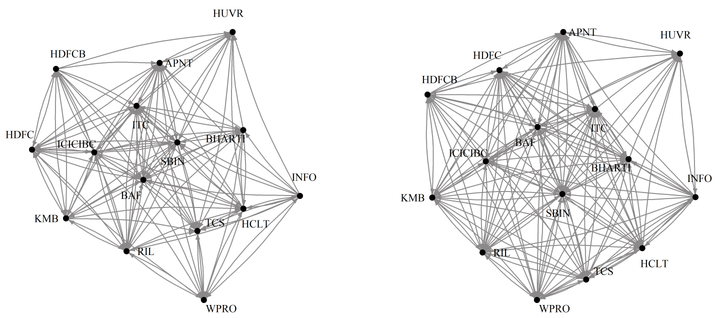

Figure 4.5 Granger causal networks of 15 Indian companies constructed from their intra-day stock returns. The left panel shows the network evaluated at 5% level of significance whereas the right panel shows the one evaluated at 10% level of significance.

correction (Duncan, 1955). This does not fully allay the concern as the Bonferroni correction can be too conservative and Duncan's correction can be too liberal. See Berry and Hochberg (1999) for a Bayesian perspective on this problem.

4.3 Analyses of non-stationary time series

So far we have analyzed stationary time series. Next, we will introduce the concept of non-stationary time series. We can motivate the reasons for studying non-stationary time series in multiple ways. But the best way is to simply refer to the famous statement by D. J. Thomson, who said: "*Experience with real-world data, however, soon convinces one that both stationarity and Gaussianity are fairy tales invented for the amusement of undergraduates*" (Thomson (1994))!

Non-stationarity might arise in a number of ways. The time series can have a deterministic trend (said to exhibit trend stationarity), there can be regime shifts in level or changes in the variance. Finally, the time series can possess unit roots (also referred to as time series with stochastic trend). In the present discussion, we will discuss time series with trends, deterministic and stochastic types of non-stationarity. For the sake of discussion, we will ignore seasonal fluctuations in the following. For example, both the GDP series shown in figure 4.1 and the evolution of market capitalization of Bitcoins shown in figure 4.2 are non-stationary. We will describe the exact meaning of the term below.

A process $\{x_t\}$ is called trend stationary if

$$x_t = f(t) + \zeta_t \tag{4.84}$$

where ζ_t is a stationary process. For example, let us say ζ_t is some stationary

ARMA(p, q) process and x_t has the form –

$$x_t = \sum_{i=0}^{k} a_i t^k + \zeta_t. \tag{4.85}$$

for some positive integer k and constant terms a_i. Then x_t is a trend stationary process. On the other hand, a process x_t is called difference stationary if

$$\Delta^d x_t = \zeta_t \tag{4.86}$$

where Δ denote difference operator such that $\Delta\omega_t = \omega_t - \omega_{t-1}$ for a process ω_t, d is the order of differencing and ζ_t is stationary. This process is very useful for many reasons. The most well known case is probably that of the random walk. Let's say, $d = 1$ and ε_t is an *i.i.d.* variable. Note that an an *i.i.d.* variable is stationary by definition and therefore, ε_t is a candidate for ζ_t in equation 4.86 above. Then clearly the above equation reduces to an AR process with the AR coefficient being one –

$$x_t = x_{t-1} + \varepsilon_t, \tag{4.87}$$

which is a simple random walk process. We will discuss below how difference stationarity can be modeled.

4.3.1 AR process with unit root

Here we will analyze the process from the point of view of the ARMA framework we have developed before. Consider the process described above in equation 4.87. Note that the corresponding lag polynomial can be written as

$$\alpha(\mathcal{B}) = 1 - \alpha_1 \mathcal{B} \tag{4.88}$$

where $\alpha_1 = 1$. The corresponding characteristic polynomial is $\alpha(z) = 1 - z$. Evidently, it has a unit root since $\alpha(1) = 0$. By backward substitution, we get (assuming that the process started finite periods back at time point $t = 0$)

$$x_t = x_0 + \varepsilon_1 + \ldots + \varepsilon_t. \tag{4.89}$$

An important point to notice here is that the effect of any shock remains forever. The best way to see this is to consider an AR(1) process with a coefficient less than one (as in equation 4.21):

$$x_t = \alpha^t x_0 + \alpha^{t-1}\varepsilon_1 + \ldots + \varepsilon_t. \tag{4.90}$$

If $|\alpha| < 1$, then the effect of any shock on x_t through ε_k for some k clearly decays over time as the multiplier α^{t-k} goes to zero. Since for a random walk we have $\alpha = 1$, the effect of the shock does not decay and is described as 'permanent'.

