MODELING AND ANALYSIS FOR COMPLEX SYSTEMS

REPORT FOR THE FINAL EXAM

Bruno Casella
Department of Economics
University of Catania
casella0798@gmail.com

June 5, 2021

ABSTRACT

This is a report made for the final exam of the course *Modeling and Analysis for Complex Systems*, taught by Prof. G. Nunnari, at University of Catania, about the analysis of Time Series and Stochastic processes. The purpose of this paper is to collect the knowledge learned during the course and show its usefulness by applying it to practical cases. In particular, the case study considered in this work is related to a molecular communication system, provided by the *ACM Nanocom 2021 Data Competition*.

Introduction

The aim of this paper is to illustrate the principal theory concepts learned during the Academic Course in *Modeling and Analysis for Complex Systems*, taught by Prof. Giuseppe Nunnari, at University of Catania, and to show how to apply these concepts to practical cases. The case study of this report is a time series taken by a set of datasets provided for the *ACM Nanocom 2021 Data Competition*, related to experimental molecular communication systems. The report is organized as follows. In Chapter 1 the main concepts about stochastic processes will be summarized. In Chapter 2, the main time series concepts will be summarized. In Chapter 3, the main features of the models learned in the Course will be summarized (MA, AR, ARMA, ARIMA, SARIMA, ARMAX, NARMAX). In Chapter 4 The model validation problem will be addressed. In Chapter 5, The available software tools will be described (Ident, Econometric...), nntool, etc). In Chapter 5, each model will be applied to the proposed case study. In Chapter 7, conclusions will be drawn. Finally, in appendix, the implemented software code will be reported.

1 Stochastic Processes

The data on which are applied the inferential techniques, especially for econometrics, are *time series*, that are series of data points indexed in time order. We need an extension of probability concepts in order to understand and apply those techniques. The reason is that time has a direction and an history. In time series, indeed, the natural tendency of lot of phenomenons, to evolve in some regular ways, leads us to think that data at instant t is more similar to data at instant $t-1$ rather than to data in farther epochs. This is because, in some sense, time series has a "memory of itself". This characteristic is generally indicated with the name **persistency** (or, sometimes, hysteresis). So, persistency refers to the fundamental importance of order of data. The tool used for combining the probability point of view with time series is the **stochastic process**. An informal definition of stochastic process is: a infinitely long sequence of random variables. So, a samples of T consecutive observations is not thought as a realization of T distinct random variables, but as a unique realization of a stochastic process, in which the memory is given by the connection between random variables that constitute it.

1.1 *Properties of a stochastic process*

A stochastic process is a sequence of random variables ordered with an index t, the time. We consider t as a discrete index $t = 0, 1, 2, ...$ The random variable associated with time t is denoted by $v(t)$. Remember that a random variable

is not a number,, but a real function of the outcome of a random experiment s, in other words, $v(t) = v(t, s)$. Thus, a stochastic process is an infinite sequence of real variables, each of which depends upon two variables, time t, and outcome s. Often the dependence upon s is omitted for simplicity. The simplest way to describe a stochastic process is to specify its **mean** $m(t) = E[v(t)]$ and its **variance** $Var(t) = E[(v(t) - m(t))^2]$ There are some properties that stochastic processes can have as well as not have: **stationarity** and **ergodicity**.

1.1.1 *Stationarity*

Stationary stochastic processes can be of two types: *strong* or *weak*. For defining strong stationarity, we take any subset of random variables of the stochastic process. These random variables can not be consecutive, but for simplicity we will consider them consecutive. So, let's consider a window of amplitude k of the process, $W_t^k = (x_t, ..., x_{t+k-1})$. This is a random variable of dimension k with its density function, that can depends on t. It it not depends on t, then the distribution of W_t^k is the same of W_{t+1}^k W_{t+2}^k and so on. We have strong stationarity if this invariance happens for every k. In other words, a process has strong stationarity when its distribution characteristics remain constant over time.

Weak stationarity instead, is only about windows with $k = 2$: a stochastic process has weak stationarity if all the double random variables $W_t^2 = (x_t), x_{t+1}$ have moments of first and second order constant over time.

Strong stationarity does not imply weak stationarity, and viceversa. A process can be strong stationary but it could not have moments; viceversa, constance over time of moments does not imply constance of the distribution characteristics. However, there is a case in which both definitions of stationarity coincide, and which is particularly useful for practical applications: the case in which the process is **gaussian**, that is when the joint distribution of any subset of elements of the process, is a multivariate normal distirbution. If a process is gaussian, saying that it is weak stationary is the same of saying that it is strong stationary. Due to the pervasiveness of gaussian processes in real data, generally it is adopted the definition of weak stationarity.

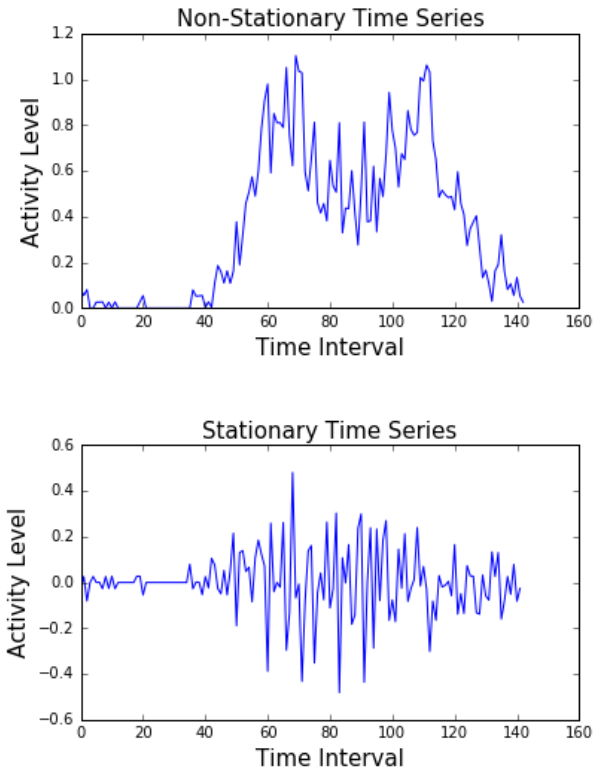


Figure 1: Stationarity and non stationarity

1.1.2 *Ergodicity*

Ergodicity is a condition that limits the memory of the process: a not ergodic process has a strong persistency such that a sequence of the process, however long, is not sufficient to say anything about the characteristics of the distribution.

Instead, in an ergodic process the memory is weak on long time horizons, and if we increase the length of the considered sequence, also the information about the process increase. In an informal way, a process is ergodic if very distant events over time can be considered independent. From a practical point of view, this can be summarized with the following formula:

$$\lim_{n\to\infty} \sum_{k=1}^{n} \mathrm{Cov}(x_t, x_{t-k}) = 0. \tag{1}$$

So, if a process is ergodic, it is possible to use information contained in its execution over time to infer its characteristics. The *Ergodic Theorem* says that if a process is ergodic, the observation of its realization enough long, is equivalent to the observation of a lot of realizations. For example, if the ergodic process x_t has an expectation μ, then its arithmetic average over time is a good estimator of μ (so, μ can be estimated like if we have a lot of realizations of the process rather than only one).

In general, we can say that inference is possible only if the stochastic process is stationary and ergodic. There are methods to understand if the process is stationary; instead the ergodicity is not testable if we have only one realization of the process, also if it is of an infinite window.

1.1.3 *Moments*

For stationary stochastic processes, we have that each element of the process x_t will have a finite expectation μ and a finite and constant variance σ^2. Moreover, we can define the covariance between different elements of the process $\gamma_k = E[(x_t - \mu)(x_{t-k} - \mu)]$ that are known as **autocovariances**. Remember that stationarity guarantees that these quantities do not depend on t, they are functions of k. Obviously the autocovariance of order 0 is the variance

Now, we can define the **autocorrelation** as $\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{\gamma_k}{\sigma^2}$. Just as correlation measures the extent of a linear relationship between two variables, autocorrelation measures the linear relationship between lagged values of a time series. Obviously, $\rho_0 = 1$. These quantities, if different from 0, describe the memory of the process, and they are the element that make stochastic processes the perfect element for representing time series characterized by consistency. Let's see an example of the autocorrelation function, or ACF. This plot is also called correlogram.

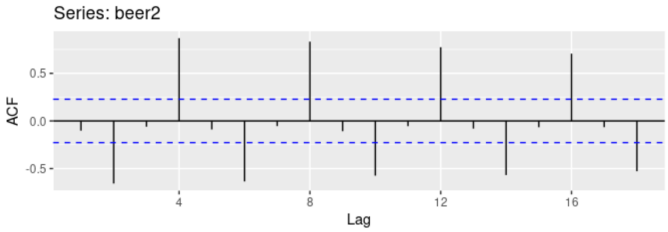


Figure 2: Autocorrelation function of quarterly beer production.

In this graph:

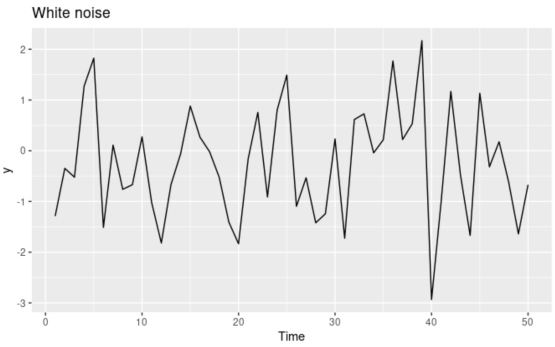
- ρ_4 is higher than for the other lags. This is probably due to the seasonal pattern in the data: the peaks tend to be four quarters apart and the troughs tend to be four quarters apart.
- ρ_2 is more negative than for the other lags probably because troughs tend to be two quarters behind peaks.
- The dashed blue lines indicate whether the correlations are significantly different from zero.

1.1.4 *White Noise*

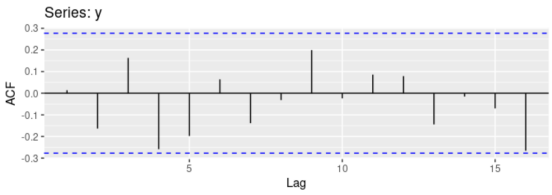
Time series that show no autocorrelation are called **white noise**. White noise is the simplest stochastic process we can imagine: it is a process with moments (at least) until the second order; these moments are constant over time (so the process is stationary), but they do not give any memory of itself to the process. In a more formal way, a white noise process, whose $t - th$ element is indicated by ϵ_t has these characteristics:

- $E(\epsilon_t) = 0$
- $E(\epsilon_t^2) = V(\epsilon_t) = \sigma^2$
- $\gamma_k = 0$ for $|k| > 0$

So, basically a white noise is a process made by an infinite number of random variables, with zero mean and constant variance. These random variables are all uncorrelated between each other; this does not mean that they are independent. A white noise process, so, is a stochastic process that does not exhibit persistence. We will see later what are the benefits given by the white noise, when we apply a polynomial in the *delay* operator to a white noise.



