José Abílio Oliveira Matos

Entropy Measures Applied to Financial Time Series - an Econophysics Approach

Tese submetida à Faculdade de Ciências da Universidade do Porto para a obtenção do grau de Doutor em Matemática Aplicada

Departamento de Matemática Aplicada
Faculdade de Ciências da Universidade do Porto
Julho 2006

Acknowledgements

I would like to thank my advisers Sílvio and Heather for their support. DMA was a nice place to work and DCU was like a second home to me. It has been a funny journey.

I would like to thank José Duarte for his initial guidance and for allowing and encouraging me to pursue my goals.

The influence of good teachers stays for life. I would like to thank them all and namely Hermínio, Eduardo and Beatriz for their work/passion for the education of their students. You helped me to find answers but more importantly to find new questions.

This thesis was supported under a Prodep III grant "Prodep III, Acção 5.3 - Formação Avançada de Docentes do Ensino Superior, Concurso nº 2/2003".

I would like to thank the different Communities to which I belong, namely Free Software and Scouting, the life experience that I had has been priceless.

I would like to thank my friends for support and for lots of interesting conversations. You know who you are.

It is fair to remember my family, parents and aunt, brother and sisters for their patience and support. The same applies in-laws. A huge hug goes to my nephews, Ana, Jorge and Beatriz for getting the priorities right.

Last but not least I would like to thank my wife, Susana, for everything. Life is a lot funnier and rich (there should an obvious and at the same time confusing reference to complex numbers and quaternions here) with you.

Acknowledgements

Abstract

Econophysics is the main topic of this thesis. As an interdisciplinary research field, it seeks to apply theories and methods originally developed in statistical physics, with others coming from non-linear dynamics, in order to solve problems in Economics.

The focus of this work is on financial time series and the study of the values of indexes of the major markets worldwide. A stock market index is a listing of stocks and a statistics reflecting the composite value of its components. The stock index values give us statistics reflecting the actual state of the market. The time series resulting from these values is a registry of the stock market performance.

In order to decipher the secrets of financial time series, several mathematical and computational techniques are used here. On the computational side an emphasis is made on the use of Free Software and the repeatability of results. In the mathematical frame the emphasis is on entropy measures, with entropy understood here in a broader sense as a measure of uncertainty.

The first application of the techniques toolbox is PSI-20 (Portuguese Stock Index - 20), a Portuguese index of the 20 most liquid assets of the Portuguese Stock market.

Following the work on PSI-20 a new method is propose for studying the Hurst exponent, which includes investigation of both time and scale dependency. This approach permits the recovery of major events, affecting worldwide markets, (such as Sept. 11th 2001) and facilitates examination of the propagation of effects produced across different scales. Such effects may include early awareness, distinctive patterns of recovery, as well as comparative behaviour distinctions in emergent/established markets. The emphasis on time dependence serves to demonstrate the importance of entropy measures as snapshots of market uncertainty, which have their own dynamic.

We developed and applied a new technique, the TSDFA (Time and Scale Detrended Fluctuation Analysis), to study the time evolution of each market. Major features may include transition from a developing to a mature state, (International Finance Corporation definition). Comparing the results obtained using TSDFA to all markets, we identify groups that display similar behaviour at any given time. This classification allow us to distinguish perturbations with global or more general effect, (e.g. Asian tiger crash, 9/11, Madrid bomb attack in 2004 and others) from local influences affecting a small set of markets or even a single market only.

Interestingly, in spite of known differences between emerging and established markets, the evidence suggests that, in recent years, entropy measures are convergent across markets

studied worldwide. This can be construed as an increasing number of markets achieving or mimicking mature behaviour relatively rapidly, irrespective of their trading capability, which suggests that windows of opportunity are narrowing for investors. The stakes are raising.

Resumo

A Econofísica é o tema principal desta tese. A Econofísica é uma área de investigação interdisciplinar que aplica teorias e métodos desenvolvidas na Física Estatística, juntamente com outros da dinámica não linear, de modo a resolver problemas em Economia.

Este trabalho foca-se nas séries temporais financeiras e estuda o valor dos indíces das principais bolsas mundiais. O índice de uma bolsa de valores é uma listagem das acções juntamente com uma estatística que reflecte o valor composto dos seus componentes. O valor do indíce dá-nos uma estatística que reflecte estado actual do mercado. A série temporal resultante é um indicador da performance do mercado.

De modo a decifrar os segredos das séries temporais financeiras usamos aqui várias técnicas matemáticas e computacionais. Do lado computacional a ênfase é colocada no uso do Software Livre e na capacidade de repetir os resultados. Na vertente matemática a ênfase vais para as medidas de entropia, com a entropia a ser entendida aqui no sentido mais alargado como uma medida da incerteza.

A primeira aplicação do conjunto de ferramentas que se reuniu é no PSI-20 (Portuguese Stock Index - 20), o índice da Bolsa de Valores portuguesa com os 20 títulos mais líquidos do mercado.

No seguimento do estudo do PSI-20 proponho um novo método de estudar o expoente de Hurst, o qual inclui estudar a dependência temporal e de escala. Esta aproximação permite recuperar acontecimentos importantes que afectam a evolução das bolsas (tal como o 11 de Setembro de 2001) e facilita o exame da propagação dos efeitos nas diferentes escalas temporais. Esses efeitos incluem uma percepção precoce, distintos padrões de recuperação, assim como uma distinção no comportamento comparativamente a mercados emergentes/maduros. A ênfase na dependência temporal serve para demonstrar as medidas de entropias como "instantâneos" da incerteza do mercado, a qual tem a sua própria dinâmica.

Aplicamos a TSDFA (Time and Scale Detrended Fluctuation Analysis) ao estudo da série temporal de cada mercado. Algumas das principais características são, por exemplo, a mudança de um estado emergente para um estado desenvolvido. Comparando os resultados obtidos usando a TSDFA para todos os mercados permite-nos identificar grupos de mercados que um comportamento semelhante nos diferentes tempos. Esta classificação permite-nos distinguir perturbações globais (por exemplo o crash dos Tigres Asiáticos, o 111 de Setembro, o ataque do 11 de Março em Madrid e outros) das perturbações locais que afectam apenas um grupo restrito de mercados, quando não só apenas um.

Resumo

É intereressante que, apesar das diferenças conhecidas entre mercados emergentes e mercados estabelecidos, as evidências sugerem que, nos últimos anos, as medidas de entropia estão a convergir ao longo de todos os mercados estudados. Isto pode ser entendido uma vez que cada vez mais mercados alcançam ou mimetizam o comportamento dos mercados maduros de forma relativamente rápida, independentemente da sua capacidade de transacção, o que sugere um estreitar das oportunidades para os investidores. A parada está a aumentar.

Résumé

L'Econophysique est le subject principal de cette thèse. L'Econophysique est un champ interdisciplinaire de recherche, appliquant des théories et des méthodes développées à l'origine dans la physique statistique, aussi d'autres qui viennent de la dynamique non linéaire, afin de résoudre des problèmes dans les sciences économiques.

Le centre de ce travail est sur la série chronologique financière, étudiant les valeurs des indices des marchés principaux dans le monde entier. Un indice de marché boursier est une liste des stocks, et une statistique reflétant la valeur composée de ses composants. La valeur d'indice des actions nous donne des statistiques reflétant l'état réel du marché. La série chronologique résultant de prendre ces valeurs est un enregistrement de l'exécution du marché boursier.

Afin de déchiffrer les secrets de la série chronologique financière, plusieurs techniques mathématiques et informatiques sont employées ici. Du côté informatique une emphase est faite sur l'utilisation du logiciel libre et la répétabilité des résultats. Sur l'armature mathématique l'emphase va aux mesures d'entropie, avec l'entropie comprise ici dans un plus large sens, comme mesure d'incertitude.

La première application de la boîte à outils de techniques est PSI-20 (indice des actions portugais - 20), un indice portugais des 20 actifs plus disponibles du marché boursier portugais.

Après le travail sur PSI-20 je propose une nouvelle méthode d'étudier l'exposant de Hurst, qui inclut la recherche sur le temps et la dépendance de balance. Cette approche permet le rétablissement des événements principaux, affectant les marchés mondiaux, (comme le 11 septembre 2001) et facilite l'examen de la propagation des effets produits à travers différentes balances. De tels effets peuvent inclure la première conscience, modèles distinctifs de rétablissement, aussi bien que des distinctions comparatives de comportement sur des marchés d'emergent/établis. L'emphase sur la dépendance de temps sert à démontrer l'importance des mesures d'entropie comme instantanés de l'incertitude du marché, qui ont leur propre dynamique.

Nous appliquons le TSDFA (*Time and Scale Detrended Fluctuation Analysis*) à une étude de l'évolution temporelle de chaque marché. Les dispositifs principaux peuvent inclure la transition de se développer à un état mûr, (définition de société de finance internationale). Comparant les résultats obtenus en utilisant TSDFA à tous les marchés, nous identifions les groupes qui montrent comportement semblable à n'importe quelle heure indiquée. Cette classification nous permettent de distinguer des perturbations avec

l'effet global ou plus général, (par exemple le krash de tigres asiatiques, l'attaque de le 11 septembre 2001, l'attaque de bombe en 2004 a Madrid et d'autres) des influences locales affectant un petit ensemble de marchés ou même un marché seulement.

Intéressant, malgré des différences connues entre l'émergence et les marchés établis, l'évidence suggère que, ces dernières années, les mesures d'entropie sont convergentes à travers des marchés aient étudié dans le monde entier. Ceci peut être interprété comme un nombre croissant de marchés réalisant ou imitant le comportement mûr relativement rapidement, indépendamment de leurs possibilités marchandes, qui suggèrent un rétrécir des fenêtres pour des investisseurs.

Contents

List of Figures

List of Tables

1. Introduction

"*Science may be described as the art of systematic oversimplification.*" - Karl Popper

1.1. Econophysics

If I had to choose a single keyword to describe the work done in this thesis that word would be Econophysics. Econophysics is a research area that applies tools from statistical physics and more recently from dynamical systems and complex systems to the study of economic and financial problems. The coining of the term econophysics occurred in 1995 as it can be read from the author in a later article [Stanley, 1999].

Econophysics is associated with the interest of mathematics and physics in the study of complex systems. As can be seen in the Historical Perspective Section below, the motivation comes from many different contributions, from physics, mathematics and even (ironic for some) from Economics [Pareto, 1897].

In this introductory Chapter the motivation, branches and an historical perspective for Econophysics are presented. It is important to note that no techniques or tools presented are exclusive of Econophysics. Those ideas and tools are presented here because they have an important part in the development of the main ideas used in this work.

After introducing Econophysics as the general scientific area, the specific contribution of this thesis is outlined in Section 1.2. An overview of the work and a short walk through the thesis structure is sketched with the purpose of conveying the relation between the different parts of this work.

1.1.1. Motivation

Why are we interested in economy and finance?

There are several possible answers to this question. We (physicists and mathematicians) can work with empirical data and construct phenomenological theories. The quantitative nature of pure sciences allows a degree of abstraction when analysing series of numbers.

One other answer is that statistical physics has useful approaches to deal with collective dynamics in systems. These can be seen in such areas as traffic analysis (cars or network packets), granular mediums, foam studies, biomedical signals, earthquakes studies and river flow analysis, amongst others.

On the other hand current prevailing economic theory assumes equilibrium, with descriptions mostly static. Using the language of field theory as the theory of the economy similar to "mean field" theory in physics, actions and reactions are balanced without taking into account the interaction between the different agents.

For a summary of the motivation of physicists and mathematicians there is one quotation from Zhang [1998] that captures the spirit:

> "*One can never hope to get a future economy theory as quantitative and predictive as physical laws. However, this should not deter us from searching a framework to understand some basic phenomena qualitatively.*"

Relation with models

There are two alternatives to problem solving in econophysics: one is to use a model and, from there, study the real data to infer the consequences; the other is to look to the data and from there infer a model.

These two complementary approaches have different applications, the former is used in physics and economics (where the models are assumed) while the latter is used in traditional time series analysis. The approach followed in Econophysics is typically to look first at the data and then to get the best model that describes it. This empirical overview of the data tends to be a first approximation to an important and complex subject.

One of the implicit goals of Econophysics, interesting and at the same time quite challenging, is to merge these two approaches and make a bridge between Econophysics and Economics: data are only useful within an interpretative framework. As with other complex systems, economics, and especially finance has lots of data available. To analyse these we have to summarise and reduce them to manage the complexity. This means the we have to choose among a myriad of paths. The use of Econophysics tools permits exploration of new areas and quantitative testing of relevant hypotheses.

Also interesting is the domain of applicability; Econophysics is mainly used where we have a huge flow of data. The empirical approach allows the study of data where no further conditions are assumed about stationarity or other features. For further arguments see e.g. Bouchaud and Potters [2001]. We have also some researchers, e.g. [Ball, 2006], warning against the arbitrary use of scaling relations, without justifying their value: the proverbial "*if you have a hammer everything looks like a nail*".

1.1.2. Branches

Econophysics is a broad scientific area and its current popularity can be attested by the daily number of related papers published in arXiv (`http://arxiv.org/`). Although all the subareas (branches) share the same basic principles it is possible to distinguish several distinct areas of study in Econophysics. One of the basic assumptions shared is the belief

in scaling relations implicit in the results (see [Bouchaud and Potters, 2001, Mantegna and Stanley, 2000]).

The main focus of this work is (financial) time series analysis, although we reference other branches for completeness. It must be said that, as with any classification of ongoing scientific research, this is probably incomplete and inevitable subject to change, yet the division presented here seems to best capture the different subareas at present.

Time series analysis (financial data) of returns

It is traditional to consider equally spaced data and those with smaller time periods (high frequency data). In this work we will consider the interval between data points as one day. A unique feature is restricting consideration to trade days, we only consider trade days and thus one trading day follows another trading day.

The frequency of data must be taken into account because as we shall see measures for different scales yield different results. This granularity effect will be also one of the main topics of this work.

As in the study of time series it should be noted that common statistics are based on tests which assume independence between samples, this clearly excludes them here since all values are generally related in time.

In Econophysics we do not study the original financial series. We focus instead on a transformed quantity, (as in the financial literature), namely the *returns*. Returns series are used to analyse and model financial markets. For an asset with an associated time series x we have the following definition.

Definition 1.1.1. Let x_i be the value of a time series x at time i. *Returns* are defined as

$$\eta_i = \log \frac{x_i}{x_{i-1}}, \tag{1.1}$$

where η_i is the return at time step i.

An asset is any good to which we can give a price. Since x_i are asset values they are positive and thus the returnsSometimes these called log-returns to distinguish them from the same quantity without the logarithm being applied. In what follows in this work, returns means always the log returns. are always well defined.

The use of of the ratio between two consecutive values makes the quantity of study dimensionless, and the use of logarithms gives a different sign to gains and losses.

Returns can (and will in this work) be used to study the inherent features of any given time series: time scale (characteristic time) for crash recovery; distribution of returns (scale dependence); measures of intrinsic risk and uncertainty. Returns can be used also to compare different series, search for patterns both exclusive to some series only or for the whole group of series. We can use them to give us a new perception of the involved correlations.

The complexity of financial time series can not be reduced to single numbers, and every technique allow us to see some part of the picture.

Study of distributions

Present in all econophysics branches is the conviction of scaling arguments, [Mantegna and Stanley, 2002] coming from the study of systems in critical states. In an ironic twist, distributions of income and wealth was a subject studied earlier (see work from Vilfredo Pareto [1897], an economist), who found that large values in these distributions follow *universal scaling behaviour* independent of the countries considered.

The empirical study of those distributions led also to the analysis of distributions of economic shocks, growth rate variations, firm and city sizes. In all these measures scaling laws appeared, thus giving confidence that the same type of analysis could be applied as those used to characterise complex systems near critical behaviour.

Networks

Networks have been studied at an early stage in the history of mathematics; the famous problem of Kronisberg bridges e.g. was solved by Euler in the 17th century. More recently we had the work of Erdös and Rényi [1959]. Yet only recently, with the enormous growth in computer power some of those problems have been looked at again from a different viewpoint. Examples of these types of networks include small worlds and scale free networks, (see Newman [2003]).

Agent based systems

The analogy between cellular automata, with simple laws that rule the interaction between neighbours, and economical systems, with all agents individually seeking profit maximisation, has led to the use of agent based systems. The agents are autonomous entities that live and interact among them usually by neighbourhood relations.

The set of ingredients for modelling markets are:

1. a large number of independent agents participate in a market;
2. each agent has alternatives in making decisions;
3. the aggregate activity results in a market price, which is known to all;
4. agents use public price history to make their decisions.

For a recent review of the use of agent based systems in econophysics see Ausloos [2006].

Another type of agent based systems is that related to Game theory where several cases are well known like the prisoner's dilemma and the Minority game.

1.1.3. Historical perspective

A first theory of stock-market fluctuations was proposed by Bachelier [1900], five years before Einstein's famous paper on Brownian Motion [Einstein, 1905], in which Einstein derived the partial differential heat/diffusion equation governing Brownian motion and estimated for the size of the molecules.

In 1900, Bachelier studied the Paris Stock Exchange in his PhD thesis introducing Brownian motion to describe the evolution of the financial assets. Bachelier gave the distribution function for what is now known as the Wiener stochastic process – the stochastic process that underlies Brownian Motion – linking it mathematically with the diffusion equation. The probabilist Feller [Feller, 1968] had originally called it the Bachelier-Wiener process. It is accepted that Einstein in 1905 was not aware of Bachelier's work. This work states that the second order moments of the increments of a heat/diffusion process scale as

$$E\left\{(X(t_2) - X(t_1))^2\right\} \propto |t_2 - t_1|\,. \tag{1.2}$$

Where X is stochastic process under study.

His thesis report was signed by Henri Poincaré, observing that "*M. Bachelier has evidenced an original and precise mind [but] the subject is somewhat remote from those our other candidates are in the habit of treating.*" Nevertheless, the thesis anticipated many of the mathematical discoveries made later by Wiener and Markov, and outlined the importance of such ideas in today's financial markets, stating that "*it is evident that the present theory solves the majority of problems in the study of speculation by the calculus of probability.*"

Seventy three years before Black and Scholes wrote their famous paper [Black and Scholes, 1973], Bachelier derived the price of an option where the share price movement is modelled by a Wiener process and derived the price of what is now called a *barrier option* (namely, the option which depends on whether the share price crosses a barrier). Black, Scholes and Morton, following the ideas of Osborne [1959, 1977] and Samuelson [1973], modelled the share price as a stochastic process known as a geometric Brownian motion (with drift). Fisher Black, Myron S. Scholes, and Robert C. Merton extended the theory into a methodology for virtually zero risk option and derivative pricing, and established the isomorphism between the standard deviation of the fluctuations in price of a financial instrument, and investment risk.

A modern version of Bachelier's theory is routinely used in financial literature. This theory predicts a Gaussian probability distribution for stock-price fluctuations.

The random walk hypothesis, with independent and identically distributed increments, is the basis of the Efficient Market Hypothesis [Fama, 1970]. It states, in simple words, that: the price variation is random as a result of the activity of the traders who attempt to make profit (arbitrage opportunities); the application of their strategies induces a feedback dynamic in the market randomising the stock-price.

In the 50's, Hurst, while analysing hydrological flows, proposed a single exponent to characterise time variation. As can be seen in Chapter 2 this approach is a generalisation of Brownian motion later called fractional Brownian motion [Mandelbrot and Van Ness, 1968], and is characterised by a single exponent, later called Hurst exponent.

Fat tails

In the 60's, Mandelbrot [1963], pointed out that the distributions of price differences are not Gaussian due to the so called *fat-tails*. Mandelbrot formulated a theory of price fluctuations in speculative markets based on the probability distributions discovered by the French mathematician Paul Lévy. As pointed out by Mandelbrot, the so-called log-normal distribution is of interest in finance, since wealth in a multi-agent system evolves into a log-normal distribution.

The Gaussian/Normal distribution is a special case of the more general Lévy distributions, and is often used as an approximation to log-normal distributions for mathematical expediency. In contrast, these distributions display power-law decay in the tails and this is related to the fractal nature of financial data, where uni-fractal processes, such as fractional Brownian motion [Bouchaud and Potters, 2001, Mantegna and Stanley, 2000] have been discussed in the literature for some years and, more recently, simple multi-fractal processes have been considered for financial data from various sources [Lux, 2004].

In problems with similar data features [Peng et al., 1994] while studying DNA patterns and their characteristics, introduced Detrended Fluctuation Analysis (DFA) as a method to estimate the Hurst exponent.

Entropy

The early notion of entropy as a measure of disorder comes from the work of Claussius in the 19th century, where entropy provides a way to state the second law of Thermodynamics (as well as a definition of temperature). Boltzman extended the idea further giving it a central role in statistical physics. Here, entropy is a measure of system multiplicity and can be visualised in terms of disorder.

Shannon [1948] gave a new meaning to entropy in the context of Information Theory, relating entropy with the absence/presence of information in a given message.

Entropy, one of the early ideas behind thermodynamics that later led the way to the emergence of statistical physics, has been shown to be pervasive and, perhaps surprisingly, well suited to crossing disciplinary boundaries (to pure mathematics), giving an easier interpretation to the previously defined concept of topological entropy. The influence of thermodynamics was such that it lent its name to the thermodynamical formalism by Bowen and Ruelle.

The theoretical concept proves to be rich and active as demonstrated in the late 80's when Tsallis [1988b] introduced the concept of non-extensive entropy, generalising further

the "traditional" concept of entropy.

1.2. Overview and structure of the thesis

The main focus of this thesis is placed on entropy measures for the following reasons:

1. they allow us to predict how the market will evolve;
2. they add to the portfolio of techniques used to study time series;
3. they allow us to characterise the specific features of each market;
4. they are measures of how markets perceive risk.

Each technique captures different *nuances* of the signal evolution. The use of different tools at the same times allow us to have more confidence in the obtained results, avoiding the several pitfalls of using a single technique.

The work carries several types of analyses/tools, entropy, time and scale dependency of the Hurst exponent, as well as correlation matrix analysis between different markets.

All analyses were performed on daily data from worldwide indices. The daily indices were used as benchmarks for the different markets studied. Only market indices were used but it should be noted that the same techniques apply to other type of financial assets data.

This thesis is divided into three parts. The methods, used in this work, are established in Chapters 2 and 3, which are applied in Chapters 4, 5 and 6. The conclusions are draw in Chapter 7.

Chapter 2 details most of the mathematical tools applied in the work. Here, we present wavelets, fractals, multifractals, fractional Brownian motion, statistical stable laws, entropy and time dependent correlation matrices.

In Chapter 3 the computational methodology used in this work is presented. This unusual layout comes from the author's personal opinion that computational methods deserve a special treatment on same level as the theoretical methods. A strong stance on Free Software is explained there. The use of free software in scientific computing is discussed, and examples of that usage are presented. Free software encourages cooperation in a way similar to scientific method. Several contributions of the author to major free software projects are stated in the last Section.

PSI-20 (Portuguese Stock Index - 20) is studied in Chapter 4 where an empirical study of the property of its return is made. The outcome of those results is compared with some simple models and their expected values.

A new method is proposed in Chapter 5 to characterise the different markets. This method extends the DFA over time and scale domains, instead of a single global index. To characterise the time series, we use Hurst exponents coming from DFA as a local

measure of fractional Brownian motion (fBm) behaviour and thus as a local measure of uncertainty both in time and scale. The variation of this across time and scale gives an indication of market reaction both to internal fluctuation as well as to external influences. This behaviour allows a better understanding of each market feature and, when used in comparison with other markets, allows a classification scheme for different markets.

Entropy measures are used in comparison with the correlation matrix as a way to show differences and similarities between markets in Chapter 6. The entropy measures used here are entropy as defined by the method developed in Chapter 5.

Finally in Chapter 7 we have the conclusions taken from this work.

In order to help the reading of this work some subjects interesting but not fundamental to the understanding of this thesis have been placed in Appendices.

2. Mathematical Tools

"*The scientist does not study nature because it is useful to do so. He studies it because he takes pleasure in it, and he takes pleasure in it because it is beautiful. If nature were not beautiful it would not be worth knowing, and life would not be worth living. I am not speaking, of course, of the beauty which strikes the senses, of the beauty of qualities and appearances. I am far from despising this, but it has nothing to do with science. What I mean is that more intimate beauty which comes from the harmonious order of its parts, and which a pure intelligence can grasp.*" - Henri Poincaré

2.1. Time series tools overview

In this Chapter we present and define, with mathematical rigour, most of the tools used in this work. Since we are interested in the study of financial time series we start with stochastic processes, firstly developed in the scope of statistical physics. Basic mathematical definitions include that for the Fourier transform and for the fractal dimension. The minimal assumption here is a knowledge of measure theory. Advancing in complexity, we introduce: multifractals; wavelets; stable laws (Lévy distributions); fractional Brownian motion; entropy and time dependent correlation matrices.

The purpose of the Chapter is to explain the relation between these different tools as well as to illustrate each mathematical tool with available free software. Example code is referenced and appears in either Appendix C or Appendix D depending if it was made during this work or if it was used from external sources.

2.2. Stochastic processes

The theory of Stochastic Processes is generally defined as the "dynamic" part of probability theory, where we study a collection of random variables, (called a *stochastic process*), from the point of view of their interdependence and limiting behaviour. We can apply a stochastic process whenever we have a process developing in time and controlled by probabilistic laws [Parzen, 1999]. In this context, it is interesting to note that many elements of the theory of stochastic processes, were first developed in connection with the study of fluctuation and noise in physical systems and financial data (Bachelier [1900], Einstein [1905]).

All the stochastic processes that will be considered in this work are time series. The notation used in this section is well known and is essentially the same as that used in Papoulis [1985].

2.2.1. Random variable measures

The expression random variable is a misnomer and an historical accident, as a random variable is not a variable, but rather a function that maps events to numbers.

Definition 2.2.1. Let $\mathcal{A}$ be a σ-algebra and Ω the space of events relative to the experiment. A function $X : (\Omega, \mathcal{A}) \to \mathbb{R}$ is a *random variable* if for every subset $A_r = \{\omega : X(\omega) \leq r\}$, $r \in \mathbb{R}$, the condition $A_r \in \mathcal{A}$ is satisfied. A random variable X is said to be *discrete* if the set $\{X(\omega) : \omega \in \Omega\}$ (i.e. the range of X) is countable. A random variable Y is said to be *continuous* if it has a cumulative distribution function which is absolutely continuous.

One useful definition is the expected value of a random variable, in a sense what we should expect if we have a repeated process. The expected value gives us the average of repeated measurements.

Definition 2.2.2. Consider a discrete random variable X. The *expected value*, or *expectation*, of X, denoted $E\{X\}$, is the weighted average of all possible values of X by their corresponding probabilities, i.e. $E\{X\} = \sum_x x f_X(x)$ ($f_X(x)$ is the probability function of X). If X is a continuous random variable, then $E\{X\} = \int_x x f_X(x)\mathrm{d}x$ ($f_X(x)$ is the probability density function of X).

Note that if the corresponding sum or integral does not converge, the expectation does not exist. One example of this situation is the Cauchy random variable.

Definition 2.2.3. Let X and Y be two random variables, then the *covariance* of X and Y is

$$C_{X,Y} = E\{(X - E\{X\})(Y - E\{Y\})\}. \tag{2.1}$$

If $X = Y$ then we get the *variance* of X:

$$Var_X = C_{X,X}. \tag{2.2}$$

The *standard deviation* of the random variable X is the square root of variance

$$\sigma_X = \sqrt{Var_X}. \tag{2.3}$$

Definition 2.2.4. The *correlation coefficient* of two random variables X and Y is

$$r_{X,Y} = \frac{C_{X,Y}}{\sigma_X \sigma_Y}. \tag{2.4}$$

2.2.2. Stochastic processes

Definition 2.2.5. Let $(\Omega, \mathcal{F}, \mathbf{P})$ be a probability space. A *stochastic process* is a collection $\{X(t) \mid t \in T\}$ of random variables $X(t)$ defined on $(\Omega, \mathcal{F}, \mathbf{P})$, where T is a set, called the index set of the process. T is usually (but not always) a subset of $\mathbb{R}$. One can also think of a stochastic process as a function $X = (X(t, \omega))$ in two variables: $t \in T$ and $\omega \in \Omega$, such that for each t, $X_t(\omega) \colon = X(t, \omega)$ is a random variable on $(\Omega, \mathcal{F}, \mathbf{P})$. Given any t, the possible values of $X(t)$ are called the states of the process at t. The set of all states (for all t) of a stochastic process is called its state space. If T is discrete, then the stochastic process is a discrete-time process. If T is an interval of $\mathbb{R}$, then $\{X(t) \mid t \in T\}$ is a continuous-time process. If T can be linearly ordered, then t is also known as the time.

Let $X(t)$ and $Y(t)$ be stochastic processes, with $t \in T$ and T being the index set.

Definition 2.2.6. The *mean* $\eta(t)$ of $X(t)$ is the expected value of the random variable $X(t)$

$$\eta_X(t) = E\{X(t)\}. \tag{2.5}$$

Definition 2.2.7. The *cross-correlation* of two processes $X(t)$ and $Y(t)$ is

$$R_{XY}(t_1, t_2) = E\{X(t_1)Y(t_2)\}. \tag{2.6}$$

Definition 2.2.8. The *autocorrelation* $R(t_1, t_2)$ of $X(t)$ is the expect value of the product $X(t_1)X(t_2)$

$$R(t_1, t_2) = E\{X(t_1)X(t_2)\}. \tag{2.7}$$

Definition 2.2.9. The *cross-covariance* of two processes $X(t)$ and $Y(t)$ is

$$C_{XY}(t_1, t_2) = E\{X(t_1)Y(t_2)\} - \eta_X(t_1)\eta_Y(t_2). \tag{2.8}$$

Definition 2.2.10. The *autocovariance* $C(t_1, t_2)$ of $X(t)$ is the covariance of the random variables $X(t_1)$ and $X(t_2)$

$$C(t_1, t_2) = R(t_1, t_2) - \eta(t_1)\eta(t_2). \tag{2.9}$$

Definition 2.2.11. The ratio

$$r(t_1, t_2) = \frac{C(t_1, t_2)}{\sqrt{C(t_1, t_1)C(t_2, t_2)}} \tag{2.10}$$

is the *correlation coefficient* of the process $X(t)$.

2.2.3. Computational implementation

Most of the data analysis made in this work was done through a Python module described in Appendix C. The sub-module *tools* provides the basic statistical tools for time series

analysis.

2.3. Fourier transform

The use of Fourier transforms in this work is indirect. Rather than using them to analyse financial time series by decomposition of the series into several components with different amplitudes and frequencies, we rely on analytical properties to develop or explain methods and tools presented later.

The useful properties of Fourier transforms referred above are:

- Fourier transforms are linear operators and, with proper normalisation, are unitary as well (a property known as Parseval's theorem);
- the transforms are invertible, and in fact the inverse transform has almost the same form as the forward transform;
- the sinusoidal basis functions are eigenfunctions of differentiation, which means that this representation transforms linear differential equations with constant coefficients into ordinary algebraic ones;
- by the convolution theorem, Fourier transforms turn the complicated convolution operation into simple multiplication, which means that they provide an efficient way to compute convolution-based operations such as polynomial multiplication.

Definition 2.3.1. We define the $L^p(\mathbb{R})$ space as the set of real functions such that

$$\int_{\mathbb{R}} |f(t)|^p \mathrm{d}t < +\infty. \tag{2.11}$$

Definition 2.3.2. The Fourier transform of function $f(x)$ is defined as ($z \in \mathbb{C}$)

$$\hat{f}(z) = \int_{\mathbb{R}} e^{izx} f(x) \mathrm{d}x. \tag{2.12}$$

The Fourier transform exists if f is Lebesgue integrable on the whole real axis.

Definition 2.3.3. The inverse Fourier transform is defined as

$$f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} e^{-izx} \hat{f}(z) \mathrm{d}z. \tag{2.13}$$

If f is Lebesgue integrable and can be divided into a finite number of continuous, monotone functions and at every point both one-sided limits exist, the Fourier transform can be inverted.

If $f(x)$ is a probability distribution we call its Fourier transform the *characteristic function* of $f(x)$.

One other definition used in several scientific areas is the dot product between functions.

Definition 2.3.4. Let f and g be two real functions, then we define the dot product as

$$< f, g >= \int f(t)\overline{g}(t)\mathrm{d}t. \tag{2.14}$$

Notice that this equality defines equivalence classes.

Definition 2.3.5. It is usual to define the *energy of a signal* as

$$\int_{\mathbb{R}} |f(t)|^2 \mathrm{d}t. \tag{2.15}$$

2.3.1. Computational implementation

The discrete version of the Fourier transform can be evaluated quickly on computers using fast Fourier transform (FFT) algorithms [Cooley and Tukey, 1965]. The computational implementation used is FFTW (see Appendix D for further details).

2.4. Wavelets

2.4.1. Introduction and motivation

Wavelets are a topic, developed essentially in the last twenty years (see Kaiser [1994], Burrus et al. [1998], Ueda and Loadha [1995], Graps [1995], Valens [1999]). Instead of focusing solely on time dependency, this is a method that highlights also the scale dependency (scale is the inverse of frequency, as seen also in Fourier transforms).

For examples of the use of wavelets in econophysics, applied to financial time series, see Vuorenmaa [2005], Bartolozzi et al. [2006] and Sharkasi et al. [2006a].

In theoretical terms wavelets constitute a basis for a functional space, using as a seed a single function (that, as we will see, has special properties). On the practical side they have shown to be numerically suitable due to several algorithms whose complexity time is on the same order as FFT.

Traditional signal analysis, (Fourier transform based), does not indicate when an "event" occurs (e.g. trends or abrupt changes). There is a lack of temporal resolution. The "time/frequency" aspect of wavelets permits us to gain information about frequency composition of the signal at a particular time. Fourier analysis does not work well on discontinuous, "bursty" data, while wavelets work well with discontinuous data and perform well when applied to non-stationary data.

Comparing Fourier and wavelets we have:

Fourier	Wavelets
• looses time (location) coordinate completely	• Localised time-frequency analysis
• Analyses the whole signal	• Short signal pieces also have significance
• Short pieces loose "frequency" meaning	• Scale = Frequency band

It should be said that wavelets are not the only technique using a time and scale approach. Some other methods [Carvalho, 2000] include short-time Fourier windows transform (Windowed Fourier transform), Gabor transform and Wiegner distribution for time-frequency analysis.

Due to these strengths, wavelets have been applied in different fields and to different applications. In a number of cases the method has assumed the role formerly taken by Fourier transforms. Application of wavelets analysis include some discussed here: multifractals and Hölder exponents, stable laws, fractional Brownian motion [Doukhan et al., 2003].

The duality concept referred to in this approach is related to the fundamental nature of the wavelets: its simultaneous analysis of frequency and time. This duality is also related to the synthesis and analysis involved (i.e. construction and deconstruction of wavelets).

2.4.2. Continuous wavelet transform

Wavelets can be used to analyse functions in $L^2(\mathbb{R})$ (the space of Lebesgue absolutely square integrable functions defined on the real numbers to the complex numbers) in much the same way the complex exponentials are used in the Fourier transform, but wavelets offer the advantage of not only describing the frequency content of a function, but also providing information on the time localisation of that frequency content. We are not restricted to a single function as the basis as it happens for Fourier Transforms.

We introduce the wavelets through the continuous wavelet transforms and from there develop the properties required for the wavelets. We start with a real function ψ, called the mother wavelet, and from there we build a family of functions

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-\tau}{s}), \tag{2.16}$$

where s is the scale and τ is the translation. The new functions are then rescaled and translated versions of the original. The factor $s^{-1/2}$ is for energy normalisation across different scales, so that $\|\psi_{s,\tau}\| = \|\psi\| = 1$.

Definition 2.4.1. The continuous wavelet transform of a real function f, over the wavelet family $\psi_{s,\tau}$, is given by

$$\gamma(s,\tau) =< f, \psi_{s,\tau} > . \tag{2.17}$$

Definition 2.4.2. The inverse continuous wavelet transform, the inverse of last definition is given by

$$f(t) = \int \int \gamma(s,\tau)\psi_{s,\tau}(t)\,\mathrm{d}\tau\mathrm{d}s. \tag{2.18}$$

Such as it were given this definitions are broad and not generally useful. We present below the properties that give a meaning to the above definitions and at the same time define some of the wavelets unique properties.

To guarantee the existence of the inverse we must require the *admissibility condition*

$$\int \frac{|\widehat{\psi}(\omega)|^2}{|\omega|}\mathrm{d}\omega < +\infty. \tag{2.19}$$

This condition allows to reconstruct the signal without loss of information. From this condition we get immediately that $\widehat{\psi}(0) = 0$, i.e. $\int \psi(t) = 0$. We see then that $\psi(t)$ behaves like a wave.

The other requirement comes from the time location, to get this behaviour we demand the function to decay quickly in the scale dependency. This is called the *regularity condition.*

Performing a Taylor expansion, of degree n in s, around $\tau = 0$ for γ we have

$$\gamma(s,0) = \frac{1}{\sqrt{s}}\left(\sum_{p=0}^{n} f^{(p)}(0)\int \frac{t^p}{p!}\psi(\frac{t}{s}))\mathrm{d}t + O(s^{n+1})\right). \tag{2.20}$$

Let M_p be the moments of ψ, i.e.

$$M_p = \int t^p\psi(t)\mathrm{d}t. \tag{2.21}$$

We have $M_0 = 0$, due to the admissibility condition. Replacing the expression for moments in the Taylor expansion we get

$$\gamma(s,0) = \frac{1}{\sqrt{s}}\left(\sum_{p=0}^{n} M_p\frac{f^{(p)}(0)}{p!}s^{p+1} + O(s^{n+2})\right) \tag{2.22}$$

If we use further vanishing moments, $M_1 = \cdots = M_n = 0$, then $\gamma(s,\tau)$ will decay as fast as s^{n+2} for a smooth signal $f(t)$.

We can resume this saying that the admissibility condition gives us the *wave* and the regularity condition brings the *let*, thus having *wavelet.*

2.4.3. Discrete wavelets

The continuous wavelet transform was interesting since it allowed to expand on the properties of wavelets, but it presents some inconveniences: redundancy of $\psi_{s,\tau}(t)$, s and τ are continuous coefficients; it would be interesting like in the Fourier transform to get a

more manageable number of base functions; and it is difficulty in practice to get a closed formula for $f(t)$.

To overcome this we discretise the scale and the translations, using minimum s_0 and τ_0. We define now the wavelets as a countable family of functions, using as a seed the mother wavelet as above.

Definition 2.4.3. A (more properly, an *orthonormal dyadic*) *wavelet* is a function $\psi(t) \in L^2(\mathbb{R})$ such that the family of functions $\psi_{jk} \equiv 2^{j/2}\psi(2^j t - k)$, where $j, k \in \mathbb{Z}$, is an orthonormal basis in the Hilbert space $L^2(\mathbb{R})$.

The scaling factor of $2^{j/2}$ ensures that $||\psi_{j,k}|| = ||\psi|| = 1$. These type of wavelets, (the most popular), are known as dyadic wavelets because the scaling factor is a power of 2.

The wavelet series decomposition gives us a series of numbers that corresponds to the coefficient of each wavelet.

Definition 2.4.4. The continuous wavelet transform of a real function f, over the wavelet family $\psi_{j,k}$, is given by

$$\gamma(j,k) =< f, \psi_{j,k} > . \tag{2.23}$$

The condition for reconstruction is that the energy of the wavelet must lie between two positive bounds

$$A||f||^2 \leq \sum_{j,k} | < f, \psi_{j,k} > |^2 \leq B||f||^2, \tag{2.24}$$

where $A > 0$ and $B < +\infty$.

If 2.24 is satisfied we call$\{\psi_{j,k}(t) : j, k \in \mathbb{Z}\}$ a frame with bounds A and B. When $A = B$ the frame is "tight" and the discrete wavelets behave like an orthonormal basis. Again this is similar to what we get with the Discrete Fourier Transform (DFT).

By properly choosing the mother wavelet, $< \psi_{j,k}, \psi_{m,n} >= \delta_{j,k}\delta_{m,n}$, where $\delta_{j,k}$ is the delta of Dirac.

The reconstruction formula is now a double sum

$$f(t) = \sum_{j,k} \gamma(j,k)\psi_{j,k}(t). \tag{2.25}$$

2.4.3.1. Filter band coding

We have progressed from the continuous wavelets transforms, but we still need an infinite number of scalings and translations to calculate the wavelet transform. Is it possible to reduce the number of wavelets to analyse a signal and still have a useful result?

At this point Fourier transforms give an interesting interpretation to wavelets, we could look at $\widehat{\psi}(0) = 0$ as a band-pass like spectrum.

Fourier transform property for a scaling is

$$\widehat{f}(at) = \frac{1}{|a|}\widehat{f}(\frac{\omega}{a}). \tag{2.26}$$

The time compression of the wavelet by a factor of 2 will stretch the frequency spectrum of the wavelet by a factor of 2 and also shift all frequency components the same factor. We can cover the finite spectrum of our signal with the spectra of dilated wavelets in the same way we covered our signal in continuous time with translated wavelets.

If one variable can be seen as a band-pass filter then a series of dilated wavelets can be seen as a band pass filter bank. If the signal has a finite energy then it will have a finite cover, this argument covers the high frequency limit, but for small frequencies we need to stop somewhere. The threshold is then is a low-pass spectrum filter and it belongs to the so called *scaling function*.

The *scaling function* is then

$$\varphi(t) = \sum_{j,k} \gamma(j,k)\psi_{j,k}(t). \tag{2.27}$$

If we analyse the signals as a combination of scaling functions and wavelets, the scaling function takes care of the spectrum covered by the wavelet up to scale j, while the remaining part is expressed by a finite number of coefficient of the wavelets.

The low-pass spectrum of the scaling function allows us to state an admissibility for scaling functions

$$\int \varphi(t)\mathrm{d}t = 1. \tag{2.28}$$

The sub-band coding we have been detailing is the basis for Mallat's algorithm [Mallat, 1989a].

If we regard the wavelet transform as a filter bank, then we can consider the wavelet transformation of signal as passing the signal through this filter bank. The output of the different filter stages are the wavelet and scaling function transform coefficients. The iterated filter bank has two passes: high pass, contains the finest details of interest (rapid or short-term fluctuations); and a low pass; still contains some details and fine details may further be extracted in another iteration.

The advantage of this scheme is that we have only two filters, the main disadvantage is that the signal spectrum coverage is limited.

The wavelet transform is the same as a sub-band coding scheme using a constant-Q filter bank. This kind of analysis is known as multi-resolution analysis [Mallat, 1989b].

2.4.4. Multi-Resolution analysis

Wavelets can be constructed from a multi-resolution analysis, define below. Here the ideas presented before, as a motivation for wavelets, are given a new formulation.

Definition 2.4.5. An orthonormal *Multi-Resolution Analysis* (MRA) $\{(V_j), \phi\}$ is made by a scaling function $\phi \in L^2(\mathbb{R})$ and a sequence (V_j), $j \in \mathbb{Z}$ of closed subspaces of $L^2(\mathbb{R})$

such that:

$$\cdots \subset V_{-1} \subset V_0 \subset V_1 \subset \cdots \tag{2.29}$$

$$\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R}) \tag{2.30}$$

$$\bigcap_{j\in\mathbb{Z}} V_j = \{0\} \tag{2.31}$$

$$v(t) \in V_j \Leftrightarrow v(2t) \in V_{j+1} \tag{2.32}$$

$$\{\phi(t-k) : k \in \mathbb{Z}\} \text{ is an orthonormed base of } V_0 \tag{2.33}$$

Observe that 2.29 tells that (V_j), $j \in \mathbb{Z}$ is a nested sequence of spaces, while 2.30 states that we have a full closure of the functional space $L^2(\mathbb{R})$. The only common element between all the sequence is the null function, according to 2.31. The scaling relation in 2.32 show that the functions from V_{j+1} are scaled versions of V_j. Finally in 2.33 tells that $\phi(t)$ and its translations form an orthonormed base of the space V_0.

Using 2.32 and 2.33 we have that

$$\phi_{j,k}(x) = \sqrt{2^j}\phi(2^j x - k)\, j, k \in \mathbb{Z} \tag{2.34}$$

is a base of V_j. We can see that scaling and translations of $\phi(t)$ form an orthonormal basis for every space of the collection (V_j).

If we define P_j as the orthogonal projection of $f(t) \in L^2(\mathbb{R})$ into subspace V_j

$$P_j(f(t)) = \sum_{k\in\mathbb{Z}} < f, \phi_{j,k} > \phi_{j,k}(t), \tag{2.35}$$

we have thus a sequence of functions that approximates $f(t)$ and such that:

$$\lim_{j\to-\infty} P_j(f(t)) = 0 \tag{2.36}$$

$$\lim_{j\to+\infty} P_j(f(t)) = f(t) \tag{2.37}$$

The projection coefficients, $s_{j,k} =< f, \phi_{j,k} >$, are called *scaling coefficients*.

2.4.4.1. Details

Instead of looking into the sequence (V_j), we can study another sequence (W_j) such that W_j is the orthogonal complement of V_j with relation to V_{j+1}

$$V_{j+1} = V_j \bigoplus W_j, \tag{2.38}$$

$$V_j \perp W_j. \tag{2.39}$$

The sequences (W_j) are the differences between each approximation level, they represent the details (or "errors") for each approximation. These detail spaces are orthogonal among themselves $W_j \perp W_k$, $j \neq k$. On the other hand the recursive definition allow us to have

$$V_j = V_i \bigoplus \left(\bigoplus_{k=i}^{j} W_k \right). \tag{2.40}$$

The relation 2.40 is important, even from piratical terms, we express a time series as the sum of the a smooth part (V_i) plus the several details taken from W_k.

From 2.30 and 2.29 we have

$$L^2(\mathbb{R}) = \bigoplus_{k \in \mathbb{Z}} W_k. \tag{2.41}$$

This property says that the reunion of the sequence (W_j) spans $L^2(\mathbb{R})$. Moreover property 2.32 still applies in subspaces W_j, since they are contained inside V_{j+1} where this property is valid.