From the above expression (equation 4.89), we can easily find the variance of the process

$$Var(x_t) = t\sigma_\varepsilon^2. \tag{4.91}$$

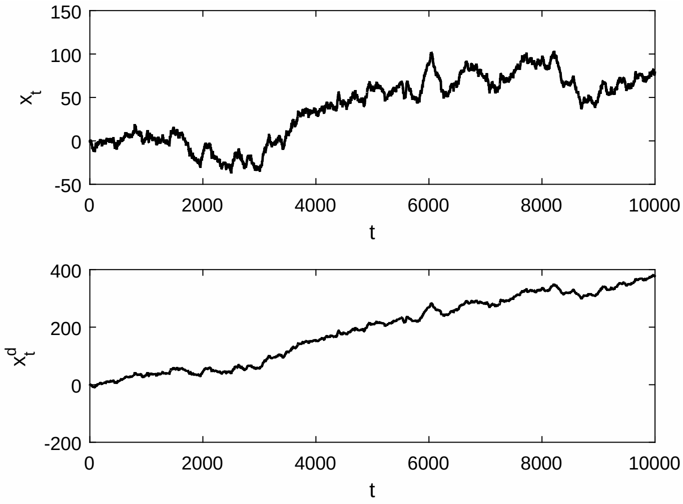

Figure 4.6 Simulation of random walk (top panel). The bottom panel shows the same time series added with a small drift in the upward direction.

Also, the autocovariance is given by (assuming $t > s$ and x_0 is known)

$$E((x_t - x_0)(x_s - x_0)|x_0) = (t-s)\sigma_\varepsilon^2. \tag{4.92}$$

Finally, autocorrelation is given by

$$\begin{aligned}\rho(x_t, x_s|x_0) &= \frac{Cov(x_t, x_s|x_0)}{\sqrt{V(x_t|x_0)V(x_s|x_0)}}\\ &= \sqrt{\frac{t-s}{t}}\end{aligned} \tag{4.93}$$

These moments are all time dependent making the process non-stationary. In particular, variance increases linearly with time. Often the random walk process is augmented with a drift term δ:

$$x_t = \delta + x_{t-1} + \varepsilon_t. \tag{4.94}$$

By backward substitution, we can write the expanded process as

$$x_t = x_0 + \delta t + \sum_{j=1}^{t} \varepsilon_j, \tag{4.95}$$

which is clearly a generalization of equation 4.89.

We show an example of random walk with Gaussian noise in figure 4.6. The divergence of the process is clearly visible. The bottom panel exhibits the same series with a small upward drift. This kind of processes visually mimic stock price data very well, although they do not reproduce a number of specific regularities found in the stock markets. We will discuss some of them below.

4.3.2 Unit root testing

Here we describe a few statistical tests that allow us to check for stationarity of a process (or the absence of non-stationarity). Fundamentally, the objective is to estimate an autoregressive model on a given set of data and test whether there is an unit root or not. While this task is seemingly trivial, non-triviality arises from the fact that the asymptotic distribution of the estimate for the unit root does not converge to normal distribution and therefore we cannot perform the inference exercise with standard t-tests (as in chapter 3)). This problem was recognized long time back and currently there is a wide array of tests. We will not get into a complete enumeration of all such tests. Instead we will describe the family of Dickey-Fuller tests that is probably the most standard solution to this problem. For more elaborate discussions and relevant econometric background, readers may consult Hamilton (1994), Brockwell et al. (1991), Neusser (2016) and Tsay (2010).

Dickey-Fuller test: Consider an AR(1) model (Dickey and Fuller (1979)) –

$$x_t = \alpha x_{t-1} + \varepsilon_t. \tag{4.96}$$

A straightforward hypothesis test can be conducted on

$$H_0 : \quad \alpha = 1 \quad \text{against} \quad H_1 : \quad \alpha \in (-1, 1). \tag{4.97}$$

This is a one-sided test. The DF test statistic is simply the $t-ratio$ of the estimated α, viz.

$$\hat{t} = \frac{\hat{\alpha} - 1}{se(\hat{\alpha})} \tag{4.98}$$

where the denominator is the standard error of the estimated coefficient. This test differs from the usual hypothesis testing on parameter estimation in that the Dicky-Fuller distribution does not converge to a normal distribution and needs to be simulated. Standard software packages simulate the Dicky-Fuller distribution automatically and provide the resulting significance.