(a) White noise



(b) Autocorrelation of white noise

Figure 3: A figure showing white noise and its autocorrelation

1.1.5 *Spectral Analysis*

Many time series show periodic behavior. This periodic behavior can be very complex. **Spectral analysis** is a technique that allows us to discover underlying periodicities. To perform spectral analysis, we first must transform data from time domain to frequency domain. There are various way of defining the spectrum. A very simple one is to define it as the Fourier transform of the covariance function:

$$\Gamma(\omega) = \sum_{\tau=-\infty}^{\infty} \gamma(\tau) e^{-j\omega\tau} \tag{2}$$

with ω that is a real variable. Given the formula above, we can recover the covariance function $\gamma(\tau$ with the anti-transform formula:

$$\gamma(\tau) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \Gamma(\omega) e^{-j\omega\tau} dw \tag{3}$$

The main properties of $\Gamma(\omega)$ are:

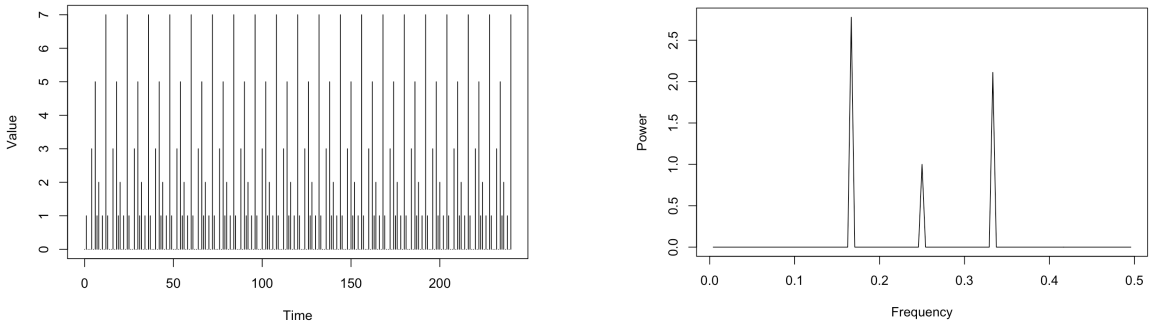
1. $\Gamma(\omega)$ is real
2. $\Gamma(\omega)$ is periodic of period 2π
3. $\Gamma(-\omega) = \Gamma(\omega)$ for every w
4. $\Gamma(\omega) \geq 0$
5. The are under the curve $\Gamma(\omega)$ between $-\pi$ and π is, up to $\frac{1}{2\pi}$, the process variance

$\Gamma(\omega)$ being periodic of period 2π, the spectrum can be represented in a graphical way for an angular frequency w ranging from $\omega = -\pi$ to $\omega = +\pi$. Being the spectrum even, the diagram can be limited between $w = 0$ and $w = +\pi$. Alternatively, in place of the angular frequency, one can make reference to the frequency $f = \frac{\omega}{2\pi}$ and draw the diagram from $f = -0.5$ and $f = +0.5$

As seen above, in discrete time, frequencies may range up to a maximum of $f = 0.5$. This can be easily interpreted. indeed, the most rapidly varying signal in discrete time is a signal changing its sign passing from one instant to the subsequent one, e.g. signal $v(t) = 1$ for t even and $v(t) = -1$ for t odd. Such function is periodic of period $T = 2$; hence, its frequency is $f0.5$ and its pulsation $w = 2\pi f = \pi$.

Let's see a simple example: small signal every three years; larger signal every four years; large signal every six years. Now calculate the frequency spectrum "by hand" using a Fast Fourier Transform. We can see very clearly that we can recover the periodic signals that we built into our toy time series. Indeed, for the first signal $T = 3$, so $f = \frac{1}{T} = \frac{1}{3} \approx 0.33$ and we can see a peak in the spectrum at frequency 0.33. For the second signal $T = 4$, so

$f = \frac{1}{T} = \frac{1}{4} \approx 0.25$ and we can see a peak in the spectrum at frequency 0.25. Finally, for the third signal $T = 6$, so $f = \frac{1}{T} = \frac{1}{6} \approx 0.167$ and we can see a peak in the spectrum at frequency 0.167.



(a) 3 periodic signal

(b) Spectrum

Figure 4: A figure showing three periodic signals and their spectrum

The spectrum of a white noise is easily computed. Indeed, being $\gamma(\tau) = 0, \forall \tau \neq 0$ all terms in the definition of the spectrum are null, with the only exception of the term associated with $\tau = 0$. Hence,

$$\Gamma(\omega) = \sum_{\tau=-\infty}^{\infty} \gamma(\tau) e^{-j\omega\tau} = \gamma(0) \tag{4}$$

Thus, the spectrum is a constant: all frequencies are equally contributing to form the signal.

Figure 5: White noise spectrum.

2 Time Series Analysis

A time series is a series of data points indexed in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus, it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.

2.1 *Structure of Time Series*

As said before, a time series is a sequence of numerical data points in successive order. It can be any data recorded over time in sequential order. We can think of stock prices, but also videos, languages, songs and MRI Scans can be thought as time series data as well. In Fig.1 there is an example image of Stock Prices, and we can observe that in the x-axis we have the time index, and in the y-axis we have the stock prices of different markets.

One obvious example use case of Time Series is predicting stock prices. (Well if this was so easy, a lot of Data Scientist would be rich.) In general we want to forecast / predict the next value when it comes to Time Series. Most time series data can be described by four components. And those are trend, seasonal, cyclical and irregular components.

A time series can be thought in two different ways: as additive model or as multiplicative model of the previous mentioned components. In the first case we have:

$$y_t = T_t + S_t + C_t + I_t \tag{5}$$

Whereas a multiplicative model would be:

$$y_t = T_t \times S_t \times C_t \times I_t \tag{6}$$

Now we are going to see what each of these components represent.

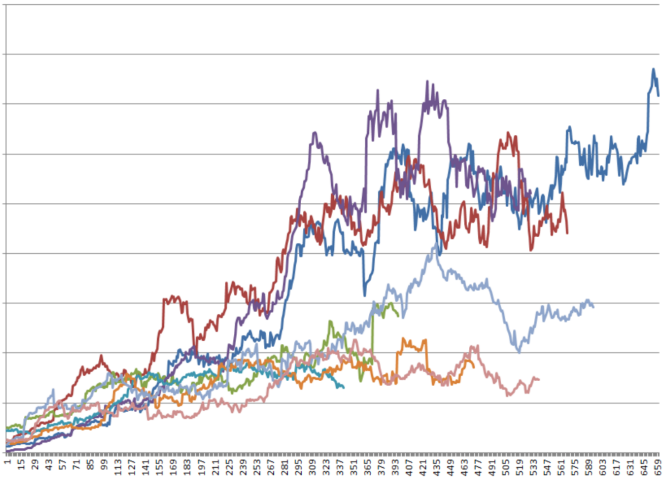


Figure 6: Example image of Time Series.

2.2 *Trends*

The trend component at time t, T_t, reflects the long-term progression of the series (secular variation). A trend exists when there is a persistent increasing or decreasing direction in the data. The trend component does not have to be linear. Sometimes we will refer to a trend as "changing direction," when it might go from an increasing trend to a decreasing trend.

2.3 *Seasonal Components*

The seasonal component at time t, S_t, reflects seasonality (seasonal variation). A seasonal pattern exists when a time series is influenced by seasonal factors. Seasonality occurs over a fixed and known period (e.g., the quarter of the year, the month, or day of the week). Seasonality may be caused by various factors, such as weather, vacation, and holidays and consists of periodic, repetitive, and generally regular and predictable patterns in the levels of a time series.

2.4 *Cyclical Component*

The cyclical component at time t, C_t, reflects repeated but non-periodic fluctuations. A cycle occurs when the data exhibit rises and falls that are not of a fixed frequency. The duration of these fluctuations depend on the nature of the time series. These fluctuations are usually due to economic conditions, and are often related to the "business cycle." The duration of these fluctuations is usually at least 2 years. Many people confuse cyclic behaviour with seasonal behaviour, but they are really quite different. If the fluctuations are not of a fixed frequency then they are cyclic; if the frequency is unchanging and associated with some aspect of the calendar, then the pattern is seasonal.

2.5 *Irregular Component*

The irregular component (or "noise") at time t, I_t, describes random, irregular influences. It represents the residuals or remainder of the time series after the other components have been removed.

Many time series include these mentioned patterns. When choosing a forecasting method, we will first need to identify the time series patterns in the data, and then choose a method that is able to capture the patterns properly. The examples in Fig. 2 show different combinations of the above components.

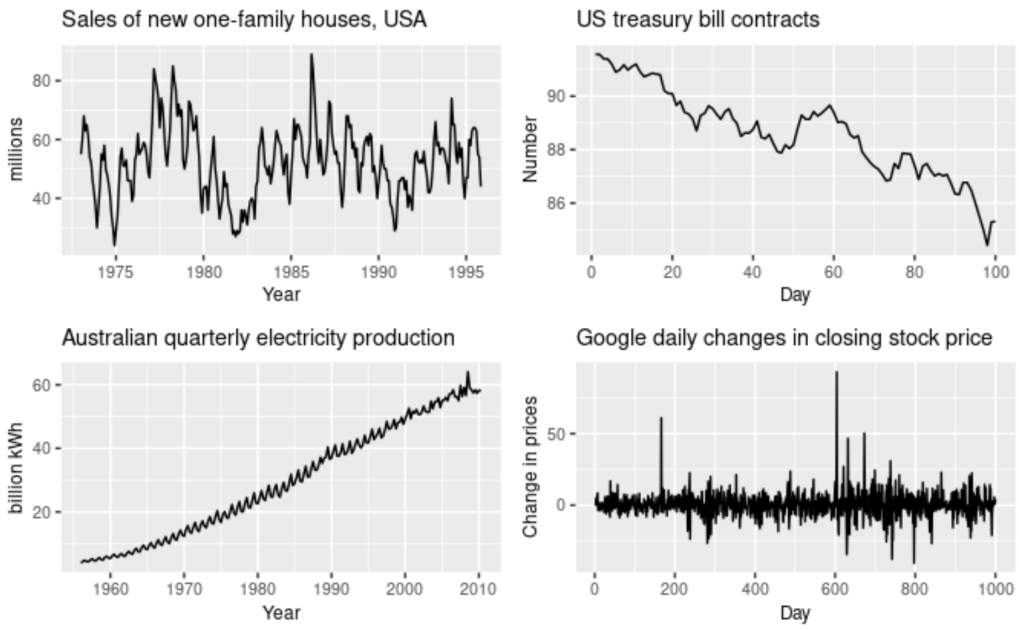


Figure 7: Example image of Time Series Components.

1. The monthly housing sales (top left) show strong seasonality within each year, as well as some strong cyclic behaviour with a period of about 6–10 years. There is no apparent trend in the data over this period.
2. The US treasury bill contracts (top right) show results from the Chicago market for 100 consecutive trading days in 1981. Here there is no seasonality, but an obvious downward trend. Possibly, if we had a much longer series, we would see that this downward trend is actually part of a long cycle, but when viewed over only 100 days it appears to be a trend.
3. The Australian quarterly electricity production (bottom left) shows a strong increasing trend, with strong seasonality. There is no evidence of any cyclic behaviour here.
4. The daily change in the Google closing stock price (bottom right) has no trend, seasonality or cyclic behaviour. There are random fluctuations which do not appear to be very predictable, and no strong patterns that would help with developing a forecasting model.

In Fig.3 we can easily see how each component is related to the others.

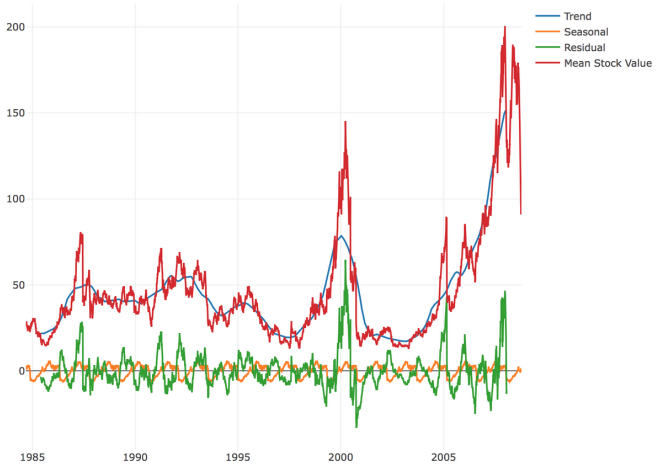


Figure 8: Example image of a Time Series decomposition.