The most important result from MRA is the that given a scaling function $\phi(t) \in L^2(\mathbb{R})$ there is a function $\psi(t) \in L^2(t)$, with the same regularity and such that integer translations generates a base of W_0. This function $\psi(t)$ is called *wavelet*.

Using 2.32 and 2.33 we get a family of functions

$$\psi_{j,k}(x) = \sqrt{2^j}\psi(2^j - k) \tag{2.42}$$

for each j which are an orthonormal base of W_j, and for all j and k an orthonormal base of $L^2(\mathbb{R})$. This is the objective of MRA to obtain an orthonormal basis of $L^2(\mathbb{R})$ using a single function, the scaling function ϕ.

2.4.4.2. Discrete wavelet transform

In practice we are interested in the different detail levels. Using equations 2.34, 2.35 and 2.42, we can express

$$f(t) = s_{j_0}(t) + \sum_{j=j_0} f_j(t) \tag{2.43}$$

where

$$s_{j_0}(t) = \sum_k c_{j_0}(k)\phi_{j_0,k}(t), \tag{2.44}$$

and

$$f_j(t) = \sum_k d_j(k)\psi_{j,k}(t). \tag{2.45}$$

If the wavelet system is orthogonal, we obtain:

$$c_{j_0}(k) =< f(t), \phi_{j_0,k}(t) >, \tag{2.46}$$

and

$$d_j(k) =< f(t), \psi_{j,k}(t) > . \tag{2.47}$$

In equation 2.43 we have decomposed the original function into a smooth part and a sum of details.

2.4.5. Examples

There are different families of wavelets, one of the distinctive behaviour inside each family is the degree of the vanishing moments. Depending on the properties desired so we can choose the scaling function. Continuity is not a requirement and some wavelets families are (piecewise) continuous functions.

Examples of wavelets families include Haar, Daubechies and general order B-spline, (see in [Ueda and Loadha, 1995, Burrus et al., 1998]).

The software used in this work related with wavelets is described in Appendix D.

2.5. Fractal dimension

One other tool used in the study of financial time series is the fractal dimension. Fractals, so named by Mandelbrot [1977, 1982], were known long before the term was coined. Initially looked upon with suspicion and considered mathematical toys, they have since been found to occur everywhere, (to the point where the exceptions are the non-fractal objects).

The following definitions can be found on Falconer [1985].

Definition 2.5.1. Let A be a compact subset of the Euclidean space $\mathbb{R}^n$. For $\epsilon > 0$, consider the subdivision of $\mathbb{R}^n$ into boxes or cubes of sides of length ϵ: for $(j_1, \ldots, j_n) \in \mathbb{Z}^n$, let

$$R_{j_1,\ldots,j_n} = \{(x_1, \ldots, x_n) : j_i\epsilon \leq x_i < (j_i + 1)\epsilon \,\text{for}\, 1 \leq i \leq n\} . \tag{2.48}$$

A box of this kind is said to be a *box from the ϵ-grid*. Let $N(\epsilon, A)$ be the number of boxes R_j among all the choices of $j \in \mathbb{Z}^n$ such that $A \cap R_j \neq \emptyset$. The Minkowski-Bouligand (or box dimension) of A is

$$\dim_b(A) = -\lim_{\epsilon \to 0^+} \frac{\log(N(\epsilon, A))}{\log(\epsilon)}. \tag{2.49}$$

Definition 2.5.2. The Hausdorff-Besicovitch dimension of an object E in a metric space is given by the formula :

$$dim_H(E) = -\lim_{\delta \to 0^+} \frac{\log N(\delta, E)}{\log \delta} \tag{2.50}$$

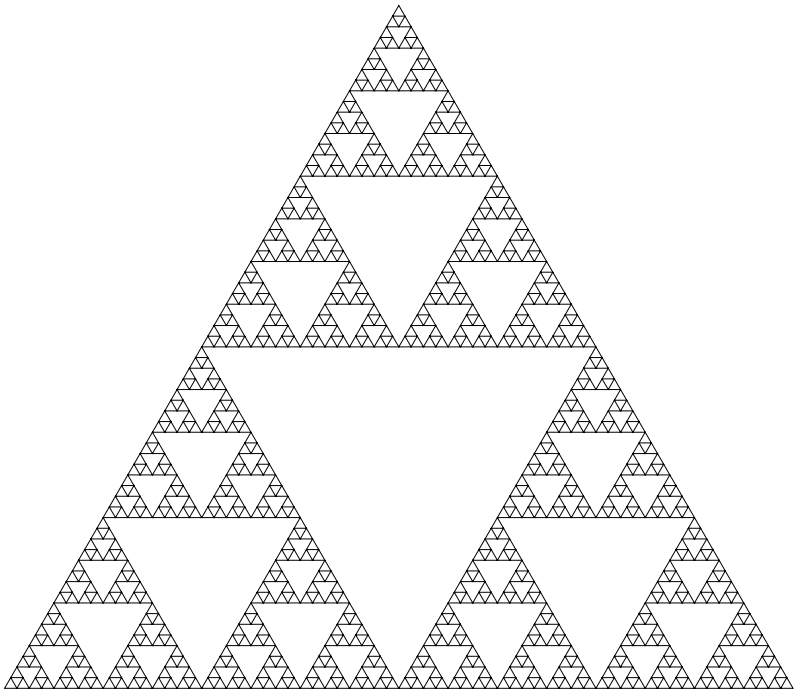


Figure 2.1.: Sierpinski triangle

where

$$N(\delta, E) = \inf\{N \in \mathbb{N} : \exists (x_1, \ldots, x_n) \in (\mathbb{R}^d)^N : E \subset \bigcup_{i=1}^{N} B(x_i, \delta)\}.$$

Remark 2.5.3. For a compact set $A \subset \mathbb{R}^n$,

$$\dim_H(A) \leq \dim_b(A) \leq n. \tag{2.51}$$

This last inequality is important because computationally it is easier to evaluate the box dimension, although usually we are interested in the Hausdorff dimension.

Technically, it should be remarked that these definitions also make sense in a metric space. Since a manifold M can be embedded in some Euclidian space $\mathbb{R}^n$, our definitions apply to compact manifolds since we are talking about local properties characterising the global structure.

Example 2.5.4. The set $A = \{1, 1/2, 1/3, 1/4, ..., 1/n, ...\}$ has $\dim_b(A) = \frac{1}{2}$ and $\dim_H(A) = 0$.

Example 2.5.5. Sierpinski triangle (Figure 2.1) has fractal dimension $\log_2 3 \approx 1.58$.

2.6. Multifractals

In financial data many records do not exhibit a simple monofractal scaling behaviour, which can be accounted for by a single exponent behaviour. The application of fractal analysis to measure theory leads to multifractals [Peitgen et al., 1992]. We recover the fractal dimension when studying the uniform distribution, as a special case.

Multifractal measures (first introduced in [Mandelbrot, 1972]) have been applied to different fields to describe the distribution of energy, matter, turbulent dissipation, stellar matter, minerals and financial returns [Mandelbrot et al., 1997]. Just like in practice where we are only interested in finite measures, so all the measures considered in these sections are finite.

Instead of having a single exponent to characterise the whole set (the fractal dimension) multifractals require a function $f(\alpha)$ to characterise the distributions. The importance of $f(\alpha)$ is reflected in the methods to evaluate $f(\alpha)$ numerically and its properties, presented in the following.

2.6.1. Multifractal measure

Taking the previous example of the Sierpinski triangle, we can construct an example of a multifractal measure. We use the same procedure as that used to obtain the original fractal, where each of the three triangles is assigned a different weight in each iteration. As an example the lower left triangle is set to 0.5, the lower right triangle to 0.2 and the upper triangle to the remainder (0.3). Using this iterative scheme it is easy to see that the mass is conserved, (so after each iteration the triangle's mass will be the same). We can consider the previous example (unifractal) as the degenerate case where each triangle gets 1/3 of the total mass. The question then becomes how to characterise this kind of structure?

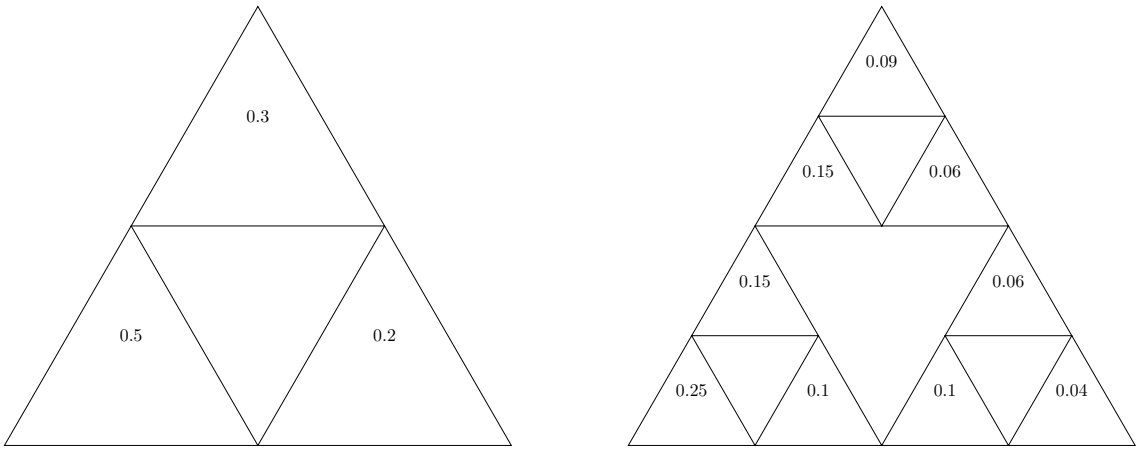
If we take the box dimension this looks like counting coins regardless of their value. In the unifractal case this was correct because each section had the same value, but here coins are not uniform and we count by face value. We should note also that the Hausdorff dimension will give the same (unifractal) result as before, because it does not take into account the different weights.

Any characterisation that describes this type of complexity should consider the weight of each setThe support of a measure is the set of points where the measure is positive.. With this restrictions the candidates are listed below:

Measure Density

$$\frac{\mu(S)}{\epsilon^E}$$

Here E is the Euclidean dimension of the embedding space, μ is the measure and ϵ is the size of the ball containing set S. The problem with this approach is the fact that multifractal measures are singular hence the results will be either 0 or $+\infty$.



a) First iteration b) second iteration

Figure 2.2.: Multifractal Sierpinski measure (first two iterations)

Local Hölder exponent

$$\alpha(x) = \lim_{\epsilon \to 0^+} \frac{\log \mu(B_x(\epsilon))}{\log \epsilon} \tag{2.52}$$

in most cases we can not take the limit, as it does not exists (x is in the support of the measure, to avoid evaluating $\log(0)$).

"Coarse-grained" Hölder exponent

$$\alpha = \frac{\log \mu(\text{box})}{\log \epsilon} \tag{2.53}$$

This apply to any finite set, and $0 \leq \alpha_{min} \leq \alpha \leq \alpha_{max} \leq \infty$.

All the previous candidates measure a local quantity and we would like to have a global quantity to characterise the measure. The choice goes to the "Coarse-grained" Hölder exponent, where we define a structure that rates the importance of exponent α in the measure.

Definition 2.6.1. Let $N_\epsilon(\alpha)$ be the number of boxes with size ϵ with Hölder exponent equal to α, then we define

$$f(\alpha) = -\lim_{\epsilon \to 0^+} \frac{\log N_\epsilon(\alpha)}{\log \epsilon}. \tag{2.54}$$

We can understand $f(\alpha)$ as the fractal dimension of the subset of boxes of size ϵ having coarse-grained Hölder exponent α in the limit $\epsilon \to 0$. The fractal dimension referred here is Hausdorff, not box-counting.

2.6.2. Multifractal spectrum calculation

The multifractal spectrum is not of much use if it can not be calculated. For some type of measures (binomial and multinomial, see Peitgen et al. [1992]) an analytical calculation is

possible but for most of the practical cases, (financial data being the one we are interested here), that is not possible.

In what follows we describe the different ways of how we can evaluate numerically the multifractal spectrum. The best known reference to the methods presented here is Halsey et al. [1986].

2.6.2.1. Histogram method

The algorithm of the histogram method is as follows:

1. Coarse-grain the measure with boxes of size ε, $\{B_i(\varepsilon)\}_{i=1}^{N(\varepsilon)}$, where $N(\varepsilon)$ is the total number of boxes needed to cover the support of measure μ.
2. For a given ϵ evaluate the weight of box i, $\mu_i = \mu(B_i) \rightarrow \alpha_i = \frac{\log \mu_i}{\log \varepsilon}$, where α_i is the coarse-grained Hölder exponent for box i.
3. Construct the histogram of α to estimate $N_\varepsilon(\alpha)$
4. Repeat 3 for different coarse-grained values of ε.
5. Since we expect $N_\varepsilon(\alpha) \sim \varepsilon^{-f(\alpha)}$, plot $-\frac{\log N_\varepsilon(\alpha)}{\log \varepsilon}$ versus α for different values of ε.

This method suggests that a measure is multifractal when the resulting plots collapse onto a curve $f(\alpha)$ if ε is small enough.

We can have some convergence problems here, there are self-similar measures [Mandelbrot, 1990] for which the collapse to a function $f(\alpha)$ is extremely slow, and largely irrelevant for any physically meaningful ϵ.

A practical application of this method can be seen in Matos and Duarte [1999].

2.6.2.2. Method of moments

Definition 2.6.2. For a given measure μ we define the *partition function* as the quantity

$$\chi_q(\varepsilon) = \sum_{i=1}^{N(\varepsilon)} \mu_i^q, \quad q \in \mathbb{R} \tag{2.55}$$

where $N(\epsilon)$ is the number of boxes of size ϵ needed to cover the support of measureμ.

Another quantity that can be defined, using the analogy from thermodynamics, is the free energy.

Definition 2.6.3. For a given measure μ we define the free energy as

$$\tau(q) = \lim_{\varepsilon \rightarrow 0^+} \frac{\log \sum_{i=1}^{N(\varepsilon)} \mu_i^q}{\log \varepsilon}. \tag{2.56}$$

The purpose now is to show that most of the contributions to the partition function can be (mostly) attributed to a single value of α.

If we consider $N_\varepsilon(\alpha)\mathrm{d}\alpha$ as the number of boxes with Hölder coarse-grained exponent between α and $\alpha + \mathrm{d}\alpha$ we can replace the sum by an integral

$$\chi_q(\varepsilon) = \int N_\varepsilon(\alpha)\,(\varepsilon^\alpha)^q\,\mathrm{d}\alpha. \tag{2.57}$$

If $N_\varepsilon(\alpha) \sim \varepsilon^{-f(\alpha)}$ then we get

$$\chi_q(\varepsilon) = \int \varepsilon^{\alpha q - f(\alpha)}\mathrm{d}\alpha \tag{2.58}$$

In the limit $\varepsilon \to 0^+$ the dominant contribution of the integral comes from the α's closes to the value that minimises $\alpha q - f(\alpha)$. If $f(\alpha)$ is differentiable the minimum $\alpha = \alpha(q)$ satisfies $\frac{\partial f(\alpha)}{\partial \alpha}\Big|_{\alpha=\alpha(q)} = q$ and $\frac{\partial^2 f(\alpha)}{\partial \alpha^2}\Big|_{\alpha=\alpha(q)} < 0$.

Considering $\tau(q) = q\alpha(q) - f(\alpha(q))$ we get $\chi_q(\varepsilon) = \varepsilon^{\tau(q)}$. We can interpret $\tau(q)$, the free energy, as the scaling behaviour of the partition function.

The algorithm used to compute the spectrum is thus the following:

1. Coarse-grain the measure with boxes of size ε, $\{B_i(\varepsilon)\}_{i=1}^{N(\varepsilon)}$, where $N(\varepsilon)$ is the total number of boxes needed to cover the support of measure μ.

2. Compute $\chi_\epsilon(q)$ for various values of ε.

3. Plot $\log \chi_\epsilon(q)$ vs ε and check that they are straight lines. If so $\tau(q)$ is the slope of the corresponding line.

4. Form $f(\alpha)$ by computing the Legendre transform of $\tau(q)$

As compared with other methods this converges faster. One of the drawbacks of these methods is that there are self-similar measures which do not have all the moments, say for $q < 0$.

2.6.3. Properties of $f(\alpha)$

Using the definition 2.54 we can obtain the properties of $f(\alpha)$.

Definition 2.6.4. Let $A^\alpha(\varepsilon)$ be the subset of boxes covering the support of the measure having a coarse Hölder exponent between α and $\alpha + \mathrm{d}\alpha$.

From equation 2.54 we get that

$$\mu(A^\alpha(\varepsilon)) = N_\varepsilon(\alpha)\varepsilon^\alpha \mathrm{d}\alpha \sim \varepsilon^{\alpha - f(\alpha)}. \tag{2.59}$$

From this we get $f(\alpha) \leq \alpha$, if not the measure would diverge.

Another important property is that $f(\alpha)$ intercepts α and since $f(\alpha)$ is convex this is a single point and corresponds to $q = 1$, where $f'(\alpha) = 1$. This corresponds to $q = 1$, (the moment of order 1), and this value is denoted by $\alpha(1)$ or α_1.

The subset of boxes $A^{\alpha_1}(\varepsilon)$ carries all the measure in the limit $\varepsilon \to 0^+$, i.e., $\mu(A^{\alpha_1}(\varepsilon)) \to 1$ for $\varepsilon \to 0$.

We have then that $\frac{\partial}{\partial q}\tau(q) = \alpha(q)$, and $f(\alpha) = q\alpha(q) - \tau(q)$, where $\tau(q)$ and $f(\alpha)$ are Legendre transforms. Both are strictly cap convex:

$$\alpha(q) = \lim_{\varepsilon \to 0^+} \frac{\sum_{i=1}^{N(\varepsilon)} \frac{\mu_i^q}{\sum_{j=1}\mu_j^q} \log \mu_i}{\log \varepsilon}, \tag{2.60}$$

$$\alpha(1) = f(\alpha(1)) = \lim_{\varepsilon \to 0^+} \frac{\sum_{i=1}^{N(\varepsilon)} \mu_i \log \mu_i}{\log \varepsilon}. \tag{2.61}$$

$\alpha(q)$ is a decreasing function of q, $\alpha_{min} = \alpha(+\infty)$ and $\alpha_{max} = \alpha(-\infty)$.

The function $\tau(q) = (q-1)D_q$, where D_q are the generalised dimensions, the most know cases are: D_0 is the fractal dimension; D_1 is the information dimension and D_2 is the correlation dimension.

2.6.4. Multifractal stochastic processes

We can take the method of moments and apply it to time series, we can then define stochastic multifractal processes, see [Mandelbrot et al., 1997, Calvet and Fisher, 2002].

Definition 2.6.5. A stochastic process $\{X(t)\}$ on an interval $T \ni 0$ of positive length is called multifractal if it has stationary increments and there exists an interval $Q \subset [0, 1]$ and functions τ and c on Q such that

$$E\left\{|X(t)|^q\right\} = c(q)t^{\tau(q)+1} \tag{2.62}$$

for all $q \in Q$.

2.7. Fractional Brownian motion

In what follows let us assume that $X(t)$ is a time series.

Definition 2.7.1. Fractional Brownian motion (fBm) [Doukhan et al., 2003] is a well-known stochastic process where the second order moments of the increments scale as

$$E\{(X(t_2) - X(t_1))^2\} \propto |t_2 - t_1|^{2H} \tag{2.63}$$

with $H \in [0, 1]$. The Brownian motion is then the particular case where $H = 1/2$.

The exponent H is called the Hurst exponent. If $H < 1/2$, then the behaviour is *anti-persistent*, that is, deviations of one sign are generally followed by deviations with the

opposite sign. The limiting case $H = 0$, corresponds to white noise, where fluctuations at all frequencies are equally present.

If $H > 1/2$, then the behaviour is *persistent* (smooth), i.e., deviations tend to keep the same sign. The limiting case $H = 1$, reflects $X(t) \propto t$, a smooth signal.

While motivation for fBm was the fat-tail characteristic of real price distributions [Mandelbrot, 1963], this $H-$threshold for persistent/anti-persistent behaviour is useful in terms of determining when trends break down.

In what follows of this Section we will study methods to estimate the Hurst exponent. For a survey of fBm generators see Bardet et al. [2003].

2.7.1. Rescaled range (R/S) calculation

An illustration of the use of Hurst measurements may be draw from the 1950's where these were first developed to explain the flow of the Nile river, calculated by the (traditional) rescaled range approach,[Hurst, 1951].

Given a series $X(i)$, $i = 1, .., n$ where n is the length of the series, classical R/S is defined as R_n/S_n where

$$R_n = \max_{1\leq i\leq n} \sum_{i=1}^{k} (X(i) - \overline{X}) - \min_{1\leq i\leq n} \sum_{i=1}^{k} (X(i) - \overline{X}), \tag{2.64}$$

$$S_n = \frac{1}{n} \sum_{j=1}^{n} (X(j) - \overline{X})^2. \tag{2.65}$$

Hurst found that, for his time series and the case of the Nile river flow, $R_n/S_n = kn^H$, where k is a constant and H the Hurst exponent. Using a linear least squares fit, with $y = \log R_n/S_n$ as a function of $x = \log n$, we obtain the Hurst exponent as the slope of the resulting graphic.

One advantage of this method is its easy interpretation, since by considering the difference between the minimum and maximum of the deviations we obtain measure of variability in the time series. It should also be noted that the numerical implementation is straightforward. One of the main problems with this method, however is that the distribution of its test statistics is not well defined, making application of hypothesis tests for statistical confidence results difficult. Other serious issues include sensitivity to short range dependence and to heterogeneity of the data series.

2.7.1.1. Modified R/S statistics and other improvements

The dependency of the original method on maximum and minimum data makes it very sensitive to outliers; this is an important weakness. In order to overcome the deficiencies of the initial method several improvements were proposed with the most important due to Lo [1991].

The formulation of time series using fBm's, therefore with properties described by Hurst exponents, has been used in a number of different areas. One of the consequences has been the study of the Hurst itself and proposals for alternative methods of computing the exponent. Some examples of this are Higushi [1988], DePetrillo et al. [1999], Chang and Chang [2002].

In the remainder of this section, we explore those methods for evaluating the Hurst exponent frequently used in econophysics papers.

2.7.2. Detrended fluctuation analysis (DFA)

The DFA technique consists in dividing a random variable sequence $X(n)$, of length N, into N/t non-overlapping boxes, each containing t points [Peng et al., 1994]. Then, the linear local trend $z(n) = a\,n + b$ in each box is defined to be the standard linear least-square fit of the data points in that box. The detrended fluctuation function F is then defined by:

$$F^2(k,t) = \frac{1}{t}\sum_{n=kt+1}^{(k+1)t} |X(n) - z(n)|^2\,, \qquad k = 0, \cdots, \frac{N}{t} - 1. \tag{2.66}$$

Averaging $F(k,t)$ over the N/t intervals gives fluctuation average $F(t)$ as a function of t

$$F(t) = \left(\sum_{k=0}^{\frac{N}{t}-1} F^2(k,t)\right).^{1/2} \tag{2.67}$$

If the observables $X(n)$ are random uncorrelated variables or short-range correlated variables, the behaviour is expected therefore to be a power law

$$F(t) \sim t^H, \tag{2.68}$$

where H is the Hurst exponent.

DFA has the advantage over standard variance analysis of being able to detect long-term dependence in non-stationary time series. Additionally, the advantages, compared to other methods, of using DFA to compute H (for instance, the Fourier transform) are:

1. inherent trends are avoided at all t scales; since those trends are discarded by the fluctuation function; for more details about the issue of trends and DFA see Hu et al. [2001] and Kantelhardt et al. [2001];

2. local correlations can be easily probed, since we are detrending at a large range of scales, see also Chen et al. [2002].

It should be noted that DFA is a crude measure, especially as it is sensitive to non-normality of data. On a practical note it should be said that it is not enough to obtain an exponent from the detrended function, but is necessary also to check the quality of the

fit. We have used the correlation coefficient (r) as a crude measure of the goodness of the fit. All results showed a high value of r giving us some confidence in the results obtained through DFA.

2.7.2.1. Computational implementation

The code used to implement DFA was the code available from the authors of the original paper Peng et al. [1994]. This software is described in Appendix D.

During this work two other implementations were made, one using C for performance, and another using Python in the sub-module *dfa*. Both implementations are described in Appendix C.

2.7.3. Multifractal generalisations

The multifractal generalisation deals with the use of moments when determining the Hurst exponent.

2.7.3.1. Multifractal DFA

One generalisation of DFA is MF-DFA where MF stands for multifractal [Kantelhardt et al., 2002]. The determination is similar to that for DFA, but instead of taking into account just the behaviour of $F(t)$ we take into account its moments. If we use $F(k,t)$ as defined in equation 2.66 we generalise:

$$F_q(t) = \left(\sum_{k=0}^{\frac{N}{t}-1} F^2(k,t)^{q/2} \right) .^{1/q} \tag{2.69}$$

The new relation becomes then

$$F_q(t) \sim t^{h(q)}. \tag{2.70}$$

We call function $h(q)$ the generalised Hurst exponent.

2.7.3.2. Generalised Hurst exponent

In the same vein it is possible to extend the Hurst exponent to include the moments of the increments for different scales as done in Di Matteo et al. [2005].

Let $X(t)$ be a time series (with $t = \nu, 2\nu, ..., k\nu, ..., T$), where τ is the time resolution and T the observation period.

A multifractal generalisation of the Hurst approach should be associated with the scaling behaviour of the statistically significant properties of the signal, since at all scales the scaling behaves differently, with possible crossovers being present.

Consequently we analyse the $q-$order moments of the distribution of increments which provide a good characterisation of the statistical evolution of a stochastic variable $X(t)$. The qth moment is given by

$$K_q(\tau) = \frac{\langle |X(t+\tau) - X(t)|^q \rangle}{\langle |X(t)|^q \rangle} \tag{2.71}$$

where τ is a time interval and can vary as $\nu \leq \tau \leq \tau_{max} < T$.

The generalised Hurst exponent $H(q)$ can be defined from the scaling behaviour of $K_q(\tau)$ which can be assumed to be given by the relation

$$K_q(\tau) \sim \left(\frac{\tau}{\nu}\right)^{qH(q)} \tag{2.72}$$

Unifractal behaviour corresponds to the case where $H(q) = H$, i.e. constant and independent of q. In the more general case the process is called multifractal or multi-scaling and with exponents $H(q)$ characterising the scaling of the different moments q.

Computational implementation The implementation of the multifractal Hurst exponent is described in Appendix C, in the sub-module *multifractal*.

2.7.4. Using wavelets for H- estimation

We use the wavelet variance to evaluate the dependency of the variance on the studied level. This can be done estimating the variance explained by the different wavelet level j, for Hurst exponent estimation.

Recalling equation 2.43 we have decomposed a function into two components, a smooth part plus the wavelet details. We estimate the Hurst exponent using the relation [Abry and Veitch, 1998]:

$$\log_2(\frac{1}{n_j}\sum_k |d_j(k)|^2) = (2\,\hat{H} - 1)\,j + \hat{c} \tag{2.73}$$

where n_j is the available number of wavelet coefficients at level j and $\hat{c}$ is a constant.

Computationally it is an easy task, since the wavelet decomposition packages described in Appendix D give directly the coefficients $d_j(k)$.

2.8. Stable laws - Lévy distributions

From the several models used to characterise the market behaviour, Brownian motion was the first [Bachelier, 1900]. Brownian motion has the property that the distribution of increments has a finite variance, and that the increments are uncorrelated at successive time steps.

In order to account the fat tails of financial data, in last section, we have generalised Brownian motion to fractional Brownian motion by dropping the second requirement, the Independence of increments.

There is another way, proposed by Mandelbrot [1963], to explain the fat tails, where the first requirement of Brownian motion, (finite variance of increments), does not hold anymore. A more general class of distributions stable under convolutions is required, the Lévy distributions. This stochastic process is called a *Lévy flight*, where the distribution of increments follows a Lévy distribution.

The Lévy distributions satisfy the relation $\mathrm{P}(X > x) \propto x^{-\alpha}$. The scale invariance property relates them clearly to fractals; this relation is further explored in Shlesinger et al. [1993, 1995].

Again, as in fBm we recover the Normal distribution as the limit case, where $\alpha = 2$, and thus the Brownian motion is a special case of a Lévy flight.

An interesting fact is that the only stable continuous distributions under convolutions are the Normal and the Lévy distributions with parameter α between 0 and 2.

Lévy flights imply that infinite variance, yet in practice all processes have finite variance and scale invariance in a limited range [Cont et al., 1997]. In order to overcome these deficiencies, truncated power laws and finite variance, a new generalisation was proposed, a truncated Lévy flight [Mantegna and Stanley, 1994, Koponen, 1995, Bouchaud and Potters, 2001].

Like fBm, Lévy stable laws have been applied in several areas, economics, finance, engineering, analysis of network traffic, physics and astronomy, (see [Nolan, 2005] for a detailed bibliography).

2.8.1. Stable distributions

In the introduction we referred stable distributions, here we make that concept accurate.

An important property of Normal distributions is their stability under addition, i.e. the sum of two independent Normal distribution is Normal. If X is Normal then for X_1, X_2 independent copies $\forall a, b \in \mathbb{R}\, \exists c, d \in \mathbb{R}$:

$$a\,X_1 + b\,X_2 \quad \stackrel{d}{=} \quad c\,X + d, \tag{2.74}$$

where $\stackrel{d}{=}$ means equal in distribution, i.e. both expressions have the same probability law.

Now suppose that $X \sim N(\mu, \sigma^2)$. We have then that $a\,X_1 \sim N(a\mu, (a\sigma)^2)$ and $b\,X_1 \sim N(b\mu, (b\sigma)^2)$ are the terms on the left-hand side of 2.74 while the right hand side is $N(c\mu + d, (c\sigma)^2)$. By the addition rule for independent normal variables, we must have $c^2 = a^2 + b^2$ and $d = (a + b - c)\mu$. In other words, what equation 2.74 says is that the shape of Normal is preserved (up to scale and shift) under addition.

We can generalise this definition, that we have shown to hold for Normal distribution.

Definition 2.8.1. A random variable X is *stable* or *stable in the broad sense* if for X_1 and X_2 independent copies of X and any positive constants a and b,

$$a\,X_1 + b\,X_2 \quad \stackrel{d}{=} \quad c\,X + d, \tag{2.75}$$

for some positive c and $d \in \mathbb{R}$. The random variable is *strictly stable* or *stable in the narrow sense* if the relations holds with $d = 0$ for all choices of a and b. A random variable is symmetric stable if it is stable and is symmetric around 0, e.g. $X \stackrel{d}{=} -X$.

Other than Normal, there are two other distributions with a close formula that satisfy this definition. Those distribution are Lévy and Cauchy and are described in Appendix B.

It easy to generalise this definition to an equivalent version with a sum of any number of distributions.

Definition 2.8.2. Non-degenerate X is *stable* if and only if for all $n > 1$, there exist constants $c_n > 0$ and $d_n \in \mathbb{R}$ such that

$$X_1 + \cdots X_n \quad \stackrel{d}{=} \quad c_n X + d_n, \tag{2.76}$$

where $X_1, \cdots, X_n$ are independent, identical copies of X. X is strictly stable if and only if $d_n = 0$ for all n.

Using the Generalised Central Limit Theorem [Nolan, 2006], we have yet another equivalent definition that has the advantage of being parametrised.

Definition 2.8.3. A random variable X is stable if and only if $X \stackrel{d}{=} a\,Z + b$, where Z is a random variable with characteristic function

$$E[\exp(iuZ)] \quad = \quad \begin{cases} \exp(-|u|^\alpha(1 - i\beta\tan\frac{\pi\alpha}{2}(\text{sign}u)) & \alpha \neq 1 \\ \exp(-|u|^\alpha(1 + i\beta\frac{2}{\pi}(\text{sign}u)\log|u|) & \alpha = 1 \end{cases} \tag{2.77}$$

and $0 < \alpha \leq 2$, $-1 \leq \beta \leq 1$, $a > 0$, $b \in \mathbb{R}$.

If $\beta = 0$ and $b = 0$ the characteristic function of $a\,Z$ has a simpler form

$$\phi(u) = \mathrm{e}^{-a^\alpha|u|^\alpha}. \tag{2.78}$$

This definition gives an explicit (closed) formula for the Fourier transform of the distributions. The only stable distributions that have a closed formula, are the Normal, Cauchy and Lévy, with the parametrisation that follows:

- $N(\mu, \sigma^2)$ is stable with $(\alpha = 2, \beta = 0, a = \sigma^2/2, b = \mu)$
- Cauchy(γ, δ) is stable with $(\alpha = 1, \beta = 0, a = \gamma, b = \delta)$
- Lévy$(\gamma, \delta)$is stable with $(\alpha = 1/2, \beta = 1, a = \gamma, b = \delta)$

In Appendix B the different possible parametrisation of Lévy stable distributions are studied. The existence of different parametrisation is both due to historical reasons and to different purposes, analytical or numerical use. The parametrisation is important since it allows us to interpret the different types of stable laws, as they are richer than to be simply classified due to the exponent α. There are three further parameters necessary to uniquely characterise each stable distribution and the general parametrisation is denoted as $\mathbf{S}(\alpha, \beta, \gamma, \delta; k)$, where β is the asymmetry coefficient, γ is the scale coefficient, δ is the location parameter and k the number of the parametrisation type.

2.8.2. Tail properties and moments

A general distribution is said to be *heavy tailed* if its tails behaviour are heavier than exponential.

The initial interest of stable laws applied to financial problems was due to the heavy tails of the distribution of price increments [Mandelbrot, 1963]. In the following we study the analytical properties of the distributions that justify that interest.

Theorem 2.8.4. ***Tail approximation.*** *Let $Z \sim \mathbf{S}(\alpha, \beta, \gamma, \delta; k)$ with $0 < \alpha < 2$, $-1 < \beta \leq 1$. Then as $x \to +\infty$,*

$$P(X > x) \sim \gamma^{\alpha} c_{\alpha} (1 + \beta) x^{-\alpha} \tag{2.79}$$

$$f(x|\alpha, \beta; 0) \sim -\alpha \gamma^{\alpha} c_{\alpha} (1 + \beta) x^{-(\alpha+1)} \tag{2.80}$$

where $c_{\alpha} = \sin(\pi\alpha/2)\Gamma(\alpha)/\pi$.

For all $\alpha < 2$ and $\beta > -1$ the upper tails probabilities and densities are asymptotically power laws. (*Pareto tails*)

For all $\alpha < 2$, stable distributions have one or both tails that are asymptotically power laws with heavy tails. One consequence of heavy tails is that not all moments exist.

Studying the moments for stable distributions we have that $E[|X|^p]$ is finite for $0 < p < \alpha$ and $E[|X|^p] = +\infty$ for $p \geq \alpha$. For all stable laws, with $\alpha < 2$, the variance is infinite. Thus the first moment $E[X]$ and variance $\mathrm{Var}[X]$ do not characterise the distribution as well as in other distributions.

2.8.3. Signal generation and analysis

Here the computational methods used to simulate and estimate the parameters for stable laws are described.

For simulation we have used the GNU Scientific Library (GSL), described in Section 3.4.1, that has two functions to generate Lévy symmetric stable distributions, one for the symmetric and another for the skewed cases. They work by taking advantage of equation 2.77, and then taking an inverse Fourier transform.

The estimation of parameters it was used the software described in Appendix D, Section D.4.2 using the Maximum Likewood Estimator for stable laws [Nolan, 2001].

2.9. Entropy

The definition of entropy is the following:

Definition 2.9.1. Let X be a discrete random variable on a finite set $\mathcal{X} = \{x_1, ..., x_n\}$, with a probability distribution function $p(x) = \mathrm{P}(X = x)$. The entropy $H(X)$ of X is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x). \tag{2.81}$$

Note that here we are using H to represent the entropy while before it was used to represent the Hurst exponent. Since both notations are traditional in their fields I opted to keep them since most of the time there is no ambiguity on what is the quantity studied.

If we apply the previous definition to a continuous time series, e.g. financial, we have to partition the signal into k symbols, in order to complete the partition we need to choose the length of the words we will be using, say size m. The Shannon entropy for symbol sequences, with an alphabet of k symbols and block length m, gets a particular form [Kantz and Schreiber, 2004].

Before presenting the formula it is necessary a short introduction on how to code the sequences. We have k^m possible sequences, we can associate any integer number j, such that $0 \leq j < k^m$, with its digit representation on base k as $j = (j_{m-1} j_{m-2} \dots j_1 j_0)_k$, where each digit $0 \leq j_i < k$ for $0 \leq i < m$. We can then associate a probability p_j to each of these sequences.

Definition 2.9.2. The Shannon entropy for blocks of size m for an alphabet of k symbols is

$$\widetilde{H}(m) = -\sum_{j=0}^{k^m - 1} p_j \log p_j, \tag{2.82}$$

the entropy of the source is then

$$\widetilde{h} = \lim_{m \to \infty} \frac{\widetilde{H}(m)}{m}. \tag{2.83}$$

This definition is attractive for several reasons: it is easy to calculate and it is well defined for a source of symbol strings. In the particular case of returns, if we choose a symmetrical partition we know that half of the symbols represent losses and half of the symbols represent gains. If the sequence is predictable, we have the same losses and gains sequences repeated everytime, the entropy will be lower; if however all sequences are equally probable the uncertainty will be higher and so it will be the entropy. Entropy is thus a good measure of uncertainty.

This particular method has problems, the entropy depends on the choice of encoding, it is not a unique characteristic for the underlying continuous time series. Also since the number of possible states grows exponentially with m, after a short number of sequences in practical terms it will become difficult to find a sequence that repeats itself. This entropy is not invariant under smooth coordinate changes, both in time and encoding. This is a strong handicap for its adoption into financial time series study.

The entropy shows a different behaviour for odd and even k if we have a large bulk in the centre of the distribution, as it usually happens for financial time series. Analysing the histogram the reason is obvious the central bulge is covered by either one or two symbols, depending on the parity of k. The sequence will have a smaller entropy for odd values of k.

2.9.1. Order-q Rényi entropies

A series of entropy-like quantities, the order-q Rényi entropies Rényi [1961], characterise the amount of information which is needed in order to specify the value of an observable with a certain precision [Kantz and Schreiber, 2004].

Definition 2.9.3. Let $\mathcal{P}_\epsilon$ be a partition of disjoint boxes $\mathcal{P}_j$, of size length $\leq \epsilon$, over the support of measure μ. If we consider $\mu(\mathcal{P}_j) = p_j$ then

$$\widetilde{H}_q(\mathcal{P}_\epsilon) = \frac{1}{1-q} \log \sum_j p_j^q \tag{2.84}$$

is the *q-order Rényi entropy* for the partition $\mathcal{P}_\epsilon$.

Note for $q = 1$ we have to apply the de l'Hospital rule where we get

$$\widetilde{H}_1(\mathcal{P}_\epsilon) = -p \sum_j p_j \log p_j. \tag{2.85}$$

$\widetilde{H}_1(\mathcal{P}_\epsilon)$ is thus the Shannon entropy as defined in equation 2.81. In contrast to the other Rényi entropies is additive, i.e. if the probabilities can be factorised into independent factors, the entropy of the joint process is the sum of the entropies of the independent processes.

Note that this generalisation of the entropy is closely related to the free energy function within the multifractal analysis $\tau(q) = (1-q)D_q$. If we analyse equations 2.60 and 2.61 we see the same relation has we have between equation 2.84 and 2.85.

2.9.2. Kolmogorov-Sinai entropy

The Rényi entropies gain even more relevances when they are applied to transition probabilities, equation 2.83. We apply the same reasoning as before, apply a partition $\mathcal{P}_\epsilon$ on the dynamic range of the observable, and introduce the joint probability $p_{i_1,i_2,\ldots,i_m}$ that at

an arbitrary time n the observable falls into the interval I_{i_1}, at time $n+1$ fall into interval I_{i_2}, and so on.

Definition 2.9.4. The block entropies of block size m is

$$H_q(m, \mathcal{P}_\epsilon) = \frac{1}{1-q} \log \sum_{i_1, i_2, \ldots, i_m} p^q_{i_1, i_2, \ldots, i_m}. \tag{2.86}$$

The *order-q entropies* are then

$$h_q = \sup_{\mathcal{P}} \lim_{m \to \infty} \frac{1}{m} H_q(m, \mathcal{P}_\epsilon) \Leftrightarrow h_q = \sup_{\mathcal{P}} \lim_{m \to \infty} h_q(m, \mathcal{P}_\epsilon), \tag{2.87}$$

where

$$h_q(m, \mathcal{P}_\epsilon) := H_q(m+1, \mathcal{P}_\epsilon) - H_q(m, \mathcal{P}_\epsilon), \quad h_q(0, \mathcal{P}_\epsilon) = H_q(0, \mathcal{P}_\epsilon). \tag{2.88}$$

In the original sense only $h_1$was called the Kolmogorov-Sinai entropy [Kolmogorov, 1958, Sinai, 1959], but since the idea is the same, the name was extended to cover all the other Rényi entropies.

Kolmogorov and Sinai where the first to consider correlations in time in information theory. The limit $q \to 0$ gives the topological entropy h_0. As D_0, the fractal dimension of the support of the measure, just counts the number of non-empty boxes in partition, h_0 gives just a measure of the different orbits, not of their relative importance as we get with h_1.

Another extension of entropy, related with Rényi entropies, is Tsallis non extensive entropy [Tsallis, 1988a], with applications to economics described in Tsallis et al. [2003].

2.9.3. Computational implementation

The implementation is described in Appendix C, in the Python sub-module *entropy*. The package only implements Shannon entropy for blocks.

2.10. Time dependent covariance matrix

All the techniques used before dealt with a single time series. The time dependent covariance matrix (see Litterman and Winkelmann [1998]) studies the multivariate case, (several random variables at once).

One of the properties that this method shares with others used in this work is the time dependent results that allows to compare with results obtained with other methods.

Definition 2.10.1. The covariance matrix with variable weights at time T, over an horizon M, $\sigma^T(M)$, is given by:

$$\sigma^T_{ij}(M) = \frac{\sum_{s=0}^{M} W_s r_{i,T-s} r_{j,T-s}}{\sum_{s=0}^{M} W_s}. \tag{2.89}$$

Where $r_{i,t}$ is the value of return r_i at time t, and W_s is the weight given for the covariance at delay s, (time $T - s$).

The weight vector,$\mathbf{W}$, has decreasing components since we give higher weights to closer times for moments closer to the time we are analysing. One example traditionally used and the same that is used in this work is $W_i = R^i$, with $0 < R < 1$. Then we have $\sum_{s=0}^{T} W_{T-s} = \frac{R^T}{1-R^T}$, and W_i corresponds to a geometric series. Typical values (see Litterman and Winkelmann [1998]) are $R = 0.9$ and $T = 20$.

According to the findings of Gallucio et al. [1998], Laloux et al. [1999], Plerou et al. [1999], Laloux et al. [2000], Plerou et al. [2001], Wilcox and Gebbie [2004], Sharifi et al. [2004] the correlation (or covariance) matrices of financial time series, apart from a few large eigenvalues and their corresponding eigenvectors, appear to contain such a large amount of noise that their structure can essentially be regard as random.

Such as in Wilcox and Gebbie [2004], Sharkasi et al. [2006a] we will consider the three larger eigenvalues and its respective eigenvectors as carrying meaningful information.

In the multivariate signal processing problem, one key issue might be when instabilities occur in signal patterns and how we might determine if the fluctuations are damped, remain at low level, or combine in some way as to cause a major event, e.g. a market crash. Crashes are also interesting since the market dynamics changes during the event, see Vilela Mendes et al. [2003], Araújo and Louçã [2006].

2.10.1. Computational implementation

The computational implementation of this method is presented in Appendix C.

Computationally there is one important issue, that does not appears when considering a single data set, sometimes for a given day different markets are closed, even if the cause is a local holiday. In the implementation we have skipped those values, ignoring both the covariance product and the respective weight.

3. Computational Implementation

> *"Turning ideas into software in this way need not be an unpleasant duty, of course: programming can be very stimulating and immensely satisfying. In addition the exercise of drafting an algorithm to the level of precision that programming requires can in itself clarify ideas and promote rigorous intellectual scrutiny. In our view it is somewhat ironic that even very substantial software contributions do not seem to attract the same academic credit as refer reed publications: in reality nearly every user of the software becomes a more meticulous and critical reviewer than most anonymous referees!"* - Venables and Ripley [2000]

3.1. Introduction

The purpose of this chapter is to introduce some of the computational methodology used in this thesis as well as to discuss some of the changes and challenges in today's scientific computing landscape.

It is my opinion that scientific computing is an interesting study in itself, to the same degree as it is needed to realise the mathematical tools and techniques demanded. The choice of computational tools and techniques applied in this work is as important or nearly so as the mathematical formulation since the results are based on their discriminating application and they serve as a basis for characterising the work.

According to wikipedia (`http://en.wikipedia.org/wiki/Scientific_computing`):

> "*Scientific Computing (or Computational Science) is the field of study concerned with constructing mathematical models and numerical solution techniques and using computers to analyze and solve scientific and engineering problems. In practical use, it is typically the application of computer simulation and other forms of computation to problems in various scientific disciplines.*
>
> *The field is distinct from computer science (the mathematical study of computation, computers and information processing). It is also different from theory and experiment which are the traditional forms of science and engineering. The scientific computing approach is to gain understanding, mainly through the analysis of mathematical models implemented on computers.*
>
> *Scientists and engineers develop computer programs, application software, that model systems being studied and run these programs with various sets of*

> *input parameters. Typically, these models require massive amounts of calculations (usually floating-point) and are often executed on supercomputers or distributed computing platforms.*
>
> *[...] Computational science may be considered as a new third mode of science, complementing and adding to experimentation/observation and theory.*"

The reason for placing the quotation above illustrates a feature of Internet, the dynamic nature, unlike books where data is presented statically the Internet is constantly changing. The quotation above is thus a frozen image of the content above at the time this Chapter was written and will certainly be different at a later time.