Note that the process given by equation 4.96 can also be written as

$$\Delta x_t = \xi x_{t-1} + \varepsilon_t. \tag{4.99}$$

where $\xi = \alpha - 1$. Therefore, the hypothesis test can also be conducted on

$$H_0 : \quad \xi = 0 \quad \text{against} \quad H_1 : \quad \xi \in (-2, 0). \tag{4.100}$$

The DF test statistic in this case can be simplified to the following expression:

$$\hat{t} = \frac{\hat{\xi}}{se(\hat{\xi})}. \tag{4.101}$$

Augmented Dickey-Fuller test:

Suppose we have an AR(p) process:

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \ldots + \alpha_p x_{t-p} + \varepsilon_t. \tag{4.102}$$

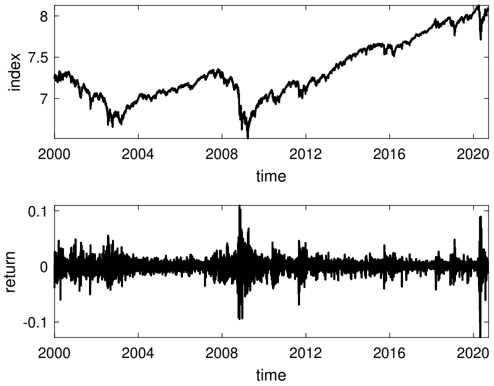

Figure 4.7 Stock index fluctuations. Upper panel: Evolution of log of S&P 500 index series from 3rd January, 2000 to 28th July, 2020. Lower panel: The corresponding return series in the form of first difference.

This can be rewritten in terms of first differences as

$$\Delta x_t = \xi x_{t-1} + \theta_1 \Delta x_{t-1} + \ldots + \theta_{p-1} \Delta x_{t-p+1} + \varepsilon_t \tag{4.103}$$

where $\xi,\ \theta_1,\ \theta_2,\ \ldots \theta_{p-1}$ are defined suitably. The hypothesis testing can be conducted on

$$H_0 : \quad \xi = 0 \quad \text{against} \quad H_1 : \quad \xi \in (-2, 0). \tag{4.104}$$

The ADF test statistic is

$$\hat{t} = \frac{\hat{\xi}}{se(\hat{\xi})}. \tag{4.105}$$

We can easily augment the process with a constant term c, viz.

$$\Delta x_t = c + \xi x_{t-1} + \theta_1 \Delta x_{t-1} + \ldots + \theta_{p-1} \Delta x_{t-p+1} + \varepsilon_t \tag{4.106}$$

or a trend term:

$$\Delta x_t = c + \kappa t + \xi x_{t-1} + \theta_1 \Delta x_{t-1} + \ldots + \theta_{p-1} \Delta x_{t-p+1} + \varepsilon_t \tag{4.107}$$

Standard time series softwares/packages allow this kind of hypothesis testing quite easily. As opposed to the Dickey-Fuller (also Phillips-Perron test – see Phillips and Perron (1988)) test which takes presence of unit roots to be the null hypothesis, one can also test for non-stationarity in the data by KPSS test (short form of Kwiatkowski-Phillips-Schmidt-Shin; see Kwiatkowski et al. (1992)) which consider presence of an unit root as the alternative hypothesis. Elliott et al. (1996) developed ADF-GLS test which is an improvement on the ADF test. Ng and Perron (2001) developed a *modified* Phillips-Perron test, which also improves upon ADF-GLS test.

4.4 Modeling fluctuations

Financial asset return data exhibit a number of interesting patterns. Figure 4.7 shows the evolution of the log of S&P 500 index daily data from 2000 to 2020 (data is obtained from Bloomberg). Looking at the data in the level (top panel) does not tell us much beyond existence of large fluctuations and an upward trend. In the bottom panel, we have drawn the first difference of the same data set. This new transformed time series represents the log return of S&P 500 index. As can be seen return data exhibits episodes of high volatility followed by period of low volatility. Two periods that stand apart from the rest are the period of 2008-09 (the time of the global financial crisis) and the period around March in 2020 (the beginning of the Coronavirus pandemic). Large fluctuations during these periods indicate high volatility in return in the financial markets.

Such features are not specific to only S&P 500 data, and in fact, can be found in stock market data from all over the world. In the literature, these are known as *stylized facts*. In a highly cited research article, Cont (2001) collected a wide range of such stylized facts in financial markets, ranging from zero autocorrelation in asset returns to properties of the second moment of asset returns. In this section, we will study two of those features, viz. *volatility clustering* and *slow decay of autocorrelation of modulus of return*. Volatility clustering refers to the pattern in the data that volatile periods tend to cluster in time. Put differently, a volatile event is more likely to be followed by another volatile event rather than a non-volatile event. Below, we will analyze such a scenario from a model point of view. In particular, we will introduce a very famous class of models called *GARCH* models that are able to replicate this feature in financial data (along with other features, like fat tailed return distribution). It is worth emphasizing here this model does not explain *why* there is volatility clustering, it merely allows us to *replicate* the pattern in the data, assuming that volatility tends to cluster. After setting up the GARCH model and describing how such models are estimated, we will discuss the phenomenon of long-range correlations and in particular, we will relate it to the observation of slow decay of autocorrelation of the magnitude of returns.