- Red Line → Apple Stock Prices from 1985
- Blue Line → Trend for Apple Stock Price
- Green Line → Residual (Noise) for Apple Stock Price
- Orange Line → Seasonal (Yearly) trend for Apple Stock Price

3 Modeling

In this chapter we will describe the main properties and limitations of the models learned in the course (MA, AR, ARMA, ARIMA, SARIMA, ARMAX, NARMAX...).

3.1 *Delay operator*

Stochastic processes and time series are sequences of numbers. Frequently, we will work on these sequences using some operators. The **delay operator** is generally indicated with letter L in econometrics (B in statistics). It is an operator applied to a sequence of general objects, like sequences of random variables (so, stochastic processes), or sequences of numbers. This operator transforms a sequence x_t in another sequence with the same values of the original, but translated over time of a period. Obviously, if the operator is applied to a constant sequence, nothing happens.

$$Lx_t = x_{t-1} \tag{7}$$

If we repeat the L operation n times, we can write L_n, and we have $L_n x_t = x_{t-n}$. For convention, we say $L_0 = 1$. L is a linear operator, so if we have two constants a and b, then we will have $L(ax_t + b) = aLx_t + b = ax_{t-1} + b$

3.2 *MA processes*

A MA process, or a **moving average** process is a sequence of random variables that can be written in the following way:

$$y_t = C(L)\epsilon_t = c_0\epsilon(t) + c_1\epsilon(t-1) + ... + c_n\epsilon(t-n) \tag{8}$$

where $c_0, c_1, ..., c_n$ are real numbers, and $C(L)$ is a polynomial of order n of the delay operator and ϵ is a white noise. So, basically a MA process is a stochastic process generated as a linear combination of the current and past values of a white noise. Let's examine its moments, starting from the mean:

$$E(y_t) = E(C(L)\epsilon(t)) = E(c_0\epsilon(t)+c_1\epsilon(t-1)+...+c_n\epsilon(t-n)) = c_0E(\epsilon(t))+c_1E(\epsilon(t-1))+...+c_nE(\epsilon(t-n)) \tag{9}$$

Since $E(\epsilon_t) = 0$ it follows that $E(y_t) = 0$. So, a MA process has zero mean. We can think that this characteristic causes some limitations to real situations, because generally time series have not a zero mean. However, for each process with x_t with $E(x_t) = \mu_t$, we can always define a new process $y_t = x_t - \mu_t$ with zero mean.

Passing to the variance, we have

$$Var(y_t) = E(y_t)^2 = c_0^2E(\epsilon(t))^2+c_1^2E(\epsilon(t-1)^2+...+c_n^2E(\epsilon(t-n)^2+c_0c_1E(\epsilon(t)\epsilon(t-1))+c_0c_2E(\epsilon(t)\epsilon(t-2))+... \tag{10}$$

Being $\epsilon(\cdot)$ a white noise, all the mean values of the cross-products of the type $E(\epsilon(i)\epsilon(j))$ with $i \neq j$ are equal to zero. So, at the end we have

$$Var(y_t) = E(y_t)^2 = (c_0^2 + c_1^2 + ... + c_n^2)\sigma^2 \tag{11}$$

With analogous reasoning, we can calculate autocovariance, that is this:

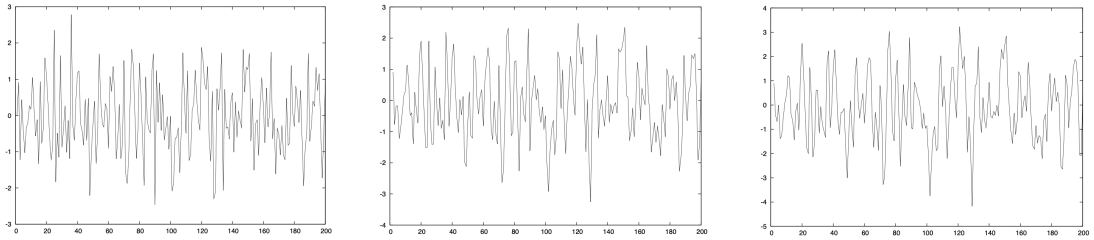
$$\gamma(k) = E(y(t), y(t+k)) = \sigma^2 \sum_{i=0}^{n} c_i c_{i+k} \tag{12}$$

We can note that the covariance between the values at any two time points, t, $t-k$, depends only on k, the difference between the two times, and not on the location of the points along the time axis. So, a MA process, is a process that is the linear combination of different elements of the same white noise with some persistency characteristics so much higher, so much is the order. Summarizing, a MA process has:

- constant mean value
- constant variance

- covariance function depending upon the distance between the two considered time points.

Example: let's take a MA process of order 1, MA(1), like $y(t) = \epsilon(t) + c_1\epsilon(t-1)$. We can plot $y(t)$ for 3 different values of c_1 to see what changes:



(a) MA(1): $c_1 = 0$ (white noise) (b) MA(1): $c_1 = 0.5$ (c) MA(1): $c_1 = 0.9$

Figure 9: MA(1) for 3 different coefficients

Obviously, when $c_1 = 0$ the process is a white noise. When c_1 increases, the persistence feature of the process becomes more visible (series is smoother) and the variance increases as well. If instead of having MA(1), we had a MA(n) with $n > 1$, this characteristic would have been more visible.

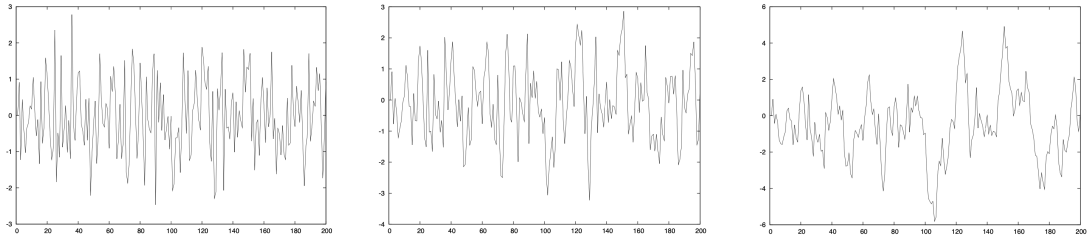
3.3 *AR processes*

Another important class of processes is the one of the AR (**Auto Regressive**). These processes are more intuitive of the MA, because the idea here is that the value of the series at time t is a linear combination of its own past values, plus a white noise. The name Auto Regressive, indeed, comes from the fact that an AR model is similar to a regression model.

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + ... + a_n y(t-n) + \epsilon(t) \tag{13}$$

AR processes are like the dual of MA processes because if a MA process is a process defined by applying the delay operator to a white noise, an AR is defined like a process on which is applied a delay operator, that produces a white noise: $A(L)y(t) = \epsilon(t)$.

Example: let's take a AR process of order 1, AR(1), like $y(t) = a_1 y(t-1) + \epsilon(t)$. We can plot $y(t)$ for 3 different values of a_1 to see what changes:



(a) AR(1): $a_1 = 0$ (white noise) (b) AR(1): $a_1 = 0.5$ (c) AR(1): $a_1 = 0.9$

Figure 10: AR(1) for 3 different coefficients

The first picture on the left is the white noise we have already seen also in the example of MA(1). Also in this case, we can notice an increase of the persistence characteristics when increasing the coefficient a_1, but here is more visible.

3.4 *ARMA processes*

The class of **ARMA** processes includes both AR and MA. An ARMA(p,q) is defined as:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + ... + a_{n_p} y(t-n_p) + c_0\epsilon(t) + c_1\epsilon(t-1) + ... + c_{n_q}\epsilon(t-n_q) \tag{14}$$

In a more compact way:

$$A(L)y(t) = C(L)\epsilon(t) \tag{15}$$

Obviously, AR and MA are special cases of ARMA processes. If $A(L)$ has all the roots $\geq |1|$ then $y(t)$ can be written in MA form: $y(t) = A(L)^{-1}C(L)\epsilon(t)$. At the same time, if $C(L)$ is invertible, then $y(t)$ can have an autoregressive representation $C(L)^{-1}A(L)y(t) = \epsilon(t)$. And we say in this case that the process is invertible.

3.5 *ARMAX processes*

Let's consider this equation:

$$\begin{aligned} y(t) = a_1 y(t-1) + a_2 y(t-2) + ... + a_{n_a} y(t-n_a) + \\ b_1 u(t-1) + b_2 u(t-2) + ... + b_{n_b} u(t-n_b) + \epsilon(t) \end{aligned} \tag{16}$$

We can notice the AR part, given by the terms with the a coefficients and the white noise, and another part, given by the terms with coefficient b. This other part is the **eXogenous** part, that is formed by the sequence of inputs $u(\cdot)$. This equation is of a *ARX* process. If we add the moving average process, we obtain:

$$\begin{aligned} y(t) = a_1 y(t-1) + a_2 y(t-2) + ... + a_{n_a} y(t-n_a) + \\ b_1 u(t-1) + b_2 u(t-2) + ... + b_{n_b} u(t-n_b) + \epsilon(t) + \\ c_1 \epsilon(t-1) + c_2 \epsilon(t-2) + ... + c_{n_c} \epsilon(t-n_c) \end{aligned} \tag{17}$$

That is the equation of an ARMAX process. Obviously, if there is no exogenous variable, the model becomes the ARMA model. ARMAX models are useful when you have dominating disturbances that enter early in the process, such as at the input. For example, a wind gust affecting an aircraft is a dominating disturbance early in the process. The ARMAX model has more flexibility than the ARX model in handling models that contain disturbances.

3.6 *ARIMA processes*

Before talking of the ARIMA processes, in particular what is the I, because we already know the AR and MA parts, we have to introduce some concepts about differencing.

3.6.1 *Differencing*

The models considered until now assume that the time series can be treated as realization of stationary stochastic processes. However, this is not true, especially when we treat macroeconomics series. For example, let's see the log of quarter italian PIL over time:

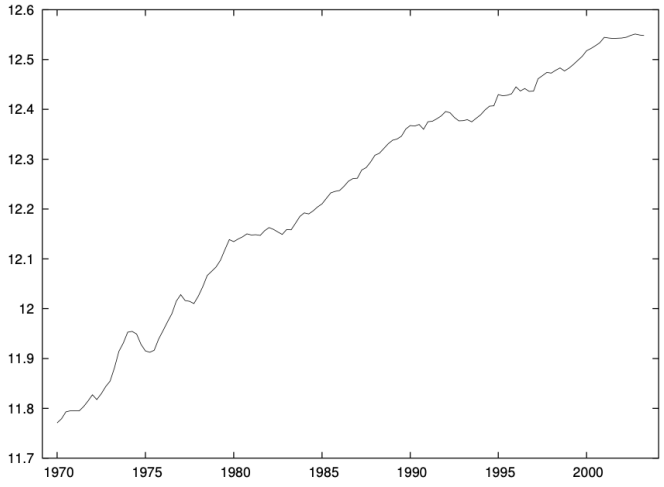


Figure 11: log(PIL)

The series has an increasing trend over time, so we can not use a stationary process for modeling. A stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends,

or with seasonality, are not stationary — the trend and seasonality will affect the value of the time series at different times. On the other hand, a white noise series is stationary — it does not matter when you observe it, it should look much the same at any point in time. Some cases can be confusing — a time series with cyclic behaviour (but with no trend or seasonality) is stationary. This is because the cycles are not of a fixed length, so before we observe the series we cannot be sure where the peaks and troughs of the cycles will be. In general, a stationary time series will have no predictable patterns in the long-term. Time plots will show the series to be roughly horizontal (although some cyclic behaviour is possible), with constant variance.