This definition is interesting for two seemingly unrelated reasons: for one it highlights the role of scientific computing and places it on the same footing as theoretical and experimental fields; another less immediately obvious reason is that the Internet has not only brought more comprehensive search and access but has realised new ways for people to coordinate and show scientific work. Wikipedia, (*Wiki* stands for What I Know Is, it was first used as a way to allow people to foster collaboration by allowing anyone to modify the content), an on-line encyclopedia that can be updated by anyone, subject to very few rules, provides one simple example; the sharing of bio-related data through on-line databases such as PDP, GEO, KEGG, ExPasy and others or the financial data available from Yahoo/Finance, used in this work, is another.

Today the use of computers is pervasive, inside and outside science, and that is also a consequence of the use of Internet. Each computer is a "laboratory" in itself since it allows the use of new methodologies in problem solving as well the exploration of new areas, while the Internet as a global network brings a new level of interaction and potential collaboration. This Chapter tries to explore this duality: the use of computers in scientific computing and opportunities provided by the collaborative platform that a global network of connected computers allows to the scientific community. One of the postulates of this Chapter is that it is increasingly difficult to distinguish between the two sides of this coin and the use at local level is essentially one end of the global gateway continuum.

We are still trying to understand how best to take advantage of computers to analyse and understand the problems posed in everyday scientific activity. This is the Internet age, where we have new ways to cooperate and develop new techniques. If we take the Wikipedia example, there are recent reports, Nat [2005], comparing the accuracy of traditional encyclopedias, with this decentralised process with results (perhaps surprisingly) showing little differences in terms of the defects/errors statistics of the articles.

Early use of computers allowed the exploration of new horizons, which have continued to expand to current time, and this change of paradigm will take some time to settle down. Meanwhile we need to signpost the route taken. For this reason the methodology section contains external references to software (at the current stage of development) that are used to implements some of the proposed methods. These signposts, directions and choice

of routes will, inevitably, be updated with time.

In what follows in this Chapter we define Free Software, how it is related to scientific computing and the analogies between both dynamics. Free Software has been used exclusively in this work and the intention is to reinforce that choice as a methodological approach, an important one in the author's opinion.

Free Software is not the single methodology that can be highlighted in this work. Equally, although computer science and scientific computing are different areas of study, this should not be a reason for them not to share methods. On the contrary, methods can be "borrowed" from each area with considerable advantages for both and those relevant here are presented in more detail in Section 3.3.

Next the tools/programs used in this work are described, and the reasons for each tool choice are presented in a general introduction while reference to a more exhaustive list of specific packages is made later. The general framework discussed in detail, permits other tools to be built.

Collaboration is intrinsic both to modern Science and the development of Free Software. Every incremental step is a gain towards the higher objective of better understanding. In the last section, projects are discussed to which the author has made a contribution during the course of this work.

3.2. Free Software

3.2.1. Introduction

Universities were some of the first places to adopt the Internet, and for long time academic centres were both its major users and its backbone. The Internet has allowed development of new tools, with email and the Web (the result of an experiment from Tim Berners Lee to make information easily available on CERN, European Organisation for Nuclear Research, a research centre on high energy (particle) physics,) being two of the best known examples.

The symbiosis between Free Software and the Internet was mutual with Free Software both a product of the Internet and its main supporter. If it was not for Free Software, it is very doubtful that we would be as advanced as we are today with the omnipresent Internet.

3.2.2. Definition

New methods for transfer of information promoted the emergence, in 1984, of the Free Software movement. Free Software existed before this date, initially sharing software was the rule that later became the exception.

UNIX is one such example, due partially to Bell Company antitrust case, the antitrust rule forbid Bell from doing business outside of telecommunications. UNIX was developed in Bell and its code was distributed on request. Only later attempts were made down

on the to close the code, but by then it was too late. The Free Software Manifesto from Richard Stallman was the first sign of a new understanding required to preserve those roots of knowledge transfer. This freedom to share has been strongly associated with the scientific method (see Stallman [2002] story of Tycho `http://www.gnu.org/philosophy/right-to-read.html`). According to Free Software Foundation (`http://www.gnu.org/philosophy/free-sw.html`):

> "*Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software.*"

It should be noted that those rules only apply to distribution, any private changes are permitted by the license and do not need to be published. This remark may seem superfluous, yet it is frequently misunderstood.

Free should not be perceived as "gratis" and it is possible to have commercial free software. It is also referred to as Open Source, and for most practical matters these are mutually interchangeable, but the philosophy behind each is a little different. The focus of Open Source is based on notion that "*It works*", while Free Software treats software as ideas and in the same spirit as that of scientific investigation: ideas develop a lot faster if they are shared.

The Free Software Foundation created the GNU project, designed to create a Free Software derivative of UNIX. At the same time a license was developed to legally uphold the ideals of Free Software; that license is GPL (General Public License), and it forms the corner stone of the Free Software movement. Most of the software projects presented here are released under this license, this applies both to the libraries created specifically for this work (Appendix C) as well as to external software used (Appendix D).

Other Free Software licenses exist, with the most important being the BSD/MIT class. The major difference is that this type of license allows the derivative code to be made proprietary, (as opposed to free).

3.2.3. Free Science - Open Access

Other movements inside apply the same philosophy to other domains, (arXiv and Wikipedia are two already cited). "Open Access" is a movement which intends to guarantee free access to scientific articles (see Moody [2006a,b,c,d], Kelty [2001], Willinsky [2005]).

In illustration, a useful resource in writing this thesis was `http://planetmath.org/` (Planet Math), a community site dedicated to the compilation of mathematical knowledge. PlanetMath's content is created collaboratively: the main feature is the mathematics encyclopedia with entries written and reviewed by members.

3.3. Methodological approach

This section illustrates the different computational methodologies used in this thesis. Methodologies here are intended as tenets for all computer-related work. The distinction between methodologies and tools, (presented in the next section), is that methodologies are related to the design data analysis and treatment and are independent of the tools used, although different tools favour different approaches.

Exclusive use of free software

A consequence of using Free Software is that programs can be ported everywhere. In this case this implies many Operating Systems, although naturally the tools are easiest to setup in the environment in which they have been developed.

Reproducibility of results

All results should be possible to regenerate easily: this usually entails the use of scripts to drive the different parts of the analysis.

Reuse of available software

Behaviour described in informatics terms, as NIH syndrome (where NIH stands for Not In Here) characterises the reaction of suspicion, by which any software not made internally is discarded. This equates to the proverbial case of "reinventing the wheel" every time. It is also unreliable in concept and practice as the other extreme, where any outcome from a computer program is taken as the "Truth".

Redundant methods

It does not matter if a program is fast if it is not correct. This tautology is easy to understand but less easy to implement.

In order to avoid single failure points every effort has been made to implement all methods using at least two different implementations. This in itself does not guarantee the correctness of the results but does increases our confidence in them.

One other technique coming from software development is "Unit testing". The idea here is that tests for the code are written first, then the code itself. There is an analogy with mathematical systems in that one of the methods we use is the identification of invariants, (quantities that remain unchanged over a given range of operations).

Unit testing advocates the writing of tests where we compare the empirical result to that expected based on known cases, in order to ensure the correctness of the code at hand.

3.4. Tools

Tools described are general and not restricted to implementation of any particular technique; they allow and encourage the creation and use of libraries related to the problems studied.

The programs and languages described where are the basis for those the software used in this work. The library developed in this work for the analysis of financial time series (Appendix C) is a Python library. The external software (Appendix D) has a broader origin and it was developed using any of the languages presented below.

3.4.1. Languages and libraries

An important distinction between different languages relates to their libraries, whether the standard library or available add-ons.

There are few, if any, "one size fits all" solutions; every language used has advantages and drawbacks, which are discussed in detail in what follows.

3.4.1.1. R

R (`http://www.r-project.org`) is a free implementation of the S language. S was primarily developed at AT&T Bell Laboratories to be a language oriented towards Statistics, (hence the name).

The repository of available packages, (almost all of which are Free Software), can be found in R homepage CRAN (Comprehensive R Archive Network, `http://cran.r-project.org`).

3.4.1.2. Python

Python (`http://www.python.org`) is a general purpose script language with a tidy syntax and with very good and appropriated features as a glue language, i.e. a language that holds together different programs.

Numeric Python `http://numpy.sourceforge.net` is the python library that adds numeric arrays to python. For this reason it has become the basis for lots of other packages.

matplotlib `http://matplotlib.sf.net` [Barrett et al., 2004] is a python library that adds support for plotting graphics.

rpy `rpy.sf.net` is a bidirectional wrapper that allows communication between python and R.

3.4.1.3. bash

bash (`http://www.gnu.org/software/bash/`) is an implementation of a shell language. It serves as a framework for further tools, used as building blocks of powerful scripts:

coreutils The GNU Core Utilities (`http://www.gnu.org/software/coreutils/`) are the basic file, shell and text manipulation utilities of the GNU operating system. These are the core utilities which are expected to exist on every operating system. Examples of tools belonging to this package are: *cut*; *head*; *tail*; *paste*; *join*; *sort* and *uniq*.

awk (`http://www.gnu.org/software/gawk/`): the GNU implementation of the AWK programming language. It allows for simple manipulation of patterns and it is best used with shell. The gawk package contains the GNU version of awk, a text processing utility. Awk interprets a special-purpose programming language to do quick and easy text pattern matching and reformatting jobs.

grep (`http://www.gnu.org/software/grep`): grep searches through textual input for lines which contain a match to a specified pattern and then prints the matching lines.

sed (`http://www.gnu.org/software/sed`) the sed (Stream EDitor) editor is a stream or batch, (non-interactive), editor. Sed takes text as input, performs an operation or set of operations on the text and outputs the modified text. The operations that sed performs, (substitutions, deletions, insertions, etc.), can be specified in a script file or from the command line.

findutils (`http://www.gnu.org/software/findutils/`): the find utility searches through a hierarchy of directories looking for files which match a certain set of criteria, (such as a filename pattern), and executing the ordered operation.

3.4.1.4. C/C++

Technically C and C++ are different languages with C being a subset of C++, (there are very small differences that will not be discussed here). Although sharing some features, the philosophy of the languages is different since each (as for all languages) employs a different conceptual approach to problem solving.

The compiler used to compile all the C and C++ programs here was gcc (`http://gcc.gnu.org/`). GCC stands for Gnu Compiler Collection as it covers other languages namely Fortran, with gfortran its frontend for Fortran 95.

Numerical Analysis Backbones Among the scientific community reference to the "Numerical Recipes" series of books on numerical methods is widespread. Several criticisms

however, have been made to these books; while a useful introduction to numerical methods, their methods are not the most effective, stable or modern. An alternative place to search for numerical method implementation is `http://www.netlib.org/`.

Atlas (`http://math-atlas.sourceforge.net/`) (Automatically Tuned Linear Algebra Software): the project is an ongoing research effort focusing on applying empirical techniques in order to provide portable performance. At present, it provides C and Fortran77 interfaces to a portably efficient BLAS implementation, as well as a few routines from LAPACK.

C Lapack

LAPACK (Linear Algebra PACKage) is a standard library for numerical linear algebra. Lapack (`http://www.netlib.org/lapack/`), clapack (`http://www.netlib.org/clapack/`) and lapack++ are the respective implementations for Fortran, C and C++.

GSL The GNU Scientific Library (`http://www.gnu.org/software/gsl/`) is a numerical library for C and C++ programmers.

pygsl (`http://pygsl.sourceforge.net/`) provides a python interface for the GNU scientific library (gsl).

3.4.2. General

3.4.2.1. Computer Algebra System

In the fortunate cases, where analytical calculus is possible, Maxima is used to perform the calculations `http://maxima.sourceforge.net`.

3.4.2.2. Plotting

Grace (`http://plasma-gate.weizmann.ac.il/Grace/`) is an application for two-dimensional data visualisation. Grace can transform the data using free equations, FFT, cross- and auto-correlation, differences, integrals, histograms, and much more. The generated figures are of high quality. Grace is a very convenient tool for data inspection, data transformation, and for publication-quality figures.

gnuplot (`http://gnuplot.info/`) is a command-line driven, interactive function plotting program especially suited for scientific data representation. Gnuplot can be used to plot functions and data points in both two and three dimensions and in many different formats.

ipython (`http://ipython.scipy.org/`) provides a replacement for the interactive Python interpreter with extra functionality.

3.5. Contributions of this work to software projects

In addition to econophysics analysis reported in the following Chapters, contributions to the software used in this work, have been extensive and in many cases have been made freely available for wider use. This is the typical "hitch and scratch" approach to Free Software.

LyX (`http://lyx.org`)

Writing, be it papers, reports or books is an integral part of scientific activity.

LyX is a document preparation system that encourages an approach to writing based on the structure of documents, not their appearance. LyX uses several backends the most important being LaTeX [Lamport, 1986], a set of macros build up on Donald Knuth TeX [Knuth, 1984].

The contributions of the author have been made to the other two backends, namely LinuxDoc and Docbook [Walsh and Muellner, 1999]. Another contribution was made to the sub-system that allows to read, and update, older versions of the LyX file format. This allowed to decouple to support for older versions from the C++ code improving the file format to clean and more robust state.

As an illustration of the capabilities of the software, this thesis was written in LyX.

Fedora (`http://fedoraproject.org`)

Fedora is an international project to build a Linux distribution. A Linux distribution is the collection of the software packages with the necessary framework to install in a new computer. Examples of other widely used distributions are Debian, Ubuntu, Suse, Gentoo or Mandriva. The mission goal of Fedora states its strong commitment to Free Software.

One of the other goals is the ability to run in different CPU's architectures, presently it runs on the Intel 32 bits architecture (and compatible like AMD), 64 bits from Intel and AMD chipmakers, PowerPc architecture (whose most famous member were the Mac's) and Sparc architecture from Sun. This allows to run (almost) the same software versions across this heterogeneous hardware.

This project has several components with the more important being Fedora Core and Fedora Extras. Fedora Core is a Linux distribution descendant of the popular Red Hat Linux. As the name implies it intends to release distributions with the base packages (Core). Fedora Extras is a community-oriented project with the goal of building general packages to run in Core.

Fedora serves as a basis for other Linux distributions (both directly and indirectly). One such project, Scientific Linux (`https://www.scientificlinux.org/`), is oriented towards Scientific Computation. Scientific Linux is a Linux release put together by Fermilab, CERN, and various other laboratories and universities around the world. Its primary

purpose is to reduce duplicated effort of the laboratories, and to have a common install base for the various experimenters.

The packages listed below were submitted and accepted for release in Fedora Extras, and are maintained by the author:

fftw-2 `http://www.fftw.org/` version 2 of FFTW is an excellent implementation of Fast Fourier Transforms.

grace `http://plasma-gate.weizmann.ac.il/Grace/` a plotting program.

python-imaging `http://www.pythonware.com/products/pil/` a python library for image manipulation.

pygsl `http://pygsl.sourceforge.net` a python library with bindings for GSL.

rpy `http://rpy.sourceforge.net` a python library with bindings for R language.

R-mAr `http://cran.r-project.org/contrib` R package for Multivariate AutoRegressive analysis.

R-waveslim `http://cran.r-project.org/contrib` R package for wavelet studies. The code provided here is based on wavelet methodology developed in Percival and Walden [2000].

R-wavetresh `http://cran.r-project.org/contrib` R package for wavelet studies. Software to perform 1-d and 2-d wavelet statistics and transforms.

tetex-dvipost `http://efeu.cybertec.at/` a LaTeX package to post-process the dvi output.

4. Portuguese Standard Index (PSI-20) Analysis

"*Jean-Luc Picard: Sometimes it's possible to make no mistakes and still lose. It is not a weakness. It is life.*" - Star_Trek: The Next Generation ("Peak Performance")

4.1. Introduction

This Chapter is an extension of the work presented in Matos et al. [2004]. Some of the previously described econophysics tools are applied to the Portuguese Standard Index PSI-20. PSI-20 index main characteristics are described in Section 4.2. The Portuguese case is chosen both for: a) regional relevance; b) relatively little previous study and c) its relevance as a showcase both as an emerging young/mature market and its relevance to discuss features on the techniques presented.

The data analysis, using multiplicative and additive stochastic models, studying the empirical distribution of data and the trend persistence analysis, is presented in Section 4.3.

Detrended fluctuation analysis (DFA), (introduced in Chapter 2), is applied to the PSI-20 daily time series and results are analysed and discussed in Section 4.4. This initial application was the forerunner and constituted the main motivation for the development of the generalisation of this method, presented in next Chapter. The results presented favour a multifractal description of the data therefore in Section 4.5 we explore the multifractal Hurst exponent.

The main conclusions relative to PSI-20 are gathered in Section 4.6.

4.2. The Portuguese Stock Index PSI-20

The Portuguese Stock Index PSI-20 is the national benchmark index, reflecting the price evolution of the 20 largest most liquid assets selected from the set of companies listed on the Portuguese Main Market. The rules for construction of PSI-20 are published [PSI, 2003], but can be summarised briefly as giving a different weight to each asset belonging to the index, such that no asset has more than 20% of the total weight.

The data used in this manuscript are the daily values at the close of session for the PSI-20 obtained through the Porto services of Bolsa de Valores de Lisboa e Porto

(BVLP). PSI-20 had its beginning in January 4th 1993 and still remains as an independent index.

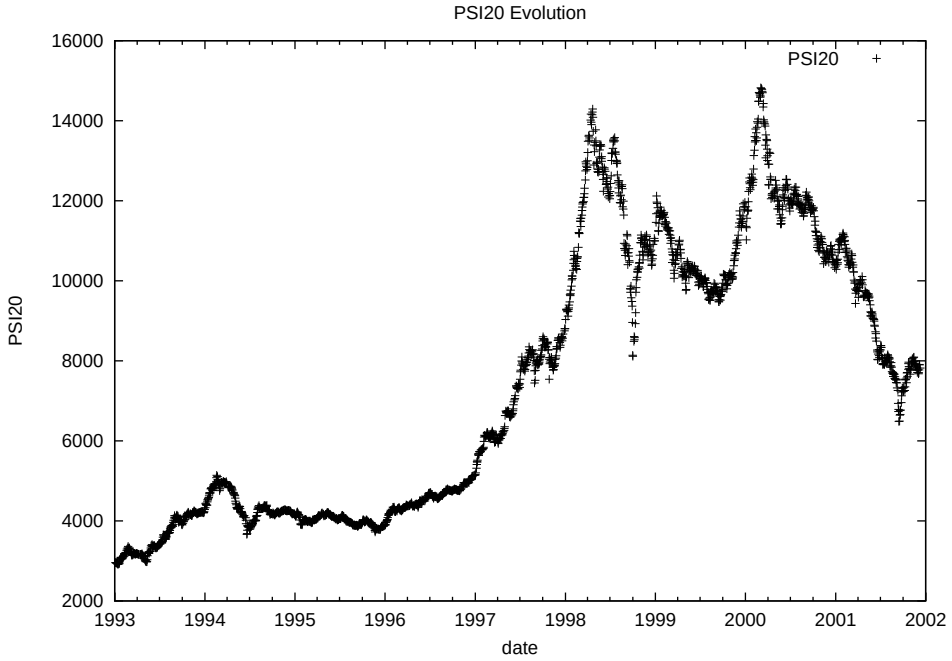


Figure 4.1.: PSI-20 evolution from 1993 to 2002.

Figure 4.1 shows the different periods present in the PSI-20 index. In the first 15 months we can observe a clear initial growth, followed by a stable (small fluctuations) period until the beginning of 1997. From 1997 to the first quarter of 1998 there is a surge, where the PSI-20 index triples its value. After this period and up to 2000, we observe a highly volatile regime characterised by strong fluctuations and short range trends. After the 2000 peak (roughly corresponding to the dot com bubble burst), we essentially have a decline.

In the following, we present an analysis of the PSI-20 signal, using time series models (the additive and multiplicative variants) and methods, (such as detrended fluctuation analysis and correlation functions), which have recently been popularised in the econophysics literature.

4.3. Data analysis

4.3.1. Stochastic models

Most of the statistical methods assume (weak or strong) stationarity to deal with time series, e.g. Chatfield [2003]. Simple forms of time series models may be written for the additive and multiplicative cases, respectively, as:

$$X_t = x_0 + \sum_{i=0}^{t} \delta X_i \tag{4.1}$$

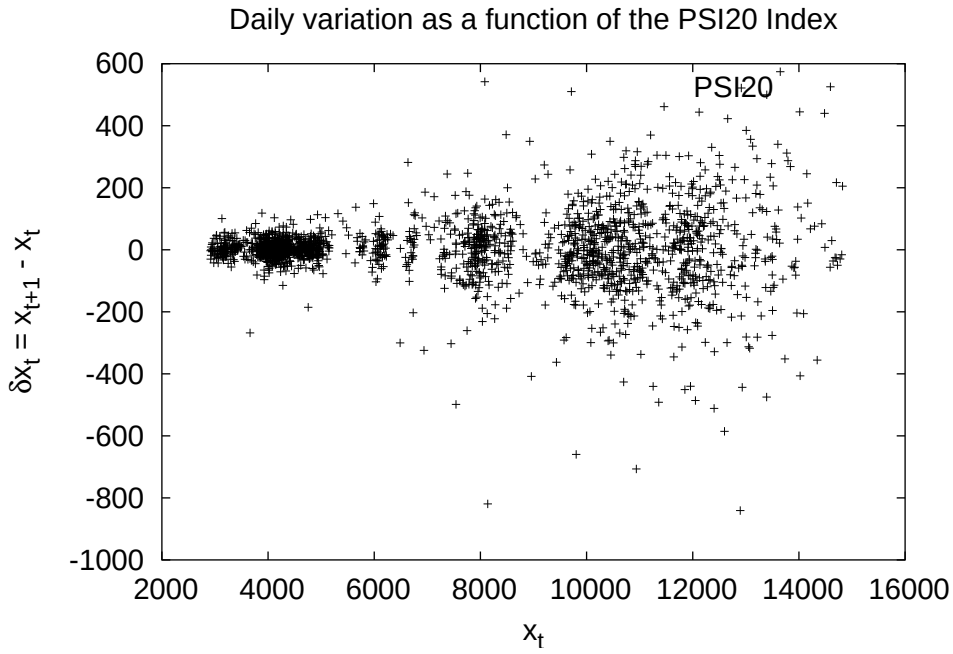


Figure 4.2.: Dispersion relation for the PSI-20 time series as a function of the index value.

or

$$X_t = x_0 \prod_{i=0}^{t} (1 + \eta_i) \ , \tag{4.2}$$

where $\delta X_i = X_{i+1} - X_i$ and $\eta_i = \frac{\delta X_i}{X_i}$, for times series values X_i $(i = 1, 2, \cdots)$, where x_0 is the initial data.

Expectations of performance, according to Bouchaud and Potters [2001] are that the best models for financial data show a combination of both short-term additive and long-term multiplicative effects.

In Figure 4.2 we can see that the index variation grows with the index values. This increased dispersion with increasing X_t is typical of financial data of this type and taking logarithms of the original series is usually required to stabilise the variance.

The next step is to quantify this variation, since this is a stochastic process and we are interested in studying the dependency of the average of fluctuations with the index value. For this we divided the range of possible values for the PSI-20 into 20 regions of equal size. It should be noted that the results obtained are robust to a variation of the number of intervals around the chosen value.

We took the expected value (average) of the square deviations for each region. The reason for this transformation over segments (TOS) with 20 intervals is a compromise between the number of points in the fit and the number of samples in the interval. The number 20 is a compromise, a greater number of regions would decrease the number of

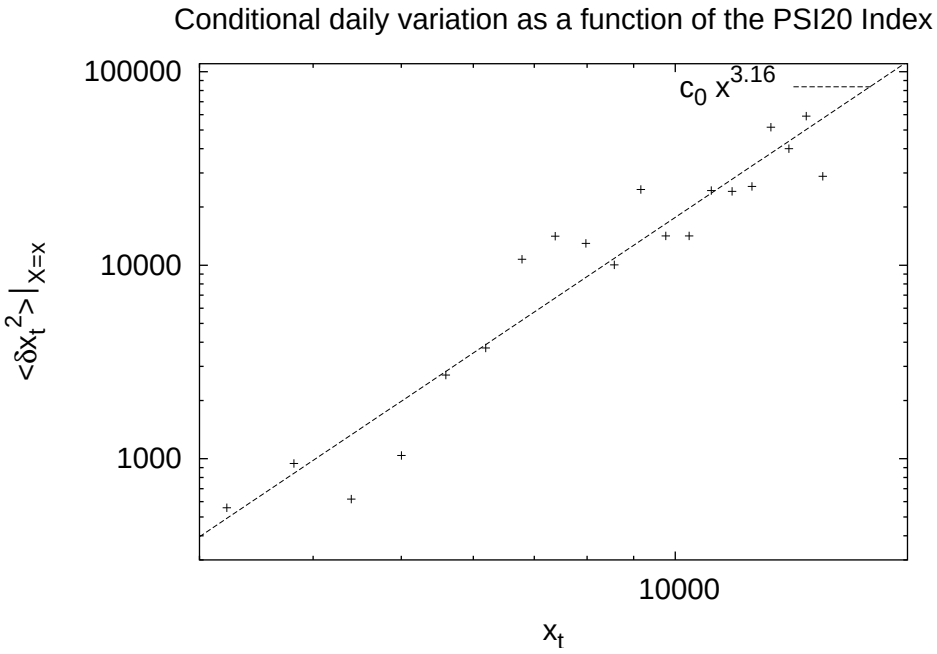


Figure 4.3.: Relation between $< \delta X^2 >$ and X, $< \delta X^2 > \propto X^\gamma$ with $\gamma = 3.16$.

points available for statistics and favour the outliers. A smaller number of regions would cover a wide range of the possible values of the index.

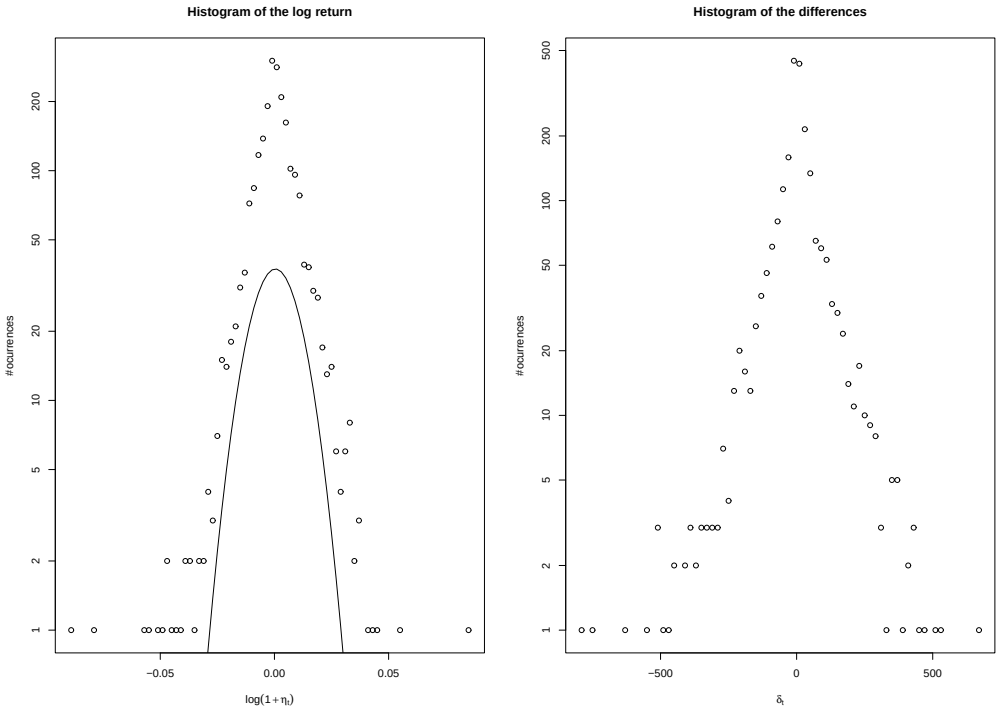
The result of the TOS analysis is plotted in Figure 4.3 and is presented on a log-log scale. A linear fit to the data presented in Figure 4.3 gives the exponent 3.16, or 1.58 for the standard deviation. Note that the value 1.58 is far from the value 1 that we would expect for a pure multiplicative model (a diffusive model with $H = 0.5$, the random walk).

4.3.2. Empirical distribution of data

Another way to characterise the PSI-20 time series data is to build the corresponding histograms and this has been done for δX, the daily difference, and for $\log(1 + \eta)$, the logarithmic return (defined in equation 1.1).

Figure 4.4 compares, in a linear-log plot, the histogram of log returns with the probability density function of a Normal distribution $\mathrm{N}(\mu, \sigma^2)$ with average and standard deviation equal to the corresponding sample average ($\mu = 4.3\,10^{-4}$) and sample standard deviation ($\sigma = 0.010$). The exponential growth and decay around the origin is clearly demonstrated, as is the presence of fat tails. It is also clear that several extreme events, (outlying values ≥ 5 standard deviation away from the mean), are also influencing the fit to a Normal distribution.

The histogram of the differences, in Figure 4.4, has similar features, although the decay near the origin is more erratic. Note that considerations about the extreme values again apply here. For the histogram of the differences the sample average is 2.21 and the sample



(a) Histogram of $\log\left(\frac{X_{t+1}}{X_t}\right)$ (return). (b) Histogram of $X_{t+1} - X_t$ (difference).

Figure 4.4.: Histograms for the return and difference from the PSI-20 time series.

standard deviation is 104.3. The standard deviation is very large and most of this comes from the two peaks occurring in 1998 and 2000, (seen in the Figure 4.1). This is illustrated again if we follow the evolution of the standard deviation with time, whit these two peaks being recovered.

Using a maximum likehood estimator, presented in Section 2.8.3, we recover the value $\alpha = 1.59$ for a Lévy stable distribution. This clearly agrees with the above description of a non-Normal distribution for logarithmic returns. The skewness coefficient $\beta = -0.001$ is again in line with the quasi-symmetry of the histogram, by which we mean that losses and gains are (almost) equally probable when comparing its absolute value.

4.3.3. Trend persistence analysis

4.3.3.1. Histogram

A further quantity of interest is the study of daily trends persistence, that is, the persistence of the same behaviour regarding gains/losses for some consecutive period of days. It is, of course, risky to infer a day-to-day trend in an index, for which only one value per day is recorded and which is itself volatile over that period. This criticism applies to many financial data, which tend to be volatile by nature, and constantly raises questions about how fine-grained or coarse-grained the intervals of recording should be. Nevertheless, for

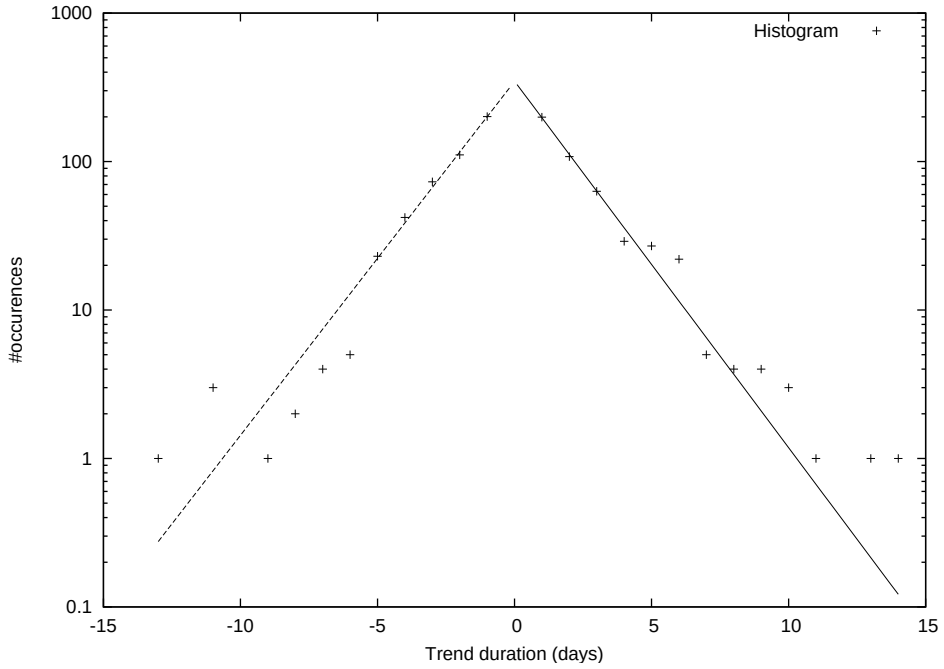


Figure 4.5.: Histogram of trends duration

the PSI-20, the available data show that the closing index value is a good indicator of its day average, in the sense that the absolute relative errors between the index at closing and the average of the maximum and minimum session values are almost everywhere less than 2%. Thus, the behaviour at closing has similar features on a day-to-day basis and constitutes a valid basis for comparison, albeit an approximate one.

Using the PSI-20 time series we build a new series with the daily trend persistence where the trading days will be distributed in clusters of different sizes. From the initial 2216 data points we get 993 trend clusters.

In Figure 4.5 we see a histogram of duration of trends or prolonged gains/losses. In order to help us to distinguish between the positive and negative trends we have considered the negative trends as corresponding to negative days. Although this is an artefact it is very effective in comparing the differences, if any, between the two behaviours, relating to gain/loss.

Considering an exponential fit to negative and positive trends, in the form

$$f(x) = a \exp(-b|x|),$$

we obtain the following results for the histogram:

	a	b
Negative	347.0 ± 11.9	0.55 ± 0.02
Positive	347.7 ± 13.5	0.57 ± 0.02

These results suggest that within the error range of the coefficients both negative and

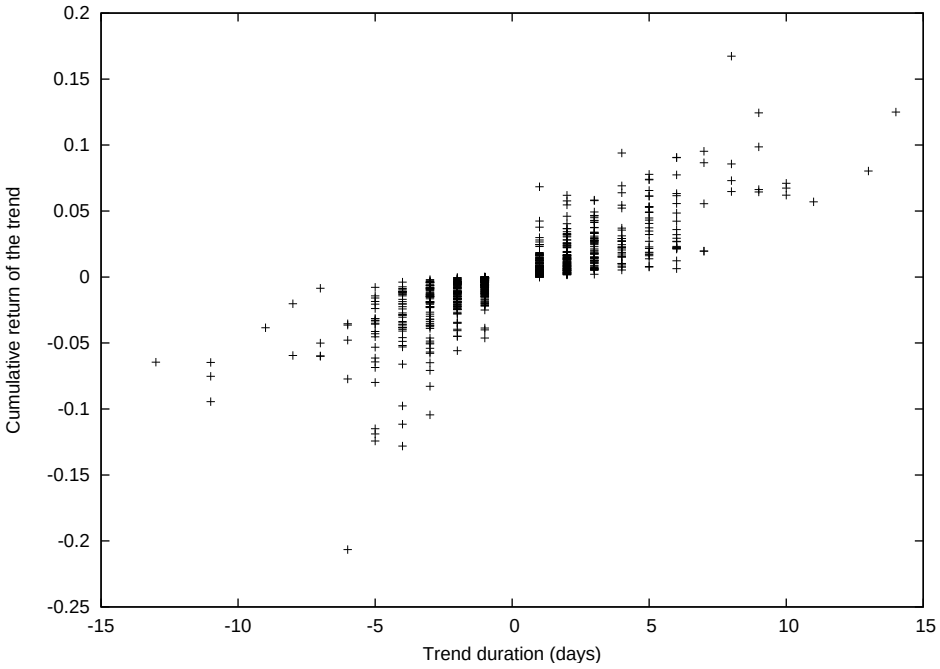


Figure 4.6.: Relative loss/gain of trends. The cumulative return is compared between the beginning and the end of the trend.

positive trends follow the same law.

4.3.3.2. Return during trends

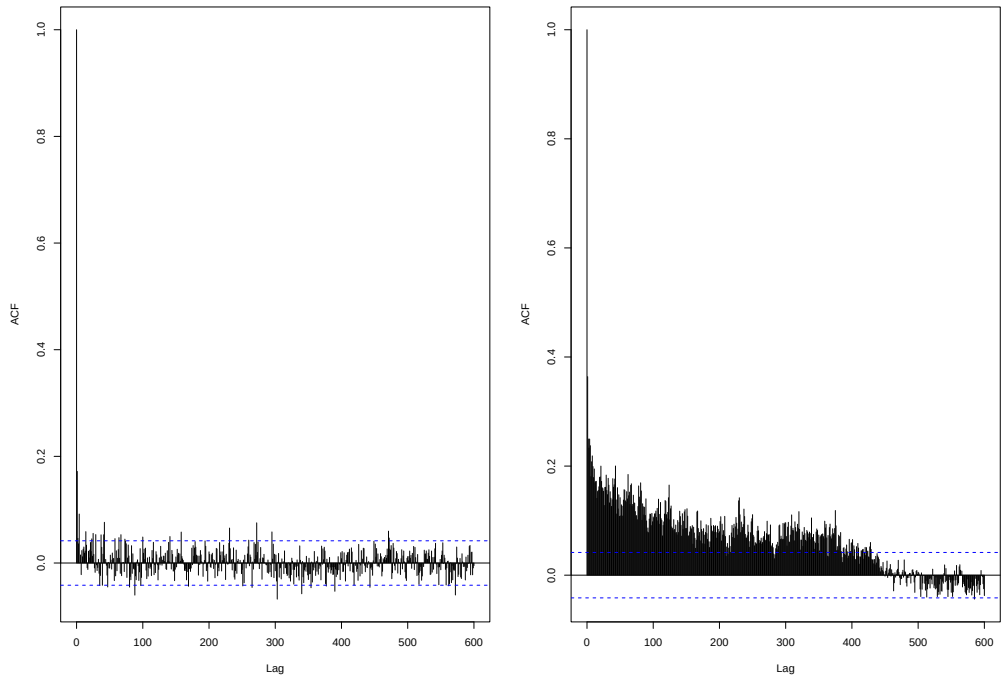
Also important is understanding how the size of the trend is related to its cumulative return as can be seen in Figure 4.6. Again we follow the same convention as above to highlight the characteristic behaviour of losses and gains.

It seems clear that returns become less cautious or conservative on longer positive trends, as do large losses on prolonged negative trends. Where quick changes are occurring (i.e. where there are a lot of short up/down runs), returns are correspondingly cautious as the market is clearly uncertain.

4.3.4. Autocorrelation function for the return series

The return series of the PSI-20 index, defined as $\eta_t = \frac{\delta X_t}{X_t}$, shown in Figure 4.7, presents a short term memory. This is typical behaviour, as observed in other financial series [Bouchaud and Potters, 2001], in liquid markets the correlation of price changes decays to negligible levels in a few minutes, consistent with the absence of long term statistical arbitrage.

If we consider the autocorrelation function for the modulus time series, $|\eta_t|$, that is the absolute variations without regarding the sign, then the series displays long-term memory. This is known in financial literature, non-linear functions of the returns exhibit significant



(a) Autocorrelation function for η_t.

(b) Autocorrelation function for $|\eta_t|$.

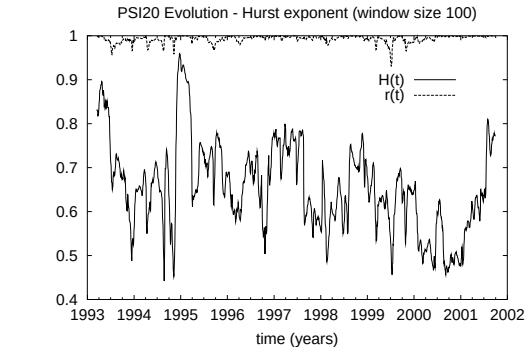
Figure 4.7.: Autocorrelation function for η_t time series.

positive autocorrelation (see Ding et al. [1993], Harvey [1993]). This is an indication that the volatility is clustered in time, that is we have periods with stronger fluctuations and others where the variations are small, but they are grouped together.

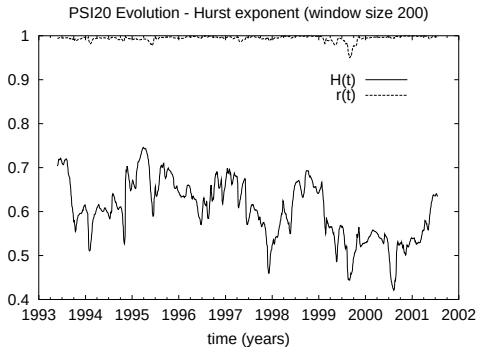
4.4. Detrended fluctuation analysis

Applying the DFA technique, described in Section 2.7.2 to the whole PSI-20 time series, we obtain the value of $H = 0.59$. This single value hides the complexity inherent in the time series, due to the periods of distinct behaviour as noted is Section 4.2, (see Figure 4.1). However, the nature of the approach (i.e. based on the interval characterisation in terms of the Hurst exponent) also means that we can apply the DFA to smaller intervals of fixed size (100, 200, and 400 points). Each one of these sub-intervals is characterised by its Hurst exponent.

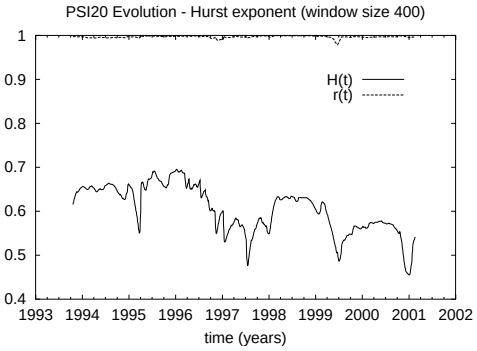
The choice 100, 200, and 400 point intervals corresponds to one half, one and two years of the PSI-20. The purpose of this analysis on different scales is to test the dependence of the results on the granularity of the data, since as seen in multifractal analysis (Section 2.6) we expect different behaviours at different scales for financial time series.



(a) Size 100.



(b) Size 200.



(c) Size 400.

Figure 4.8.: The $H(t)$ exponent obtained for different sizes of the "sliding" window.

4.4.1. Graphical analysis for sliding windows

In Figure 4.8, each point represents the centre of a *sliding* window, moved along the series, and its correspondent Hurst exponent. The latter is obtained by fitting a power law to the DFA function $< F(t) >$ computed in the sliding window. Regression coefficients are computed for the fit in each case.

The regression coefficient $r(t)$ is also plotted for each point revealing the quality of the fit where the H exponent is evaluated; in all graphics the regression coefficient is near 1. All regression coefficients, $r(t)$, may be seen to fall in the range $0.95 - 1$, giving us confidence in the power law behaviour of $< F(t) >$.

We also see that, for all window sizes, the exponents evolve to values close to $H = 1/2$. This fact can be interpreted, according to the Efficient Market Hypothesis [Fama, 1970], in terms of maturation of the market. Maturity, here, is used in the sense of increased stability or reduced liability to extreme fluctuations, with improved ability to sustain absorption of/or response to other external market influences.

Also it can be seen from the evolution of the Hurst exponent that, for the different window sizes, there is a rich structure at different scales reinforcing our previous remarks that a multiplicative model, (or more generally any uni-fractal model), does not provide

scale	100	200	400
	$1998, 1Q$	$1997, 4Q$	$1997, 2Q$
	$1999, 2Q$	$1999, 3Q$	$1999, 2Q$
	$1999, 3Q$	$2000, 3Q$	$2000, 3Q$

Table 4.1.: Common events in all scales where $H(t)$ drops below 0.5.

a complete and simple explanation for the PSI-20 data.

Smaller scales are more sensitive, and react quickly, to changes in the market. The graphics are becoming smoother as the scale increases. The correlation coefficient becomes almost indistinguishable from 1 at the larger scale, supporting the description of the intervals by the Hurst exponent.

Searching for commonalities in the three scales we see three events, where $H(t)$ drops below 0.5 in scale 400, can be found as well in the other scales. Those events are summarised in Table 4.1.

Comparing those dates with Figure 4.1 we see that around that period the index presents a highly volatile behaviour in the random walk sense, when compared with surrounding regions, an indication of a more mature behaviour at that time.

4.5. Multifractal Hurst exponent

The fBm approach is essentially uni-fractal and is predominantly used for insight on persistent/anti-persistent behaviour in this instance. Applying to PSI-20 series the generalised Hurst exponent, defined in Section 2.7.3.2, $H(q)$ where q is the moment, we get the graphic in Figure 4.9.

It is easy to see the decreasing exponent for higher values of moments. As it can be seen in the Figure 4.9 we recover the value obtained from DFA for moment $q = 2$, $H(2) \approx 0.59$. For a discussion of similar results of FX (Foreign Exchange rates) markets see Di Matteo et al. [2005].

4.6. Conclusions

For the Portuguese PSI-20, we have demonstrated that the daily variation in the index value exhibits a power-law exponent of value 1.58, in contrast to that predicted for pure additive or multiplicative models. The time series for these data shows distinct periods, characteristic of an emerging marketThe classification of markets according to emerging/-mature dichotomy is discussed in Appendix A., with much of the variation in the series due to the period after 1998. The series demonstrates the typical fat tails, though additional variation is noticeable, particularly in the period $1998 - 2000$.