4.4.1 GARCH model: Stock returns and volatility clustering

The full form of GARCH model is *Generalized Autoregressive Conditional Heteroscedasticity* model. This model captures two features of financial data. One, asset return data often shows time-varying volatility. In fact, it is rare to find asset return data having a constant variance over time. Two, volatility itself seems to be quite persistent, implying that a volatile period is often followed by another volatile period. Notice that the ARMA model that we have developed in the earlier section, is not an useful tool in this kind of a scenario as the model does not allow for time varying conditional moments.

To visually examine the idea of volatility clustering, we present the autocorrela-

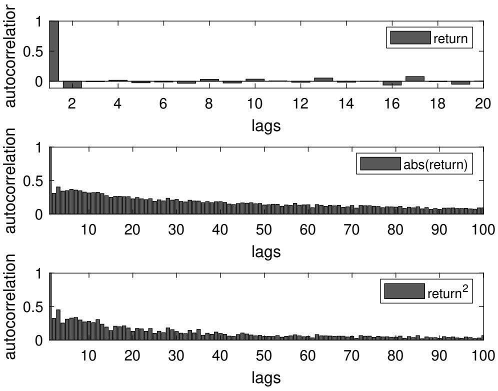

Figure 4.8

Volatility clustering in financial data. Upper panel: Sample autocorrelation function of the raw log return data from Figure 4.7. Middle and lower panels: Sample autocorrelation functions of the absolute returns and the squared returns respectively.

tion functions estimated from the daily closing price data in S&P 500 times series (same as in figure 4.7) in figure 4.8. The top panel shows the autocorrelation function of the log return, the middle panel shows the same for absolute values of the log return series and the bottom panel shows the same for squared log return series. As is evident from the figure, the autocorrelation function drops to zero almost immediately in case of log return; however, it tends to decay very slowly for the absolute values as well as squared values of the return series. This is an unique feature of financial data that indicates persistence of volatility.

The GARCH model, which is a generalization of the ARCH model (as the name suggests, the full form is *Autoregressive Conditional Heteroscedasticity* model) is a clever way to capture this statistical pattern in the data. Essentially it imposes an ARMA kind of a structure on the conditional volatility of the variable being modeled. Thus high persistence in the ARMA structure implies the volatility series itself would be persistent. We must be careful here to differentiate between conditional and unconditional volatility. As the name suggests, GARCH models the evolution of *conditional volatility* i.e., volatility conditional on past realization of the volatility and the observed variable (Andersen et al. (2014)).

Let us first see the structure of a GARCH model to see how it works. A simple example of GARCH(1,1) is as follows:

$$\begin{aligned} r_t &= \sigma_t \varepsilon_t \\ \sigma_t^2 &= \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2. \end{aligned} \tag{4.108}$$

where r_t is the asset return at time t (time is discrete here as was the case in the ARMA setup), ε_t is an independent standard normal random variable, σ_t^2 is the conditional variance that evolves as a function of squared past return r_{t-1}^2 and σ_{t-1}^2. The term ω is a positive constant.

The important point to note here is that if α and β are zeros, then we are back in the world of constant variance. In order to induce conditional *heteroscedasticity* (which means different degree of variability across observations), we need non-zero α and β. In particular, positive and high values of them would indicate high persistence in the conditional volatility.

Anther important feature of the model is the way it treats conditional volatility. Note that from the data we cannot directly observe volatility. For example, if we consider daily closing return data $\{r_t\}$, then for a given day t there is only one observation r_t. Therefore, obviously the corresponding variance of that single data point is zero. If we calculate day-wise in-sample volatility, then we will simply get a sequence of zeros, which clearly is not an useful way to think about how volatility evolves. A non-trivial way to use this approach would be to consider volatility of observations on a moving window, say across a fixed number of days. The problem with such an approach is that it is too ad-hoc with no clean way to select the window length, and also potentially the resulting volatility would be non-robust with respect to the window-length. The way this model considers volatility is to treat it as a *latent* variable which evolves over time, but is not directly observable. Hence, given a set of data, we have to estimate its values over time.