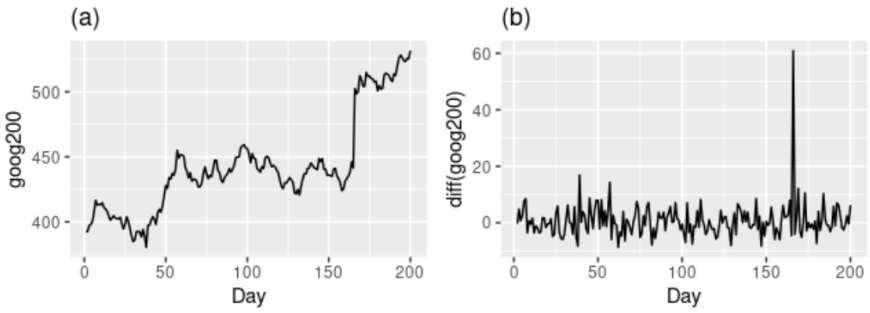


Figure 12: (a) Google stock price for 200 consecutive days; (b) Daily change in the Google stock price for 200 consecutive days;

In the last figure, note that the Google stock price was non-stationary in panel (a), but the daily changes were stationary in panel (b). This shows one way to make a non-stationary time series stationary — compute the differences between consecutive observations. This is known as **differencing**. Transformations such as logarithms can help to stabilise the variance of a time series. Differencing can help stabilise the mean of a time series by removing changes in the level of a time series, and therefore eliminating (or reducing) trend and seasonality.

As well as looking at the time plot of the data, the ACF plot is also useful for identifying non-stationary time series. For a stationary time series, the ACF will drop to zero relatively quickly, while the ACF of non-stationary data decreases slowly. Also, for non-stationary data, the value of γ_1 is often large and positive.

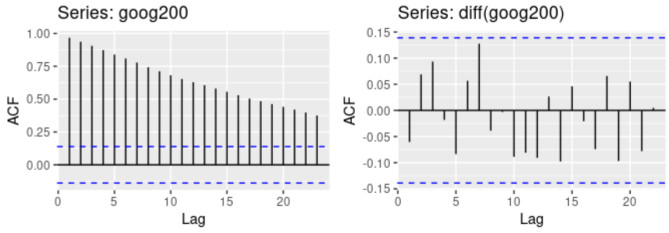


Figure 13: The ACF of the Google stock price (left) and of the daily changes in Google stock price (right).

The ACF of the differenced Google stock price looks just like that of a white noise series. There are no autocorrelations lying outside the 95% limits. This suggests that the daily change in the Google stock price is essentially a random amount which is uncorrelated with that of previous days.

The differenced series is the change between consecutive observations in the original series, and can be written as: $y(t)^{'} = y(t) - y(t-1)$. The differenced series will have only $T-1$ values, since it is not possible to calculate a difference $y(1)^{'}$ for the first observation. When the differenced series is white noise, the model for the original series can be written as $y(t) - y(t-1) = \epsilon(t)$. Rearranging this leads to the **random walk** model: $y(t) = y(t-1) + \epsilon(t)$. Random walk models are widely used for non-stationary data, particularly financial and economic data. Random walks typically have: long periods of apparent trends up or down, sudden and unpredictable changes in direction. The forecasts from a random walk model are equal to the last observation, as future movements are unpredictable, and are equally likely to be up or down.

Occasionally the differenced data will not appear to be stationary and it may be necessary to difference the data a second time to obtain a stationary series (*second order differencing*): $y(t)^{''} = y(t)^{'} - y(t-1)^{'} = y(t) - y(t-1) -$

$y(t-1) - y(t-2) = y(t-2)y(t-1) + y(t-2)$. In this case, $y(t)^{''}$ will have $T-2$ values. Then, we would model the "change in the changes" of the original data. In practice, it is almost never necessary to go beyond second-order differences.

One way to determine more objectively whether differencing is required is to use a *unit root test*. These are statistical hypothesis tests of stationarity that are designed for determining whether differencing is required. A number of unit root tests are available, which are based on different assumptions and may lead to conflicting answers. One of these test, for example, is the *KPSS test*. In this test, the null hypothesis is that the data are stationary, and we look for evidence that the null hypothesis is false. Consequently, small p-values (e.g., less than 0.05) suggest that differencing is required.

If we combine differencing with autoregression and a moving average model, we obtain a non-seasonal **ARIMA** model. ARIMA is an acronym for AutoRegressive Integrated Moving Average (in this context, "integration" is the reverse of differencing). The full model can be written as

$$y(t)^{'} = a_1 y^{'}(t-1) + a_2 y^{'}(t-2) + ... + a_{n_p} y^{'}(t-n_p) + c_0\epsilon(t) + c_1\epsilon(t-1) + ... + c_{n_q}\epsilon(t-n_q) \tag{18}$$

where $y^{'}(t)$ is the differenced series (it may have been differenced more than once). The "predictors" on the right hand side include both lagged values of $y(t)$ and lagged errors. We call this an ARIMA(p,d,q) model, where p = order of the autoregressive part, d = degree of first differencing involved, q = order of the moving average part.

Many of the models we have already discussed are special cases of the ARIMA model, as shown in table:

Table 1: Special cases of ARIMA models

White noise	ARIMA(0,0,0)
Random Walk	ARIMA(0,1,0)
Autoregression	ARIMA(p,0,0)
Moving Average	ARIMA(0,0,q)

It is usually not possible to tell, simply from a time plot, what values of p and q are appropriate for the data. However, it is sometimes possible to use the ACF plot, and the closely related PACF plot, to determine appropriate values for p and q.

Recall that an ACF plot shows the autocorrelations which measure the relationship between $y(t)$ and $y(t-k)$ for different values of k. Now if $y(t)$ and $y(t-1)$ are correlated, then $y(t-1)$ and $y(t-2)$ must also be correlated. However, then $y(t)$ and $y(t-2)$ might be correlated, simply because they are both connected to $y(t-1)$, rather than because of any new information contained in $y(t-2)$ that could be used in forecasting $y(t)$. To overcome this problem, we can use **partial autocorrelations**. These measure the relationship between $y(t)$ and $y(t-k)$ after removing the effects of lags 1,2,3,...,k-1. So the first partial autocorrelation is identical to the first autocorrelation, because there is nothing between them to remove. Each partial autocorrelation can be estimated as the last coefficient in an autoregressive model. Specifically, α_k, the k^{th} partial autocorrelation coefficient, is equal to the estimate of $a(k)$ in an AR(k) model. In practice, there are more efficient algorithms for computing α_k than fitting all of these autoregressions, but they give the same results. Let's look at an example: consider this time series about US consumption:

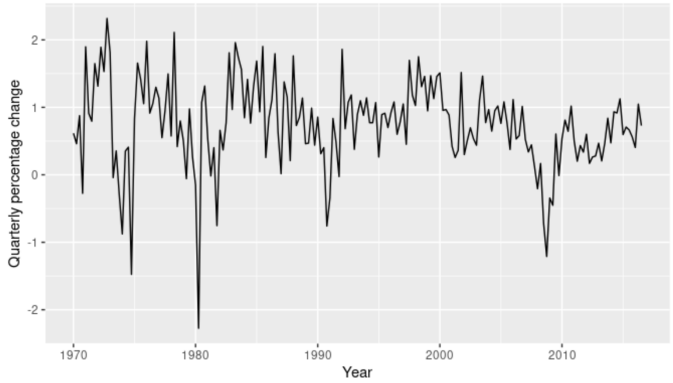
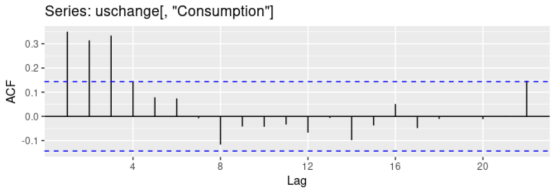
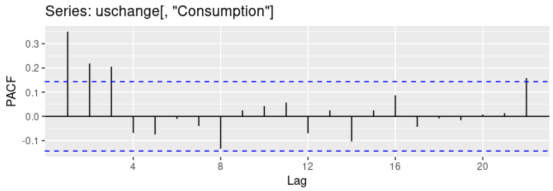


Figure 14: Quarterly percentage change in US consumption expenditure.

Figures below show the ACF and PACF plots for the US consumption data shown in Figure 8.7.



(a) ACF of quarterly percentage change in US consumption.



(b) PACF of quarterly percentage change in US consumption.

If the data are from an ARIMA(p,d,0) or ARIMA(0,d,q) model, then the ACF and PACF plots can be helpful in determining the value of p or q. If p and q are both positive, then the plots do not help in finding suitable values of p and q. The data may follow an ARIMA(p,d,0) model if the ACF and PACF plots of the differenced data show the following patterns:

- the ACF is exponentially decaying or sinusoidal;
- there is a significant spike at lag p in the PACF, but none beyond lag p.

The data may follow an ARIMA(0,d,q) model if the ACF and PACF plots of the differenced data show the following patterns:

- the PACF is exponentially decaying or sinusoidal;
- there is a significant spike at lag q in the ACF, but none beyond lag q.

In the ACF plot, we see that there are three spikes, followed by an almost significant spike at lag 4. In the PACF, there are three significant spikes, and then no significant spikes thereafter (apart from one just outside the bounds at lag 22). We can ignore one significant spike in each plot if it is just outside the limits, and not in the first few lags. After all, the probability of a spike being significant by chance is about one in twenty, and we are plotting 22 spikes in each plot. The pattern in the first three spikes is what we would expect from an ARIMA(3,0,0), as the PACF tends to decrease. So in this case, the ACF and PACF lead us to think an ARIMA(3,0,0) model might be appropriate.

3.7 *SARIMA processes*

So far, we have restricted our attention to non-seasonal data and non-seasonal ARIMA models. However, ARIMA models are also capable of modelling a wide range of seasonal data. A **seasonal ARIMA** (SARIMA) model is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows:

$$\text{ARIMA}\quad \underbrace{(p,d,q)}_{\text{Non-seasonal part of the model}}\quad \underbrace{(P,D,Q)_m}_{\text{Seasonal part of of the model}}$$

Figure 16: SARIMA structure.

where m=number of observations per year. We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model. The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model, but involve backshifts (delay) of the seasonal period. For example, an $SARIMA(1,1,1)(1,1,1)_4$ model is for quarterly data (m=4), and can be written as

$$(1-\phi_1 B)(1-\phi_1 B^4)(1-B)(1-B^4)y(t) = (1+\theta_1 B)(1+\theta_1 B^4)\epsilon(t). \tag{19}$$

The additional seasonal terms are simply multiplied by the non-seasonal terms.

The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF. For example, a $SARIMA(0,0,0)(0,0,1)_{12}$ model will show:

- a spike at lag 12 in the ACF but no other significant spikes;
- exponential decay in the seasonal lags of the PACF (i.e., at lags 12, 24, 36, ...).

Similarly, a $SARIMA(0,0,0)(1,0,0)_{12}$ model will show:

- exponential decay in the seasonal lags of the ACF;
- a single significant spike at lag 12 in the PACF.