The use of DFA to analyse the behaviour of the Hurst exponent in a fBm approach indicates that the daily series exhibit short-term persistent behaviour, though persistence

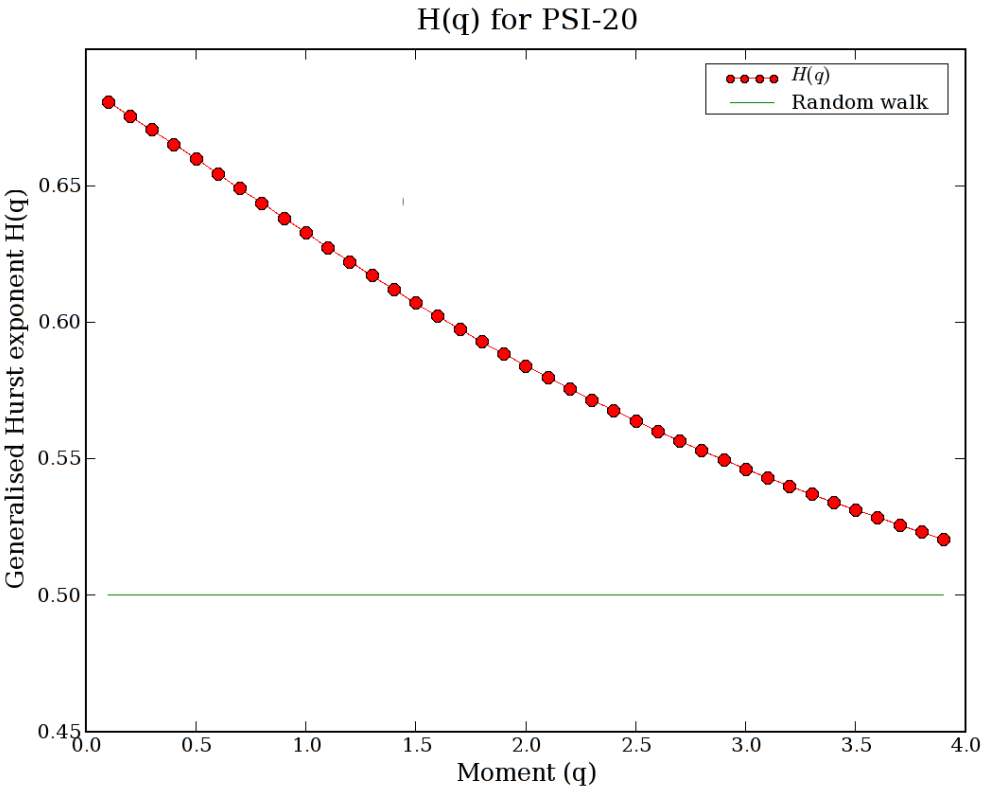


Figure 4.9.: Generalised Hurst exponent applied to PSI-20 time series.

degenerates over long periods. In particular, over intervals of 400 time points, the H exponent shows a gradual decline to anti-persistent behaviour. Much of this is due to the noted erratic period in the daily index $(1998 - 2000)$ and coincides with the Portuguese market emergence in the global context.

There is some evidence that stability has improved towards the end of the series. In other words the maturation of the Portuguese market suggests alignment with the global tendency of other reference markets.

5. Time and Scale Detrended Fluctuation Analysis (TSDFA)

"*I react pragmatically. Where the market works, I'm for that. Where the government is necessary, I'm for that. I'm deeply suspicious of somebody who says, 'I'm in favor of privatization,' or, 'I'm deeply in favor of public ownership.' I'm in favor of whatever works in the particular case.*" - John Kenneth Galbraith

5.1. Introduction

In the previous Chapter we have applied several econophysics tools to the study of the Portuguese Stock Index (PSI20). We have applied DFA to "sliding windows" of different sizes. The motivation and importance of this kind of analysis is the well known multifractal behaviour that financial data exhibits (see Lux [2004]). This was reflected in the output for 100, 200 and 400 trading days windows, as seen in Figure 4.8.

A natural extension of this analysis is to consider other window sizes, i.e. to go from $H(t)$ at scale s to $H(s,t)$. The results may then be combined in a $3D$ graphic where the scale and temporal dependencies of the Hurst exponent can be displayed for the time series studied. The previous analysis is thus a particular case, a cross section of $H(s,t)$ with the scale fixed.

In this Chapter we formalise this generalisation of the DFA to scale and time dependencies, (Section 5.2).

The results of the analysis are presented in Section 5.3, where we classify different markets according to the resulting patterns, that turned to be more richer than the traditional distinction between developed/emergent. The traditional distinction between developed and emergent is discussed here as well. The explicit time and scale dependency allows us to draw more confident conclusions about the class types and what markets these contain.

The last section thus details the principal conclusions and points directions for further study.

5.2. Generalisation of time and scale for the Hurst exponent

The dependency of the Hurst exponent on time and scale, $H(s,t)$, is akin to the Continuous Wavelet Transforms (CWT), described in Section 2.4.2. In both of these transforms, there is data redundancy, since we are moving from 1 to 2 dimensions, but this allow us to detect several features in these data that would otherwise be hidden.

Another resemblance with CWT is that both techniques display the time and scale dependency of the results.

5.2.1. Method characterisation

The general idea behind this method is the study of the Hurst exponent as a function of both time and scale. In practical terms this method is a simple expansion of the "windowed" DFA applied in Matos et al. [2004]. Instead of fixing s we let it be a variable. The Hurst exponent, $H(t,s)$, for time t and scale s, is evaluated as the Hurst exponent obtained using the DFA, (described in detail in Section 2.7.2), for the interval $[t-\frac{s}{2}; t+\frac{s}{2}]$.

Implications are wider than for a simple DFA. The general idea is to essentially invert the process and take $H(s,t)$ as the focus of the analysis with the DFA being an implementation detail. The other candidate to evaluate the Hurst exponent in the sub-intervals is the wavelet based method described in Section 2.7.4. In both cases H is recovered as a power of the scale, inside each sub-interval, (see equations 2.68 and 2.73).

Recalling the most important equation in DFA we have the detrended fluctuation function as (Equation 2.68):

$$F(t) \sim t^H,$$

where H is the Hurst exponent.

From the above condition we know that $s/2 + 1 \leq t \leq T - s/2$, where T is the time series length. In what follows the maximum scale we consider is $s = T/4$ as for large scales we essentially recover the Hurst exponent for the whole series.

A major concern in this work was to guarantee that exponents obtained through DFA were meaningful. For that reason we have used the same procedure as in Matos et al. [2004], we have controlled the quality of the fits assuring that the regression coefficients of the linear least squares fits were near unity for all studied markets. If we would not do this, the results would be unreliable, since the underlying time series is not well described by a fractional Brownian motion. To this combination of the DFA with time and scale dependency, we apply the term TSDFA (Time and scale DFA).

5.2.2. Examples

Here we study some examples of the technique applied to several international markets. We choose these because they display details that are either unique or shared with other

Date	Events
1997/07	Asian crash
1997/11	Asian crash
1998/10	Global crash
1999/10	Memory of a crash (no crash)
2000/03	DotCom crash
2001/09/11	Terrorist attack (New York)
2002/05	Stock Market Downturn
2003/12	General Threat level raised
2004/03/11	Terrorist attack (Madrid)

Table 5.1.: Major events for global markets.

markets and contribute to understand the differences and similarities that TSDFA emphasises.

Traditionally we distinguish between developed and emergent markets, the distinction varies depending on the source and of the applied criteria. A more in depth discussion of this issue is found in Section 5.3.

In order to better understand the results we display in Table 5.1 a list of major events that have affected international markets (see Sharkasi et al. [2006a]).

5.2.2.1. Nikkei

As an illustration of the method we worked with Nikkei 225 data ranging from 1984 to 2005. The evolution of the index over the last 20 years is represented in Figure 5.1. Nikkei was chosen because it is a well known and studied financial index.

The graph resulting from application of the TSDFA method is shown in Figure 5.2. The graphic represents as a contour plot, with exponents in range $[0.3; 0.9]$, the series studied from $1990-2005$ and the scale between 100 and 400 trading days. In this work we adopted these fixed ranges since this representation permits ready comparison with other indices calculated, (this applies both to the examples in this Chapter and in Appendix A).

In the Nikkei graphic (Figure 5.2) we can see that persistence is exhibited with the index normally around 0.5. This reflects a healthy blue/pink borderline and is to be expected since Nikkei is a mature market. In recent years we see a red stripe that crosses all scales in year 2000, at the same time as the DotCom crash.

We have another stripe that starts in the fourth quarter of 2001 but does not go through all scales. Another period of high values of H starts for short scales in the third quarter of 2002, after a global crash and reaches large scales in 2004.

5.2.2.2. FTSE (UK)

In Figure 5.3 we see the method applied to FTSE, a well known mature market. Just like in the Nikkei case, blue dominates the graphic. As can be seen there the Hurst exponent

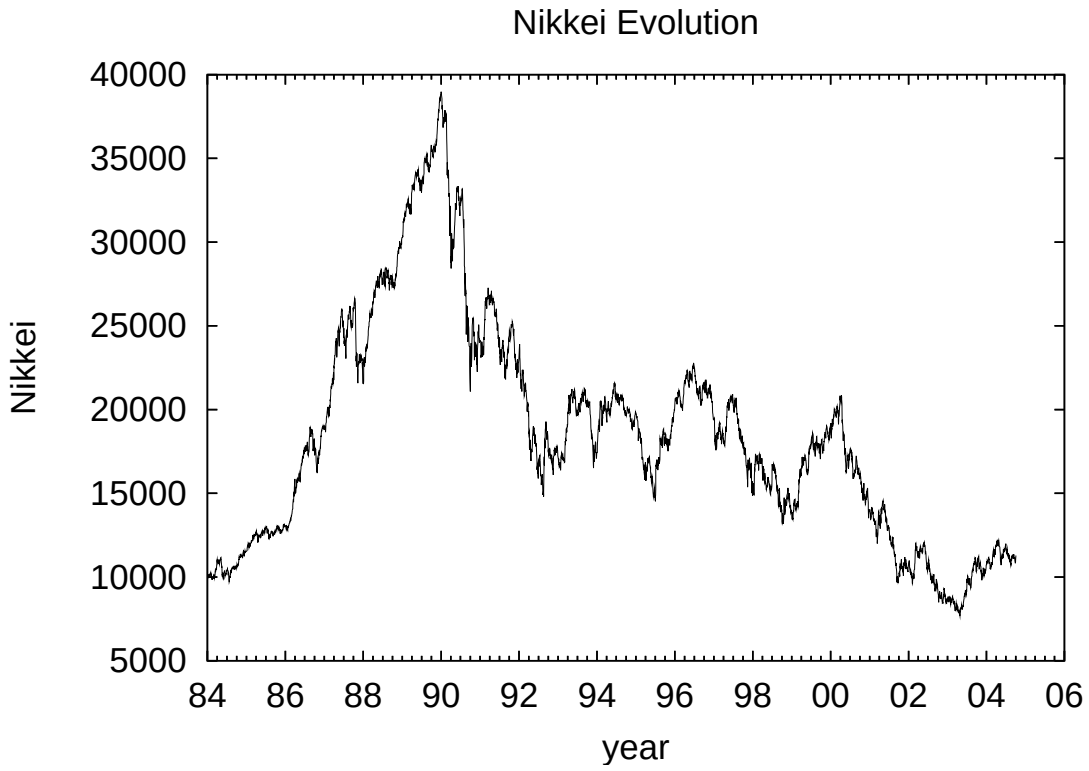


Figure 5.1.: Nikkei 225 evolution.

has been decreasing over time, most recently the H value has been frequently below the 0.5 barrier.

There is a stripe, for values of H greater than 0.5, that crosses all scales in 1997. The events 1997 presented in Table 5.1 are the Asian Tigers crashes.

More recently we have another stripe that starts for short scales around September 2001. A small scale stripe showed in the third quarter of 2003.

5.2.2.3. GSTPSE (Canada)

As seen in Figure 5.4, the market shows two distinct periods, before and after 1997. Before 1997 we see high values of Hurst exponent over all scales. After that time, all the regions of high Hurst exponents are bounded in time and the background turns out to be what we expect from a mature market, with the Hurst exponent around 0.5.

There are two red stripes after 1997 that cross all scales, one in 1998 and another starting around September 2001 and travelling forward for higher scales in time.

5.2.2.4. Bovespa (Brazil)

Bovespa, the São Paulo Stock Exchange Index, is known for its high volatility and is generally considered an emergent market. In Figure 5.5 we see an erratic behaviour with

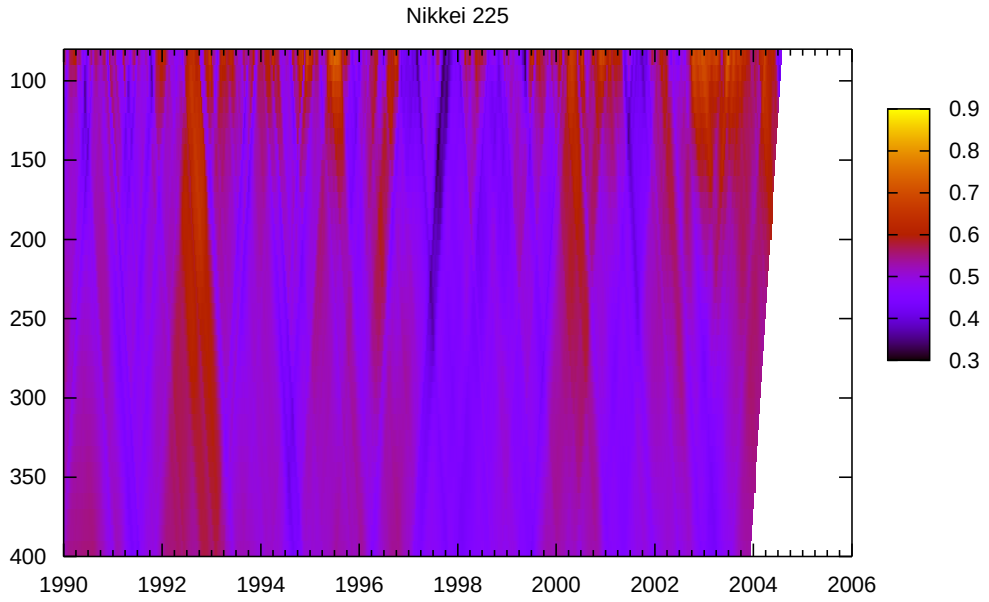


Figure 5.2.: TSDFA applied to Nikkei 225. The scale (in trading days) is represented by the y axis; the time is represented in x axis (years).

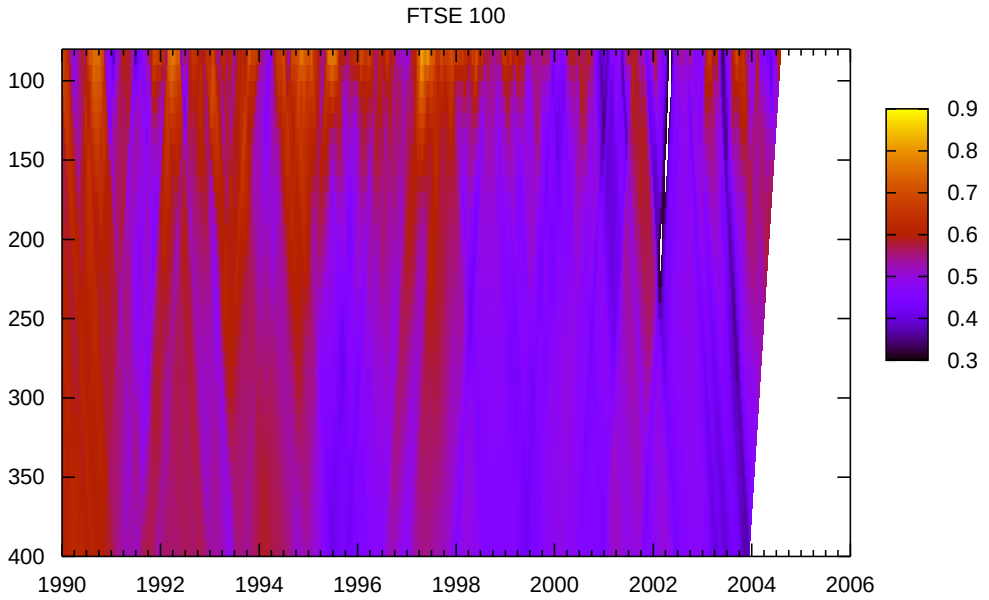


Figure 5.3.: TSDFA applied to FTSE.

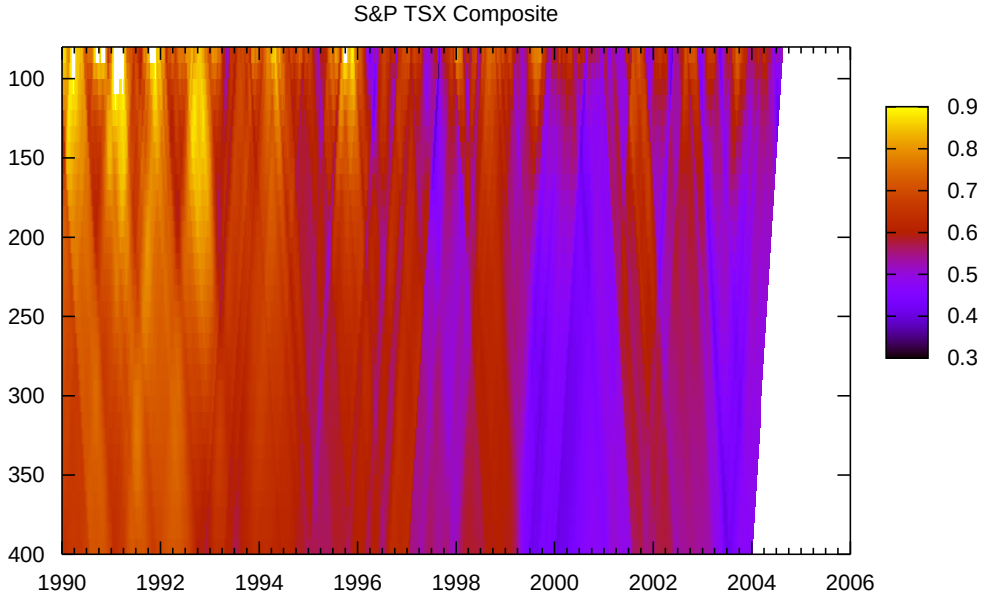


Figure 5.4.: TSDFA applied to GSTPSE.

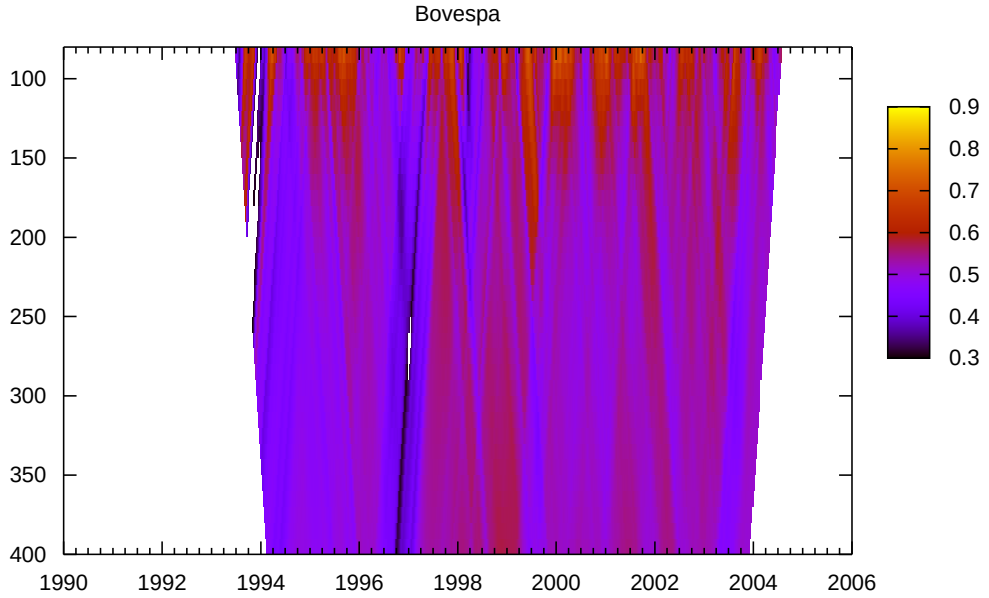


Figure 5.5.: TSDFA applied to Bovespa.

H either near or above 0.5 and the corresponding stripes crossing together, back and forward in time, at all scales. There are two red stripes that start from short scales respectively in 1997 (Asian crashes) and 1998 (global crash) which merge for large scales. There is another red stripe that walks through all scales and starts for short scales around September 2001.

5.2.2.5. PSI-20 (Portugal)

Unmodified DFA, the predecessor of TSDFA, was applied to PSI-20 in Chapter 4.1. In Figure 5.6 we see the results of applying TSDFA to this market, from establishment of series in 1993.

Initial stages are both antipersistent and subject to extreme values of the Hurst exponent. Comparing this graphic with Table 4.1, we recover the blue stripes, $H \simeq 0.5$, for the same dates found there. This graphic allows us to identify the time of the second and third stripes and as being the same but with then "travelling" in opposite directions in time when going to higher scales. We can identify two stripes with a stable (higher) value of the Hurst exponent, during 1998, and another walking forward in time starting, for short scales, next to September 2001. Notice that this stripe is so strong that it overlaps other stripes forming in the neighbourhood.

The overall strength of the TSDFA is to provide further conclusions over those drawn

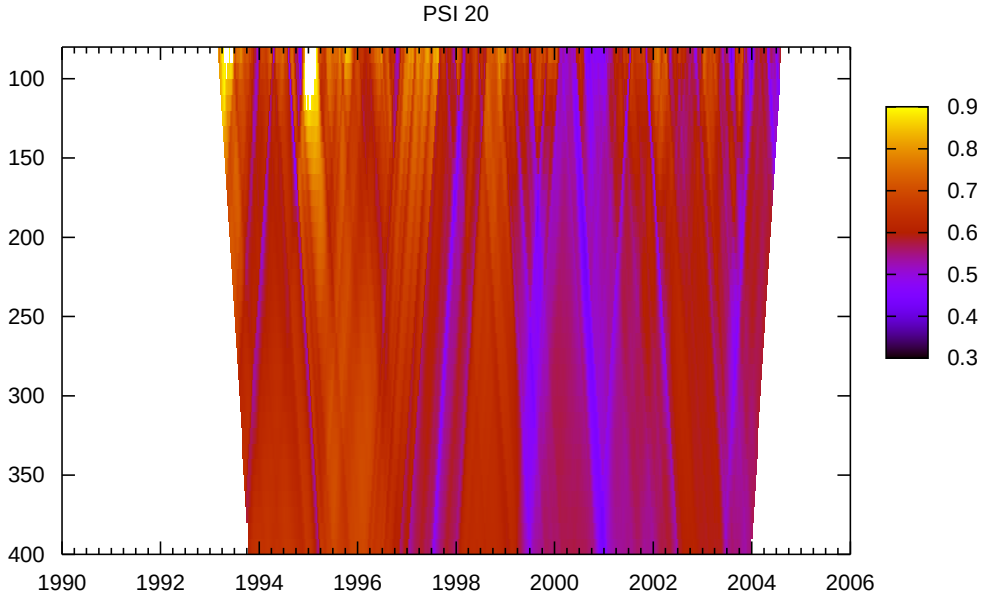


Figure 5.6.: TSDFA applied to PSI-20.

earlier concerning the market progression to mature behaviour and its responses times, clearly different from the initial position.

5.2.3. Features

As can be seen in Figure 5.2 there are several notable features of the plots produced by TSDFA. The support for these claims is reinforced with the results presented in Figures 5.3, 5.4, 5.5, 5.6, as well as with those presented in Appendix A:

- We can distinguish mature markets by the persistence and stability of H values around 0.5, most of the time.
- We can distinguish emergent markets by the persistence and stability of H values above 0.5.
- For some periods, a phase transition appears to occur, sometimes observable across all scales, sometimes across partial scales only. This is reflected in the spikes which either point to lower or to large scales;
- *A priori*, we expected smooth variations of H for large scales since we are taking into account more data values and therefore we expect greater robustness to sudden

changes of the data. This was already observed in the results obtained for PSI-20, (Figure 4.8) and is confirmed by all the examples.

- Markets evolve in time, the Canadian case is a notable example of this, where we observe a shift from emergent to mature features. Although not so dramatic for all other cases we see over time a decrease in the values of the Hurst exponent.

- There are events that change the Hurst exponent behaviour that can be seen in most/all markets. The September 11th 2001 is the most striking case that can be seen in all Figures, as discussed in each of them as it can be seen for all other markets in Appendix A.

- Clearly, the behaviour is dependent both on time and scale, indicative of the multifractal background, so that details obtained are richer than those obtained by calculation of the Hurst exponent directly. This is to be expected since the Hurst exponent is a summary measure, or index, of the data and this is the observed behaviour for financial markets (see Lux [2004]).

5.3. Results

5.3.1. Data

All the data on the respective market indices is public and came from Yahoo Finance (`finance.yahoo.com`). We have considered the daily closure as the value for the day, to obviate any time zone difficulties.

The choice of the markets used in this study was driven by the goal of studying major markets across the world in an effort to ensure that tests and conclusions could be as general as possible. Hence Table 5.2 contains some of the more important worldwide markets as well as new markets representing all continents.

In table 5.2 we summarise the markets used in this study. Data and summary statistics (index value, returns, parameters for Lévy stable distribution) on the markets studied are recorded and are presented in Appendix A. We have considered, in this study, the major and most active markets worldwide from America (North and South), Asia, Africa, Europe and Oceania.

5.3.2. Traditional classification of market maturity

The classification of markets into mature or emergent is not a simple issue. The International Finance Corporation (IFC) uses income per capita and market capitalisation relative to GNP for classifying equitity markets. If either 1) a market resides in a low or middle-income economy, or 2) the ratio of the investable market capitalisation to GNP is low, then the IFC classifies the market as emerging, otherwise the classification is mature.

Abrev.	Index Name	Country	Region	Status
^AEX	AEX General	Netherlands	Europe	mature
^AORD	All Ordinaries	Australia	Oceania	hybrid
^ATX	ATX	Austria	Europe	emerging
^BFX	BEL-20	Belgium	Europe	emerging
^BSESN	BSE 30	India	Asia/Pacific	emerging
^BVSP	Bovespa	Brazil	America	hybrid
^CCSI	CMA	Egypt	Africa/Middle East	emerging
^CSE	All Share	Sri Lanka	Asia/Pacific	emerging
^DJI	Dow Jones	United States	America	mature
^FCHI	CAC 40	France	Europe	mature
^FTSE	FTSE 100	United Kingdom	Europe	mature
^GDAXI	DAX	Germany	Europe	mature
^GSPC	500 Index	United States	America	mature
^GSPTSE	S&P TSX Composite	Canada	America	hybrid
^HSI	Hang Seng	Hong Kong	Asia/Pacific	emerging
^IPSA	IPSA	Chile	America	emerging
^ISCI	ISEC Small Cap	Ireland	Europe	emerging
^ISCT	ISEC Small Cap Techno	Ireland	Europe	emerging
^ISEQ	Irish SE Index	Ireland	Europe	emerging
^IXIC	Nas/NMS Composite (Nasdaq)	United States	America	mature
^JKSE	Jakarta Composite	Indonesia	Asia/Pacific	emerging
^KFX	KFX	Denmark	Europe	emerging
^OMXC20	OMXC20	Denmark	Europe	mature
^KLSE	KLSE Composite	Malaysia	Asia/Pacific	emerging
^KS11	Seoul Composite	South Korea	Asia/Pacific	mature
^KSE	Karachi 100	Pakistan	Asia/Pacific	emerging
^MERV	MerVal	Argentina	America	emerging
^MIBTEL	MIBTel	Italy	Europe	emerging
^MTMS	Moscow Times	Russia	Europe	hybrid
^MXX	IPC	Mexico	America	emerging
^N225	Nikkei 225	Japan	Asia/Pacific	mature
^NYA	NYSE COMPOSITE INDEX	United States	America	mature
^NZ10	NZSE 10	New Zealand	Oceania	hybrid
^OSEAX	OSE All Share	Norway	Europe	emerging
^PSI20	PSI 20	Portugal	Europe	emerging
^PSI	PSE Composite	Philippines	Asia/Pacific	emerging
^PX50	PX50	Czech Republic	Europe	emerging
^SETI	SET	Thailand	Asia/Pacific	emerging
^SMSI	Madrid General	Spain	Europe	hybrid
^SSEC	Shanghai Composite	China	Asia/Pacific	emerging
^SSMI	Swiss Market	Switzerland	Europe	hybrid
^STI	Straits Times	Singapore	Asia/Pacific	emerging
^OMXSPI	Stockholm General	Sweden	Europe	mature
^TA100	TA-100	Israel	Africa/Middle East	emerging
^TWII	Taiwan Weighted	Taiwan	Asia/Pacific	emerging
^XU100	ISE National-100	Turkey	Europe	emerging

Table 5.2.: Markets studied.

Two examples where this classification is not always followed are `http://globaledge.msu.edu/ibrd/marketpot.asp` where where Hong Kong and Singapore are considered as emerging markets and `http://www.msci.com/equity/indexdesc.html` where Austria, Belgium, Denmark, Finland, Greece, Hong Kong, Ireland, Norway, Portugal, Singapore are considered as developed markets.

5.3.3. Classification of global markets (TSDFA)

It seems clear from the results that we can distinguish different markets classes. The difference in behaviour is visible with the application of TSDFA. The most active, and mature, markets show a persistence of behaviour near $H = 0.5$ while the newer, emergent, markets show a persistence of higher values of H. The diversity of behaviours does not stop here, there are markets which show an hybrid behaviour between these two states.

The classification that we propose has thus three states:

(clearly) mature these market have a persistence of H around 0.5. The presence of regions with higher values of H is limited to small periods and is well defined both in time and scale.

(clearly) emergent these market have a persistence of H well above 0.5. The presence of regions with values of H around 0.5 is well defined both in time and scale.

hybrid unlike the two previous case the distinction between the mature and emergent phases is not well determined, with the behaviour seemingly mixing at all scales.

This classification is in agreement with another based on wavelet analysis proposed in Sharkasi et al. [2006b].

We have taken this classification and have applied it to the markets present in Appendix A where the markets are grouped according to it.

5.4. Conclusions

Our results, presented here and in Matos et al. [2006b], clearly show that the differences between worldwide markets can not simply be reduced to the simple distinction between emerging and mature markets. In some cases, an evolution or a change of regime from one state to the other can be seen clearly from this analysis.

There are certain events that are clearly reflected in all markets, as expected since most events are due to external causes, and thus independent of the specific market. One event where this is clearly seen is the 9/11 (September 11th 2001) attack against the World Trade Center towers (NY). In all the markets this is clearly seen, both in markets present here and in Appendix A, where the same type of analysis reveals the same dominant stripe appearing around September 2001.

In general our results show also that mature markets tend to absorb shocks more quickly and to learn with them. For hybrid markets, the borders are not so well defined. We see entangled stripes of mature and emergent behaviour, while for mature and emergent markets the regions are better resolved with a clear dominance of mature or emergent behaviours, respectively.

A trend common to most markets (mature and emergent) is the progressive "maturation", i.e. $H(t, s)$ has been decreasing over time for most of the studied markets. One possible reason to this is the progressive globalisation of markets, where the arbitrage opportunities are reduced thus producing more efficient markets.

6. Entropy Measures

"*If knowledge can create problems, it is not through ignorance that we can solve them.*" - Isaac Asimov

6.1. Introduction

The subject of this Chapter was previously explored in Matos et al. [2006a] and is expanded here.

Two techniques, not previously used in this thesis, are applied here: entropy (Section 6.2) and covariance matrices (Section 6.3). The purpose of this analysis is to search for signs of coherence and/or synchronisation across the set of studied global markets. The use of covariance matrices generalises the techniques applied previously to a multivariate framework, the study of several stochastic processes at once. We use the covariance matrices also to study the dependence of the results on the time granularity considered.

The purpose of entropy and coherence techniques is to examine the market behaviour. We apply econophysics techniques related to measures of "disorder"/complexity (entropy) and also discuss the relation between those results and the results from TSDFA, studied in the previous Chapter.

Finally, in Section 6.4 we present the conclusions.

6.2. Entropy

We have applied the Shannon entropy for blocks of size 5 and an alphabet of 50 symbols, as described in Section 2.9, to a set of markets previously studied. We should recall that using blocks of size 5 corresponds to a week in trading time. Notice also that we have only considered trading days, like what we do in all other analysis, so we ignore any holidays or days where the market was closed.

It should be noted that results are robust to the choice of the total number of bins (the size of our alphabet). That is, we have repeated the analysis with a different choice of the number of partitions yielding similar results.

In order to enhance the time dependence of results we have evaluated the entropy of the set for periods of 100 trading days (roughly corresponding to half a year). The motivation for this analysis is the same used in Chapters 4 and 5, to study the time evolution of entropy.

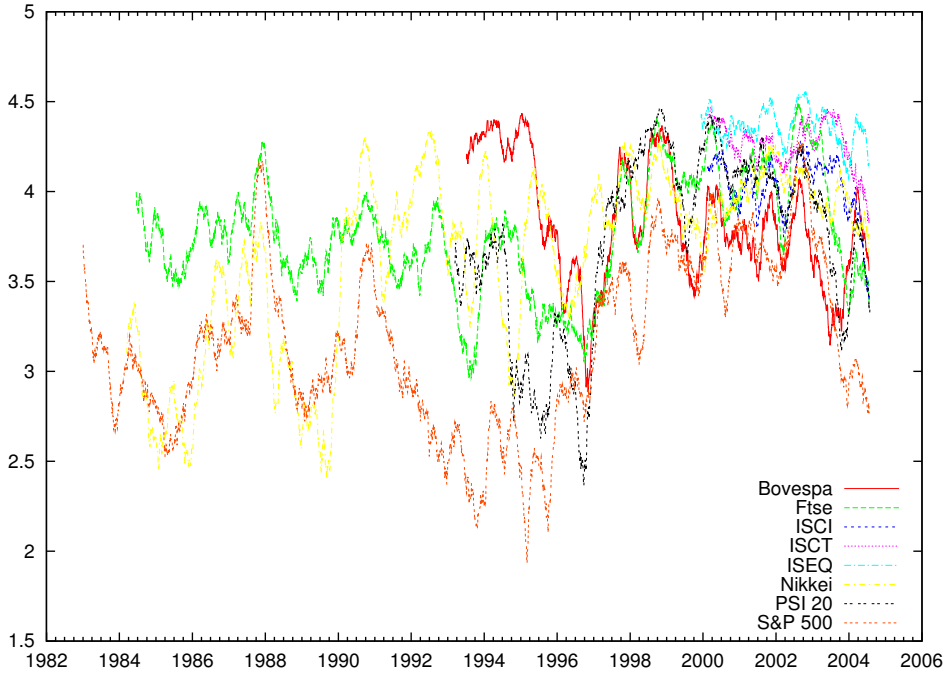


Figure 6.1.: Weekly entropy for various market indexes.

The results displayed in Figure 6.1 show improved coherence (i.e. reduced entropy) after 1997 as compared with previous periods for all markets. Higher entropy implies less predictability, in general, although the nature of shocks qualifies this statement to some extent. The notable feature of this graphic is that both mature and developing markets are affected similarly which suggests that global behaviour patterns are becoming more coherent or linked because of the progressive globalisation of markets. This is in line with the findings of Chapter 5 where we found the Hurst exponent for different markets to be decreasing with time.

6.3. Covariance matrices

In the previous section we have used the block entropy applied to several markets. The analysis of co-movements suggested a multivariate analysis. This method shares with block entropy applied in the previous Section the emphasis on time dependency.

The work explored here was developed, by the author and collaborators, in Sharkasi et al. [2006a]. We use the covariance matrix to study the coherence of various set of markets, with different degrees of maturity, (for this study we have considered the traditional distinction between mature and emerging markets as the initial point). We are interested in the time dependency of the (three) most significant eigenvalues of the covariance matrix, since as seen in Section 2.10, those are the only eigenvalues which carry meaningful information.

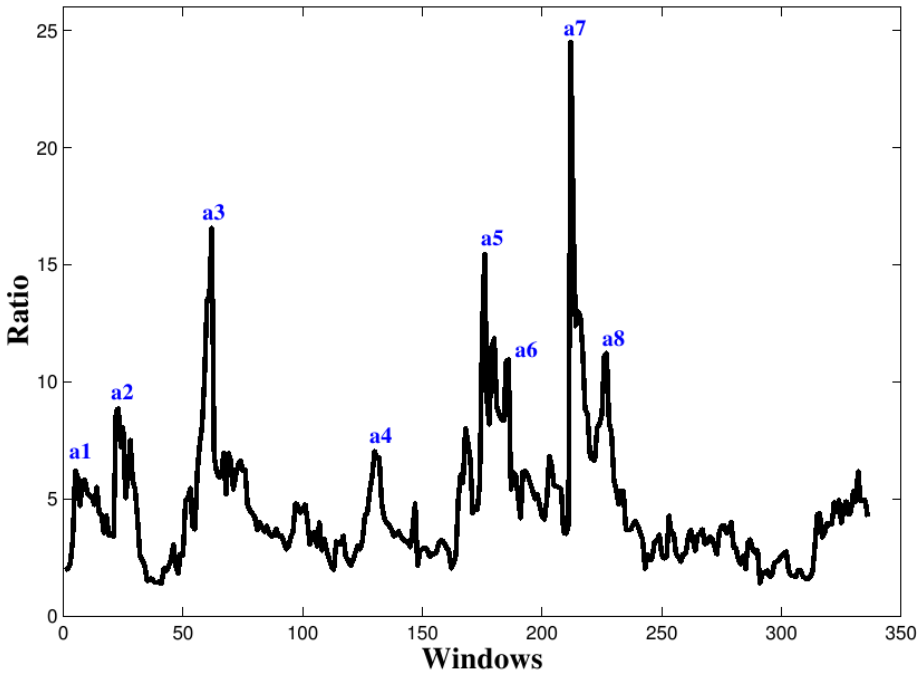


Figure 6.2.: Evolution of for $\frac{\lambda_1}{\lambda_3}$ emerging markets.

We have use the covariance matrix as defined in Section 2.10. We have used the typical value of parameters, $R = 0.9$ and an horizon of 20 trading days, (for details see Litterman and Winkelmann [1998]). In line with the analysis of the previous Section, weekly periods have been used to estimate the returns.

In Figure 6.2, we represent the ratio between the first and the third most important eigenvalues ($\frac{\lambda_1}{\lambda_3}$) for a given set of emerging markets. The same analysis applies for mature markets, see Figure 6.3.

Again, interest lies in the fact that spikes in Figures 6.2 and 6.3 correlate with real events, as summarised in Tables 6.1 and 6.2, respectively.

We have considered the evolution of the major eigenvalues assuming weekly data. Ap-

Mark	Window No	Last week included	Events
$a1$	5	first week of 7/1997	Asian Crash
$a2$	23	second week of 11/1997	Asian Crash
$a3$	62	fourth week of 8/1998	Global Crash
$a4$	130	second week of 1/2000	
$a5$	176	second week of 12/2000	Effects of DotCom Crash
$a6$	186	second week of 3/2001	
$a7$	212	second week of 9/2001	September 11th Crash
$a8$	227	fourth week of 1/2002	

Table 6.1.: Table of events (emerging).

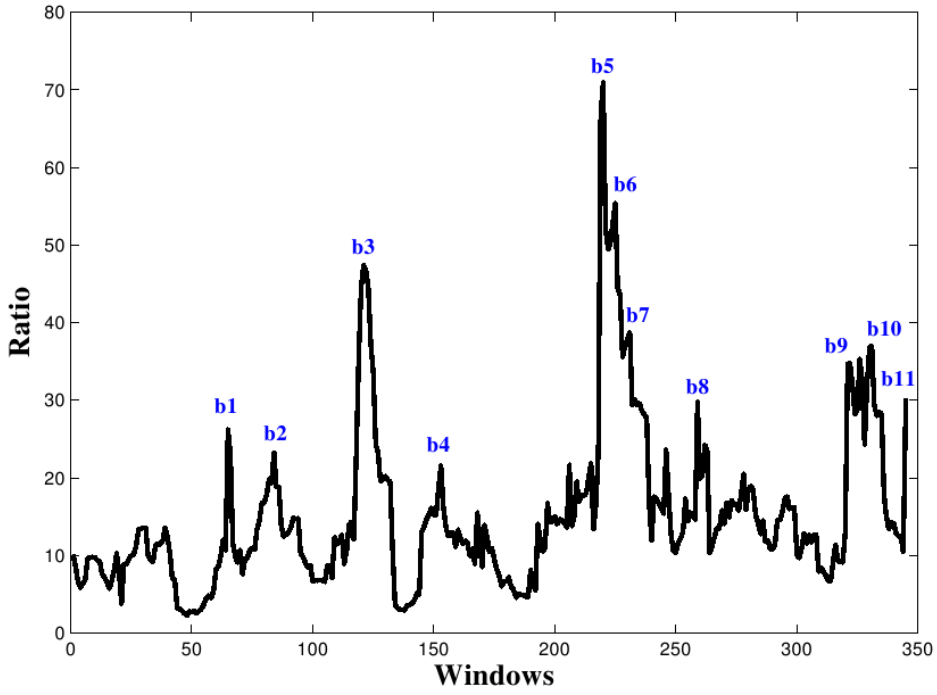


Figure 6.3.: Evolution of $\frac{\lambda_1}{\lambda_3}$ for mature markets.

Mark	Window No	Last week included	Events
$b1$	65	first week of 9/1998	Global Crash
$b2$	84	fourth week of 12/1998	Global Crash
$b3$	121	third week of 10/1999	Last October in the 20th Century
$b4$	153	second week of 6/2000	DotCom Crash
$b5$	220	second week of 9/2001	September 11th Crash
$b6$	225	first week of 11/2001	Effects of 9/11 Crash
$b7$	231	second week of 12/2001	Effects of 9/11 Crash
$b8$	259	first week of 5/2002	The Stock Market Downturn
$b9$	322	first week of 10/2003	
$b10$	331	first week of 12/2003	General Threat Level Raised
$b11$	345	third week of 3/2004	Madrid Bomb

Table 6.2.: Table of events (mature).

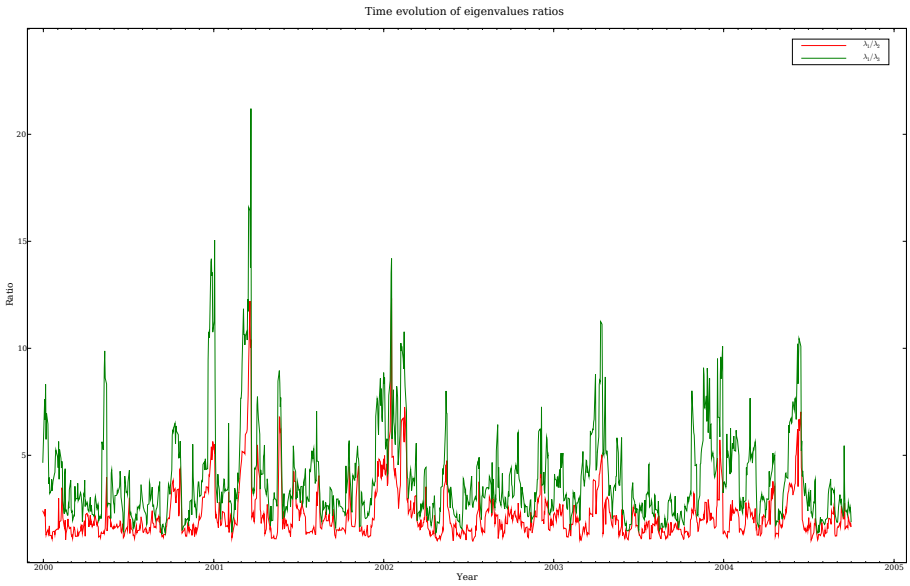


Figure 6.4.: Evolution of eigenvalue ratios for emergent markets (daily data).

plying the same analysis for daily data we get the results displayed in Figures 6.4 and 6.5. This analysis highlights the role of the data granularity, the coarse grained approach, in the results. We have a better resolution on the events and the results are qualitatively the same.

6.4. Conclusions

We have focused on aspects of time dependence, explored by several econophysics techniques, applied to markets, categorised as emerging or mature and subject to diverse levels of disorder or volatility in their financial series. The outcome shows clear synchronisation of world markets, observed in the weekly entropy of individual markets or groups. The results show that world markets tend to influence each other and reduce individual market levels of disorder (i.e. reduced entropy) demonstrating a clear synchronism of responses which is more or less robust depending on the nature of the market. The entropy measure here is considered over a week, a fairly long time in terms of market behaviour, but the results obtained for daily results show the same qualitatively behaviour.

Despite evidence that stability is linked to this synchronisation and low energy or equilibrium state, it is evident that shocks upset the balance and disorder increases with very high entropy levels in some instances. These occurrences correspond usually to crashes in markets, as it can be seen associating the events in Tables 6.1 and 6.2 with their corresponding spikes in Figures 6.2, 6.3, 6.4 and 6.5. Nevertheless it is a characteristic of the more mature markets that this period of increased entropy is relatively short, with

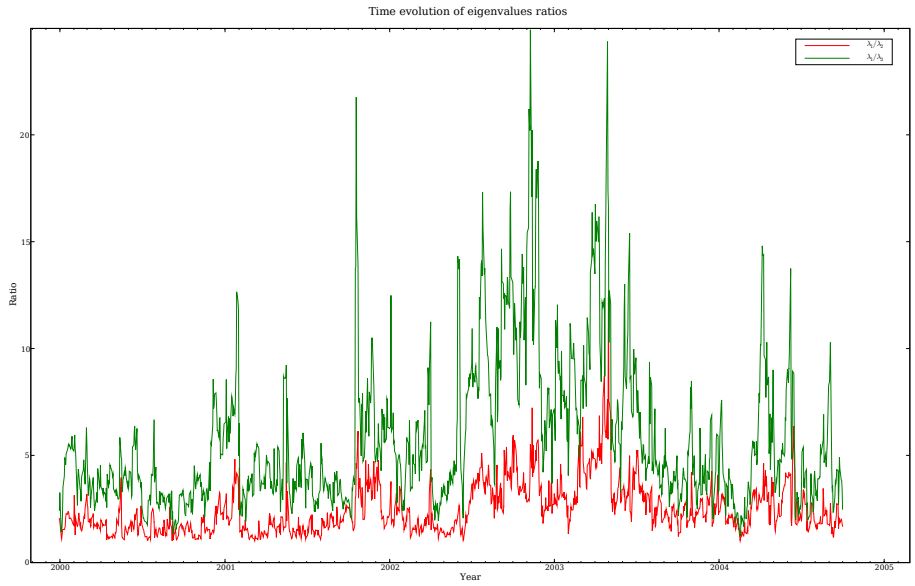


Figure 6.5.: Evolution of eigenvalue ratios for mature markets (daily data).

smaller recovery times. See both in Figures 6.2 and 6.4 how it takes almost two months for emerging markets to reflect 9/11 effects while for mature markets (Figures 6.3 and 6.5) this effect is instantaneous.

This distinction is not always clearcut, however and under different conditions markets may exhibit more than one type of behaviour, (see in Tables 6.1 and 6.2 where for certain peaks we were not able to associate any known event).