A GARCH process with generalized number of lags can be written as (Engle and Bollerslev (1986)):

$$\begin{aligned} r_t &= \sigma_t \varepsilon_t \\ \sigma_t^2 &= w + \sum_{i=1}^{p} \alpha_i r_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \end{aligned} \tag{4.109}$$

which can be recognized to be a straightforward generalization of equation 4.108 by allowing p and q lags on the past values. It is customary to assume that $\omega > 0$, $\alpha_i \geq 0$ and $\beta_j \geq 0$ for $i = 1, \ldots, p$ and $j = 1, \ldots, q$.

We need to impose some other restrictions in order to estimate the model. As usual, ε_t is a white noise i.e., $\{\varepsilon_i\} \sim \textit{i.i.d.}\ (0, 1)$. We also need $(\sum_i^p \alpha_i + \sum_j^q \beta_j) < 1$ for uniqueness and stationarity. Here we should clarify one more point: While GARCH models account for time-varying volatility, the resulting time-series are stationary. The time-varying feature of volatility is applied only to the *conditional* volatility. It can be shown that the unconditional variance of the GARCH(p, q) (equation 4.108) is

$$E(r_t^2) = \frac{\omega}{1 - \sum_i^p \alpha_i - \sum_j^q \beta_j}, \tag{4.110}$$

which would be a positive constant if the above assumptions hold. With zero mean and zero serial correlation of r_t (we will not prove these properties here; see for example Tsay (2010) for a more elaborate treatment), this indicates that the process would in fact be weakly stationary.

Estimation of GARCH models: Similar to the earlier class of models, GARCH models can also be estimated using maximum likelihood (Tsay (2010)). Most of the standard time series software/packages will contain routines to perform the estimation exercise.

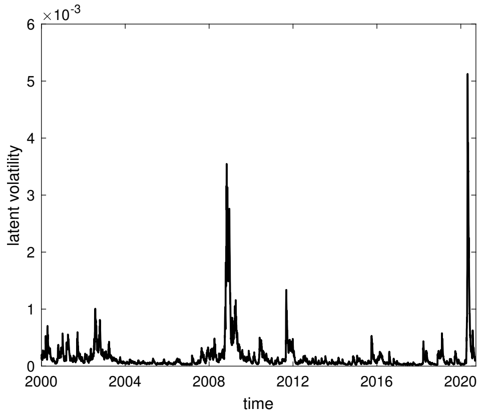

Figure 4.9

Estimated latent volatility from the log return series shown in Figure 4.7. Volatility spikes are obvious during the times of the financial crisis and the Covid-19 pandemic.

In Figure 4.9 we show the estimated latent volatility from the same log return dataset as depicted in figure 4.7. The estimated latent volatility indeed captures the periods of the 2008-09 crisis in the financial markets and the Covid-19 pandemic well.

Latent volatility is known to influence pairwise correlations in the stock market. In figure 4.10 we show the same network as in figure 4.4. The only difference is that in this case, we have normalized each return series by their corresponding volatility series estimated with a GARCH(1,1) model. The network retains its general properties even after the adjustment.

4.5 Scaling and long memory

The idea of scaling has been explored in great details in the physics literature. Over time, it found its applications in widely different fields ranging from geophysics to finance. Benoit Mandelbrot was the first scientist who brought the concept of scaling in the financial domain, in his 1963 article titled *The variation of certain speculative prices* (Mandelbrot (1997)). Interested readers may consult, for example, Mandelbrot (2013) and Mandelbrot and Hudson (2007), for discussions on the usefulness of fractals, self-similarity and scaling in financial time series data (see LeBaron (2016) for an analytical discussion on Mandelbrot's legacy).

At this point it is instructive to note that the concept of scaling has been used in more than one contexts in financial data to refer to different phenomena. The following classification is motivated by the discussion in Di Matteo et al. (2005). There are two types of scaling behaviors that scientists have studied in financial data (and also in other systems). The first-type is concerned with the time series

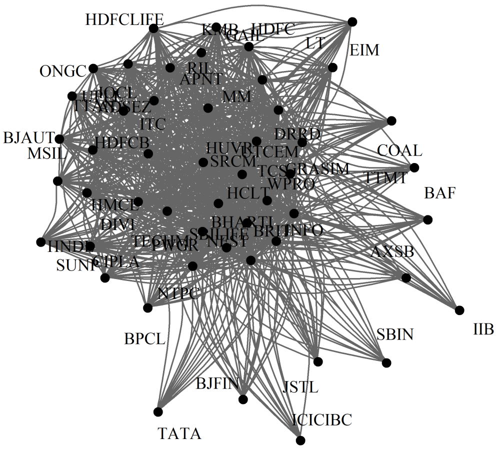

Figure 4.10 Return comovement network shown in Figure 4.4 after normalizing each return series by their corresponding estimated latent volatility series.

properties, where long range correlation exists that decay very slowly with lags. The way such scaling behavior manifests is through empirically documented slow decay in the autocorrelation function of absolute returns. The second type is concerned with fat tailed distribution of asset returns. As far as estimation is concerned, the first type of scaling behavior leads to *scaling exponents* for decay whereas the second type leads to power law for the tails of the distribution.