In considering the appropriate seasonal orders for a seasonal ARIMA model, restrict attention to the seasonal lags. The modelling procedure is almost the same as for non-seasonal data, except that we need to select seasonal AR and MA terms as well as the non-seasonal components of the model. The process is best illustrated via an example.

We will describe the seasonal ARIMA modelling procedure using quarterly European retail trade data from 1996 to 2011. The data are plotted in the following picture:

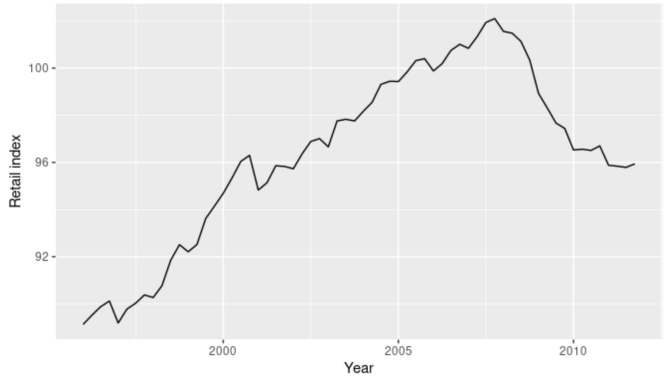


Figure 17: Quarterly retail trade index in the Euro area (17 countries), 1996–2011, covering wholesale and retail trade, and the repair of motor vehicles and motorcycles.

The data are clearly non-stationary, with some seasonality, so we will first take a seasonal difference, and an additional first difference, because the data after the seasonal difference appeared to be non-stationary.

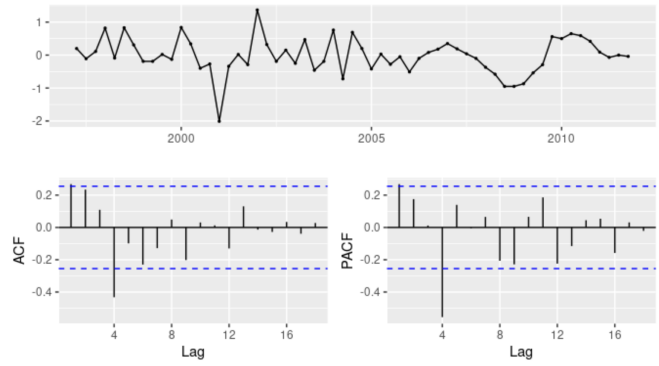


Figure 18: Double differenced European retail trade index.

The significant spike at lag 1 in the ACF suggests a non-seasonal MA(1) component, and the significant spike at lag 4 in the ACF suggests a seasonal MA(1) component. Consequently, we begin with a $SARIMA(0,1,1)(0,1,1)_4$ model, indicating a first and seasonal difference, and non-seasonal and seasonal MA(1) components. By analogous logic applied to the PACF, we could also have started with a $SARIMA(1,1,0)(1,1,0)_4$ model.

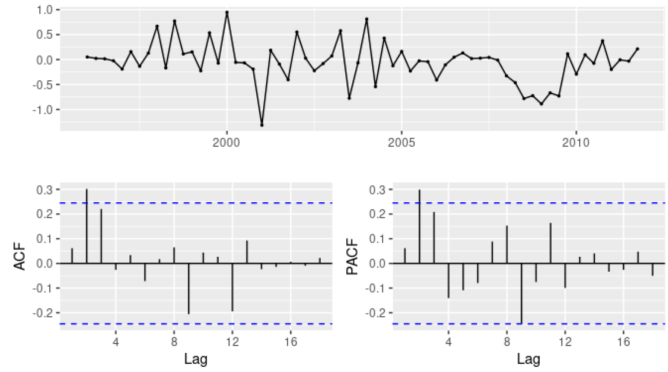


Figure 19: Residuals from the fitted $SARIMA(0,1,1)(0,1,1)_4$ model for the European retail trade index data.

Both the ACF and PACF show significant spikes at lag 2, and almost significant spikes at lag 3, indicating that some additional non-seasonal terms need to be included in the model.

3.8 *Multivariate Time Series - VAR*

A univariate time series, as the name suggests, is a series with a single time-dependent variable. A Multivariate time series has more than one time-dependent variable. Each variable depends not only on its past values but also has some dependency on other variables. Rarely, a complex phenomenon can be described with only one variable. We need to use multiple random variables, or random vectors. In a **VAR** model, each variable is a linear function of the past values of itself and the past values of all the other variables. There are some pros and cons. For example, one problem is the curse of dimensionality: the number of hyperparameters increases exponentially. Let's try to extend some known concepts to the multivariate case. A model is *k-varied* if it made of k series. Given a multivariate series $X_t = (X_{t_1}, ..., X_{t_k})^{'}$ we define the expectation μ_t as $\mu_t = E(X_t) = (E(X_{t_1}), ..., E(X_{t_k}))^{'}$. So, we can have that X_t is a matrix and μ_t is a vector. We can define also the covariance matrix $\Gamma(h) = cov(X_{t+h}, X_t)$ as:

$$\begin{pmatrix} \gamma_{11}(t+h,t) & ... & \gamma_{1k}(t+h,t) \\ ... & ... & ... \\ \gamma_{k1}(t+h,t) & ... & \gamma_{kk}(t+h,t) \end{pmatrix}$$

With these two moments we can extend the concept of stationarity to the multivariate case. If μ_t and $\Gamma(t+h,t)$ are independent by t, then the series is stationary. We can define the matrix of autocorrelation $R(H)$ as

$$\begin{pmatrix} \rho_{11}(h) & ... & \rho_{1k}(h) \\ ... & ... & ... \\ \rho_{k1}(h) & ... & \rho_{kk}(h) \end{pmatrix}$$

The elements of this matrix are functions of h, the lag. This R matrix represents the relationship between the single time series. If matrix R was an identity matrix (all 1s in principal diagonal), it would not be useful building a multivariate model, while it would be good to work on each series individually. For example, a bivariate series: $X_{t1} = Z_t$ and $X_{t2} = Z_t + 0.75Z_{7-10}$ with Z_t that is a white noise with zero mean and unit variance. In this case, $\mu = 0$, but we can calculate the covariance matrix $\Gamma(0) = cov(X_t, X_t)$:

$$\mathbf{\Gamma}(0) = \mathrm{E}\left[\begin{pmatrix} Z_t \\ Z_t + 0.75Z_{t-10} \end{pmatrix}(Z_t, Z_t + 0.75Z_{t-10})\right] = \begin{pmatrix} 1 & 1 \\ 1 & 1 + (0.75)^2 \end{pmatrix}$$

Similarly,

$$\mathbf{\Gamma}(-10) = \begin{pmatrix} 0 & 0.75 \\ 0 & 0.75 \end{pmatrix} \quad and \quad \mathbf{\Gamma}(10) = \begin{pmatrix} 0 & 0 \\ 0.75 & 0.75 \end{pmatrix}$$

A multivariate process is a white noise, if and only if Z_t is stationary with zero mean and covariance matrix equal to the autocovariance matrix for $h = 0$, or 0 otherwise.

Z_t is i.i.d. $(0, \Sigma)$ if Z_t is identically and independently distributed with zero mean and covariance Σ.

X_t is a linear process if it can be expressed as $X_t = \sum_{j=-\infty}^{\infty} C_j Z_{t-j}$.

Now, what is it an ARMA process for a multivariate series? As in the univariate case, we can define an extremely useful class of multivariate stationary processes X_t by requiring that X_t should satisfy a set of linear difference equations with constant coefficients. The multivariate white noise Z_t constitutes the fundamental building block for constructing vector ARMA processes. X_t is an ARMA(p,q) process if X_t is stationary and if for every t,

$$X_t - \phi_1 X_{t-1} - ... - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + ... + \theta_q Z_{t-q} \tag{20}$$

where Z_t is a white noise with zero mean and covariance matrix. X_t is an ARMA(p,q) process with mean μ if $X_t - \mu$ is an ARMA(p,q) process.

We understand here the curse of dimensionality: dimensions explode because the unknown variables are not vectors but matrices:

$$\phi(z) = I_k - \phi_1 z - ... - \phi_p z^p \tag{21}$$

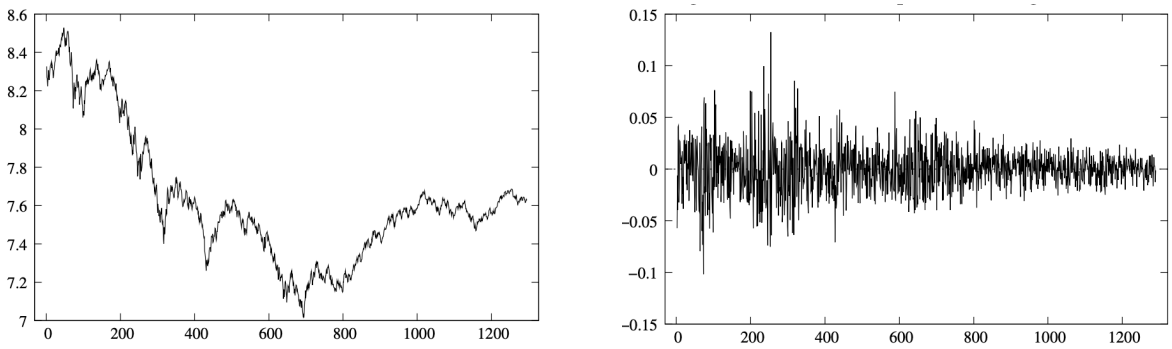
and

$$\theta(z) = I_k - \theta_1 z - ... - \theta_p z^p \tag{22}$$

A model for a multivariate time series makes sense if the model is *causal* and *invertible*. For causality we need that $det(\phi(z)) \neq 0$ for every z such that $|z| \leq 1$. For the invertibility we need $det(\theta(z)) \neq 0$ for every z such that $|z| \leq 1$. In a causal system is impossible to say in which way the process will evolve, using only information until time t. It seems obvious: in real systems, the future evolution has no influence on the past of the system.

3.9 *GARCH*

Let's see the pictures below:



(a) log of Nasdaq index

(b) diff-log of Nasdaq index

The picture of the left shows the natural logarithm of the daily Nasdaq index from 03/01/2000 to 28/02/2005. We take the difference of the series, and we obtain the **returns** (picture on the right). We don't see any significative correlation in the returns; however, we can notice that the amplitude of the oscillations changes over time, and that periods (also of long length) of low volatility are alternated with periods of high volatility. These clusters of volatility are called *volatility clusters*. This suggests that if there is no persistence in the mean, there could be persistence in the variance, or in general in the volatility. This is an important characteristic, indeed the market volatility is related to its risk level; so the possibility of predicting the volatility of a market is very important in every asset allocation. Persistence in volatility is very visible in the above picture that shows the absolute values of the returns (another index of volatility)

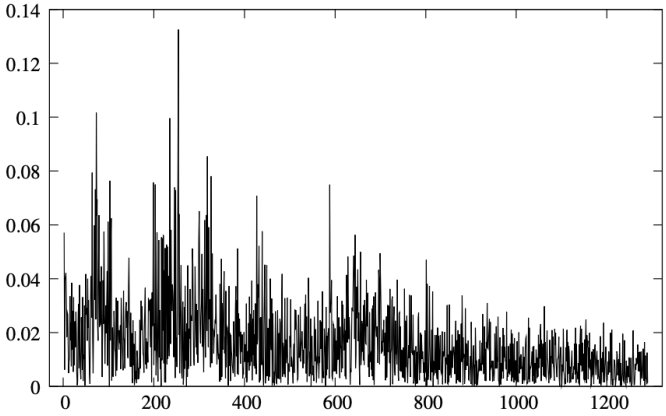


Figure 21: Absolute values of Nasdaq returns

Another important thing about financial series is the shape of their marginal distribution: in an ARMA process if the joint distribution of $\epsilon(t)$ is a multivariate normal, it is the same also for the distribution of $y(t)$. Instead, we can see from the returns of the Nasdaq index, that there are traces of non-normality, like a lot of kurtosis. So, if we want to use an ARMA process, we would have to use another type of distribution for the noise, instead of using gaussian.