7. Conclusions

"*Prediction is very difficult, especially about the future*" - Niels Bohr

In this work we have addressed the analysis of financial time series from an econophysical point of view. Financial data presents complex behaviour which needs to be decomposed effectively. After the first order approximation the granularity in time needs to be refined in order to determine the nature and persistence of the fluctuations observed. In particular, a number of techniques discussed are discussed and applied to the study of memory effects, the reaction of markets to internal and external perturbations in terms of sensitivity, recovering times and so on.

When studying stock markets there are two useful properties to measure, the uncertainty and the risk. These concepts are related to arbitrage, a trading strategy that generates profit without risk, from a zero initial investment.

Entropy, described in Chapter 2, is a measure of uncertainty. This notion can be generalised to cover other techniques used in this work. Taking a slightly different perspective on the breakdown of financial signals into component elements we can consider several of the techniques studied in Chapter 2, (wavelets; multifractals; fBm; stable laws; entropy and time dependent covariance matrix), as entropy measures, the subject of this thesis.

The econophysics techniques applied in this work are twofold: measures of "disorder"/complexity and measures of coherence, (for a discussion of coherence and persistence in the scope of financial time series see Ausloos [2001]). These techniques are in a sense complementary, i.e. each provides a different view over the financial data studied, but they can be placed under the umbrella of entropy measures.

The measures of complex disorder are the entropy, as presented in Section 2.9, and the fractional Brownian motion, (see Section 2.7), using the generalisation defined in the last Chapter. The connection of fBm with entropy is simple. For values of H larger than 0.5 the increments are correlated and if we increase the exponent H that implies smaller uncertainty and thus smaller entropy. If we decrease H then entropy increases. Notice that these measures are not equivalent, fBm does not take into account the clusters of volatility while the entropy does, (for a discussion of entropy when compared with other uncertainty measures see Maasoumi and Racine [2002], McCauley [2003]).

Another measure of entropy, presented in Chapter 2, is multifractal analysis. The relation between entropy and multifractals can be established through D_1, the information dimension, or through the relation between the moments that we explore in both multifractals and the Rényi entropies, or Kolmogorov-Sinai entropies.

Wavelets, (Section 2.4), allow the decomposition of the signal into components for the same scale. This decomposition allows to study the entropy content carried by each scale detail. For multifractals this dependence is expressed in the moments, (this relation can carried further along, see Doukhan et al. [2003] for methods of evaluating multifractals using wavelets). For fBm we have explored a similar path examining the $H(s,t)$ dependence in the previous Chapter.

If entropy is disorder, implying lack of a common trading strategy, then coherence implies cooperative, or at least common tendencies in, behaviour. We use the covariance matrix, (see Section 2.10), as a measure of coherence among a closely related set of markets. Coherence can be either observed between each time series, like in TSDFA, or between different time series as we study in the covariance matrix analysis. Since world markets are correlated the entropy of the set of markets is smaller than the sum of individual entropies.

In Chapter 3 the emphasis is made on the use of Free Software and the repeatability of results. Appendices C and D, discuss into further detail the options chosen.

The first application of the techniques toolbox is PSI-20 (Portuguese Stock Index - 20), a Portuguese index of the 20 most liquid assets of the Portuguese Stock market.

Following the work on PSI-20 a new method is propose for studying the Hurst exponent, which includes investigation of both time and scale dependency. This approach permits the recovery of major events, affecting worldwide markets, (such as Sept. 11th 2001) and facilitates examination of the propagation of effects produced across different scales. Such effects may include early awareness, distinctive patterns of recovery, as well as comparative behaviour distinctions in emergent/established markets. The emphasis on time dependence serves to demonstrate the importance of entropy measures as snapshots of market uncertainty, which have their own dynamic.

We developed and applied a new technique, the TSDFA (Time and Scale Detrended Fluctuation Analysis), to study the time evolution of each market. Major features may include transition from a developing to a mature state, (International Finance Corporation definition). Comparing the results obtained using TSDFA to all markets, we identify groups that display similar behaviour at any given time. This classification allow us to distinguish perturbations with global or more general effect, (e.g. Asian tiger crash, 9/11, Madrid bomb attack in 2004 and others) from local influences affecting a small set of markets or even a single market only.

Interestingly, in spite of known differences between emerging and established markets, the evidence suggests that, in recent years, entropy measures are convergent across markets studied worldwide. This can be construed as an increasing number of markets achieving or mimicking mature behaviour relatively rapidly, irrespectively of their trading capability, which suggests that windows of opportunity are narrowing for investors since the arbitrage opportunities are reduced due to more efficient markets.

7.1. Future work

Variants of the techniques presented in this work were not explored but show potential for further studies. Those topics, with a small discussion, are raised next.

In this thesis all analyses were applied to market indices yet all the measures can be applied to individual assets.

As stated in the begin of Chapter 5, DFA is an implementation issue. An interesting variant is to repeat the analysis using wavelet estimation to determine $H(s,t)$.

The scale dependency can be further extended into comparing the detail levels from wavelet decomposition, instead of the whole time series, using the time dependent covariance matrix.

Finally, when studying the covariance matrix and its most significant eigenvalues, we could study the evolution of eigenvectors. This type of analysis should be useful to pick sudden jumps when the main eigenvectors changes suddenly, instead of smooth time dependency.

A. Classification of Global Markets

"*Ex nihilo nihil fit - Nothing comes out of nothing.*" - René Descartes

In this Appendix we classify each of the markets studied, (Table 5.2), according to the classification defined in Chapter 5. We present for each market studied:

- Country and name of the index
- Lévy parameter α and β. α is the exponent of the tail of the distribution and β is the asymmetry parameter, $\beta = -1$ means that the distribution is a left side, i.e. there is a value X such that for $P(x > X) = 0$. $\beta = 1$ means a right side distribution, similar to the previous case but on the other side. $\beta = 0$ means that the distribution is symmetric.
- Historical index values.
- Historical return values.
- TSDFA applied to the time series.

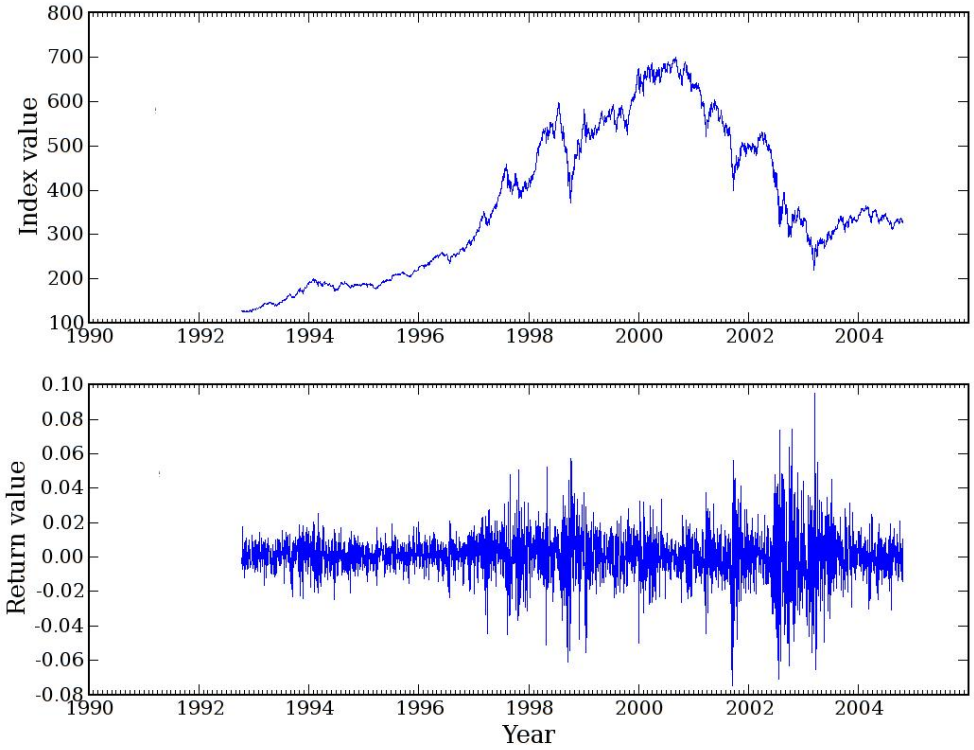
As previously described, all analyses deal with returns, as e.g. prices can be problematical due to currency exchanges. For each market therefore, we illustrate the original time series and the returns. The same scale is used for all plots to place comparisons in a context where they can be understood. These plots are for the entire length of the time series in each case and clearly cover a longer period for some countries compared to others. The plot of returns allows us to determine the volatility clusters.

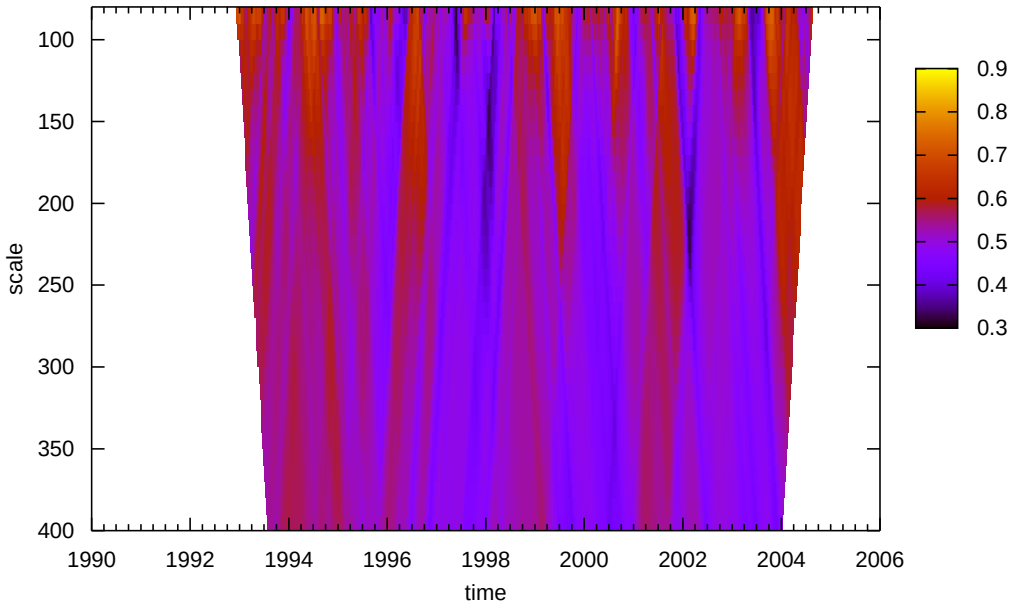
We also plot the resulting graphics of applying the TSDFA starting only at 1990. A rigid scale is used to ease the comparison between markets, of major features/events occurring during this period.

A.1. Mature

Netherlands (AEX General)

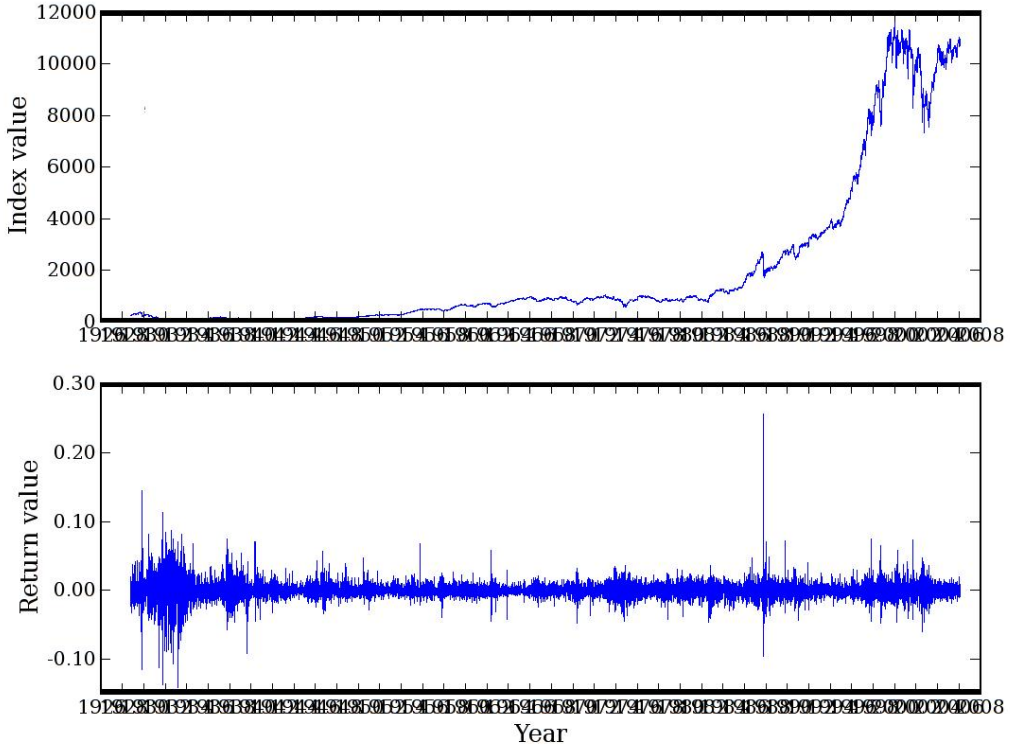
Lévy:$(\alpha, \beta) = (1.583, 0.146)$

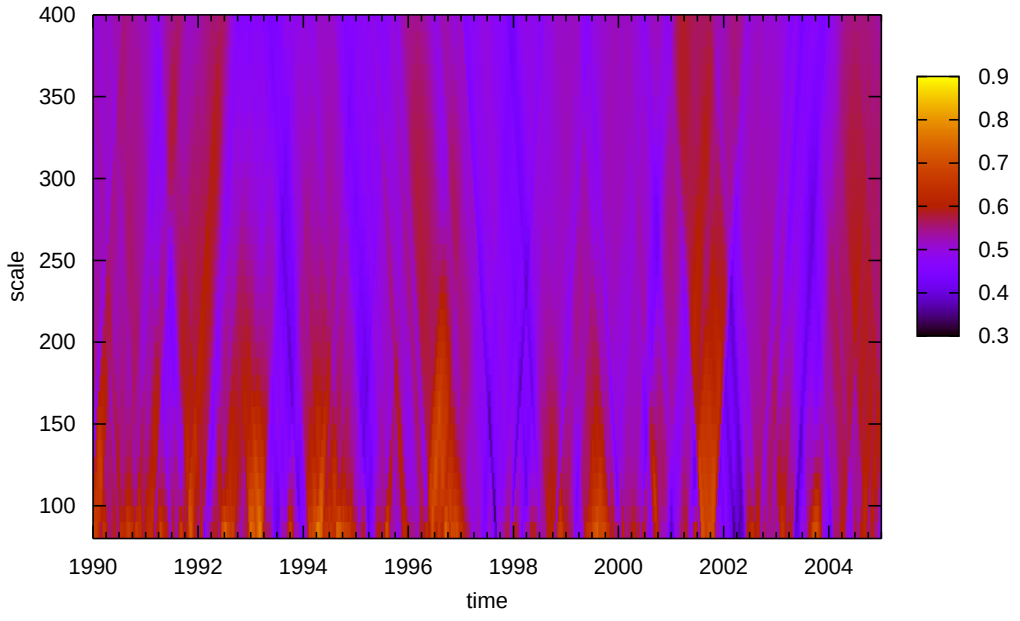




United States (Dow Jones)

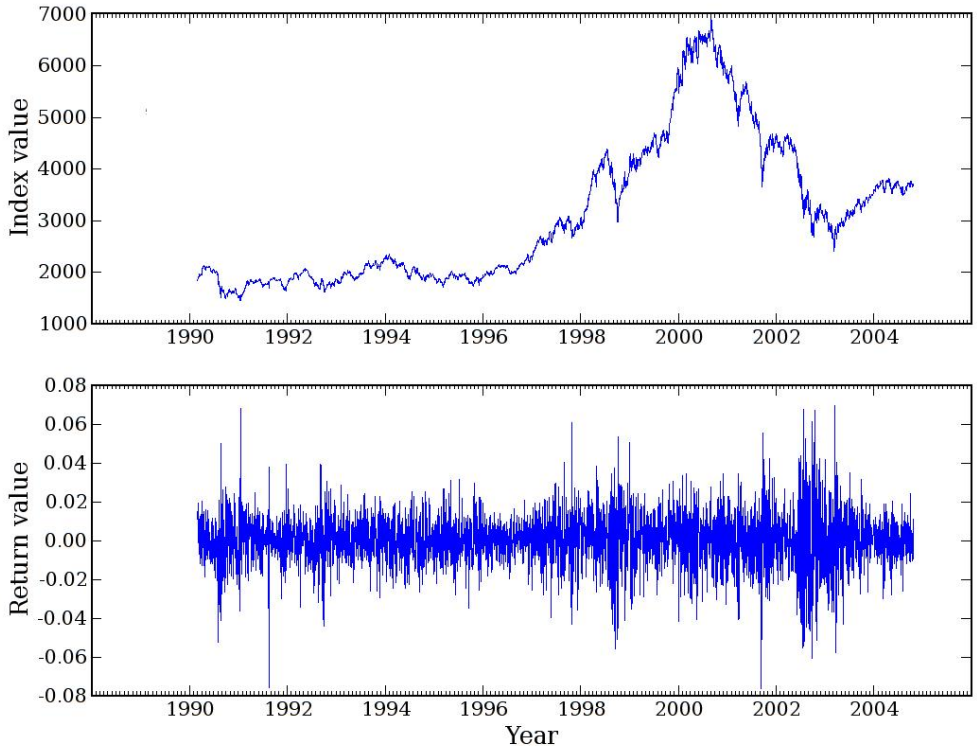
Lévy: $(\alpha, \beta) = (1.563, -0.079)$

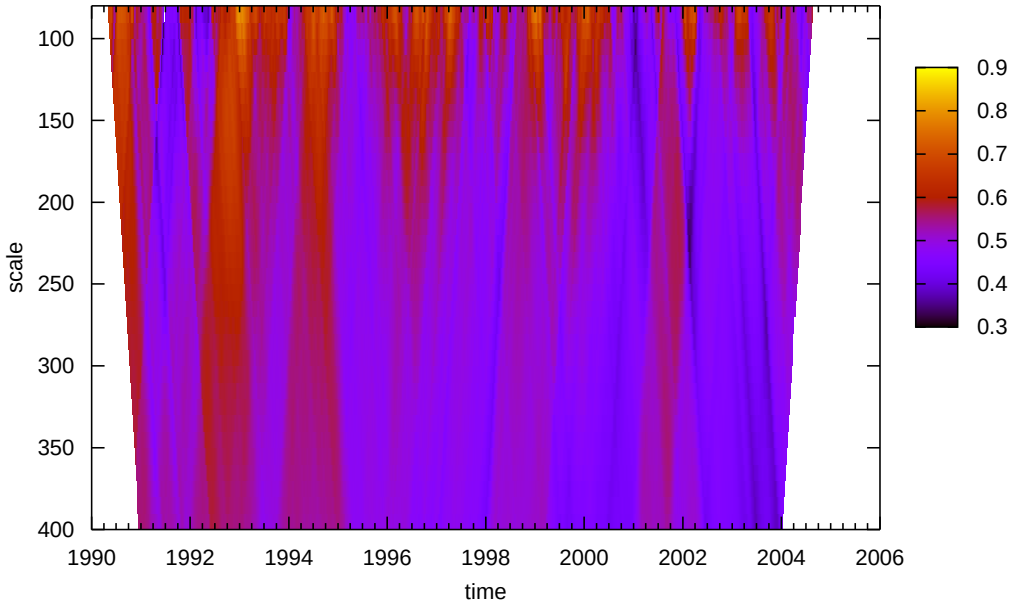




France (CAC 40)

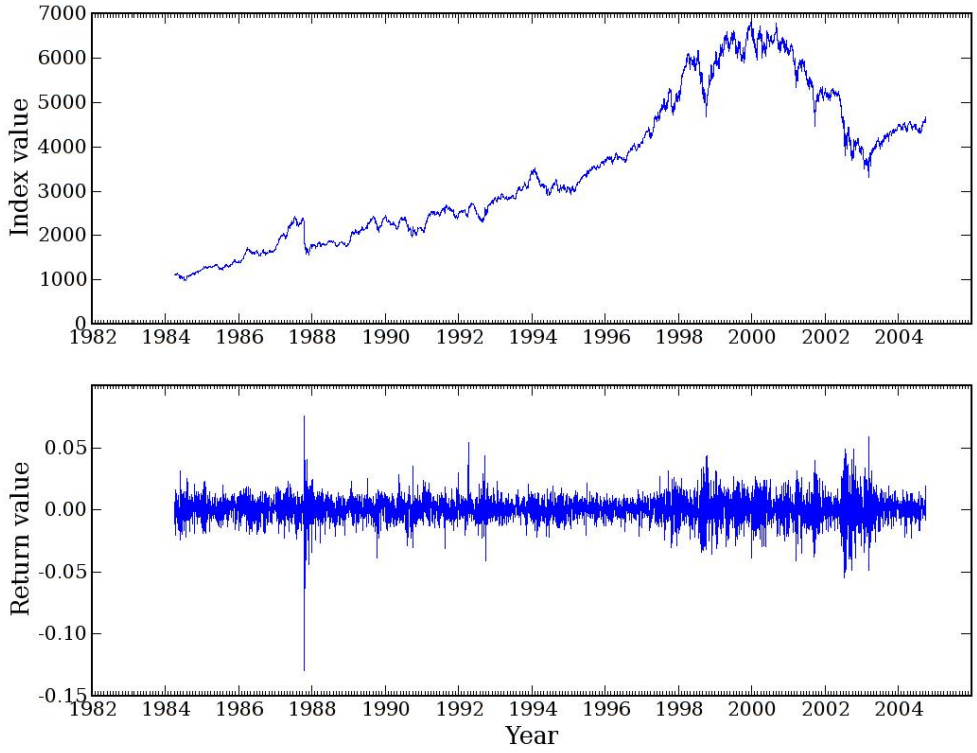
Lévy:$(\alpha, \beta) = (1.764, 0.143)$

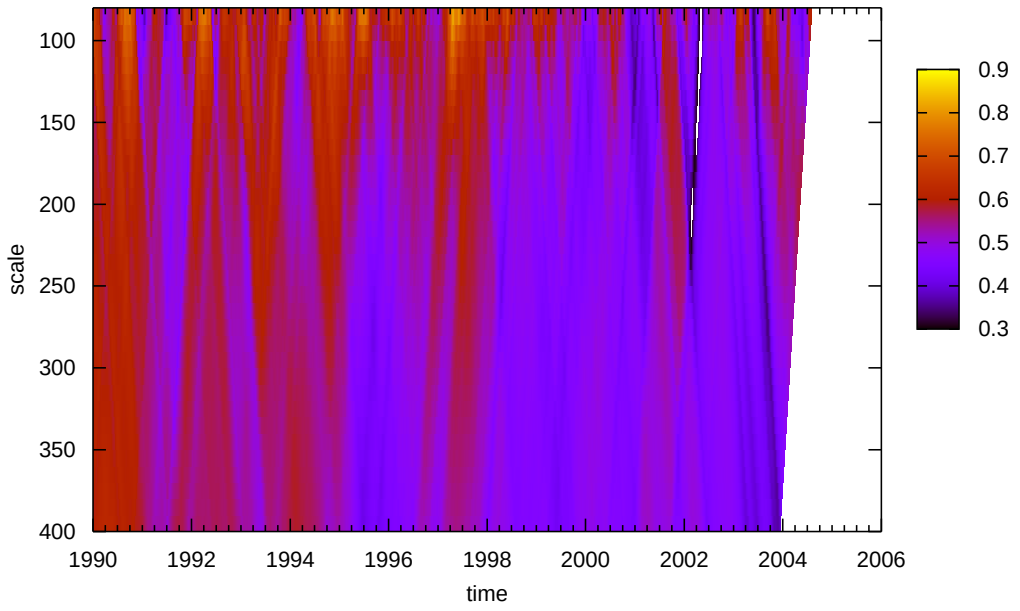




United Kingdom (FTSE 100)

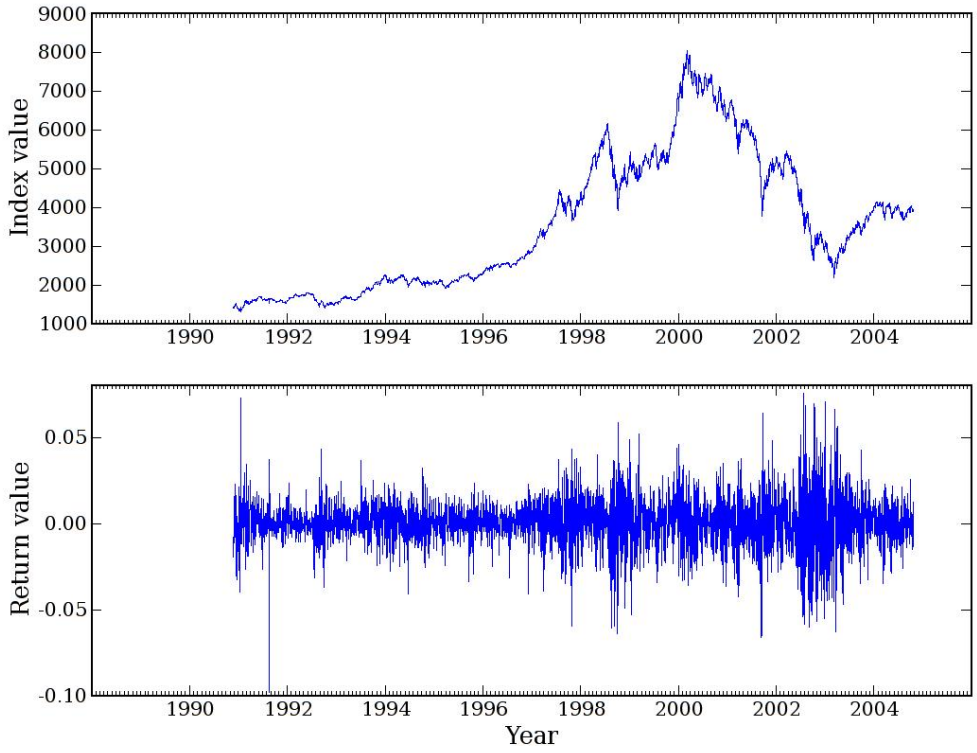
Lévy: $(\alpha, \beta) = (1.769, 0.001)$

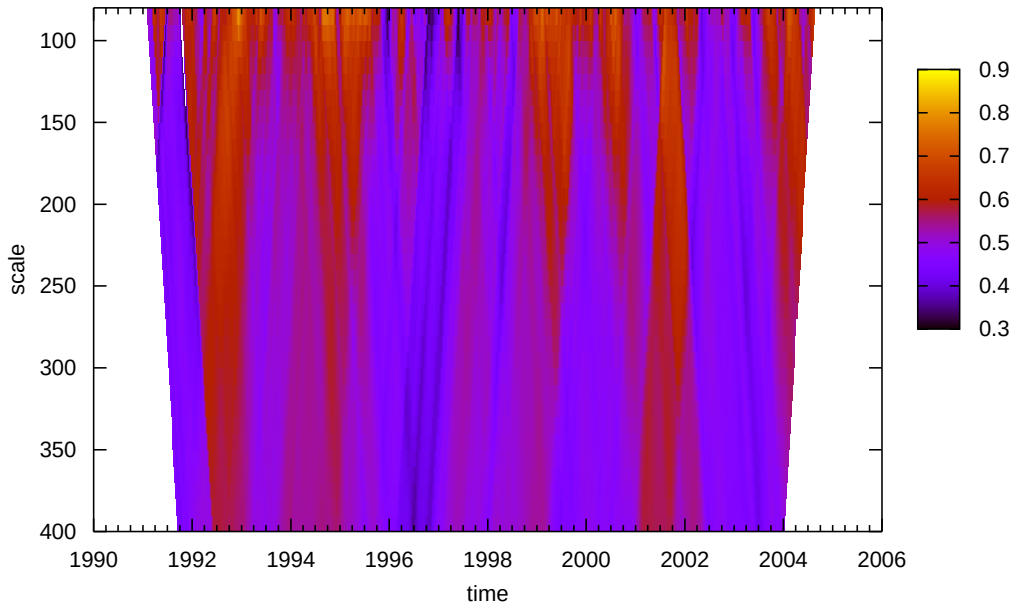




Germany (DAX)

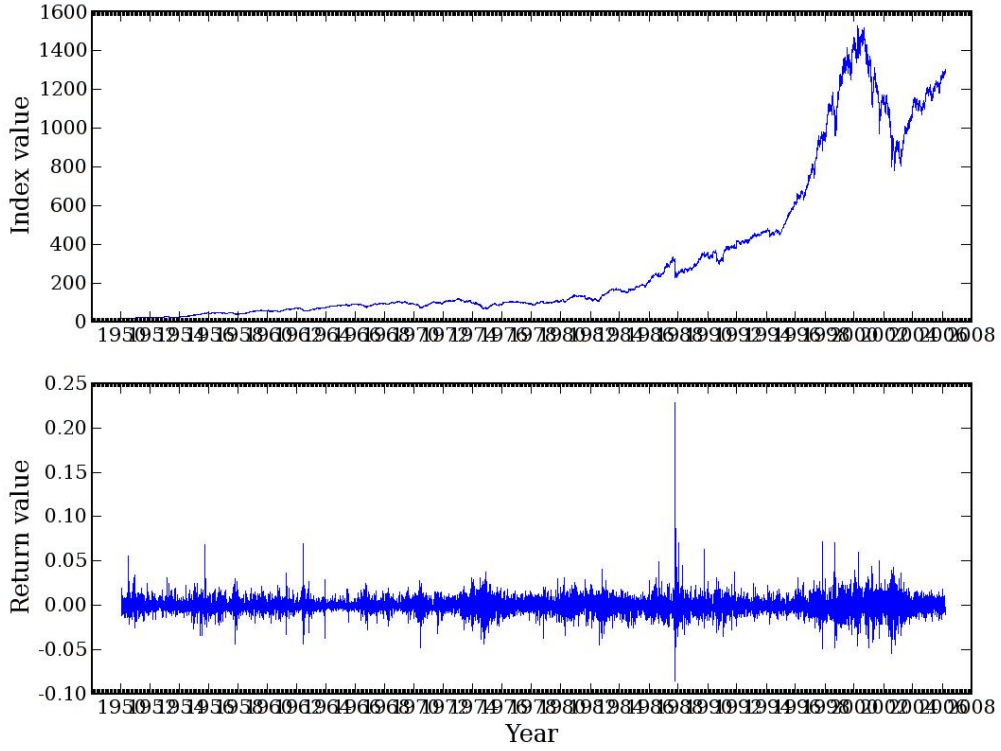
Lévy:$(\alpha, \beta) = (1.665, 0.142)$

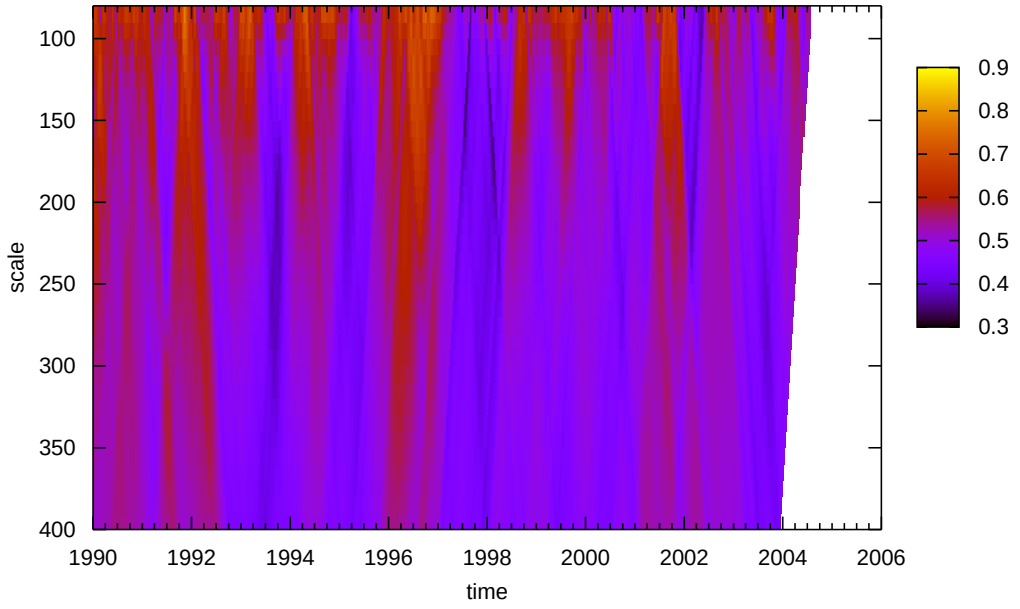




United States (500 Index)

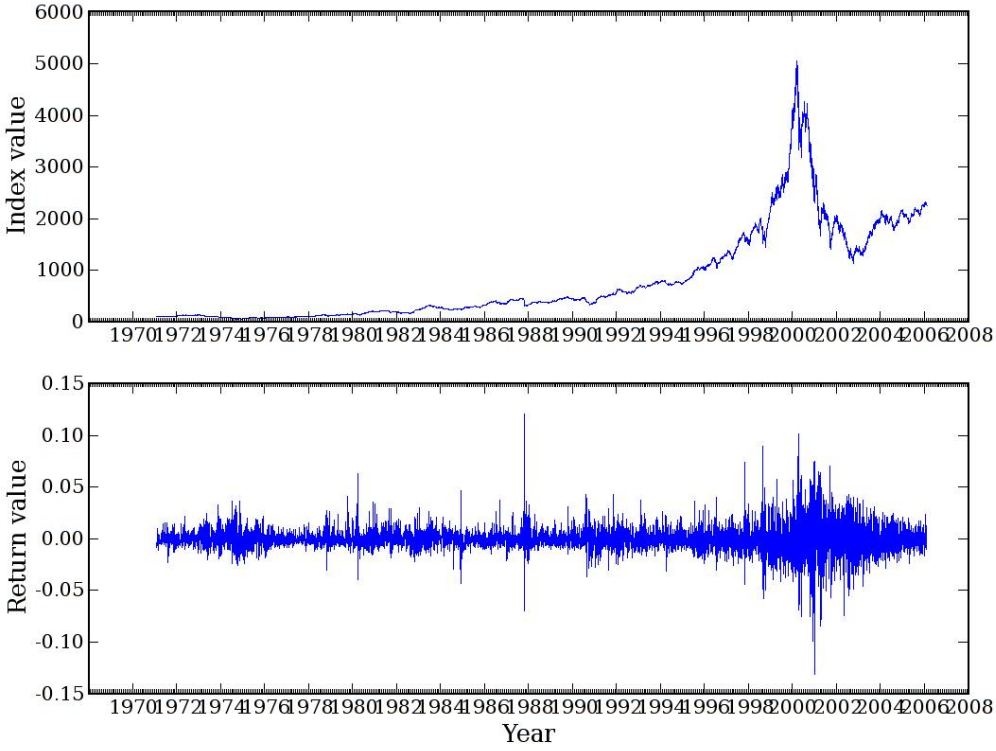
Lévy: $(\alpha, \beta) = (1.676, -0.085)$

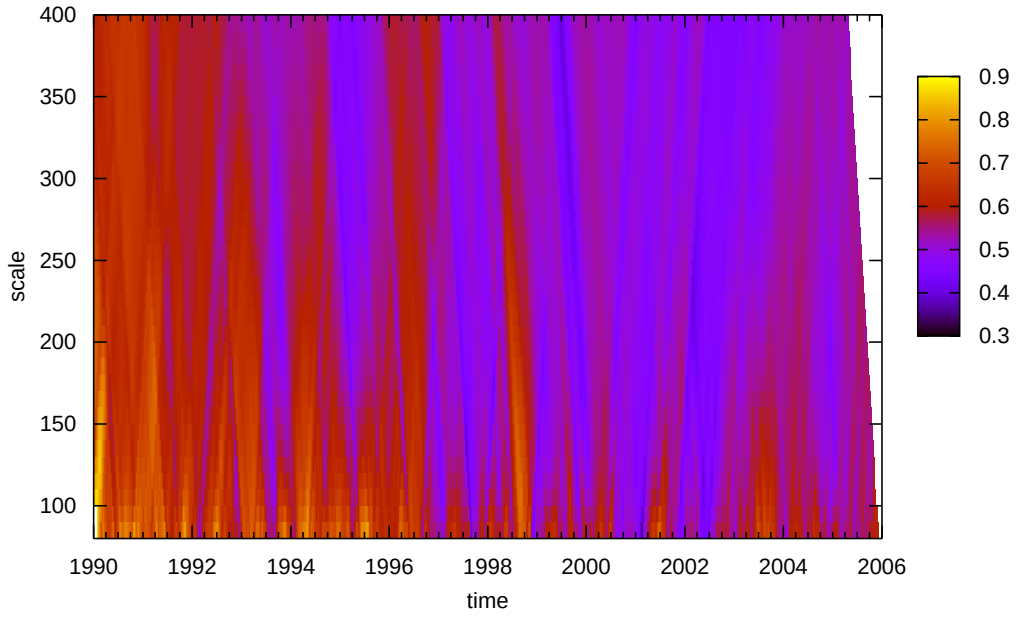




United States (Nas/NMS Composite (Nasdaq))

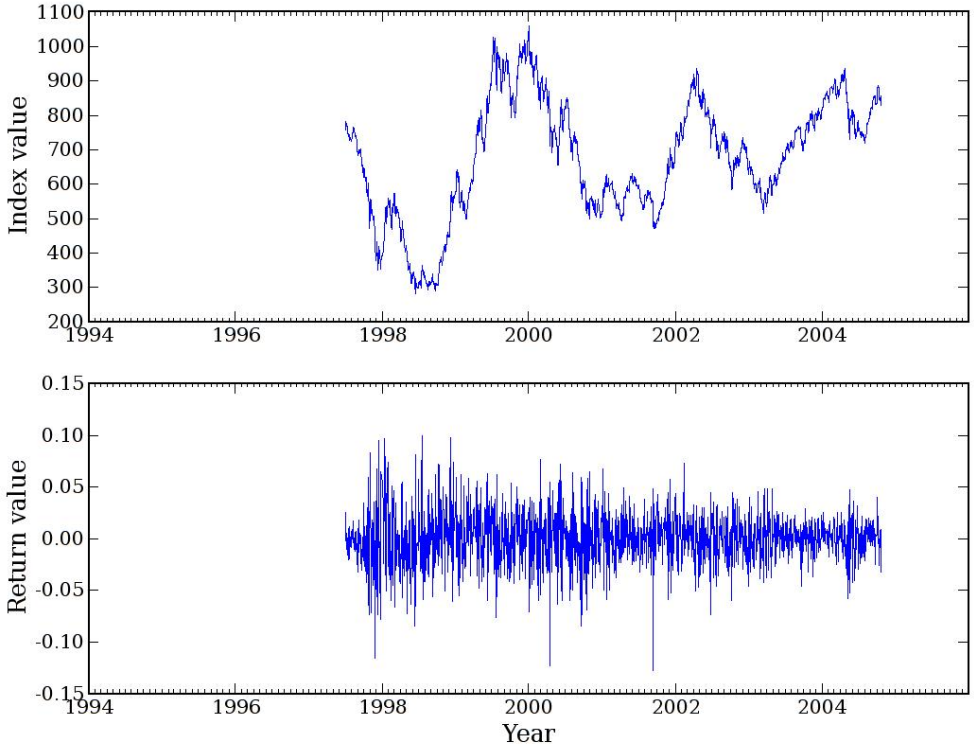
Lévy:$(\alpha, \beta) = (1.453, -0.243)$

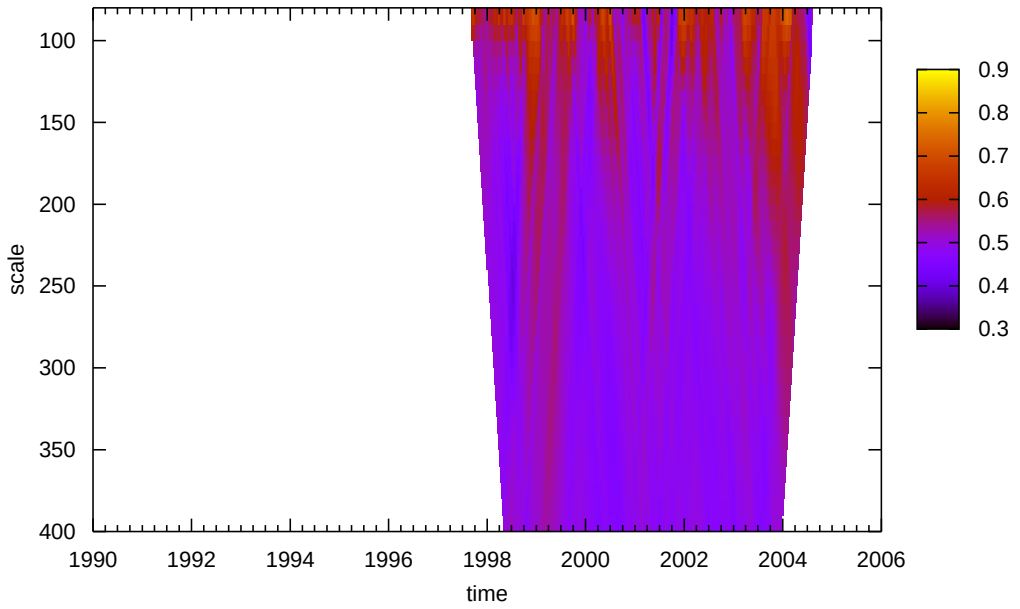




South Korea (Seoul Composite)

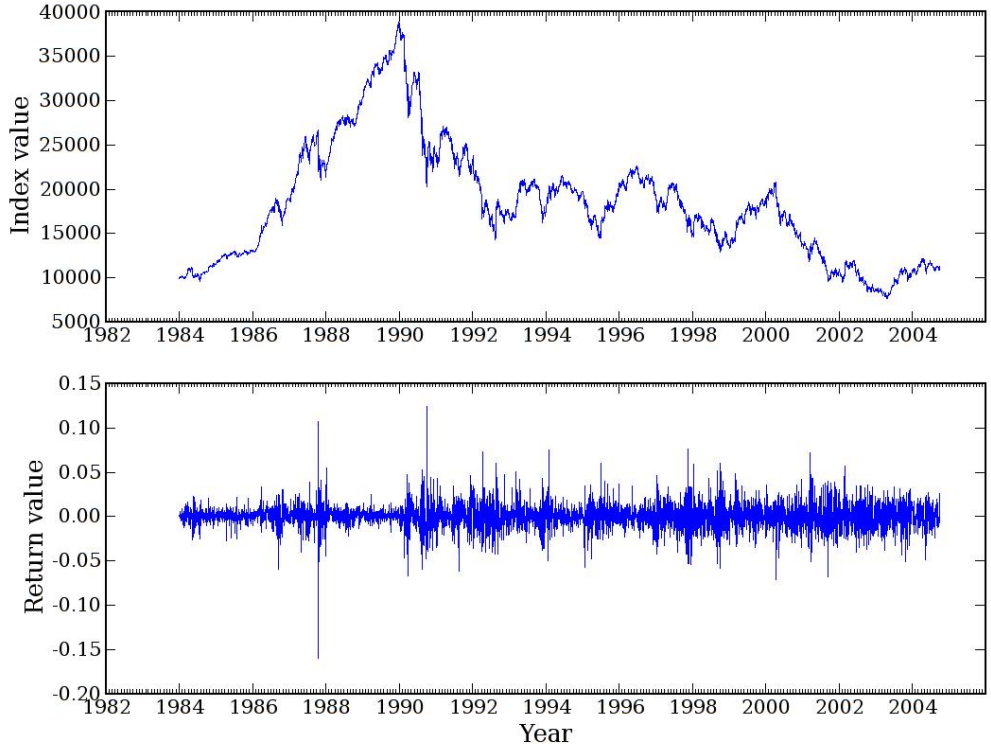
Lévy:$(\alpha, \beta) = (1.698, 0.004)$

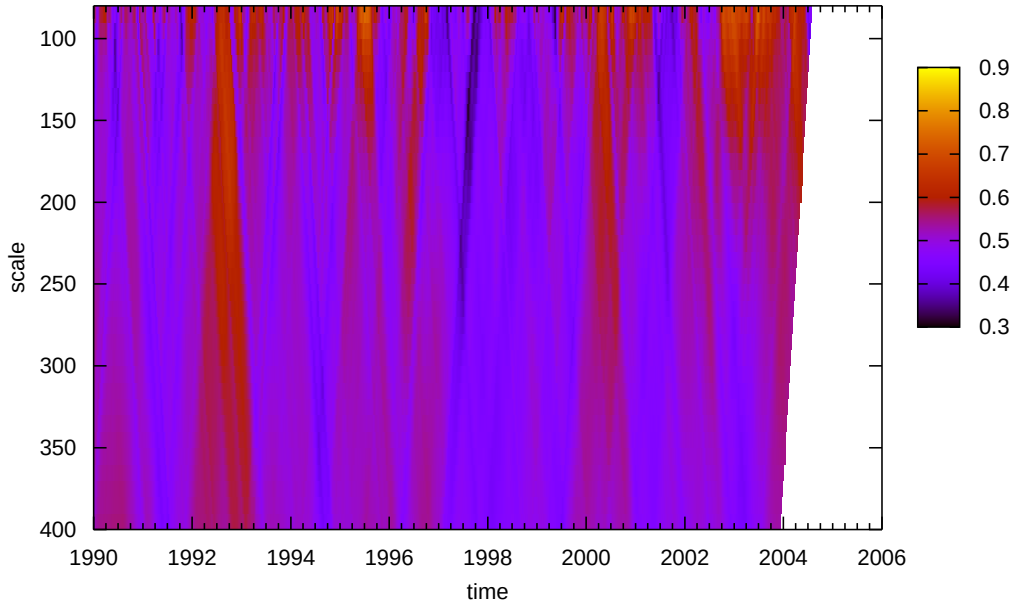




Japan (Nikkei 225)