We can view scaling in the second sense as dilation. For a function $f(t)$ and an arbitrary constant β, scaling property implies

$$f(\beta t) = \gamma(\beta) f(t) \tag{4.111}$$

where $\gamma(.)$ is a function that determines the *degree* of stretching or dilation. These kind of functions are called scale invariant or self-similar. We can guess a solution:

$$f(t) = \theta t^{\phi}. \tag{4.112}$$

With such an assumption, we get

$$f(\beta t) = \theta \beta^{\phi} t^{\phi}, \tag{4.113}$$

implying $\gamma(\beta) = \beta^{\phi}$. In the remainder of the chapter we will not develop the idea of fractality in time-series and we will also not discuss power law distributions in returns (which itself is considered to be a *stylized fact* as listed in Cont (2001)). Instead, we will focus exclusively on long range correlations by building the earlier ideas about large and significant autocorrelations of absolute returns even at large lags. Weron (2002) present a systematic treatment of the statistical features of long-range dependence in financial data. In particular, three methods can be pursued here – R/S analysis, detrended fluctuation analysis and periodogram regression. The third technique has had lesser success in the context of long-range dependence. Below, we explore the first two methods as they have become relatively more common toolkits in such analysis. Finally, we will review ARFIMA and FIGARCH class of models.

4.5.1 R/S analysis and Hurst exponent

Mandelbrot and Wallis (1969) came up with the technique of R/S analysis around half a century back in the context of long range statistical dependence (they called it 'long run' instead of 'long range' which is a more commonly used term now). The eponymous Hurst exponent is the characteristic value of rescaled range analysis, which is named after Harold Hurst. Theoretically a time series with independent draws would have an exponent H of 1/2. A positively autocorrelated series would have $H > 1/2$, a negatively autocorrelated series would have $H < 1/2$. $H > 1$ reflects non-stationary series and in particular, $H = 3/2$ indicates a random walk. While this concept is intimately related and in fact, has had very successful applications in fractal geometry (see for example Mandelbrot (2013)), we will present it in an elementary fashion purely from the point of view of statistical estimation.

Let's imagine that we have a time series vector $X_T = \{x_1,\ x_2,\ ...,\ x_T\}$. First, we construct n non-overlapping subsets of equal length t_n such that $n \times t_n = T$. Let us denote the $j-$th subset by S_j where $1 \leq j \leq n$. For the ease of description, let us denote the i-th term in the original time series vector X_T corresponding to the j-th window as $\tilde{x}_{i,j}$ where $1 \leq i \leq t_n$. For the j-th window, we calculate the first two moments:

$$\begin{aligned} \mu_j &= \frac{\sum_{i=1}^{t_n} \tilde{x}_{i,j}}{t_n}, \\ \sigma_j &= \sqrt{\frac{\sum_{i=1}^{t_n} (\tilde{x}_{i,j} - \mu_j)}{t_n}} \qquad \text{for } 1 \leq j \leq n. \end{aligned} \tag{4.114}$$

Then we find the cumulative sum of the deviations of the terms from the sample mean:

$$y_{i,j} = \sum_{i=1}^{t_n} (\tilde{x}_{i,j} - \mu_j). \tag{4.115}$$

Then we define a *range* term capturing the spread between the minimum and the maximum of all such deviations within the j-th window:

$$R_j = \max_i \{y_{i,j}\} - \min_i \{y_{i,j}\}. \tag{4.116}$$

We find the *rescaled range* (denoted by R/S_n) of the n windows by calculating the average of the ranges of all windows scaled by the corresponding standard deviations:

$$R/S_n = \frac{1}{n}\sum_{j=1}^{n}\frac{R_j}{\sigma_j}. \tag{4.117}$$

Next, we vary the values of n, i.e. the number of divisions we created in the time series. A good candidate would be to consider the powers of two, i.e. $n = 1,\ 2,\ 4,\ 8$ etc. Obviously, after a while, the number of data-points in each window (t_n) would be so small that the above calculation would not be credible. Typically, t_n is taken to be at least 8. Then we utilize the original expression proposed by Hurst to estimate the corresponding exponent:

$$R/S_n = c.t_n^H. \tag{4.118}$$

A straight forward approach would be to take log on both sides and find the Hurst exponent H by linear regression with the sample analog of the left hand side:

$$\log\ (R/S_n) = C + \beta_H \log\ t_n. \tag{4.119}$$

A simple ordinary least square estimation will give the value of β_H which is the estimated value of the Hurst exponent H.