Let's say for a moment what are the returns: they are the profits, and they are defined in this way:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} \tag{23}$$

where t is the time, P is the price of an asset; $R \in [-1; +\infty[$, because $P \geq 0$. This if we consider 1 time step; instead, if we consider k time steps, then the returns are defined as:

$$R_{t-k:t} = \frac{P_t - P_{t-k}}{P_{t-k}} \tag{24}$$

This is equal to:

$$1 + R_{t-k:t} = \frac{P_t}{P_{t-k}} \tag{25}$$

We prefer to work with the log of returns:

$$\epsilon_t = log(P_t) - log(P_{t-1}) = log(\frac{P_t}{P_{t-1}}) \tag{26}$$

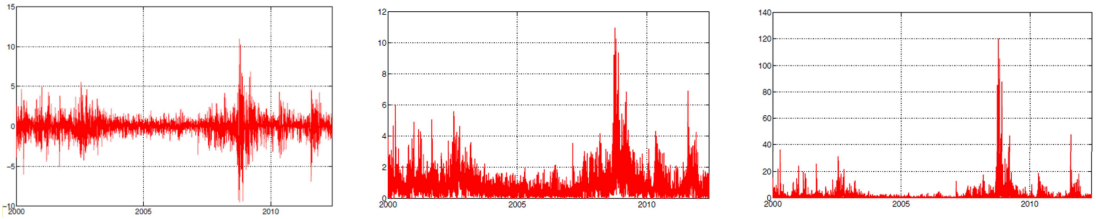
And the same is true for k time steps. Often we work with

$$\epsilon_t = 100(log(P_t) - log(P_{t-1})) = 100log(\frac{P_t}{P_{t-1}}) \tag{27}$$

to work with bigger numbers.

Which kind of financial assets can we consider? We can work on common stocks (IBM, Apple, ...), on stock indices (Dow Jones, S&P,...), on ETFs (Exchange Traded Funds).

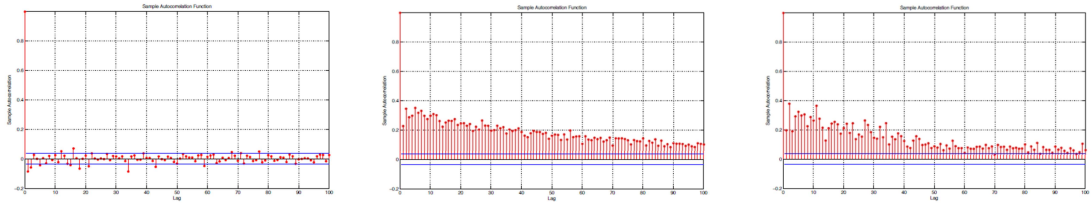
The return series are similar to random walks. Remember that random walks are time series in which future values depend at most by the present sample. This means that in the ACF, as always the first value will be 1, but already the second value will be inside the non-significance range.



(a) Daily returns (b) Daily Absolute returns (c) Daily squared returns

Figure 22: S&P500 returns

The inspection of the daily return time series plot suggests that returns appear to have weak or no serial dependence, and that absolute and squared returns appear to have strong serial dependence.



(a) Daily returns ACF (b) Daily Absolute returns ACF (c) Daily squared returns ACF

Figure 23: S&P500 returns ACF

We can not use ARMA models, because the hp of stationarity is not satisfied. We will use GARCH models. First of all, the conditional mean is defined as $\mu_t = E[y_t|I_{t-1}]$, that basically is the mean of y_t using only the Information until $t-1$. The same for the conditional variance: $\sigma_t^2 = Var[y_t|I_{t-1}]$

Can we use an ARMA model? For example, an ARMA(1,1) is: $y_t = \phi_0 + \phi_1 y_{t-1} + \epsilon_t - \theta_1 \epsilon_{t-1}$

This ARMA model will have $\mu_t = \phi_0 + \phi_1 y_{t-1} - \theta_1 \epsilon_{t-1}$ and $\sigma_t^2 = \sigma_\epsilon^2$.

Thus, while the conditional mean of an ARMA is time varying, the conditional variance of an ARMA is constant. An ARMA(p,q) is not able to capture time varying volatility. There are two approaches in modelling the conditional variance σ_t^2: the ARCH approach (where σ_t^2 is a deterministic equation) and the stochastic volatility approach (where σ_t^2 is a stochastic equation). ARCH approaches are more popoular than Stochastic Volatility Models are typically harder to work with. ARCH is the acronym of **AutoRegressive Conditional heteroskedasticity**. The simplest linear model able to model processes with non-constant variance (*heteroskedasticity*) is: $y_t = \epsilon_t = \sigma_t z_t$ with $E[z_t] = 0$ and $E[z_t^2] = 1$, where z_t is called *innovation* and is the new part, that is unpredictable. σ_t is the volatility (standard deviation). For example, and ARCH(1) is defined as: $\epsilon_t = \sqrt{\sigma_t^2} z_t$ where z_t has a distribution with zero mean and unit variance, and $\sigma_t^2 = w + \alpha \epsilon_{t-1}^2$ with $w, \alpha \geq 0$. In general, ARCH(q) have $\sigma_t^2 = w + \alpha_1 \epsilon_{t-1}^2 + ... + \alpha_q \epsilon_{t-q}^2$. Generally coefficients are estimated with the ML (Maximum Likelihood). Remember that the model will not try to predict the price, but the volatility, how varies the variance.

ARCH models are not perfect for some applications: only ARCH models with lots of parameters can fit financial series adequately, but at the same time, largely parameterized models can be unstable in forecasting. In order to overcome these problems, there is a generalization of the ARCH, called GARCH, that fits adequately financial returns while keeping the number of parameters small. A GARCH(1,1) model is $\epsilon_t = \sqrt{\sigma_t^2} z_t$ where z_t has a distribution with zero mean and unit variance, and $\sigma_t^2 = w + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2$ with $w > 0, \alpha \geq 0, \beta > 0 and \alpha + \beta < 1$ (in order to have stationarity). Another generalization is when $\alpha + \beta = 1$: in this case we have IGARCH.

3.10 *NARMAX*

Seasonal ARIMAX are the family of models commonly used to model linear systems. Linear is the key word here. For non-linear scenarios we have the **NARMAX** class (Non-linear Autoregressive Models with Moving Average and Exogenous Input). NARMAX models are not, however, a simple extension of ARMAX models. NARMAX models

are able to represent the most different and complex nonlinear systems. Introduced in 1981 by the Electrical Engineer Stephen A. Billings, NARMAX models can be described as:

$$y(t) = F^l[y(t-1), ..., y(t-n_y), x(t-d), x(t-d-1), ..., x(t-d-n_x) + \epsilon(t) + \epsilon(t-1), ..., \epsilon(t-n_\epsilon)] \quad (28)$$

where $n_y, n_x and n_\epsilon$ are the maximum lags for the system output and input respectively; In this case, F^l is some nonlinear function of the input and output regressors with non-linearity degree $l \in N$ and d is a time delay typically set to $d = 1$.

If we do not include noise terms, $\epsilon(k - n_\epsilon)$, we have NARX models. If we set $l = 1$ then we deal with ARMAX models; if $l = 1$ and we do not include input and noise terms, it turns to AR model (ARX if we include inputs, ARMA if we include noise terms instead); if $l > 1$ and there is no input terms, we have the NARMA. If there is no input or noise terms, we have NAR. There are several variants, but that is sufficient for now.

4 Model Identification and Validation

According to the family of the selected model, there are better identification and validation techniques. The two standard tools used to identify an appropriate ARMA model for a given stationary time series are the ACF and PACF. Several other potentially useful diagnostic tools are also available. An alternative to the partial ACF is the inverse ACF for example. Then, instead of subjectively examining functions like the ACF, an alternative type of approach is to choose the ARMA model which optimizes a suitably chosen function of the data. One approach, based on **Akaike's final prediction error** (**FPE**) criterion, is concerned with comparing AR processes of different order. The order p is essentially selected so as to get the estimated one-step-agead predictor with the smallest mean squared error. A more general criterion for model selection is to minimize a quantity called **Akaike's information criterion** (**AIC**). The AIC can be used to compare ARMA models as well as AR models. Akaika has also developed a Bayesian modification of AIC, denoted by **BIC**, which penalizes models with large number of parameters in a more severe way than the AIC.

Another commonly used statistic to measure goodness of fit of a stationary model is the R square (R^2) defined as $R^2 = 1 - \frac{residual sum of squares}{total sum of squares}$. It is easy to show that $0 \leq R^2 \leq 1$. Typically, a larger R^2 indicates that the model provides a closer fit to the data. However, this is only true for a stationary time series. For a given data set, it is well known that R^2 is a non decreasing function of the number of parameters used. To overcome this weakness, an *adjusted* R^2 is proposed, which is defined as: $Adj - R^2 = 1 - \frac{variance of residuals}{variance of r_t}$. Remember that residuals are the difference between the observed values y_i and the corresponding fitted values. Other indices can be MSE (Mean Squared Error), MAE (Mean Absolute Value), and a lot of other different types of losses.

5 Software Tools

The software used during the course of *Modeling and Analysis for Complex Systems* is MATLAB. Inside MATLAB there are thousands of functions and toolboxes: we mainly used 3 toolboxes: *System Identification Toolbox* (Ident), *Econometrics Toolbox* and *Deep Learning Toolbox*.

System Identification Toolbox provides MATLAB functions, Simulink blocks, and an app for constructing mathematical models of dynamic systems from measured input-output data. It lets you create and use models of dynamic systems not easily modeled from first principles or specifications. You can use time-domain and frequency-domain input-output data to identify continuous-time and discrete-time transfer functions, process models, and state-space models. The toolbox also provides algorithms for embedded online parameter estimation. The System Identification toolbox provides identification techniques such as maximum likelihood, prediction-error minimization (PEM), and subspace system identification. To represent nonlinear system dynamics, you can estimate Hammerstein-Weiner models and nonlinear ARX models with wavelet network, tree-partition, and sigmoid network nonlinearities. The toolbox performs grey-box system identification for estimating parameters of a user-defined model. You can use the identified model for system response prediction and plant modeling in Simulink. The toolbox also supports time-series data modeling and time-series forecasting.

Econometrics Toolbox provides functions for analyzing and modeling time series data. It offers a wide range of visualizations and diagnostics for model selection, including tests for autocorrelation and heteroscedasticity, unit roots and stationarity, cointegration, causality, and structural change. You can estimate, simulate, and forecast economic systems using a variety of modeling frameworks. These frameworks include regression, ARIMA, state-space, GARCH, multivariate VAR and VEC, and switching models. The toolbox also provides Bayesian tools for developing time-varying models that learn from new data.