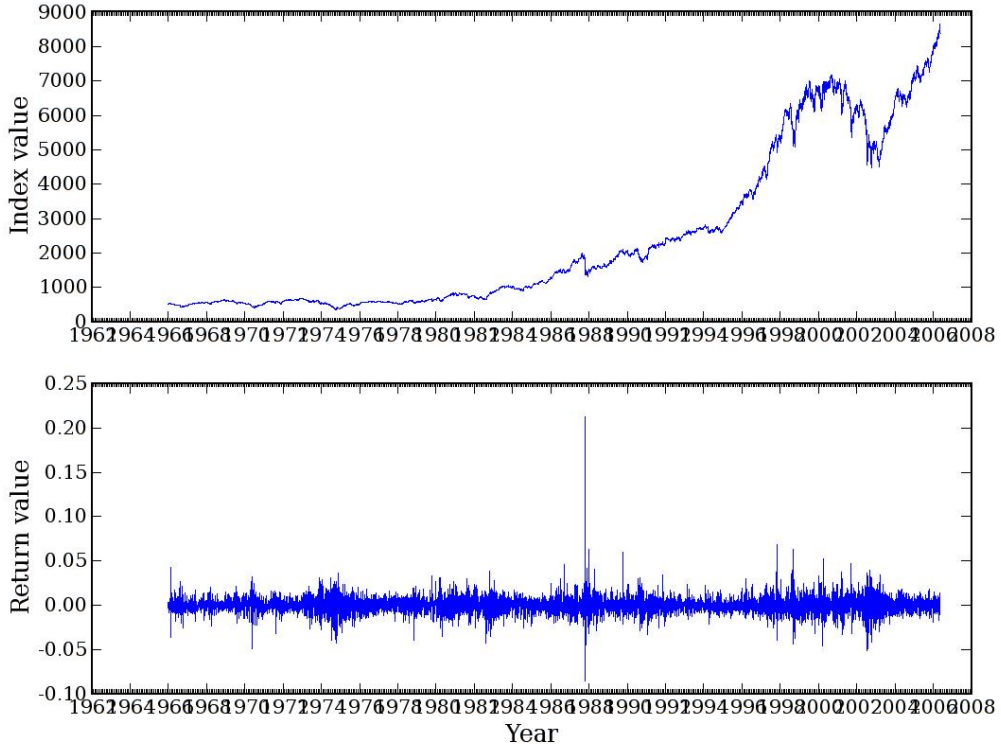
Lévy:$(\alpha, \beta) = (1.648, 0.118)$

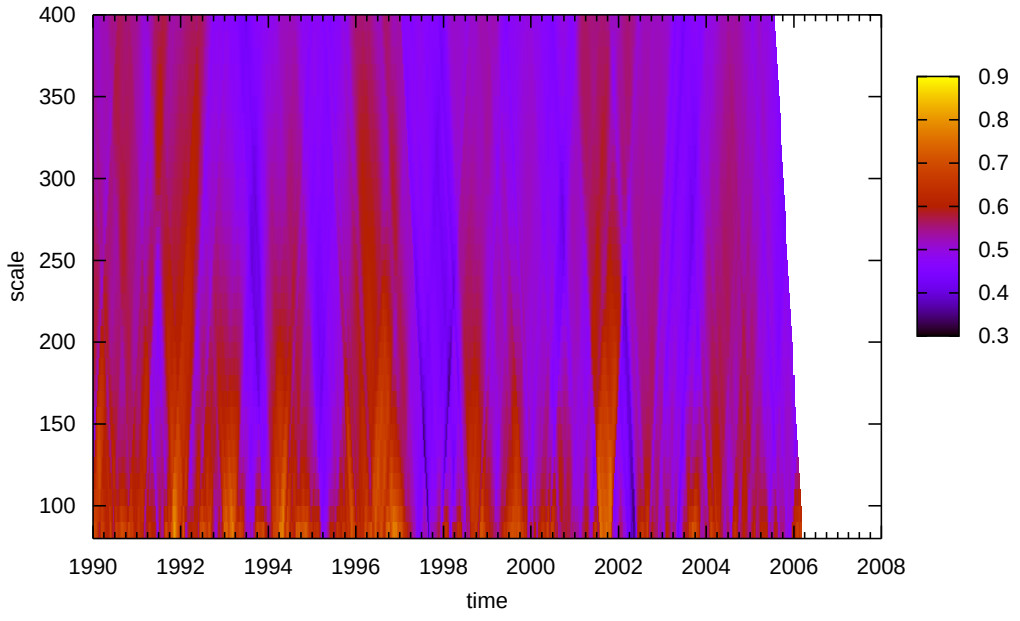




United States (NYSE COMPOSITE INDEX)

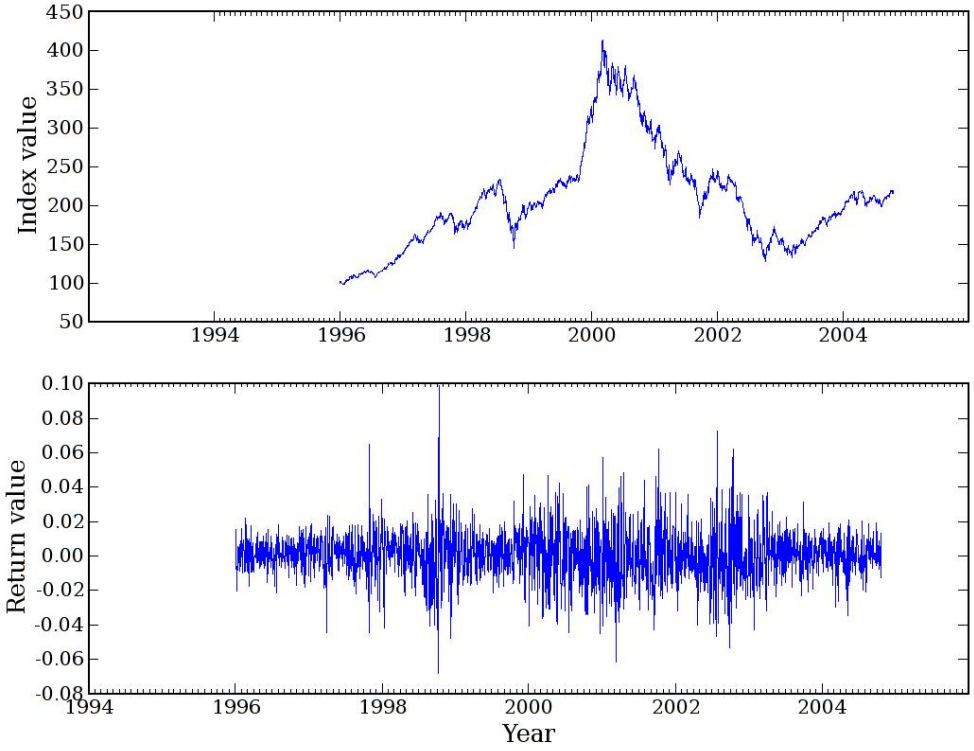
Lévy: $(\alpha, \beta) = (1.720, -0.084)$

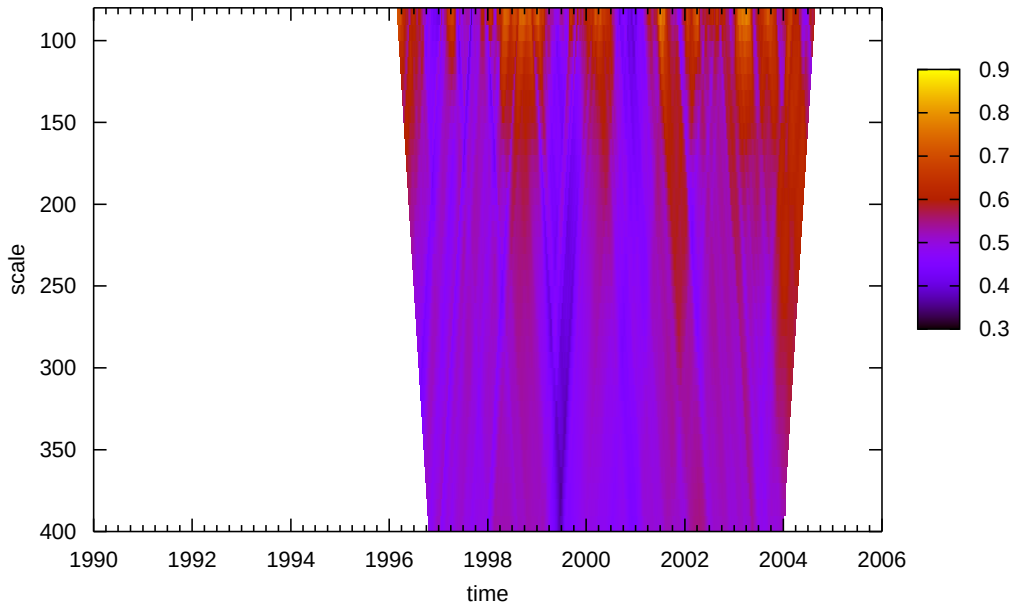




Sweden (Stockholm General)

Lévy:$(\alpha, \beta) = (1.758, 0.000)$

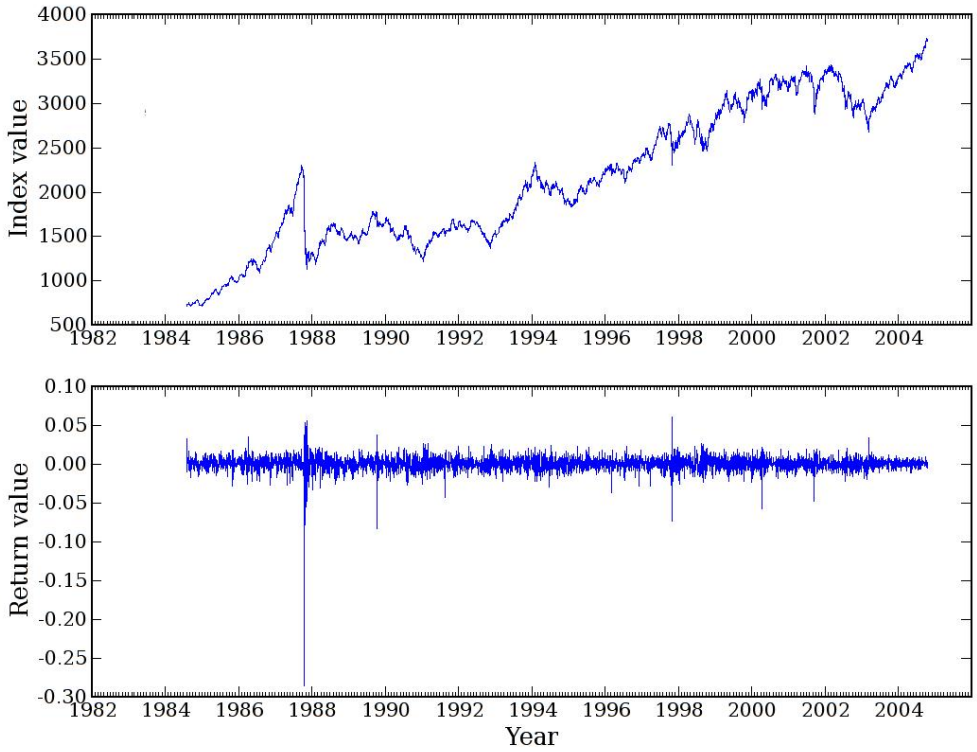


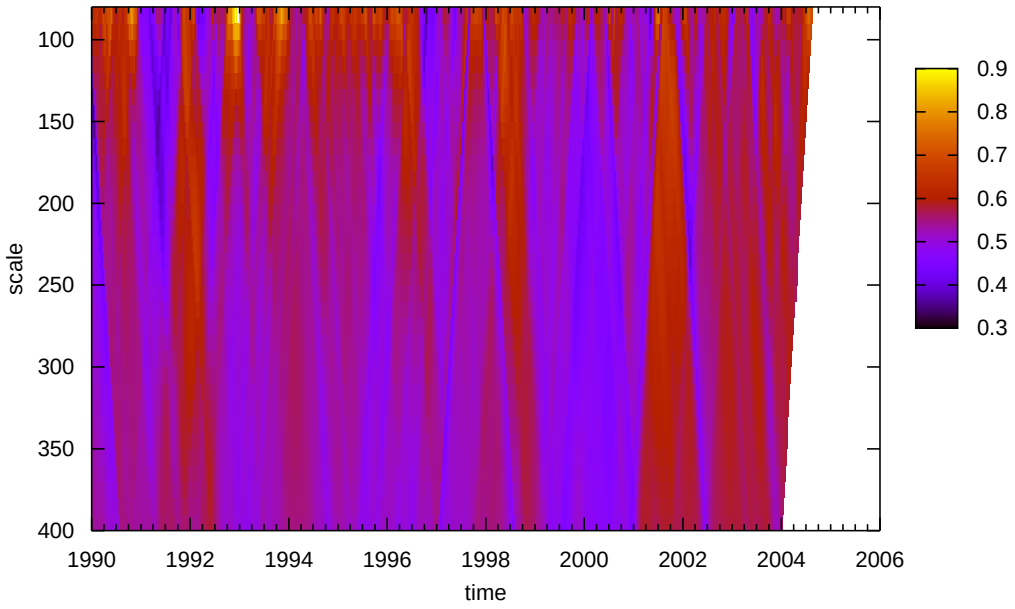


A.2. Hybrid

Australia (All Ordinaries)

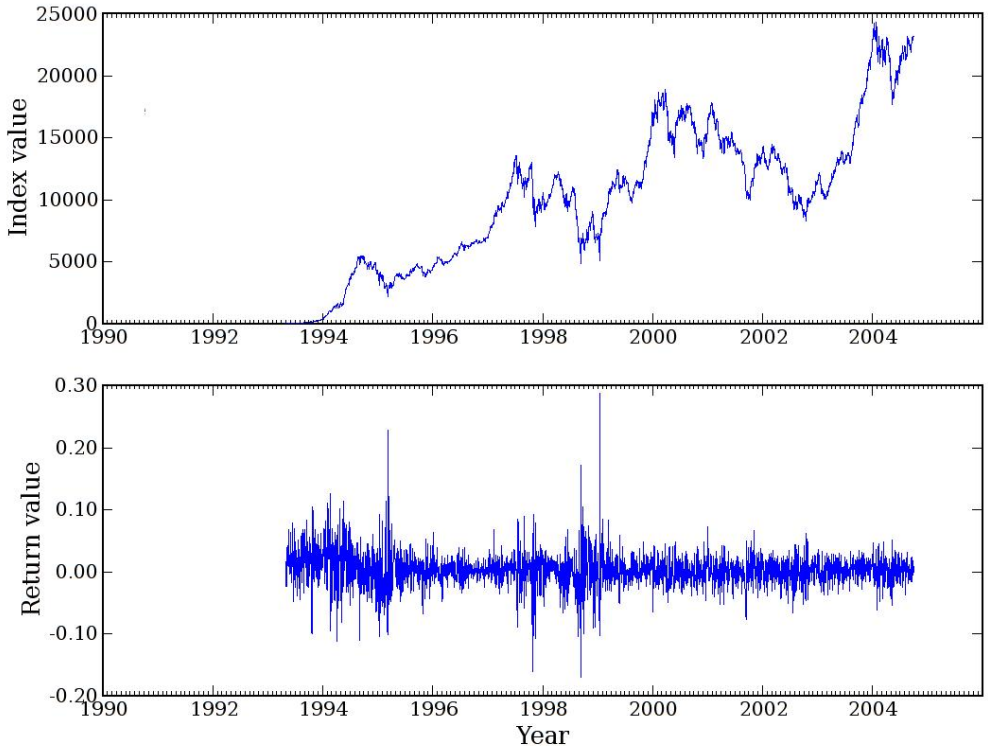
Lévy:$(\alpha, \beta) = (1.827, 0.257)$

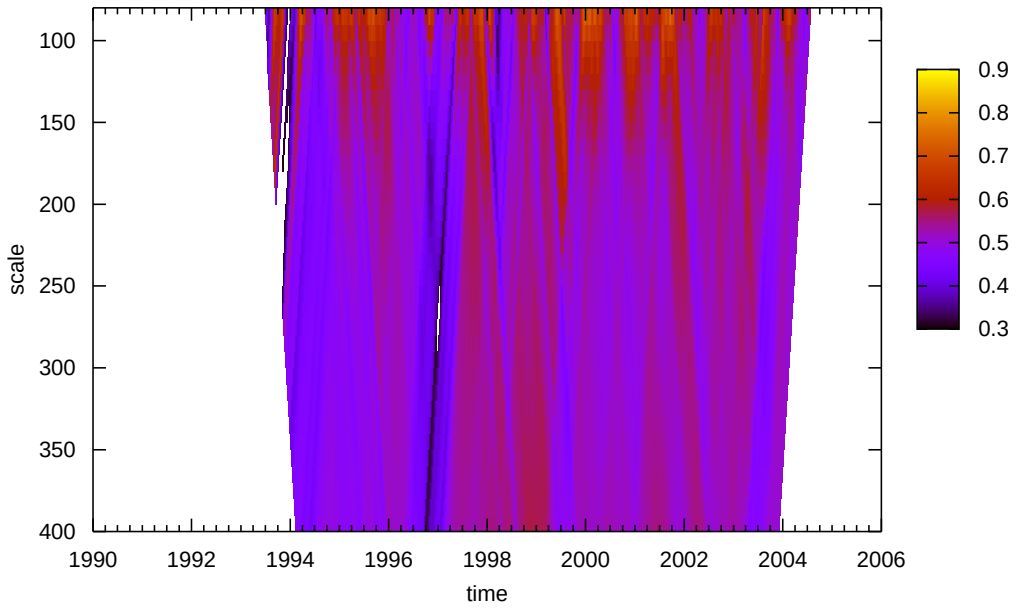




Brazil (Bovespa)

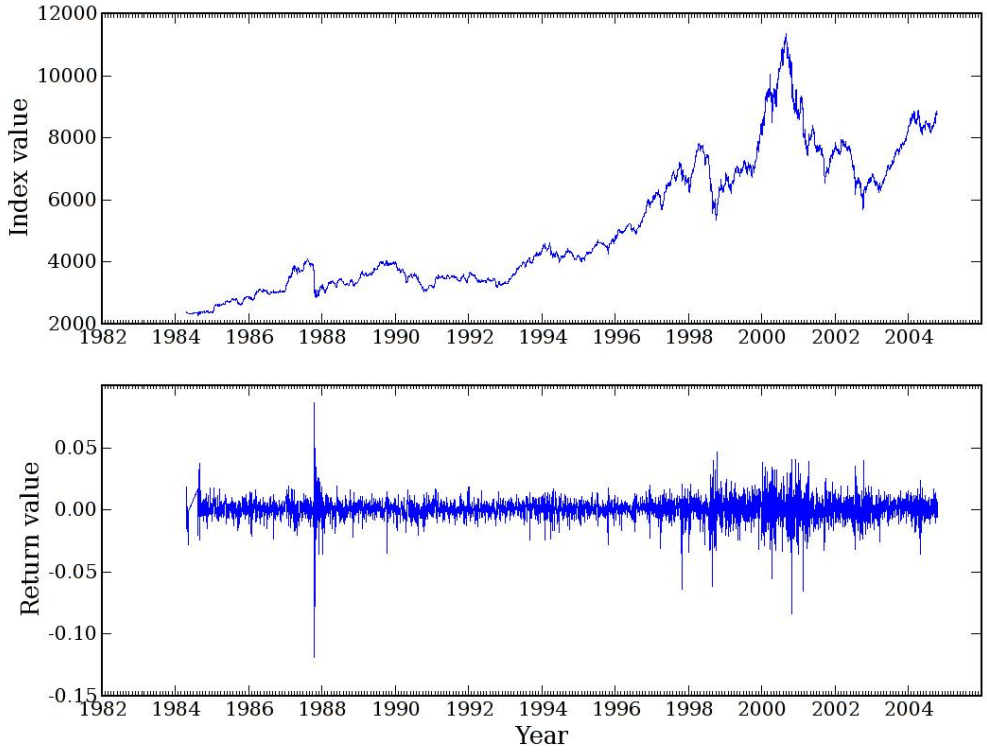
Lévy: $(\alpha, \beta) = (1.670, -0.002)$

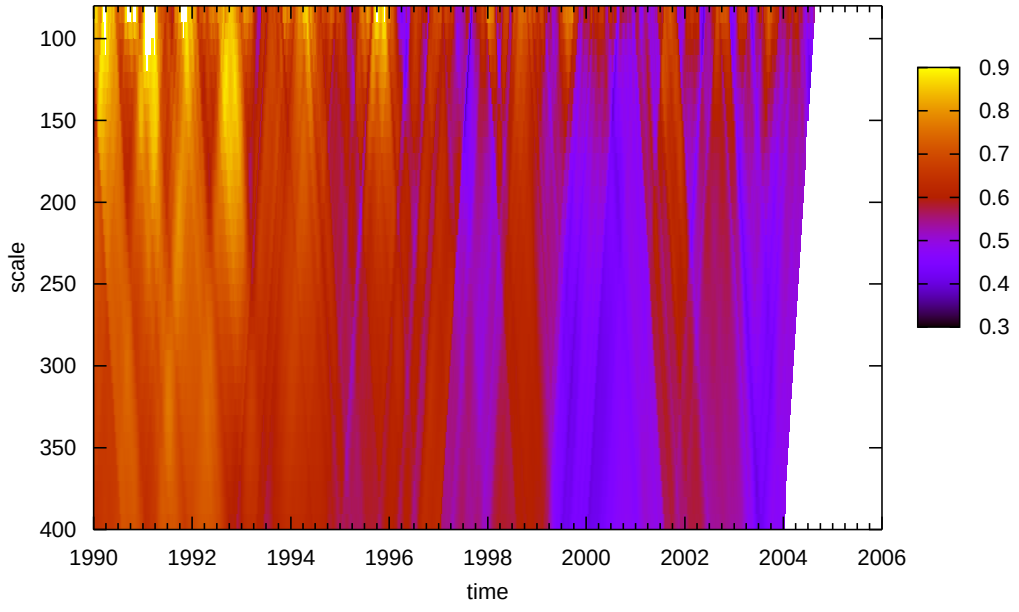




Canada (S&P TSX Composite)

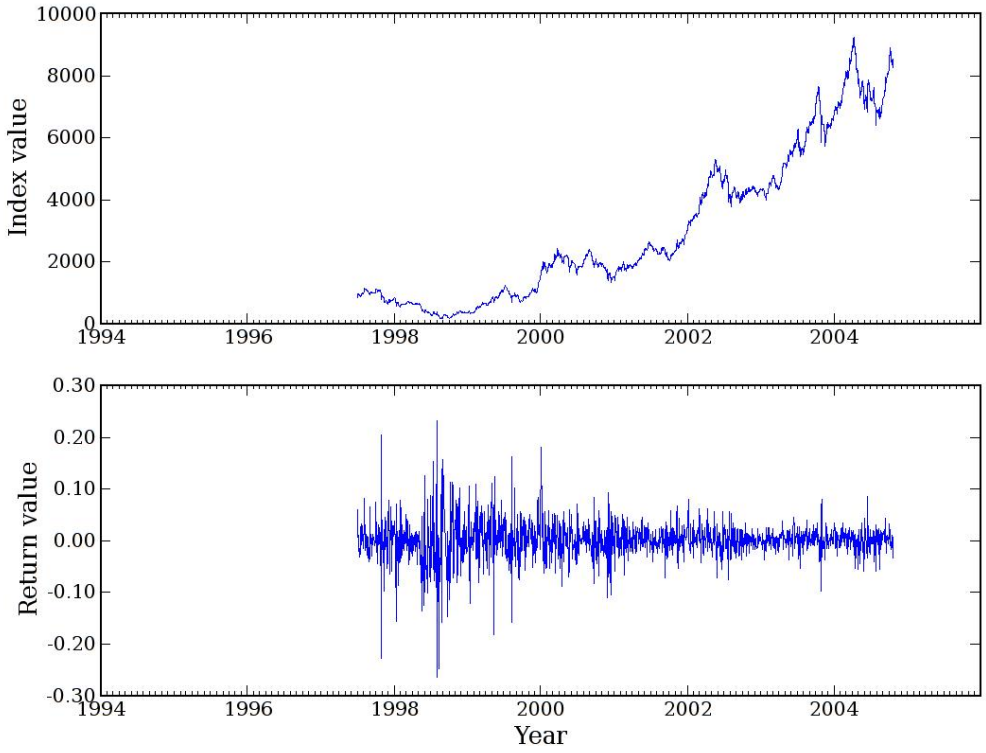
Lévy:$(\alpha, \beta) = (1.625, -0.001)$

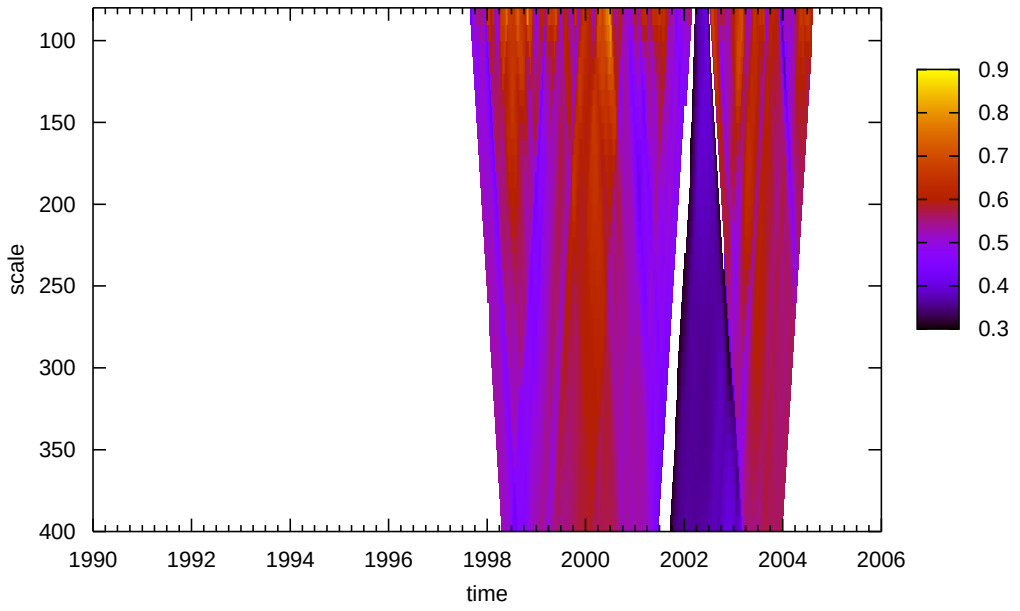




Russia (Moscow Times)

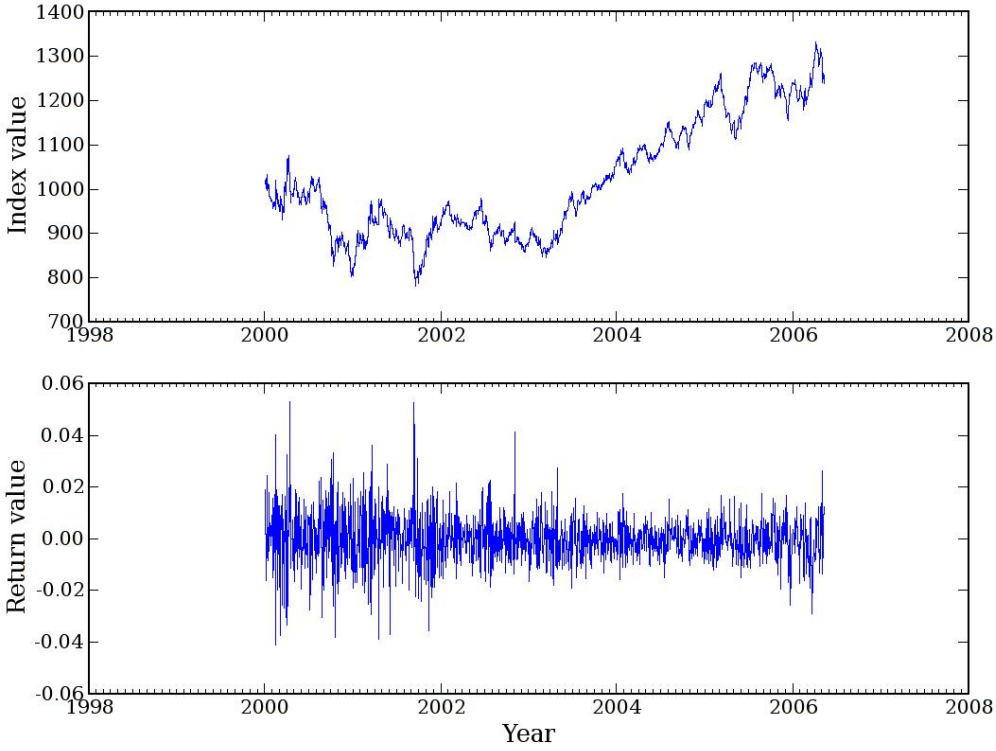
Lévy:$(\alpha, \beta) = (1.545, 0.054)$

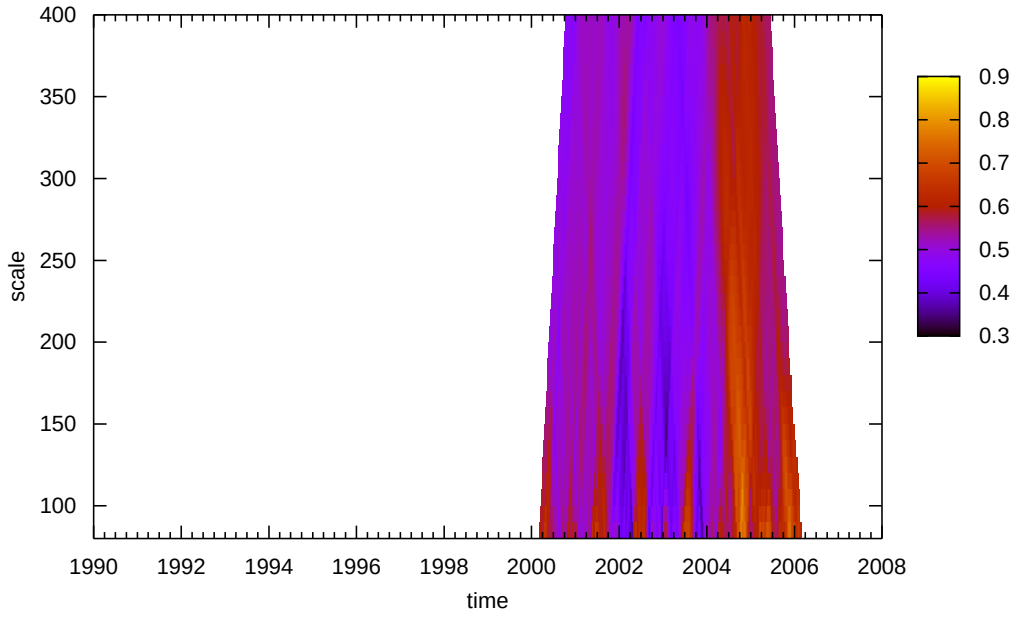




New Zealand (NZSE 10)

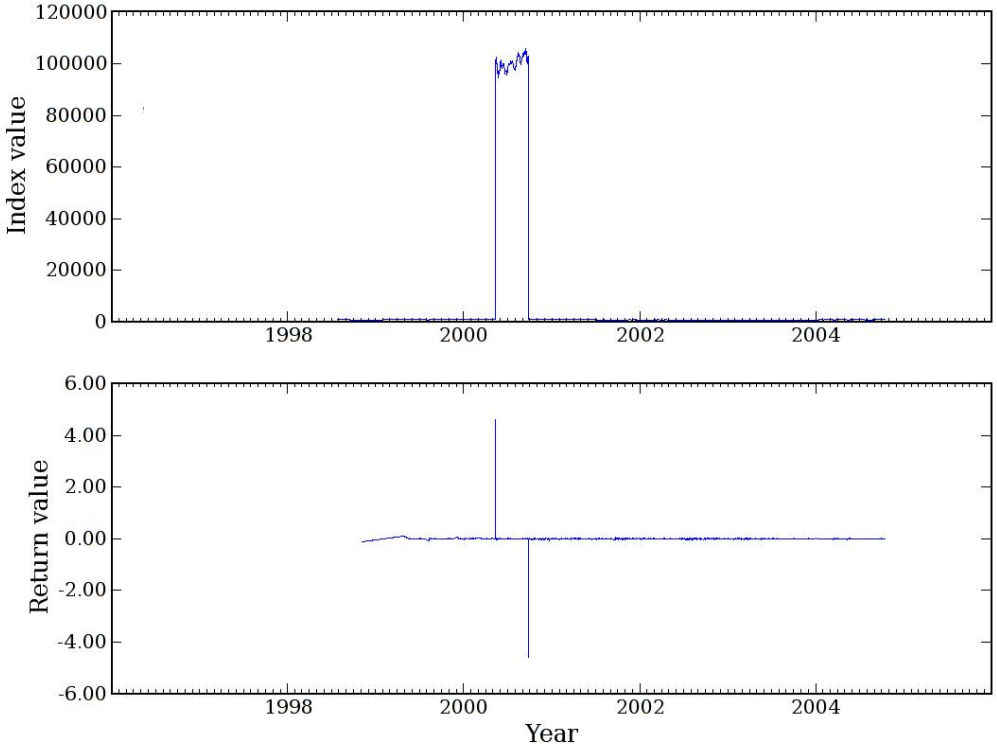
Lévy:$(\alpha, \beta) = (1.743, -0.006)$

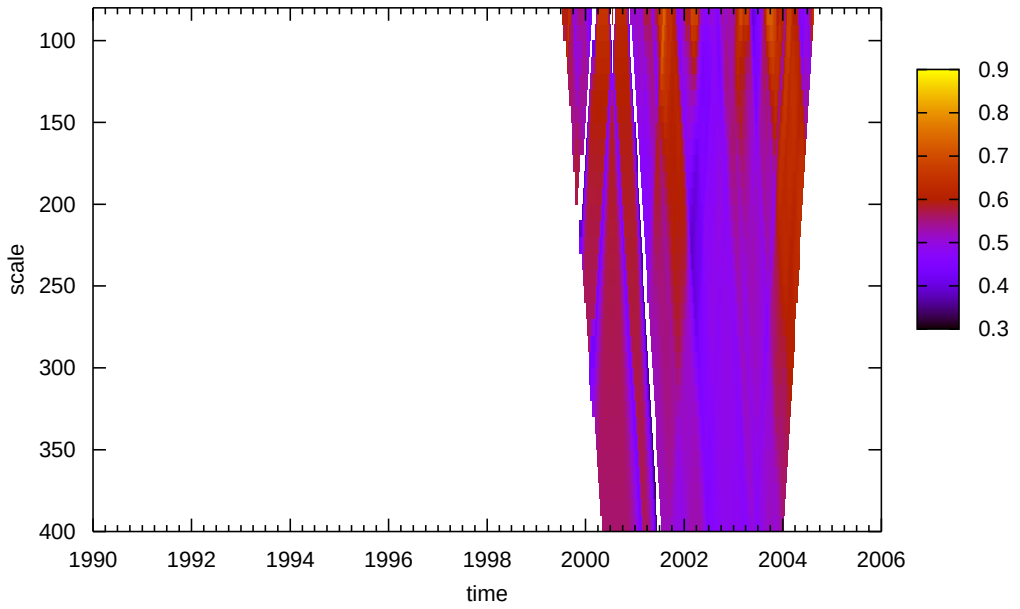




Spain (Madrid General)

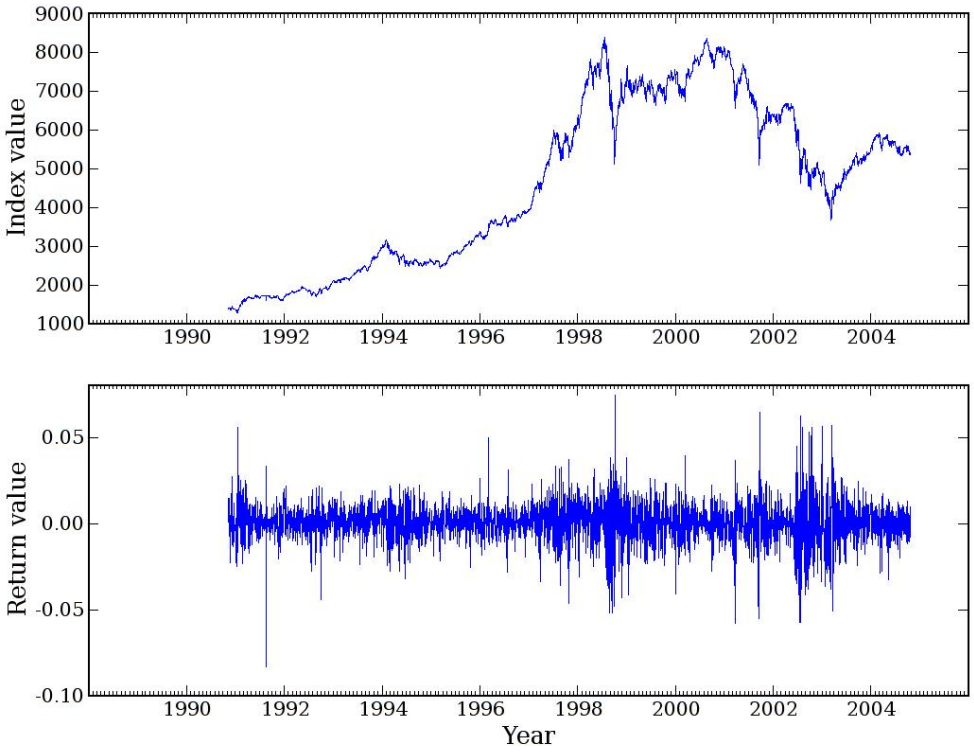
Lévy:$(\alpha, \beta) = (1.785, 0.003)$

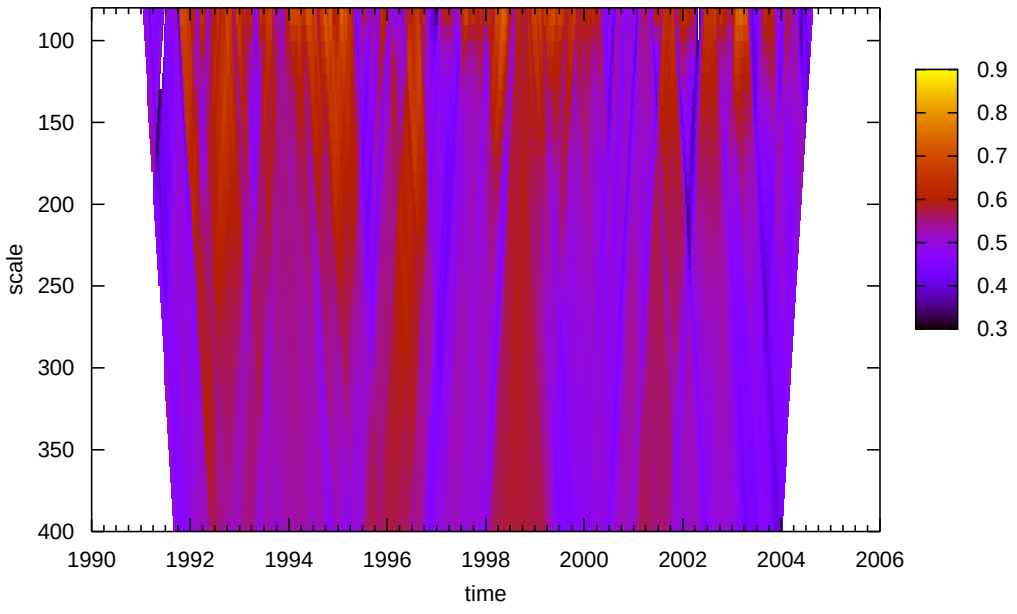




Switzerland (Swiss Market)

Lévy:$(\alpha, \beta) = (1.701, 0.179)$

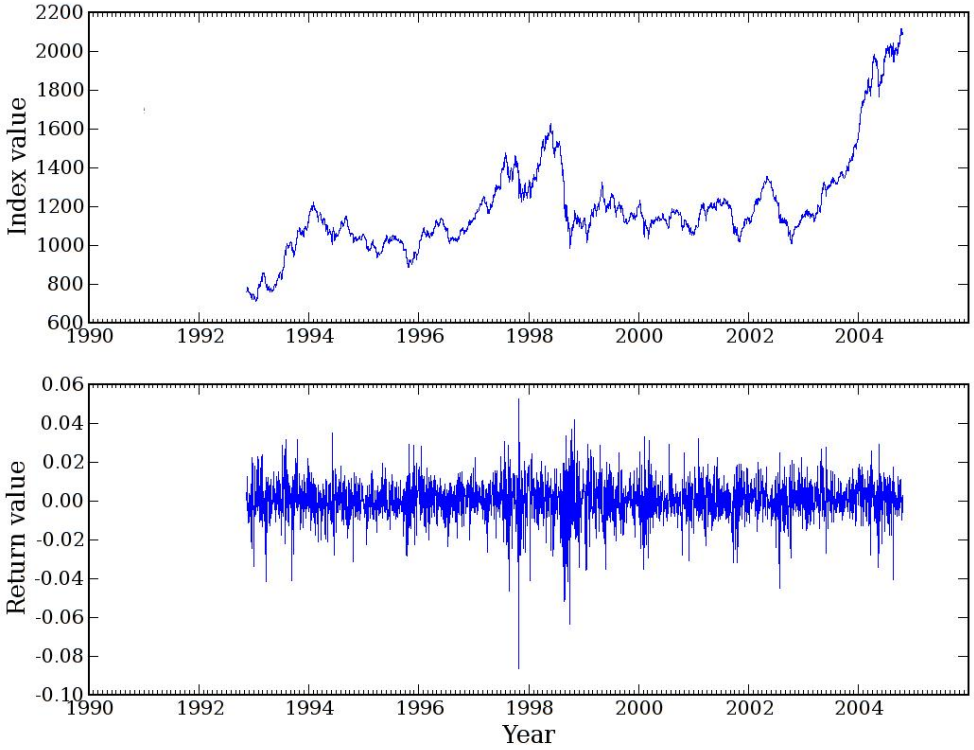


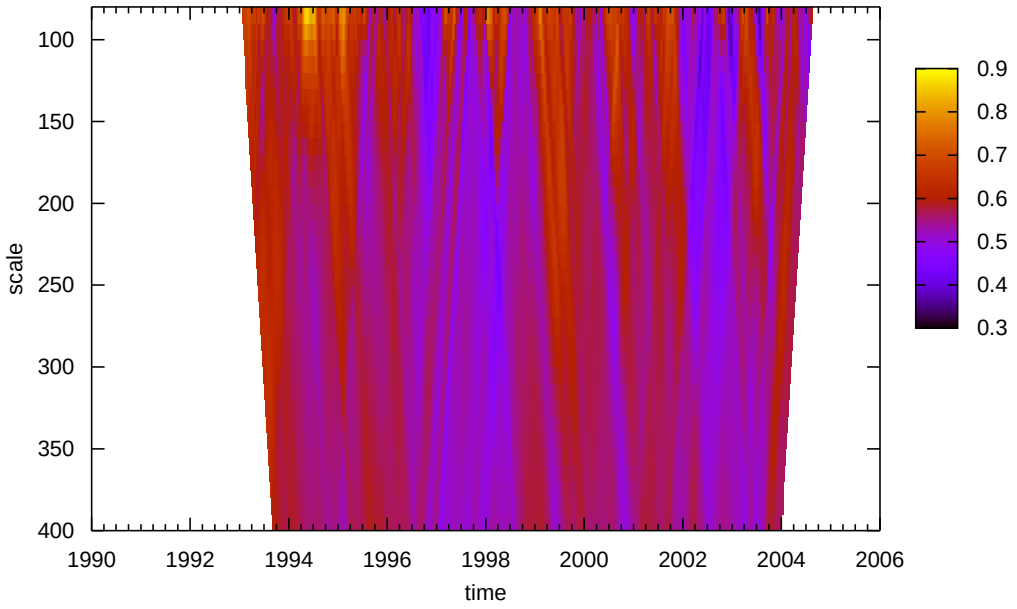


A.3. Emerging

Austria (ATX)