4.5.2 Detrended fluctuation analysis

While this technique has its origin in studying long-range correlations in DNA molecules (Peng et al., 1994), this technique is much more widely applicable and in fact, over the course of the previous two decades it has become a staple technique even in the context of financial data. Consider a time series vector $Y_T = \{y_1,\ y_2,\ ...,\ y_T\}$. First we find the sample mean

$$\mu = \frac{\sum_{i=1}^{T} y_i}{T}, \tag{4.120}$$

and we create a new series based on the cumulative sum of the deviations from the mean:

$$x_t = \sum_{i=1}^{t}(y_i - \mu). \tag{4.121}$$

Then just like the rescaled range analysis, we construct n non-overlapping subsets of equal length t_n such that $n \times t_n = T$. As we have done above, let us denote the $j-$th subset by S_j where $1 \le j \le n$ and let us denote the i-th term in the original time series vector X_T corresponding to the j-th window as $\tilde{x}_{i,j}$ where $1 \le i \le t_n$.

For the j-th window, we fit a straight line with the following model:

$$\tilde{x}_{i,j} = \alpha_j + \beta_j t + \varepsilon_t \tag{4.122}$$

where $t = 1,\ ...,\ t_n$. Let us denote the best fit line by z so that

$$z_{i,j} = \hat{\alpha}_j + \hat{\beta}_j t, \tag{4.123}$$

where $\hat{\alpha}$ and $\hat{\beta}$ denotes the estimated α and β. Now, we find the *fluctuations* of the $\tilde{x}$ series around this line of best fit z:

$$F_n = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(x_t - z_t)^2}. \tag{4.124}$$

We compute the F_n for many values of n and then a linear regression of the form

$$\log\ (F_n) = C + \beta_{dfa} \log\ n + \varepsilon_{dfa,n} \tag{4.125}$$

will give us the estimated value of β_{dfa} which represents the Hurst exponent found by analyzing detrended fluctuations.

4.5.3 Fractional integration and long memory: ARFIMA and FIGARCH

Finally, we discuss a generalization of ARMA(p, q) model which can be represented as ARIMA(p, d, q) where d is the number of differences required to make the series stationary. Using notations similar to equation 4.30, we can write

$$\alpha(\mathcal{B})(1 - \mathcal{B})^d x_t = \beta(\mathcal{B})\varepsilon_t. \tag{4.126}$$

In case of ARIMA(p, d, q) model, the order d is taken to be an integer.

Sowell (1992) considered a generalization by taking the parameter d as a fraction (see Granger and Joyeux (1980) and Hosking (1984) for prior work). The resulting process of called *autoregressive fractionally integrated moving average* process. Conceptually it lies between an $I(0)$ and $I(1)$ process, i.e. between a stationary and an unit root process. The main empirical motivation arises from the fact that often financial time series exhibit long memory in mean or in variance. Fractionally integrated processes can nicely replicate those features.

First, we need a way to operationalize fractional d. Consider a standard binomial expansion:

$$(1 - \mathcal{B})^d = \sum_{j=0}^{\infty} \binom{d}{j} (-\mathcal{B})^j. \tag{4.127}$$

For fractional d, the infinite series serves the purpose of the lag polynomial. However, to ensure weak stationarity, we need $-0.5 < d < 0.5$. In order to see how $\binom{d}{j}$ can be computed for such values of d, we can write the above equation in the form of Gamma function which generalizes the idea of factorial to fractional numbers:

$$(1 - \mathcal{B})^d = \sum_{j=0}^{\infty} \frac{\Gamma(j - d)\mathcal{B}^j}{\Gamma(-d)\Gamma(j + 1)}, \tag{4.128}$$

where the Γ function is defined as

$$\Gamma(d) = \int_0^{\infty} x^{d-1} exp(-x)dx. \tag{4.129}$$

Similar to equation 4.126 above, we can use the same notation to write an ARFIMA model as

$$\alpha(\mathcal{B})(1-\mathcal{B})^d x_t = \beta(\mathcal{B})\varepsilon_t, \tag{4.130}$$

except that here d is a fraction. The autocorrelation function exhibits very slow decay and for large lags, it can be shown that the autocorrelation decays in a power law:

$$\rho_j \approx c \times j^{2d-1}. \tag{4.131}$$

This expression gives a crude way to estimate d by noting that a log transform of the above equation (with absolute value of ρ_j to avoid taking log of negative numbers) will produce

$$log\ |\rho_j| \approx log\ c + (2d-1)log\ j. \tag{4.132}$$

The initial attempts in this literature focused on linear regression to find $\hat{d}$, i.e. the estimate of d. However, this estimation is not accurate and Sowell (1992) showed that the maximum likelihood approach works better.