Deep Learning Toolbox provides a framework for designing and implementing deep neural networks with algorithms, pretrained models, and apps. You can use convolutional neural networks (ConvNets, CNNs) and long short-term memory (LSTM) networks to perform classification and regression on image, time-series, and text data. You can build network architectures such as generative adversarial networks (GANs) and Siamese networks using automatic differentiation, custom training loops, and shared weights. With the Deep Network Designer app, you can design, analyze, and train networks graphically. The Experiment Manager app helps you manage multiple deep learning experiments, keep track of training parameters, analyze results, and compare code from different experiments. You can visualize layer activations and graphically monitor training progress.

6 Case study

Let's see the case study of this report. I've chosen a time series coming from the *ACM Nanocom Data Competition* that is related to experimental molecular communication systems. Data were collected using E. Coli bacteria that express the light-driven proton pump gloeorhodopsin. Upon an external light stimulus, these bacteria (as a transmitter) export protons (as the signaling molecules) into the channel. This changes the pH level of the environment, which can be detected with a pH sensor (as the receiver). In this way, a sequence of light pulses can be converted into a pH signal which carries the underlying information. This signal conversion yields a received signal which is highly non-linear, time-variant, and noisy. Moreover, due to the unpredictable bacteria behavior, each realization can be considered random. To provide a consistent data set for this competition, the training data is generated by simulation based on parameters obtained by experiments with real bacteria. The transmission parameters are as follows. The symbol period is 60 seconds, and the sam- pling period is 1 second, i.e., there are 60 samples per symbol. The measurement includes the pH before and after transmission.

Let's see the time series:

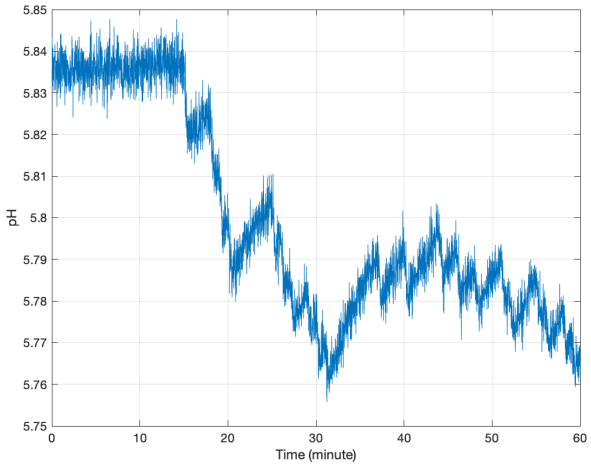


Figure 24: pH time series

From the plot, it seems that the series is not stationary: we can notice some relevant trends and changing levels, but in a while, we will check if it is stationary or not, to be sure. Some values of the series: mean $\mu = 5.7642, \sigma = 0.0399, \sigma^2 = 0.0016$.

So, the first way to check stationarity, is seeing ACF:

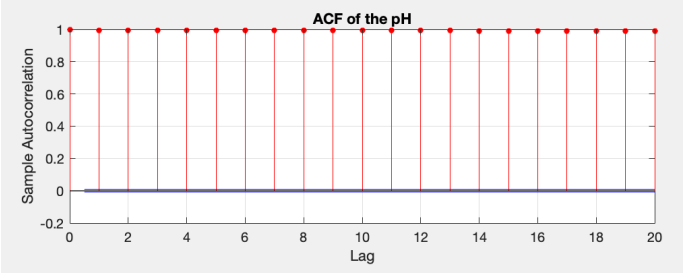
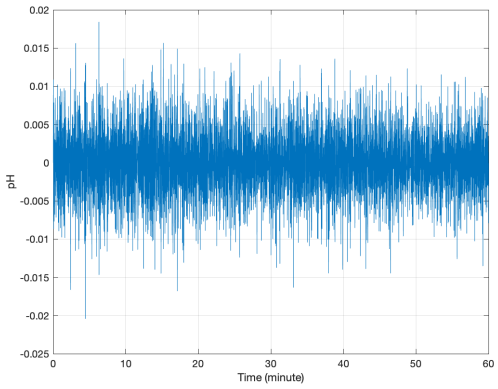


Figure 25: ACF of pH time series

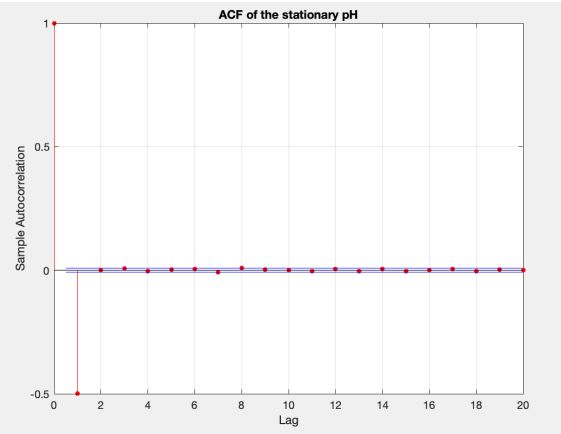
When plotting the value of the ACF for increasing lags (a plot called a correlogram), the values tend to degrade to zero quickly for stationary time series), while for non-stationary data the degradation will happen more slowly. In this case we can see a very slow degradation, so we have the confirm that the time series is non-stationary.

Another, more rigorous approach, to detecting stationarity in time series data is using statistical tests developed to detect specific types of stationarity, namely those brought about by simple parametric models of the generating stochastic process. The test performed in MATLAB are the *Augmented Dickey-Fuller test* and the *KPSS*, and both of them confirmed that the series is non stationary.

In order to make the time series stationary, I took the difference. So, for each value in our time series we subtract the previous value. This will give us a NaN value at first place because there is no previous value: to avoid this I will pad the new vector with a 0.



(a) Diff - pH time series



(b) ACF of the diff - pH time series

Now we have a stationary time series, the diff-pH. Stationarity is confirmed also by the ACF that degraded to 0 very quickly (at lag 1), and by the *adftest* function. For this new time series, the mean is a value very close to 0, as well as variance and standard deviation.

This picture shows the spectrum. No relevant information can be extracted from this periodogram. There are not characteristic peaks.

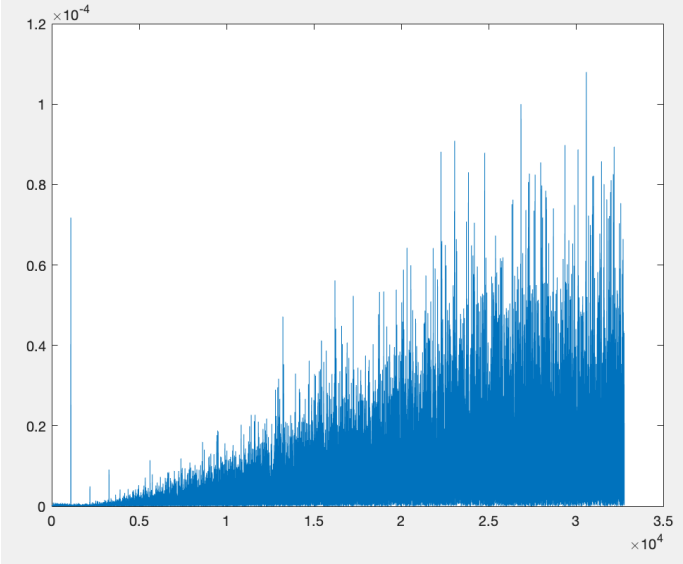
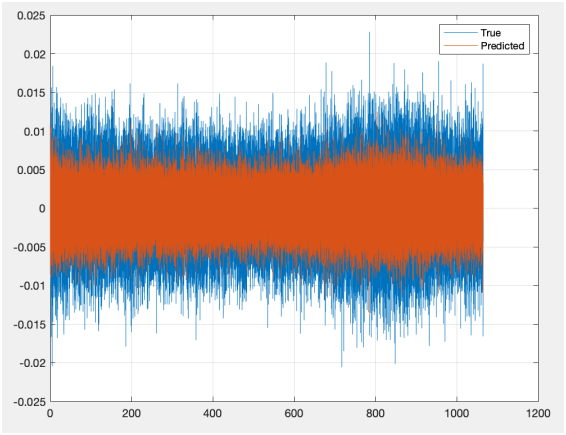
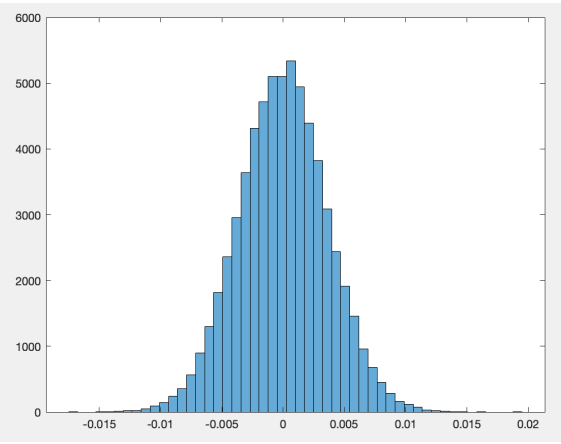


Figure 27: pH spectrum

Now, I started modeling the series using a simple AR model, using the *ar* function of the System identification Toolbox. Looking at the ACF of the differenced time series, the order of the model should have been 1, however, doing different attempts, I found a better fit using an order model of 3. The prediction was maded using one step prediction.



(a) True and predicted time series



(b) Residual analysis

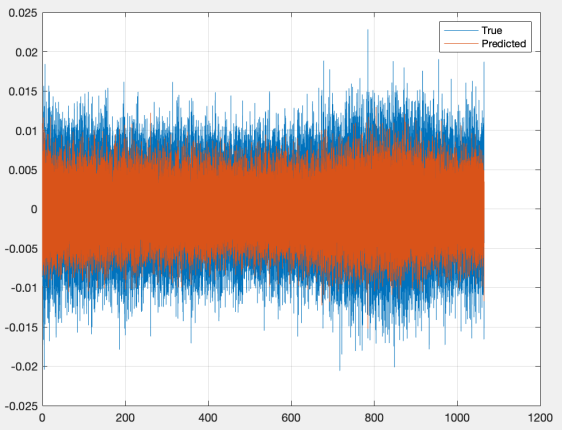
From the residual analysis we can see that the errors are normally distributed.

```
ARMdl =
Discrete-time AR model: A(z)y(t) = e(t)
  A(z) = 1 + 0.7401 z^-1 + 0.4853 z^-2 + 0.2353 z^-3

Sample time: 1 seconds

Parameterization:
   Polynomial orders:   na=3
   Number of free coefficients: 3
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
```

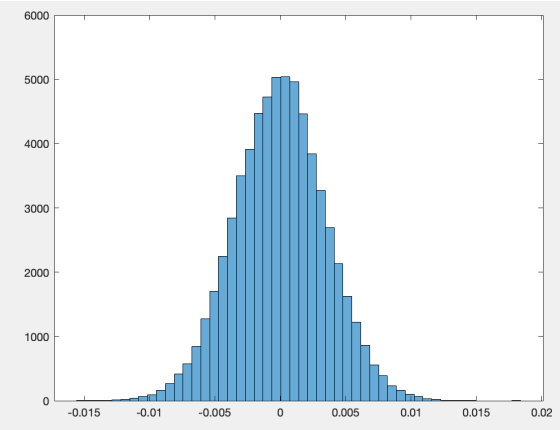
```
Estimated using AR ('fb/now') on time domain data "Y".
Fit to estimation data: 20.45%
FPE: 1.31e-05, MSE: 1.309e-05
```

The second model is an ARMA, with $n_a = 3$ and $n_c = 1$:

$$y(t) + a_1 y(t-1) + ... + a_3 y(t-3) = \epsilon(t) + c_1 \epsilon(t-1) \tag{29}$$



(a) True and predicted time series



(b) Residual analysis

```
ARMAMdl =
Discrete-time ARMA model: A(z)y(t) = C(z)e(t)
  A(z) = 1 + 0.07068 z^-1 + 0.04761 z^-2 + 0.02659 z^-3

  C(z) = 1 - 0.7736 z^-1

Sample time: 1 seconds

Parameterization:
   Polynomial orders:   na=3   nc=1
   Number of free coefficients: 4
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using ARMAX on time domain data.
Fit to estimation data: 23.61% (prediction focus)
FPE: 1.208e-05, MSE: 1.208e-05
```

Considerations are the same of AR model. We can note a slightly better fit, and a lower FPE and MSE.