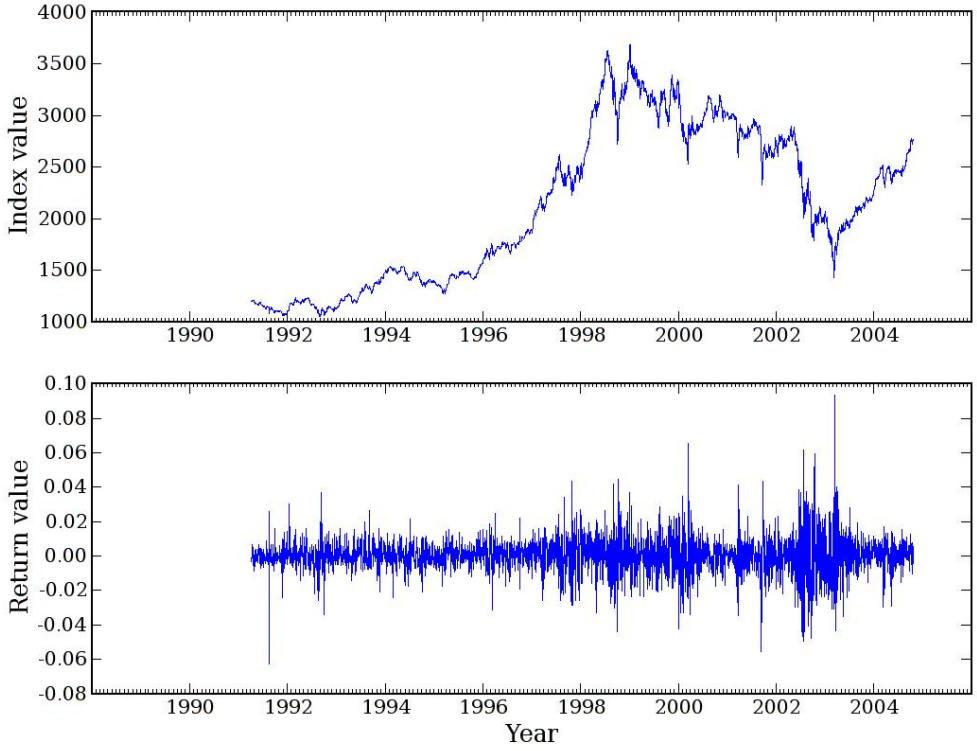
Lévy:$(\alpha, \beta) = (1.749, 0.220)$

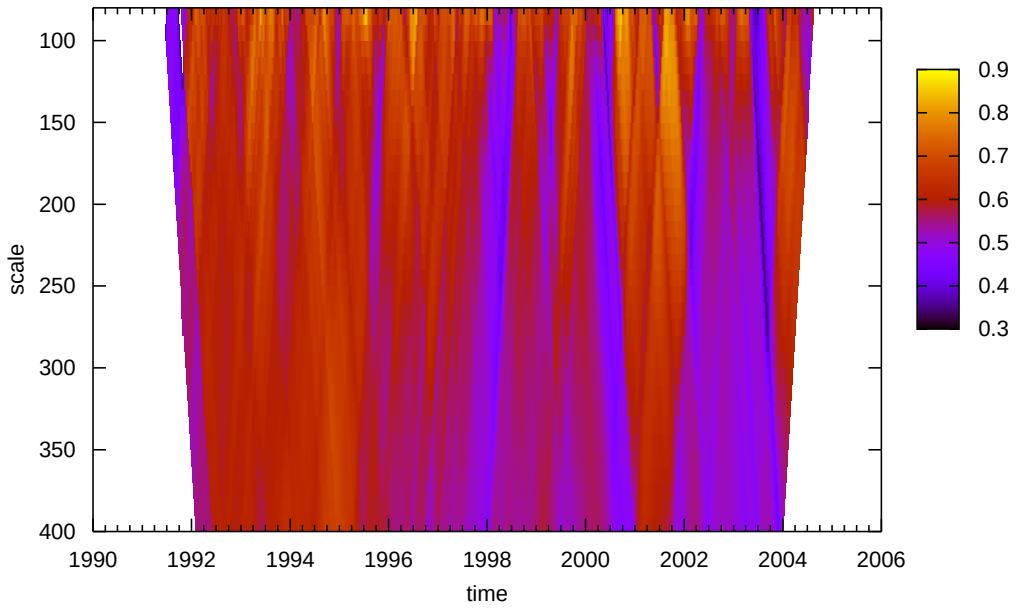




Belgium (BEL-20)

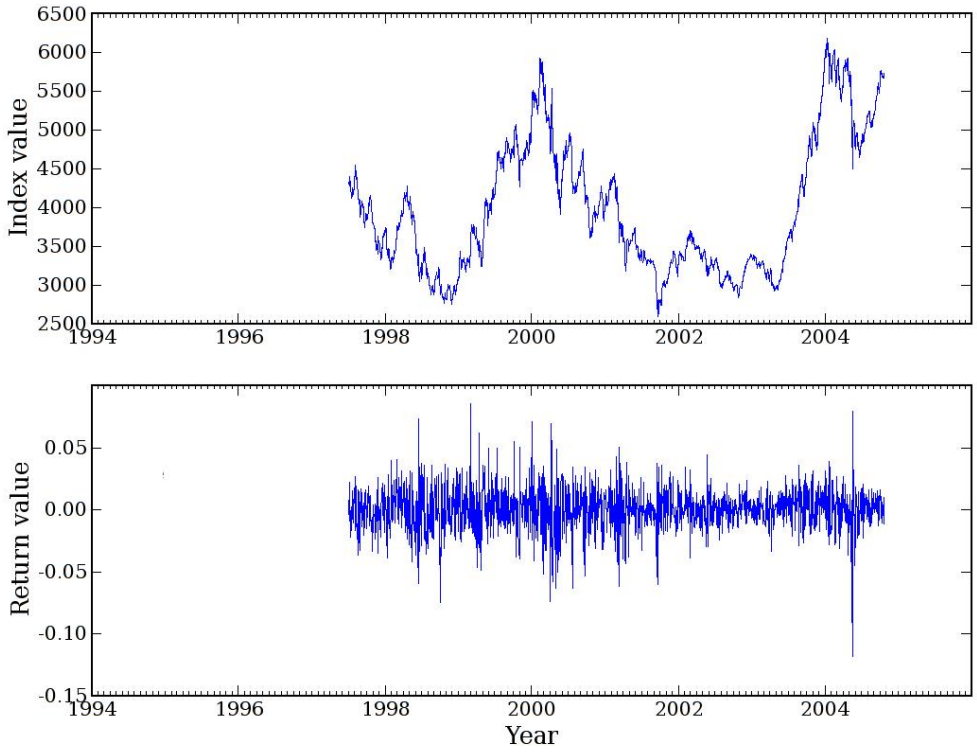
Lévy: $(\alpha, \beta) = (1.578, -0.002)$

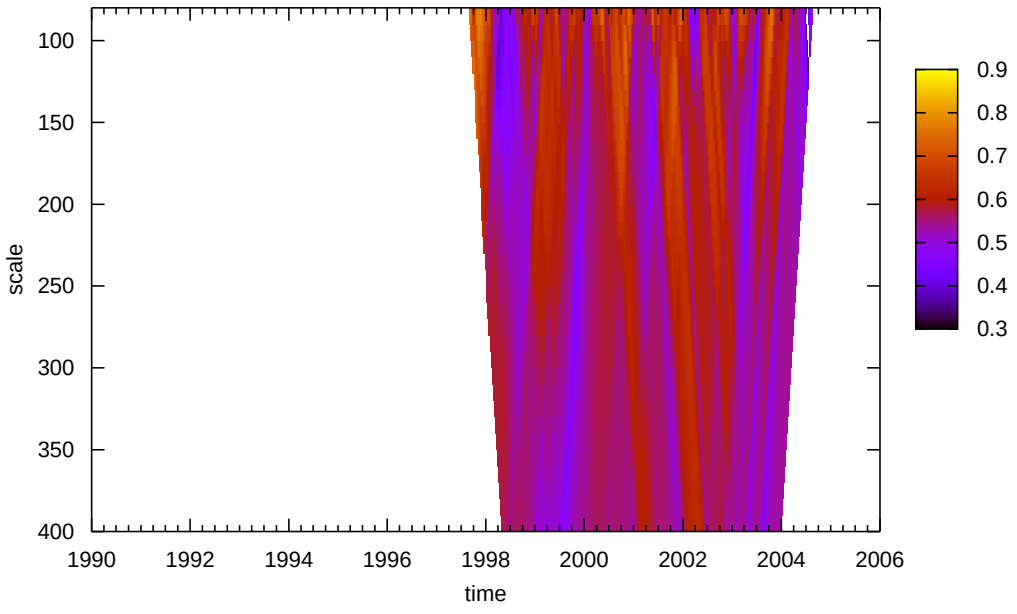




India (BSE 30)

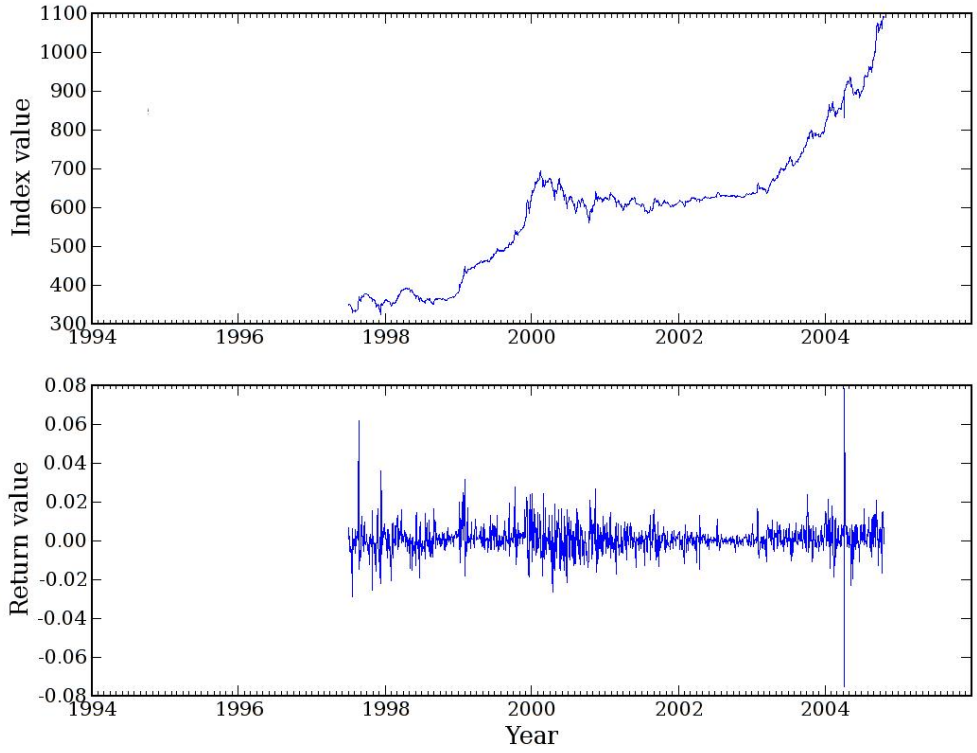
Lévy:$(\alpha, \beta) = (1.789, 0.266)$

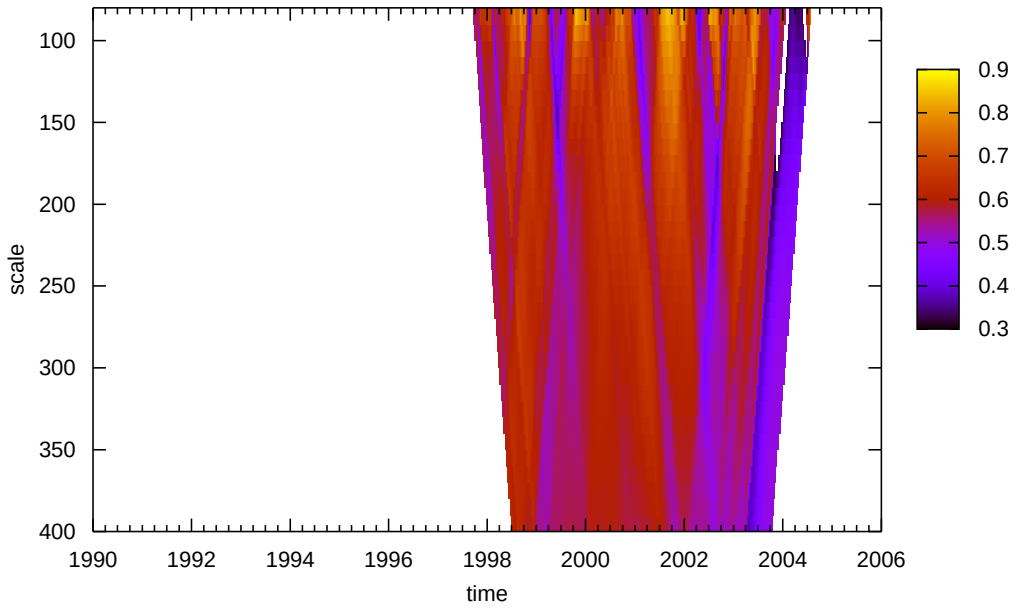




Egypt (CMA)

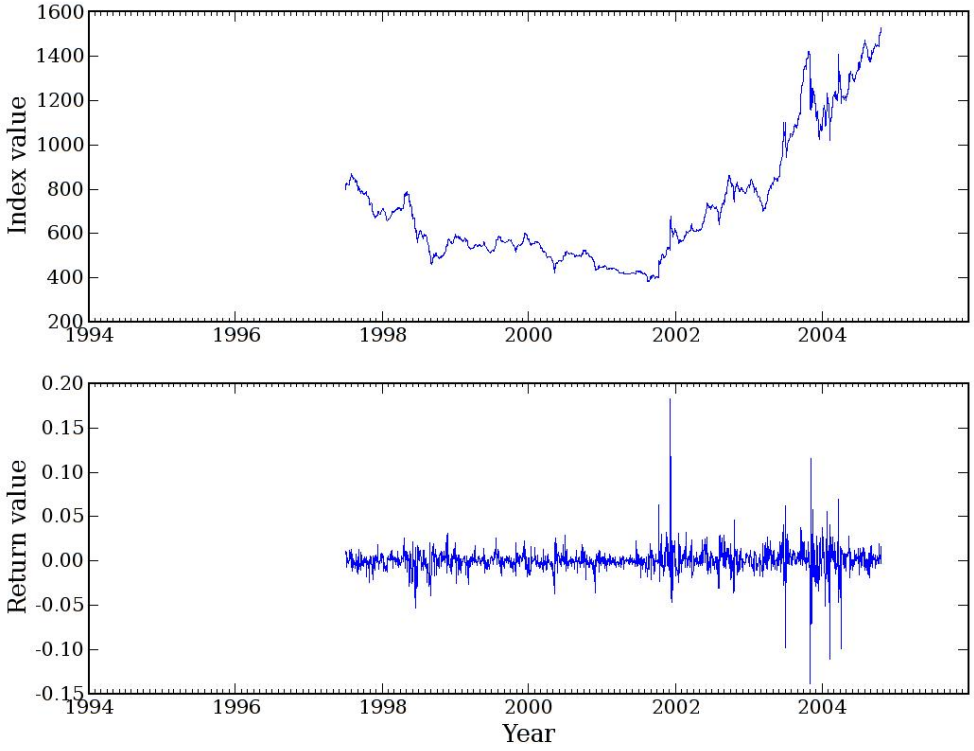
Lévy:$(\alpha, \beta) = (1.466, -0.154)$

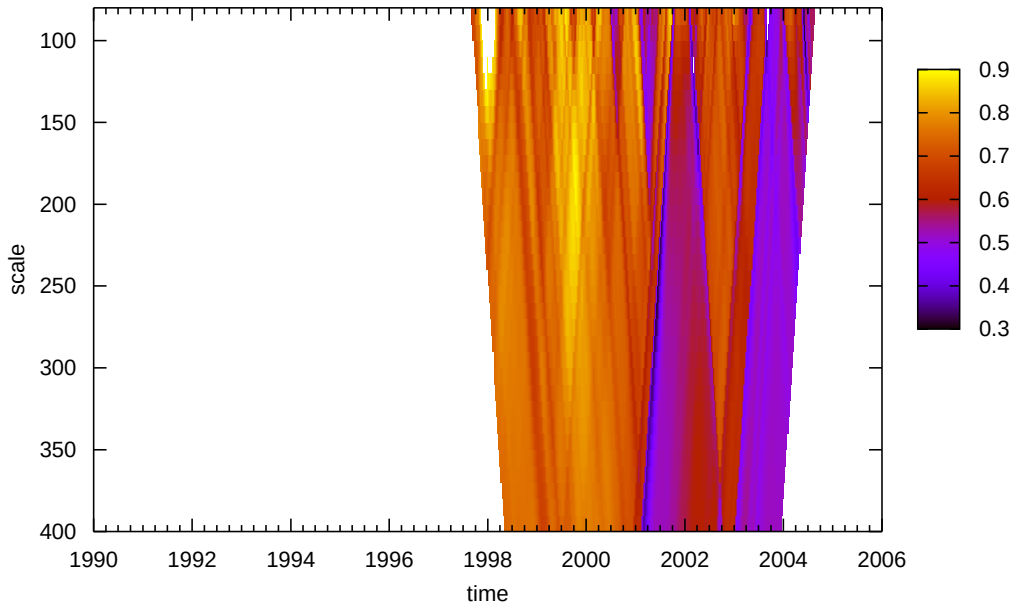




Sri Lanka (All Share)

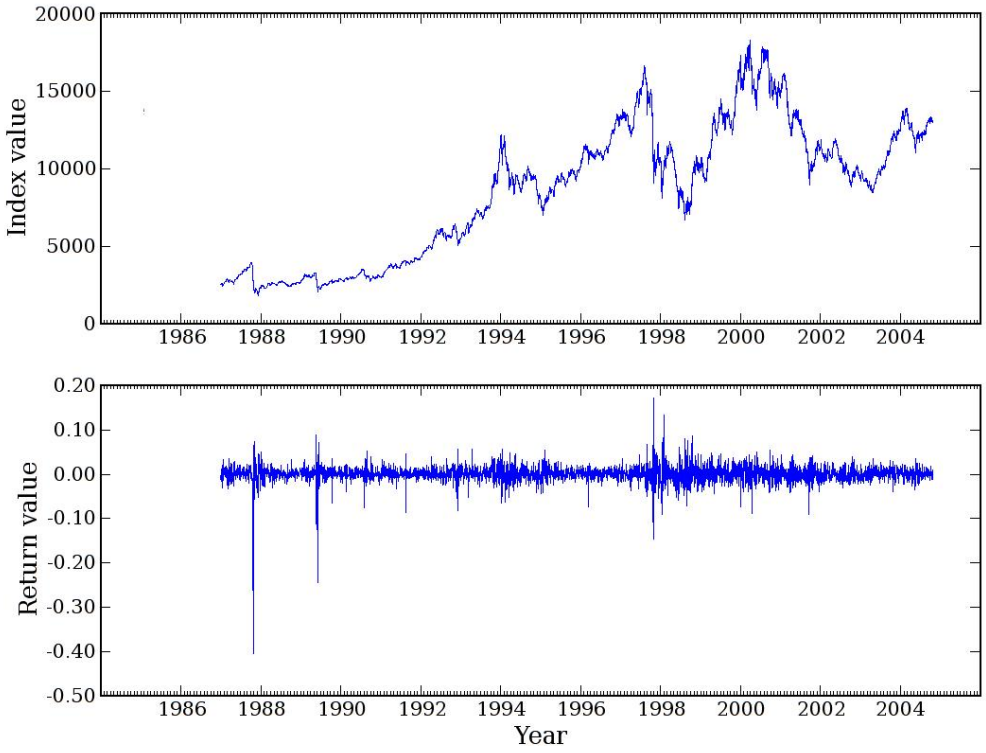
Lévy:$(\alpha, \beta) = (1.457, -0.038)$

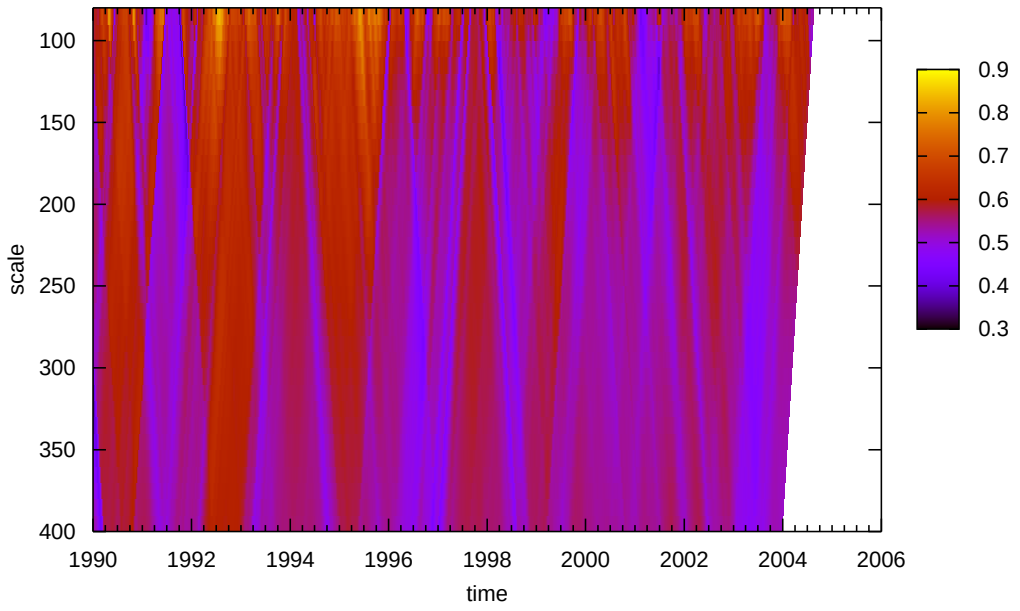




Hong Kong (Hang Seng)

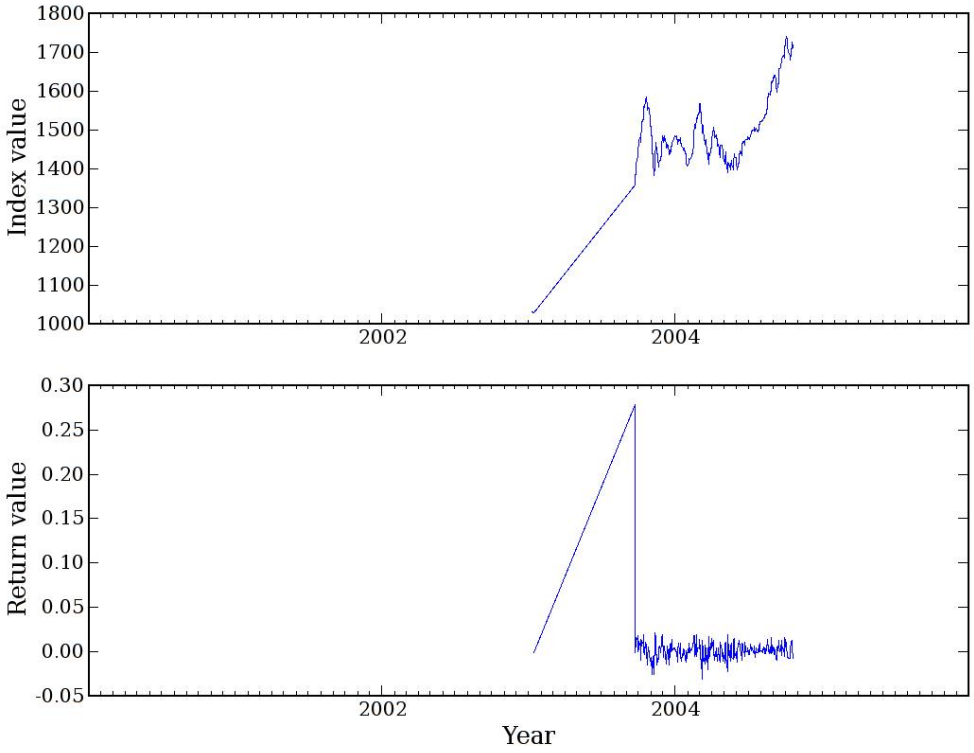
Lévy:$(\alpha, \beta) = (1.620, -0.002)$

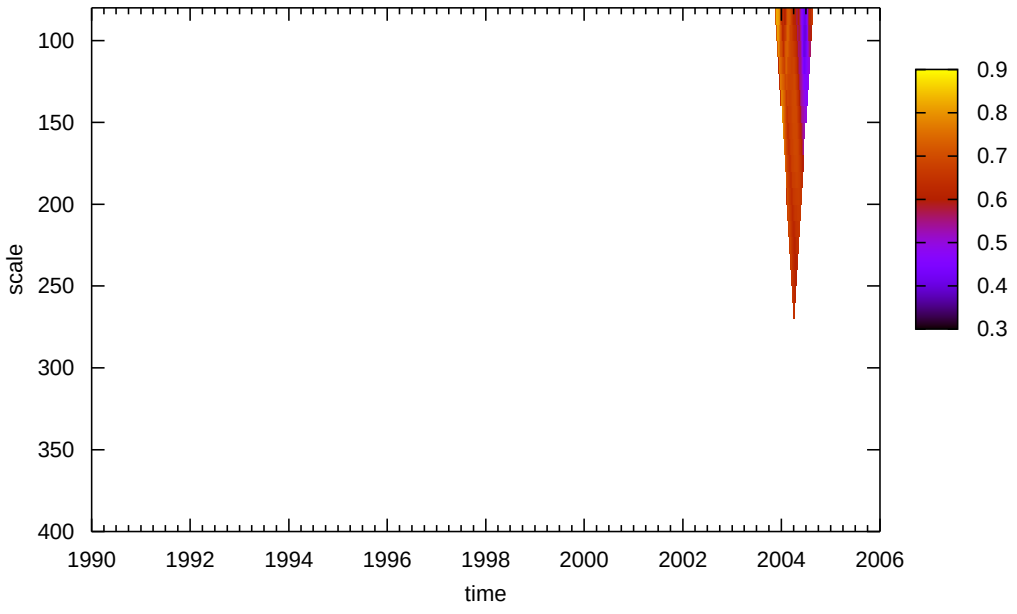




Chile (IPSA)

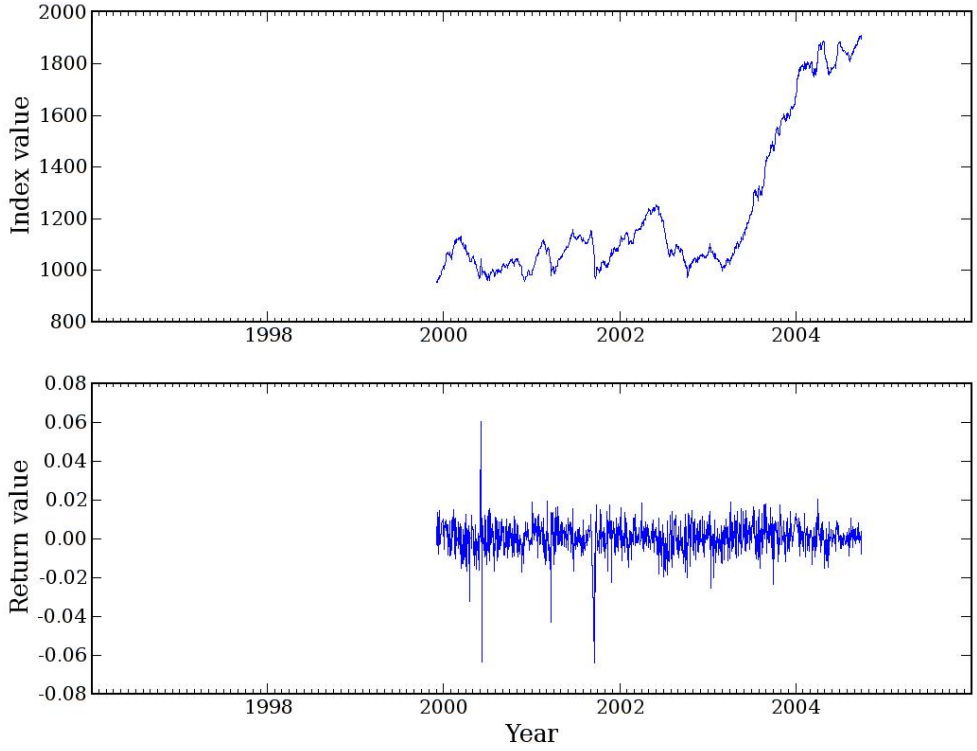
Lévy:$(\alpha, \beta) = (1.897, -0.011)$

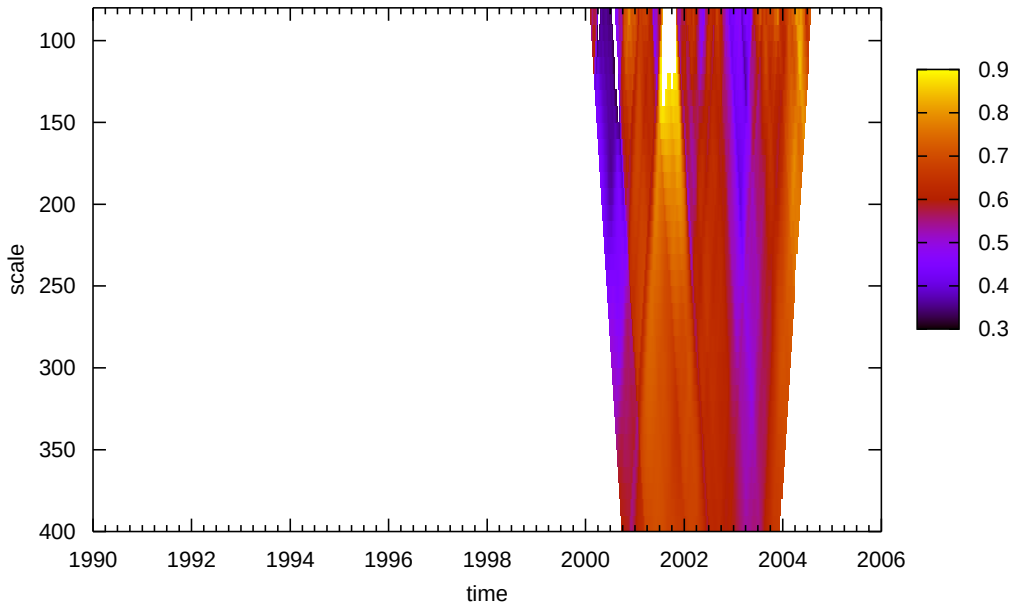




Ireland (ISEC Small Cap)

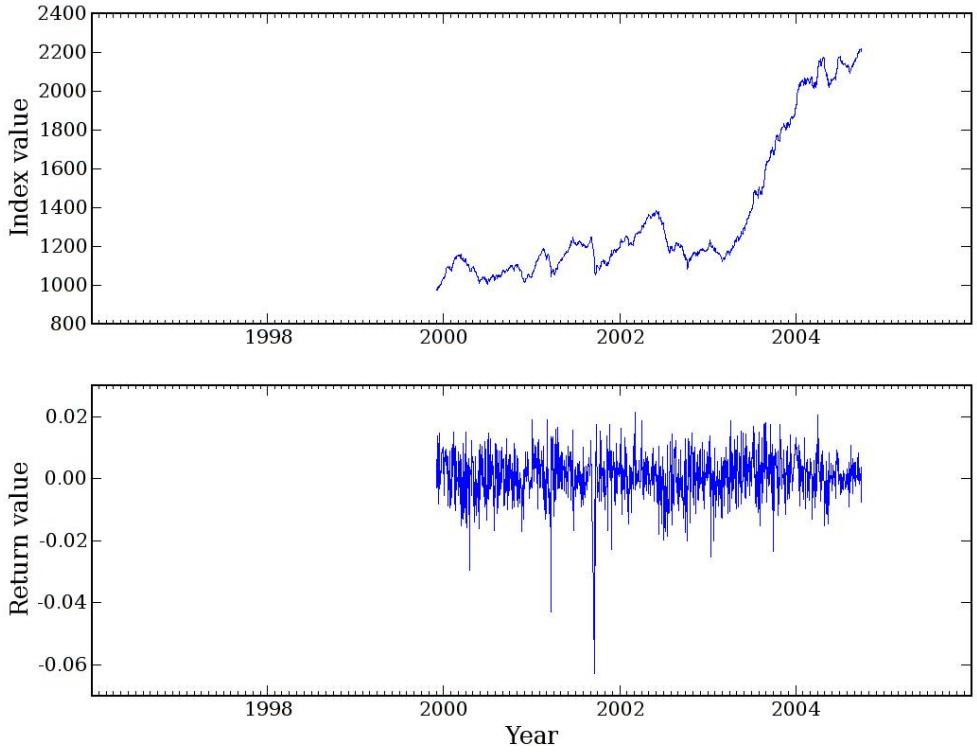
Lévy:$(\alpha, \beta) = (1.863, -0.001)$

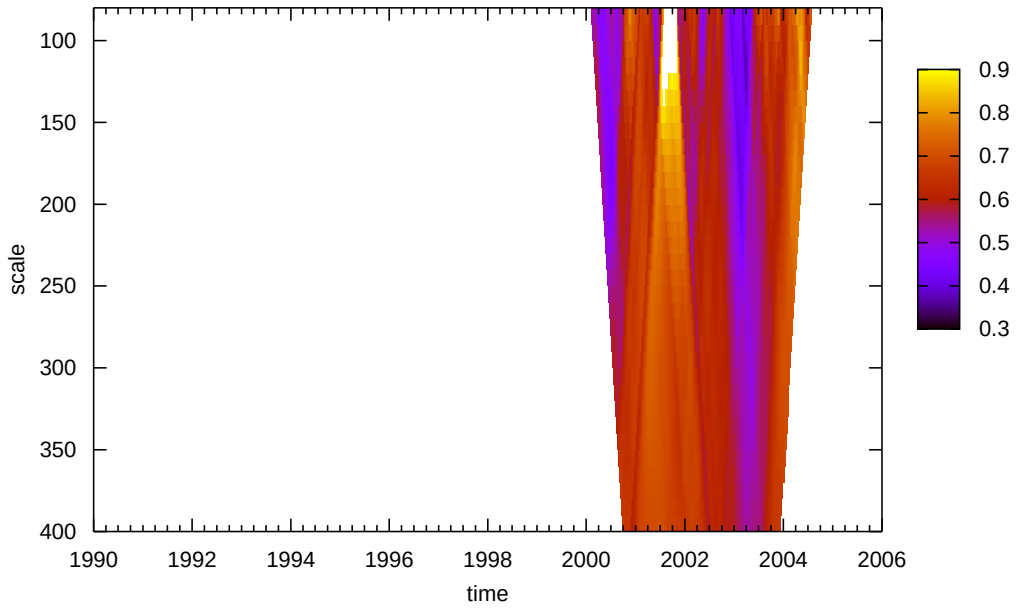




Ireland (ISEC Small Cap Techno)

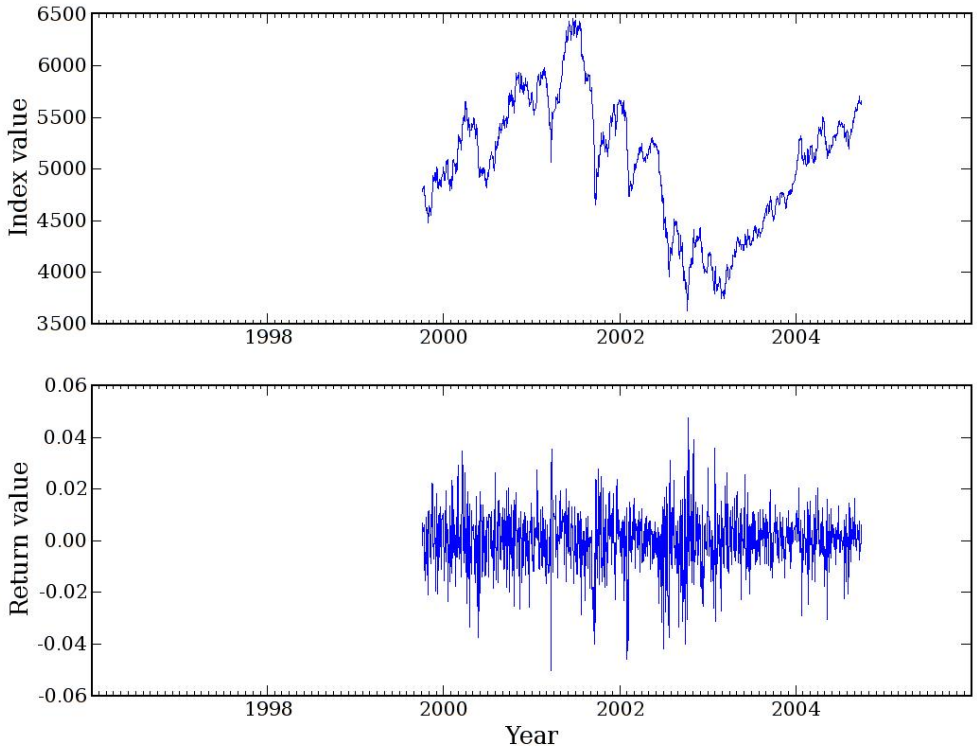
Lévy:$(\alpha, \beta) = (1.911, 1.000)$

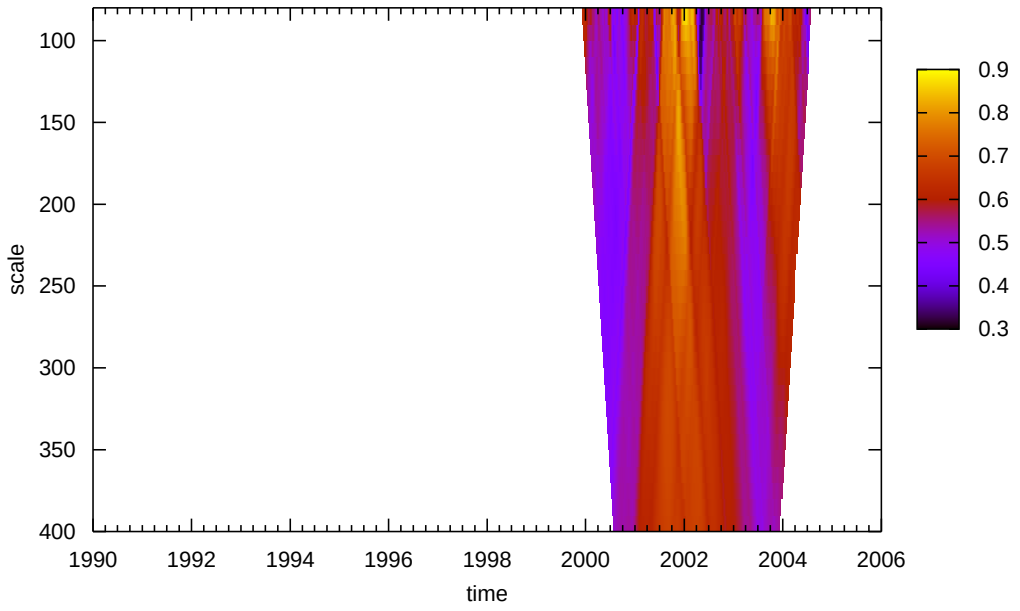




Ireland (Irish SE Index)

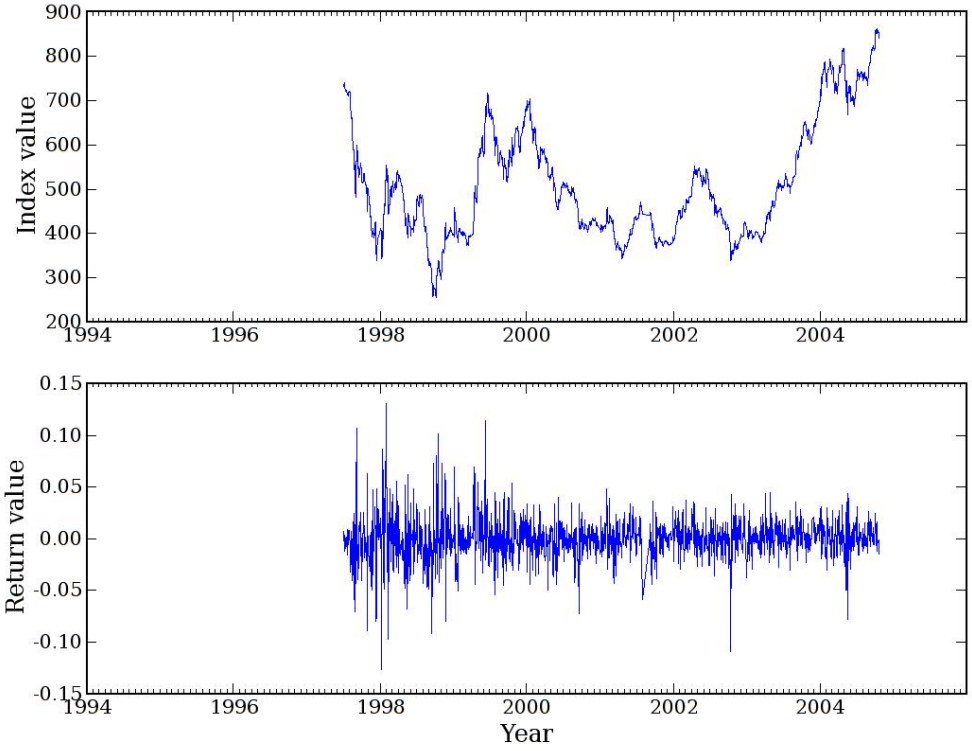
Lévy:$(\alpha, \beta) = (1.724, 0.331)$

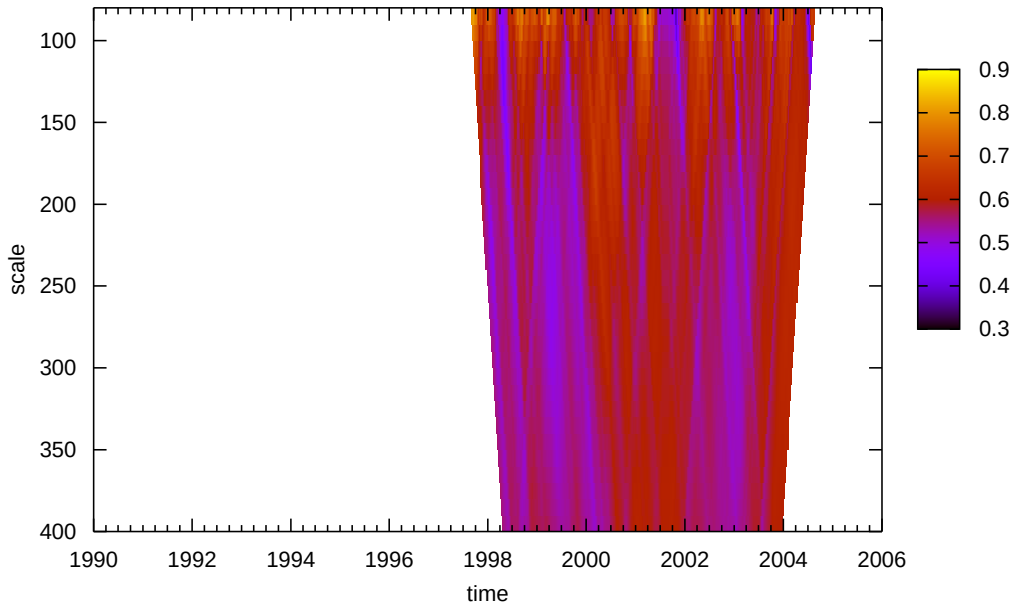




Indonesia (Jakarta Composite)

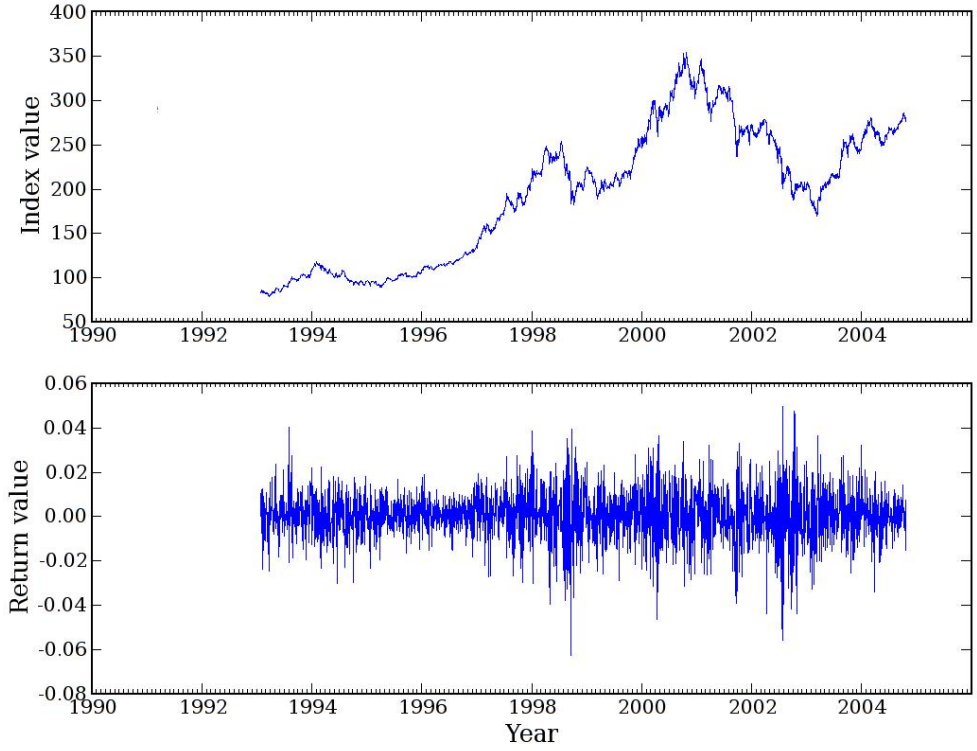
Lévy:$(\alpha, \beta) = (1.532, -0.064)$

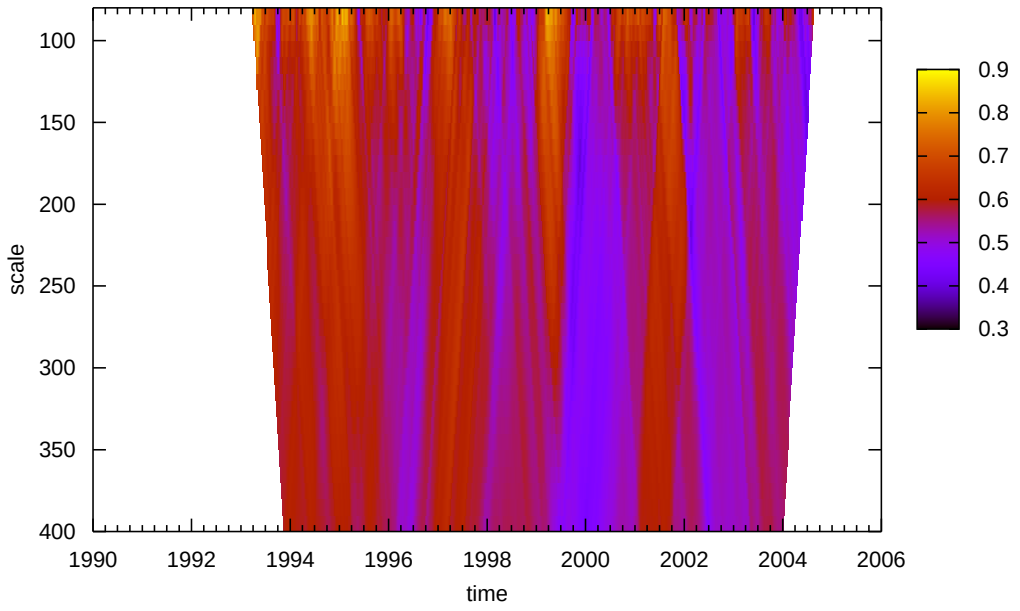




Denmark (KFX)