Next, we consider long memory in variance rather than in level. This model was introduced by Baillie et al. (1996). Consider the GARCH(1,1) model described in equation 4.108. The same model can be written as an ARMA process in x_t^2 as follows:

$$x_t^2 = \omega + (\alpha+\beta)x_{t-1}^2 + \xi_t - \beta\xi_{t-1} \tag{4.133}$$

where $\xi_t = x_t^2 - \sigma_t^2$. Similar to the ARFIMA model, we can now generalize this process by introducing fractional differencing as follows:

$$(1-\mathcal{B})^d x_t^2 = \omega + (\alpha+\beta)(1-\mathcal{B})^d x_{t-1}^2 + \xi_t - \beta\xi_{t-1}. \tag{4.134}$$

To ensure positivity of conditional variance, we require (1) $\omega > 0$, (2) $(\alpha + d) \geq 0$ and (3) $0 \leq d \leq 1 - 2(\alpha+\beta)$.

The literature on long memory has seen an amalgamation of ideas from very different origins. Graves et al. (2017) provides a concise description of the evolution of the modeling in the context of long memory starting from the work by Hurst and Mandelbrot and eventually how it led to the econometrics literature via fractional integration models. While our motivating examples came mostly from economic and financial contexts where long-range correlation in volatility is observable, there is a rich literature around applications of such models to biological and physical systems where similar long range correlations are observable (see Peng et al. (1992), Peng et al. (1995) and Stanley et al. (1993)). Gao et al. (2006) provides some useful tips on detecting long range correlations from time series data from a general systemic perspective.

4.6 Taking stock and further readings

In this chapter, we have discussed some concepts in time series analysis, mostly from an econometric point of view. We have introduced *white noise* as the fundamental building block of a time series model. We have shown that a linear combination of white noise, in the form of autoregressive moving average model, can capture a wide range of behavior. In particular, in order to model stationary time series, the ARMA class of models suffices thanks to the Wold decomposition theorem. Then we have discussed non-stationary time series and described statistical tests to detect presence of non-stationarity in the data. Finally, we have extended the discussion of non-stationarity into fractional integration, which allows us to model long range correlations. The discussion on modeling the first moment is supplemented by a discussion on GARCH-type models that allows us to model time-varying conditional volatility.

This coverage is introductory in nature. There are many interesting and useful topics in time series analysis that we have not covered here. For example, with the toolkit we have described, one can get into a discussion on forecasting which is very useful from an econometric point of view (Elliott and Timmermann (2016)). Here we have deliberately skipped this topic as it does not directly link to the models of socio-economic complex systems although recent advances in the dynamical systems theory provides a link in a more general context (Wang et al. (2016)). There are some excellent textbooks on time-series econometrics that interested readers can consult, viz. Hamilton (1994), Brockwell and Davis (2016), Neusser (2016), Enders (2008) and Tsay (2010). Some earlier attempts to discuss a few of the time series topics in the context of complex systems can be found in Mantegna and Stanley (2007) and Sinha et al. (2010).

There is a number of important directions that have been pursued in the recent (and not so-recent anymore) literature. Here we briefly describe a few of them along with some references. Autoregressive processes with distributed lags (ARDL) has a well developed toolkit to estimate dynamic relationship between multiple variables (Cho et al., 2021). Advances in modeling sampling with mixed-frequencies (MIDAS) have made it possible to incorporate variety of data with different time series frequencies in the same model for the purpose of forecasting (Ghysels et al., 2007). High frequency financial data analysis has become a very important topic in itself (see e.g. Aït-Sahalia and Jacod (2014)). The idea of scaling that Mandelbrot introduced (Mandelbrot (1997)) has now been extended to multi-scaling e.g. in Di Matteo (2007). Finally, time series analysis has also led to a much deeper understanding of financial networks and their fragility-robustness with respect to endogenous and exogenous shocks. Diebold and Yılmaz (2015) has developed a network theoretic way to model shock propagation across financial and economic entities, where they infer the connectivity across entities by applying a vector autoregression model. In complementary developments, use of state space models like

Kalman filters and particle filters (and more sophisticated variants thereof) have become a very useful way for estimation of agent-based models of financial dynamics that sheds light on scaling and other critical behaviors (e.g. Lux and Marchesi (1999) and Lux (2018)).