Then I created an ARMAX model. However, I had not exogenous data (I also asked to creator of the data competition, but they refused my request to get additional data like the light stimulus), and so, obviously the created model is identical to the ARMA model of before. It was necessary only to declare variable $n_b and n_k$ that are the lags (k) and the coefficients for the exogenous input data. I will not report here the same graphs and results, but they are available in the MATLAB code.

Then I tried to model a SARIMA, using the function *arima* from Econometrics Toolbox. However, from the previous graphical analysis of the time series, and of the spectrum, I did not note any seasonality, so the parameters related to the seasonality are all 0, and it is basically an ARIMA model. Here, as parameter of the function I did not passed the differentiated time series, but the original one, because I selected 1 as differencing parameter (the I in ARIMA).

```
    ARIMA(3,1,2) Model (Gaussian Distribution):

                  Value       StandardError    TStatistic      PValue
              -----------    -------------    ----------    ----------
```

```
    Constant     -2.0823e-06     4.3078e-06       -0.48338         0.62883
    AR{1}          -0.98958      0.0071736         -137.95               0
    AR{2}         -0.022923      0.0076523         -2.9955       0.0027398
    AR{3}         -0.0046314     0.0051645       -0.89677         0.36984
    MA{1}          0.075594      0.0058106          13.01       1.0782e-38
    MA{2}          -0.77296      0.0054776         -141.11               0
    Variance      1.2628e-05     5.5255e-08          228.55               0


SARIMAEst =

  arima with properties:

     Description: "ARIMA(3,1,2) Model (Gaussian Distribution)"
    Distribution: Name = "Gaussian"
               P: 4
               D: 1
               Q: 2
        Constant: -2.08229e-06
              AR: {-0.989583 -0.0229226 -0.00463139} at lags [1 2 3]
             SAR: {}
              MA: {0.0755943 -0.772965} at lags [1 2]
             SMA: {}
     Seasonality: 0
            Beta: [1×0]
        Variance: 1.26283e-05
```

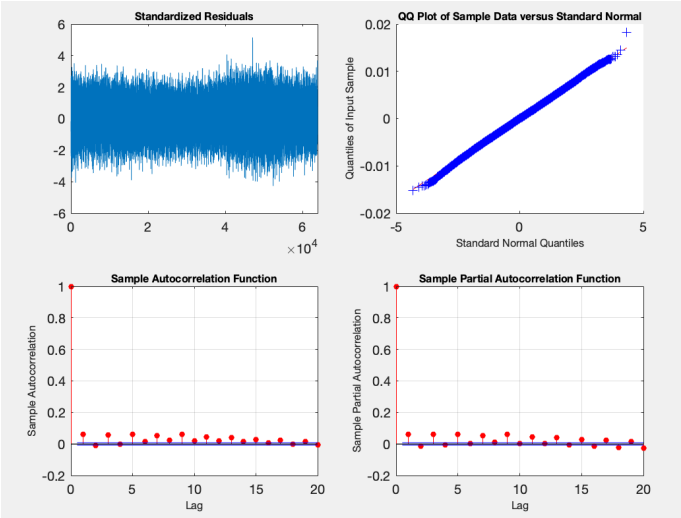
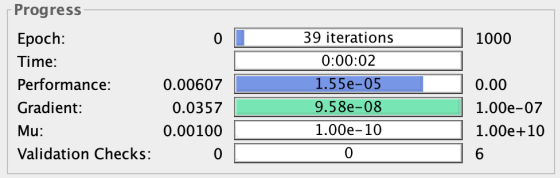


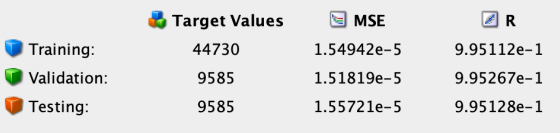
Figure 30: SARIMA results

In the picture above we can see the residuals, the q-q plot (we will see in a while what is it), ACF and PACF. A very simple method of checking the normality assumption is to construct a Q-Q plot of the residuals, which is a graph designed so that the cumulative normal distribution will plot as a straight line. In other words, the Q-Q plot, or quantile-quantile plot, is a graphical tool to help us assess if a set of data plausibly came from some theoretical distribution such as a Normal or exponential.

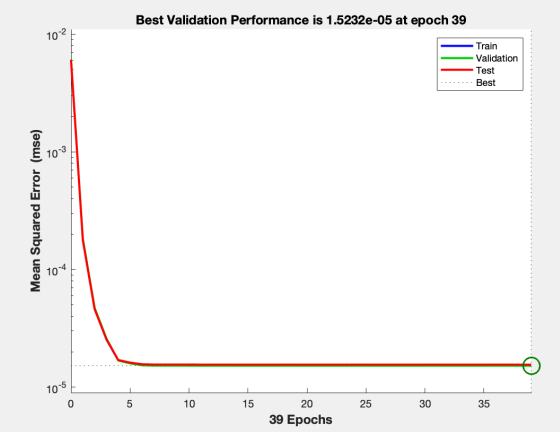
Then I used the *ntstool* (Neural Network Time Series Toolbox) to build a NAR model. Data were split into training (70%), validation (15%) and test (15%). Number of hidden neurons was set to 10, and number of delays to 2.



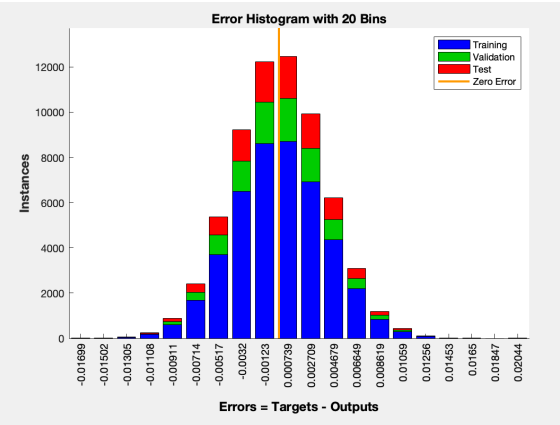
(a) Progress of NAR



(b) Train, val and test results



(a) Mean Squared Error across epochs



(b) Residual analysis

As we can see from the pictures, the training had a duration of 39 epochs. The number of epochs is decided by the tool. Performance (MSE) reached to 1.55e-05 on test set. We can see also that the errors are normally distributed for each of the sets.

Finally, the last employed model was a GARCH, from the Econometrics Toolbox. I used a GARCH component coefficient of 2, and an ARCH component coefficient of 1:

```
Description: "GARCH(2,1) Conditional Variance Model (Gaussian Distribution)"
Distribution: Name = "Gaussian"
           P: 2
           Q: 1
    Constant: NaN
       GARCH: {NaN} at lag [2]
        ARCH: {NaN} at lag [1]
      Offset: 0
```

So, the GARCH(2,1) conditional variance process σ_t^2 has the form:

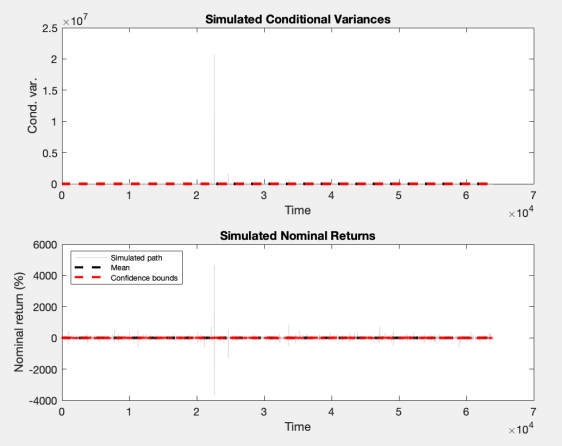
$$\sigma_t^2 = \gamma_1 \sigma_{t-1} + \gamma_2 \sigma_{t-2} + \alpha_1 \epsilon_{t-1}^2 \tag{30}$$

where $\epsilon = \sigma_t z_t$ and z_t is an independent and identically distributed standard Gaussian process.

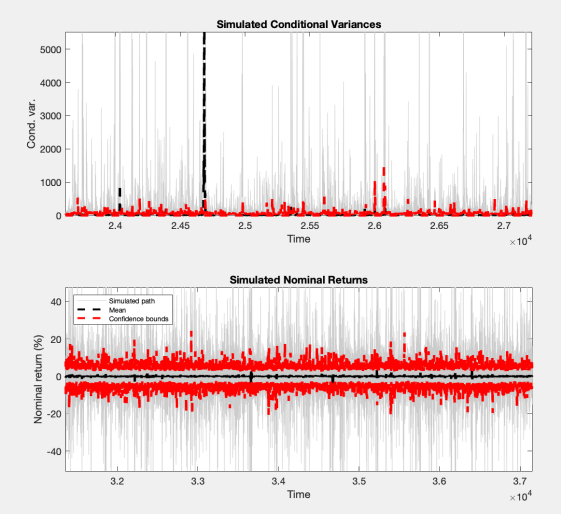
The estimation:

```
GARCH(2,1) Conditional Variance Model (Gaussian Distribution):

             Value      StandardError    TStatistic     PValue
            --------    -------------    ----------     -------

Constant      1.1921         11128        0.00010712    0.99991
GARCH{2}    0.017462        2691.6        6.4875e-06    0.99999
ARCH{1}      0.94665        2662.6        0.00035553    0.99972
```

We can simulate some steps of the model. I did 100 steps, obtaining VSim and YSim that are matrices with row dimension equal to the number of observations of the time series, and number of columns equal to the steps. So, rows correspond to a sample per period and columns correspond to a simulated path. Plot the average and the 97.5% and 2.5% percentiles of the simulated paths on the left, and the same image but zoomed in some point on the right.



(a) Simulation of GARCH



(b) Simulation of GARCH zoomed

In the MATLAB script you can find also some line of code to forecast using this model.

7 Conclusions

It was fun working on this project, and it was challenging this time series. Indeed, results were not good, like the fit estimation about 20-25% that is very low. MSE had low values, but it was due to the values of the time series. Reason why these models seem to not work well is that is very difficult to model natural phenomenons like the one of this case study, in which there are light pulses that are converted into pH values. All the natural phenomenon like weather, earthquakes, eruptions and so on, are very difficult to model. Instead, these stochastic processes are good for financial time series, like the ones used as example in the theory's chapter.

Appendix

You can find the dataset and the MATLAB script on this Github repository: https://github.com/CasellaJr/Modeling-and-Analysis-for-Complex-Systems

References

[1] *Slides of the course of prof. g. nunnari.*

[2] *Towards data science: link.*

[3] *Wikipedia: link.*

[4] C. CHATFIELD, *The analysis of time series, an introduction*, 1975.

[5] G. A. ROB J HYNDMAN, *Forecasting: Principles and practice.*

[6] TSAY, *Analysis of financial time series*, 2003.

(6) (4) (5) (1) (2) (3)