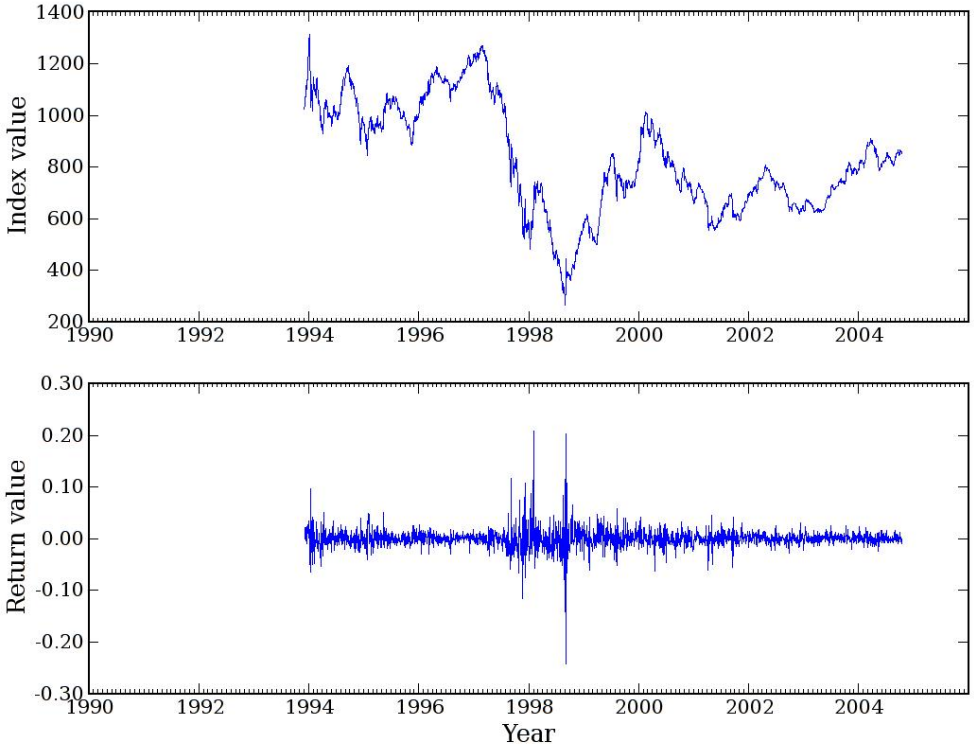
Lévy:$(\alpha, \beta) = (1.744, 0.149)$

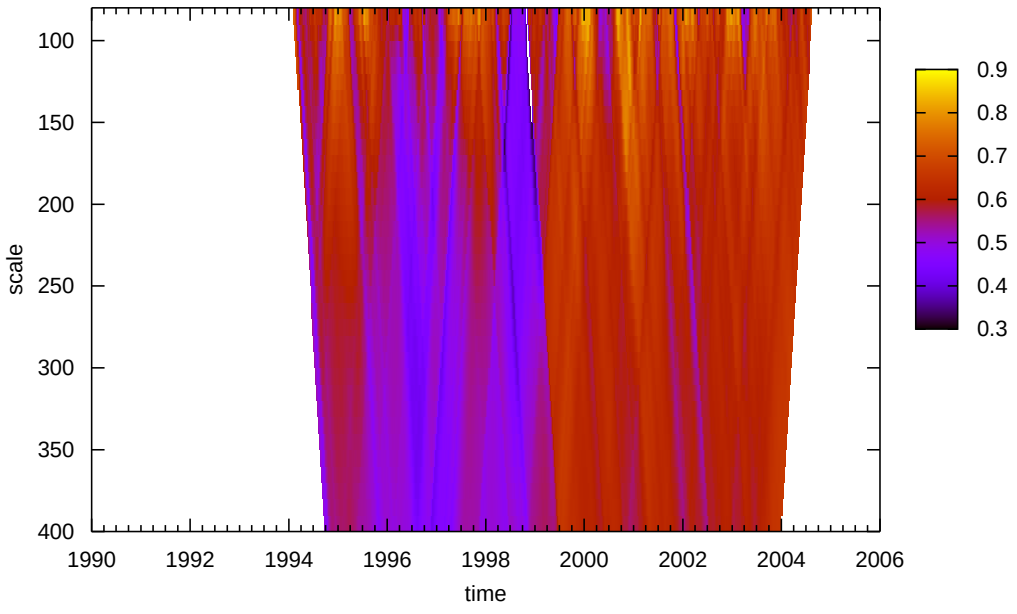




Malaysia (KLSE Composite)

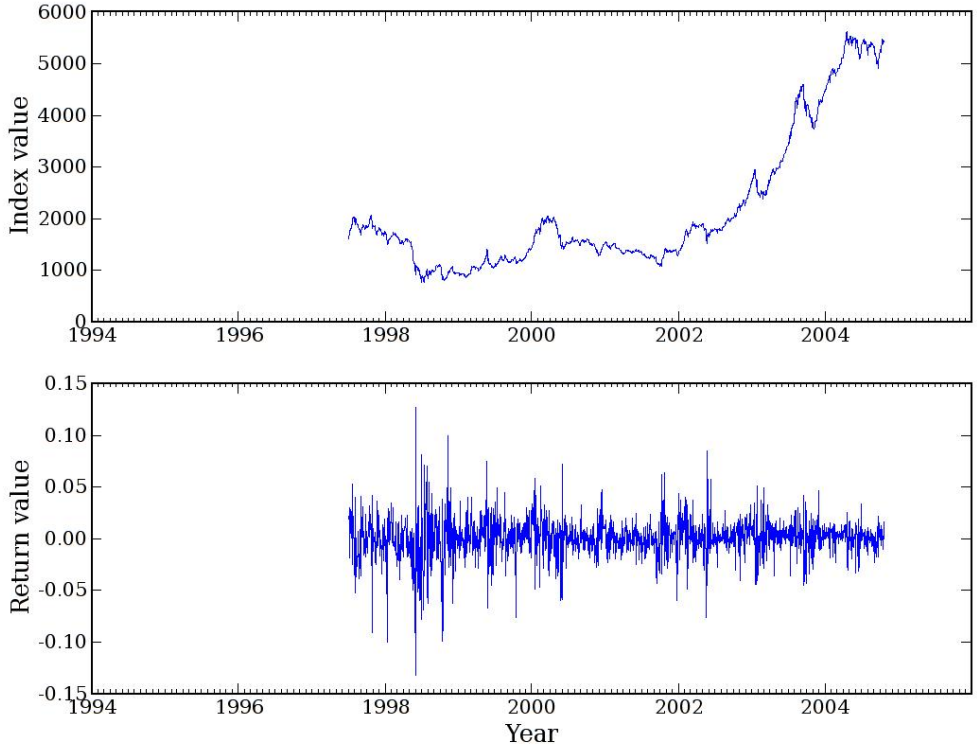
Lévy:$(\alpha, \beta) = (1.482, -0.003)$

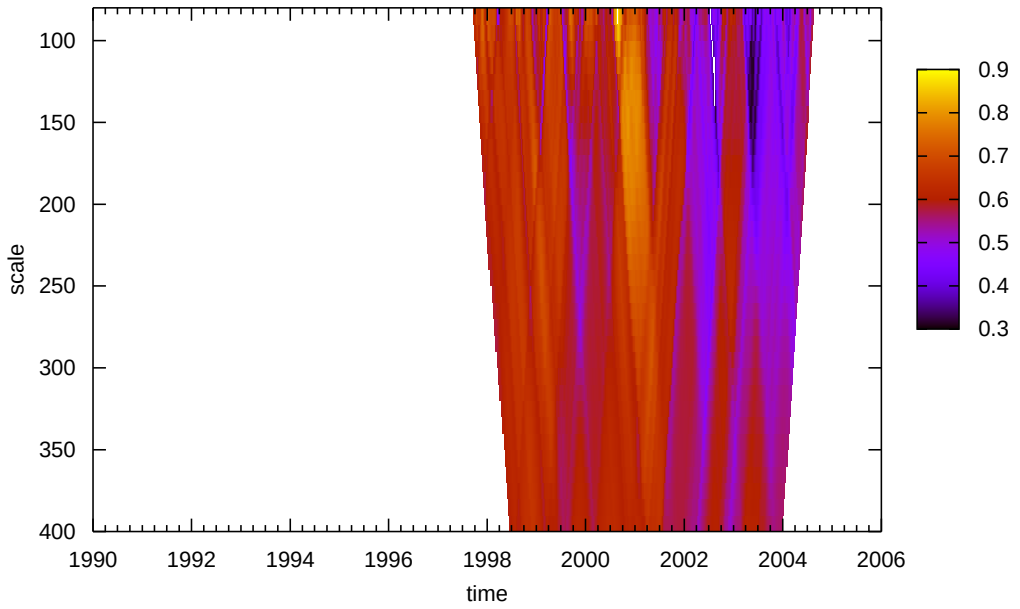




Pakistan (Karachi 100)

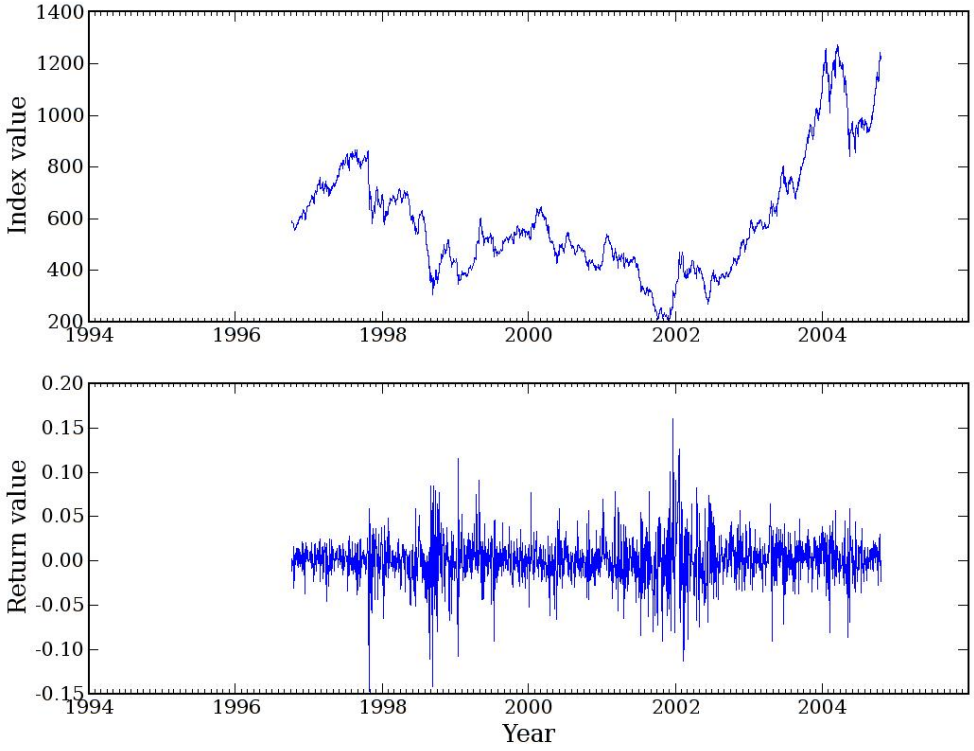
Lévy: $(\alpha, \beta) = (1.560, 0.189)$

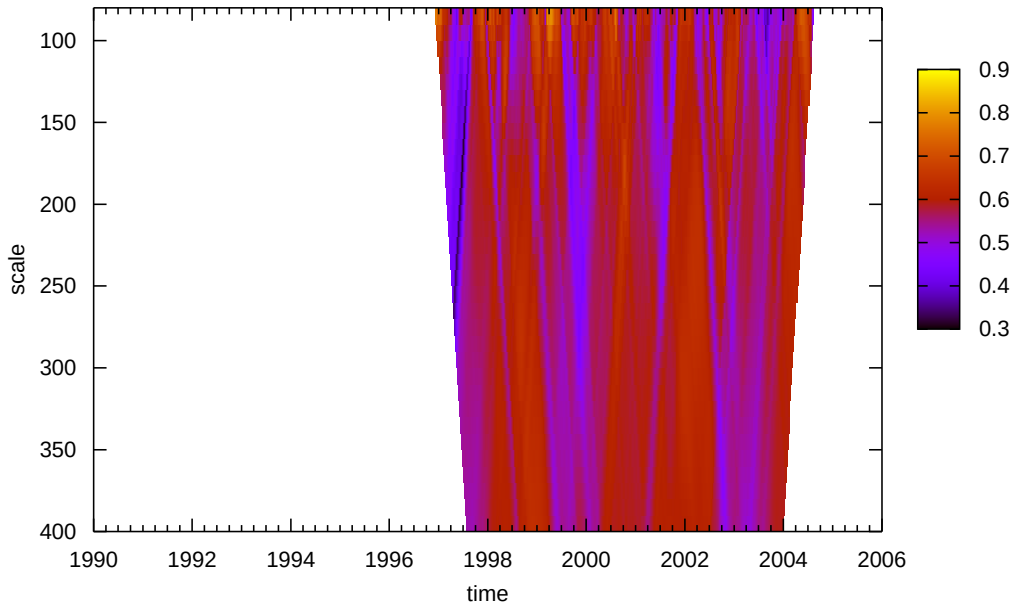




Argentina (MerVal)

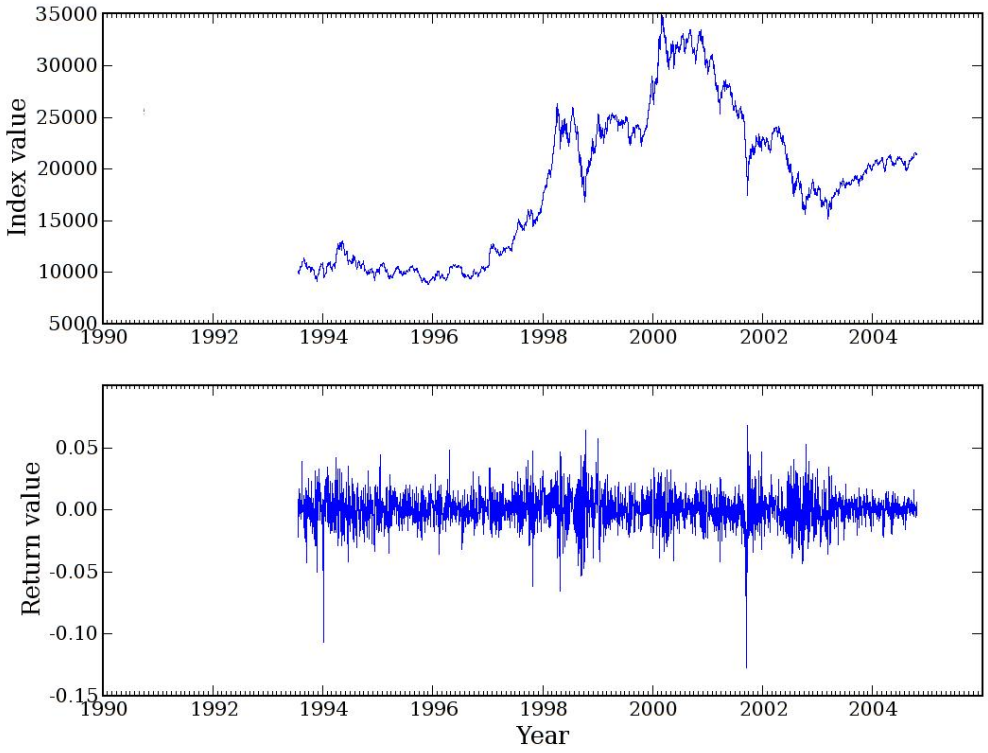
Lévy:$(\alpha, \beta) = (1.554, 0.008)$

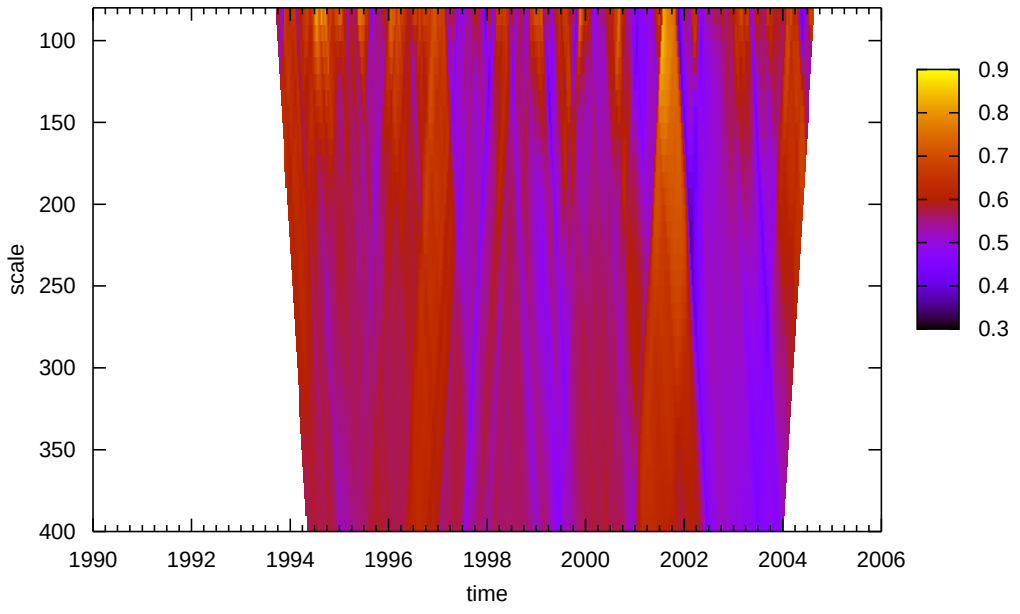




Italy (MIBTel)

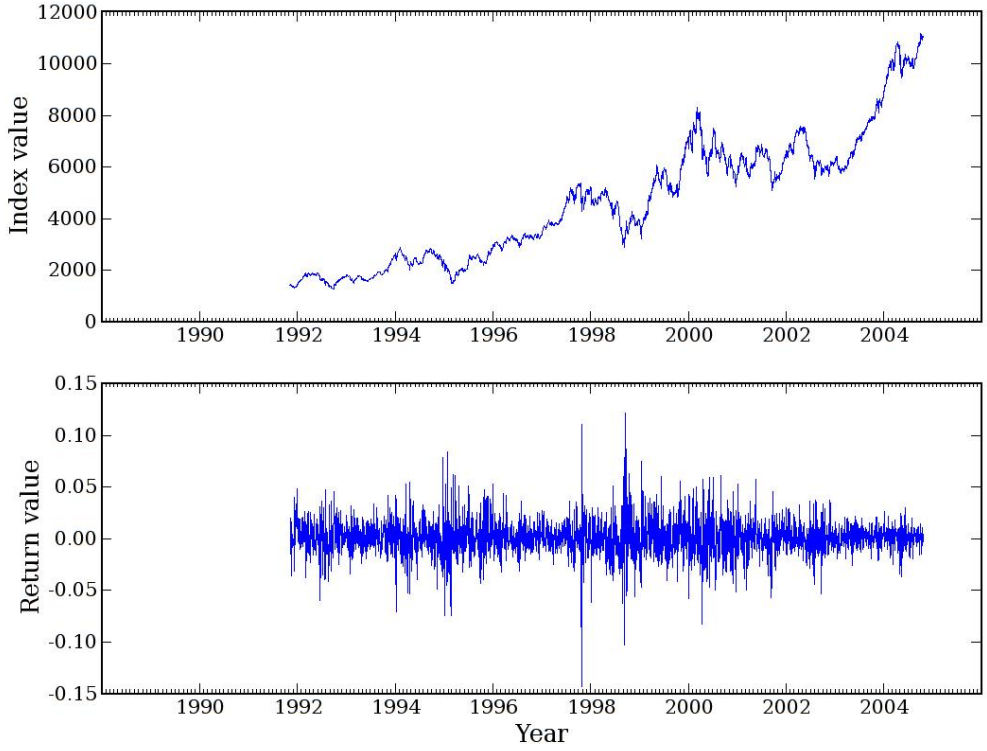
Lévy:$(\alpha, \beta) = (1.781, -0.002)$

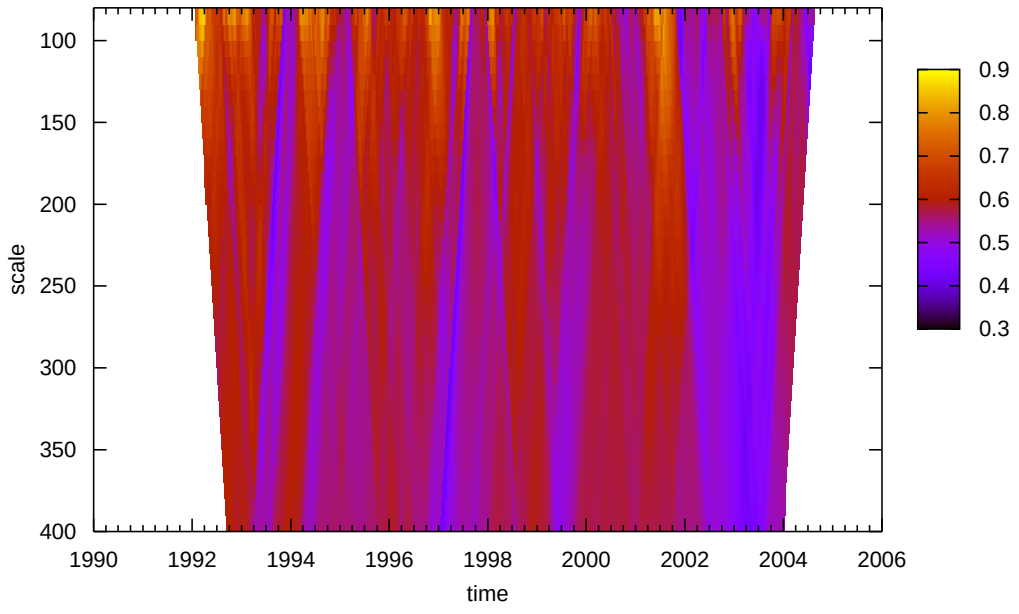




Mexico (IPC)

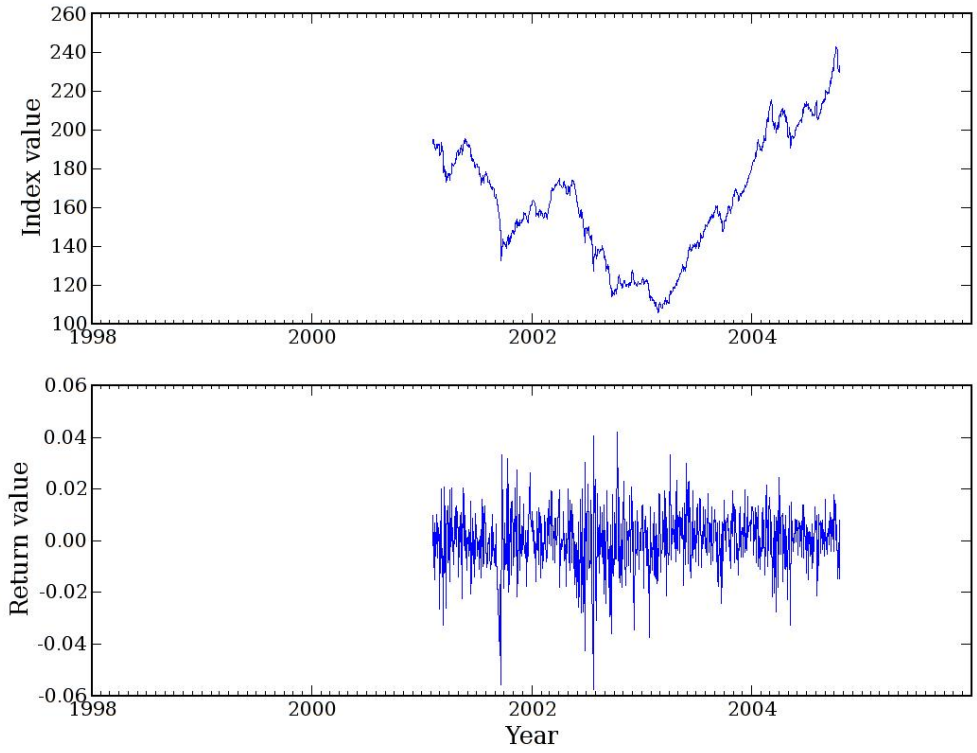
Lévy:$(\alpha, \beta) = (1.685, -0.005)$

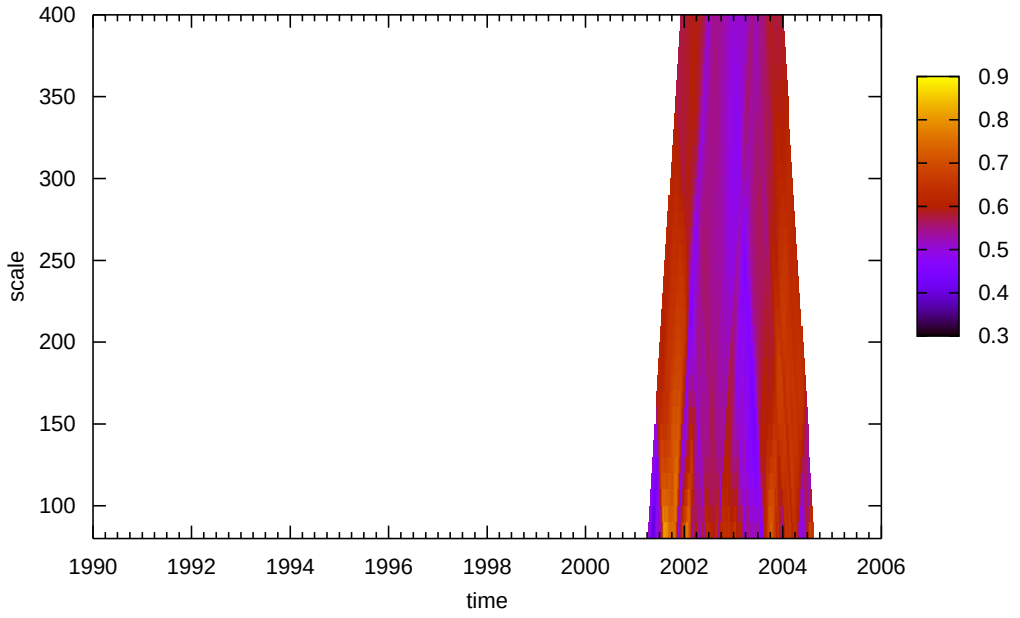




Norway (OSE All Share)

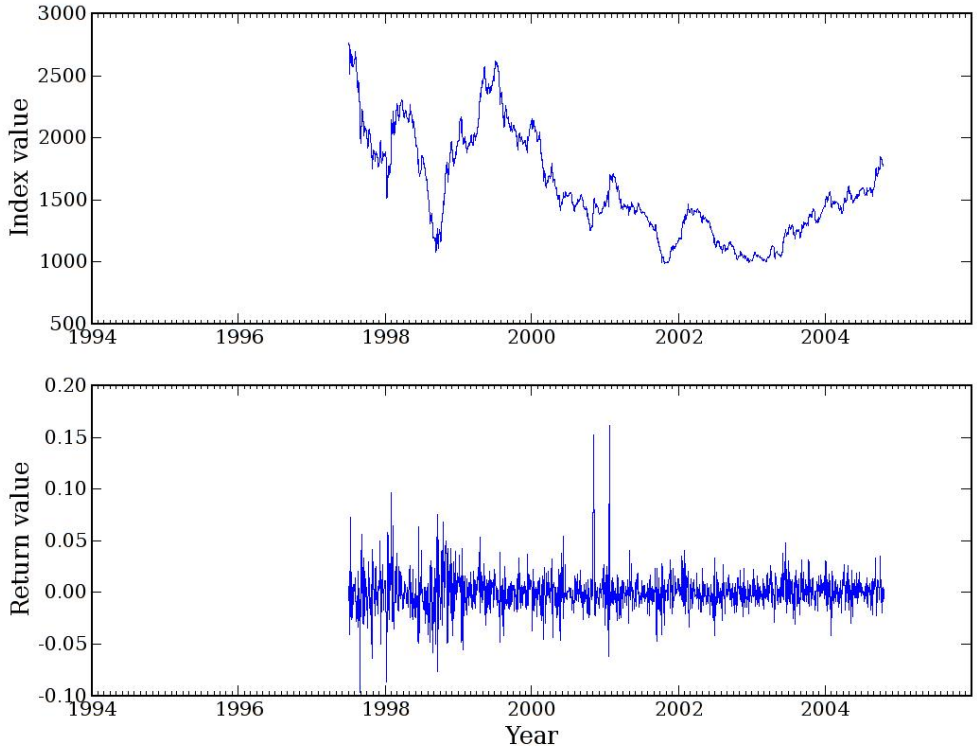
Lévy:$(\alpha, \beta) = (1.810, -0.002)$

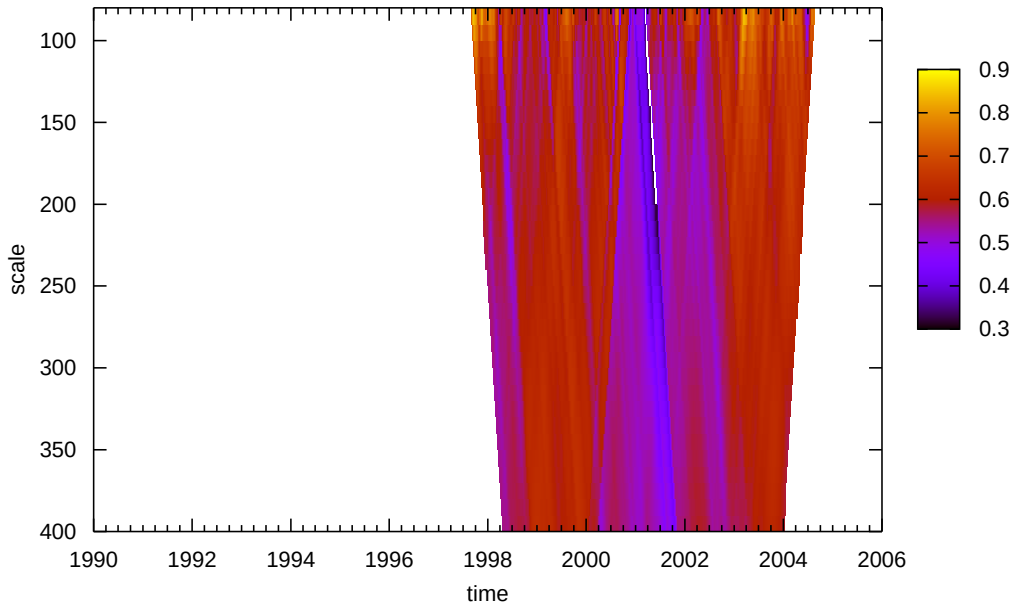




Philippines (PSE Composite)

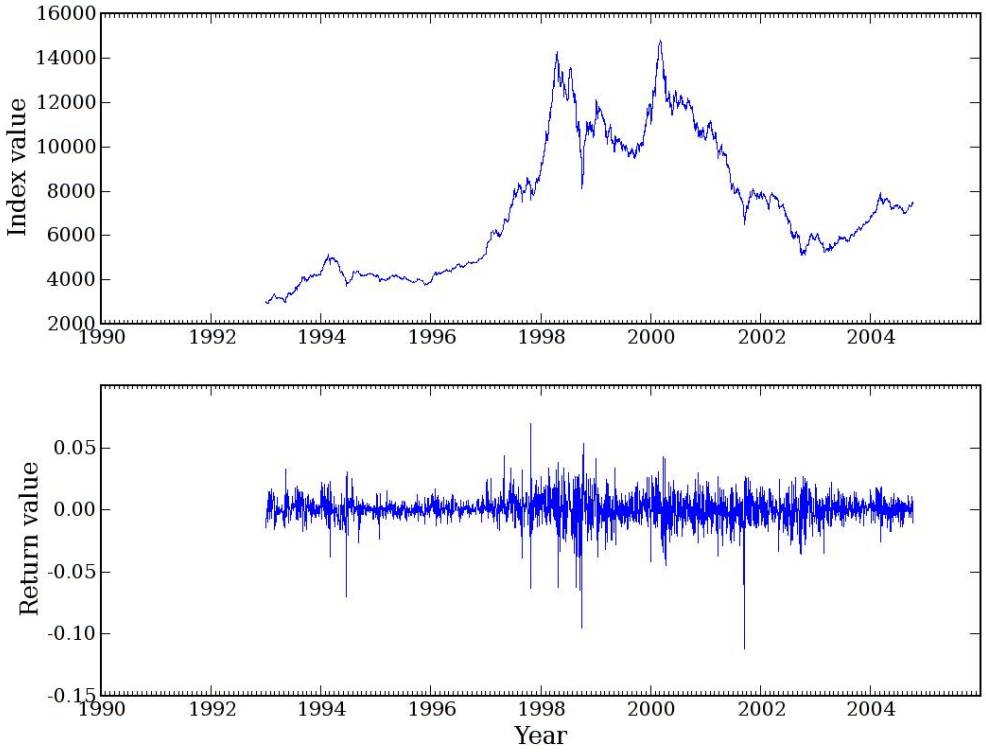
Lévy:$(\alpha, \beta) = (1.592, -0.003)$

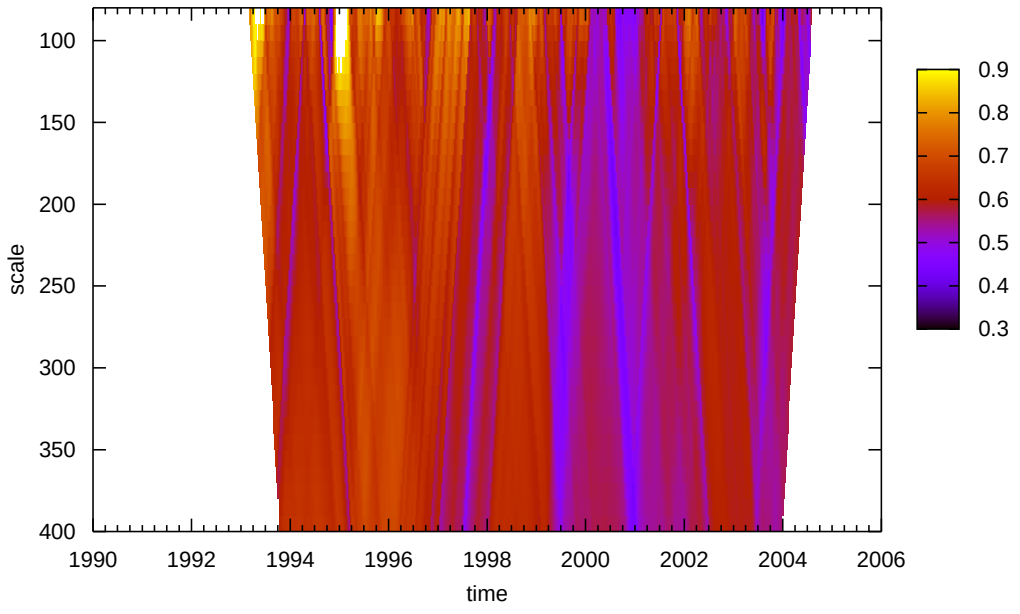




Portugal (PSI 20)

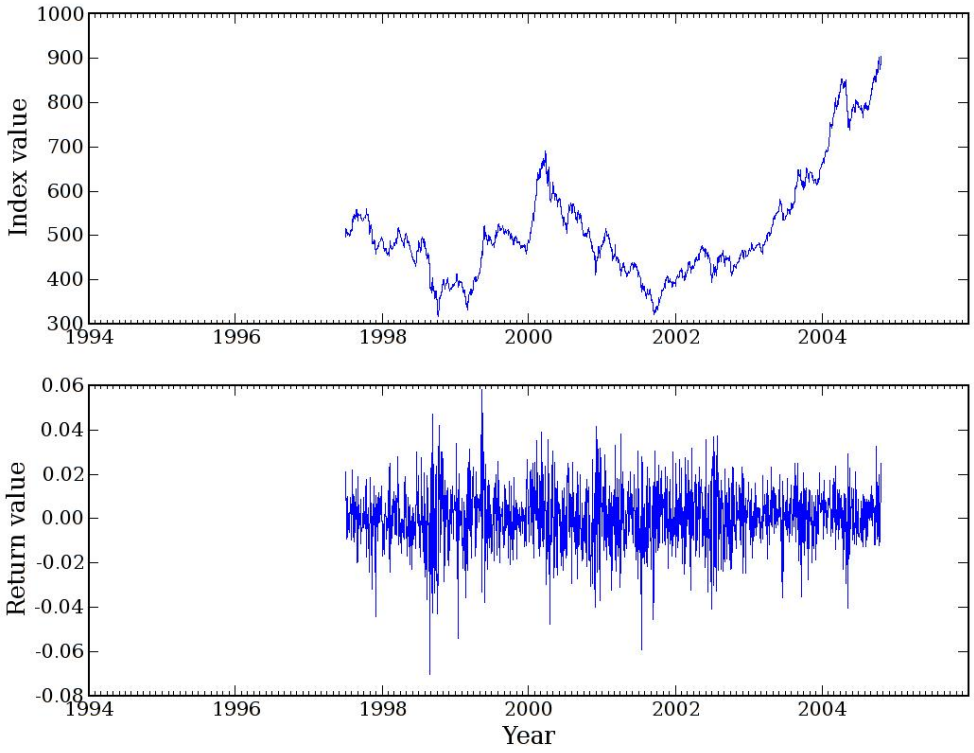
Lévy: $(\alpha, \beta) = (1.595, -0.001)$

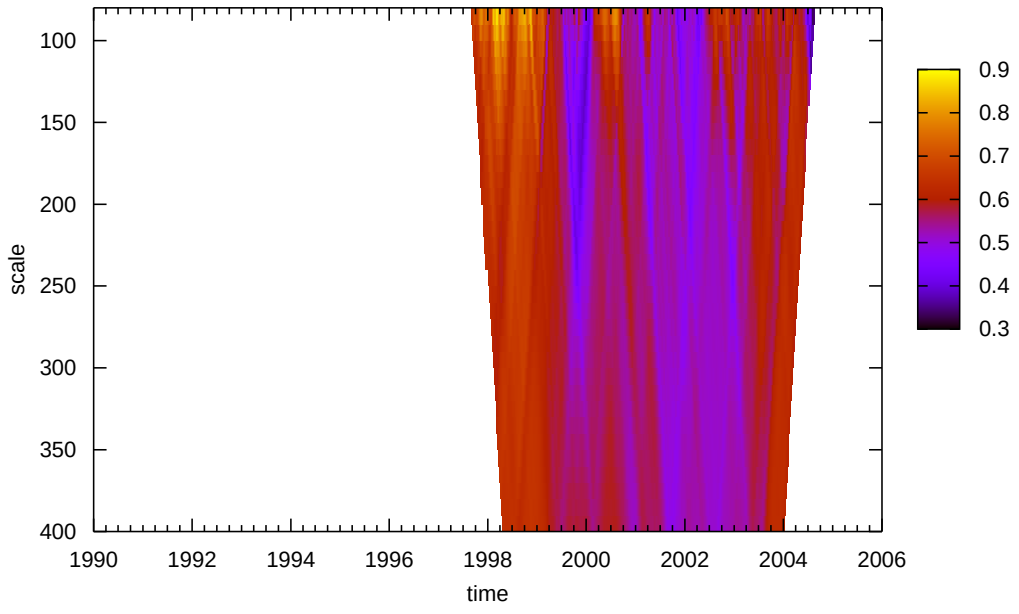




Czech Republic (PX50)

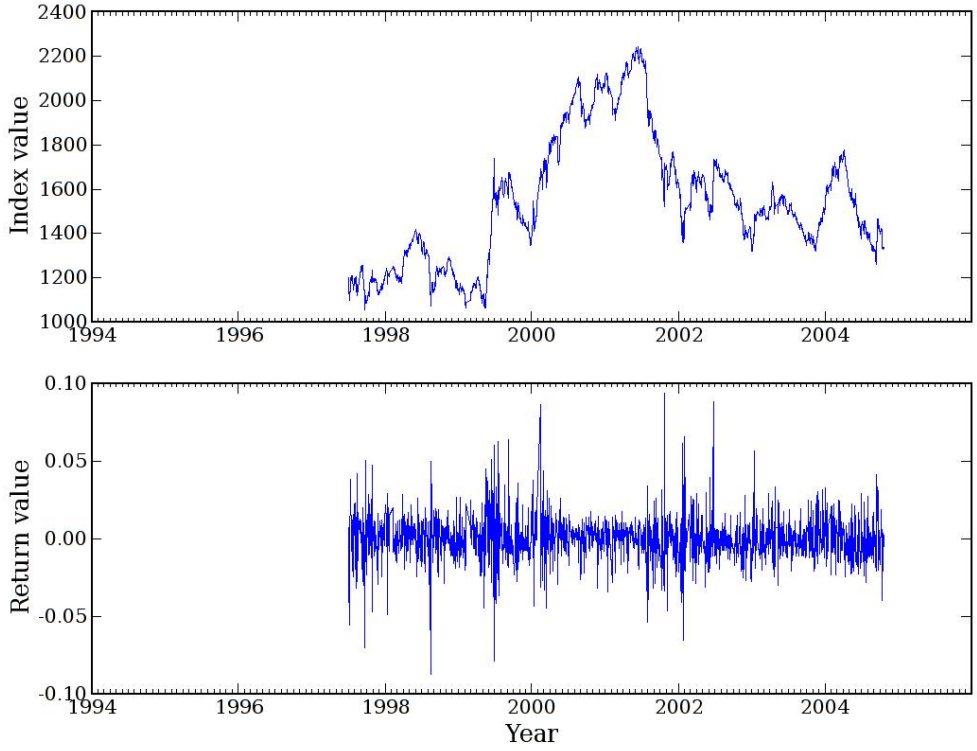
Lévy:$(\alpha, \beta) = (1.855, 0.288)$

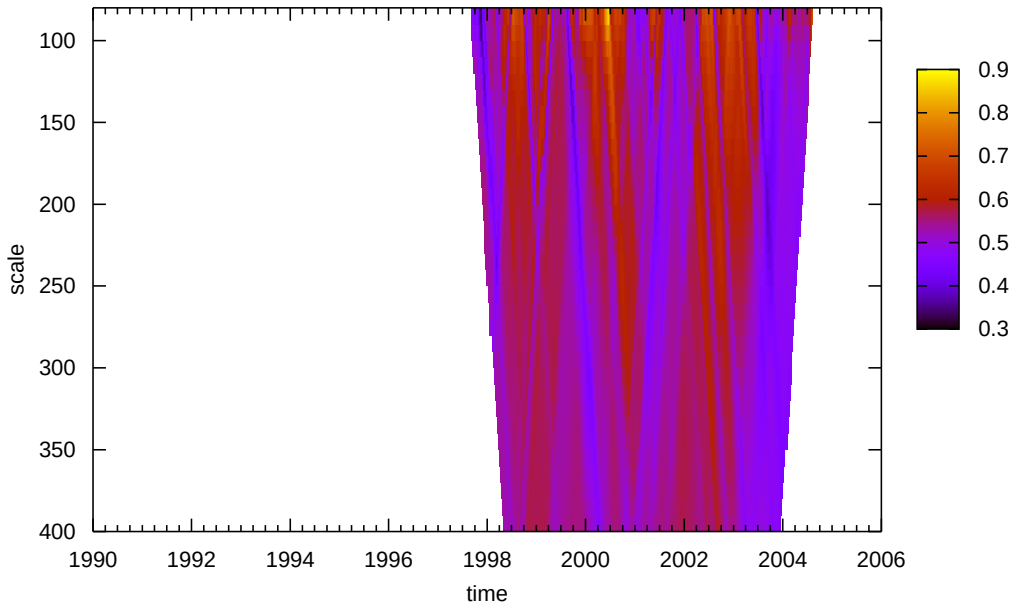




China (Shanghai Composite)

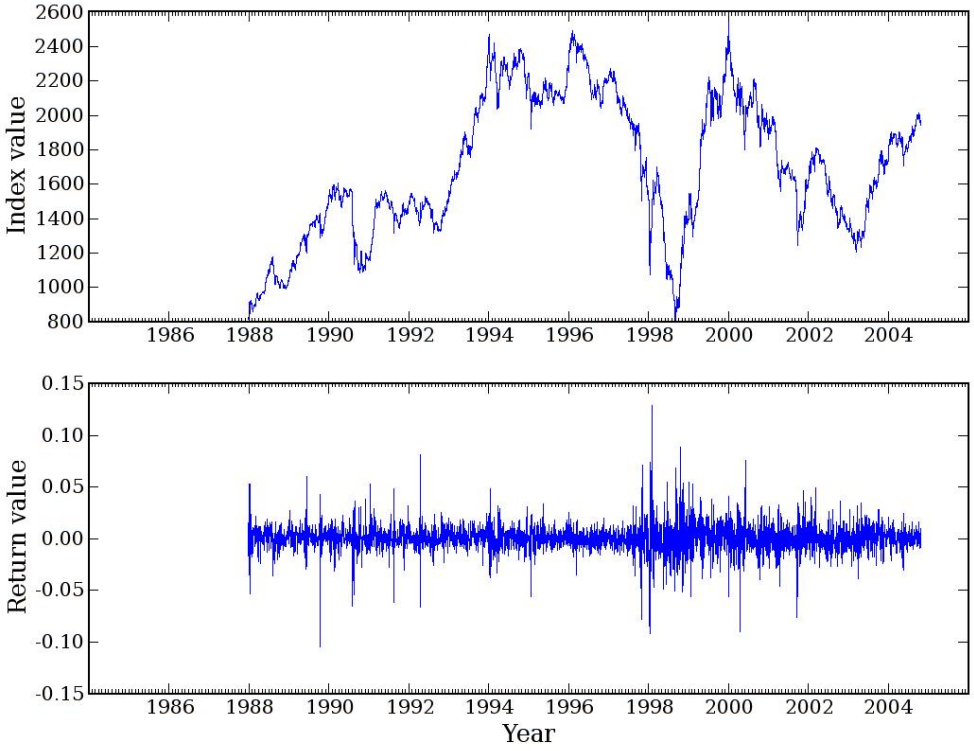
Lévy:$(\alpha, \beta) = (1.641, 0.010)$

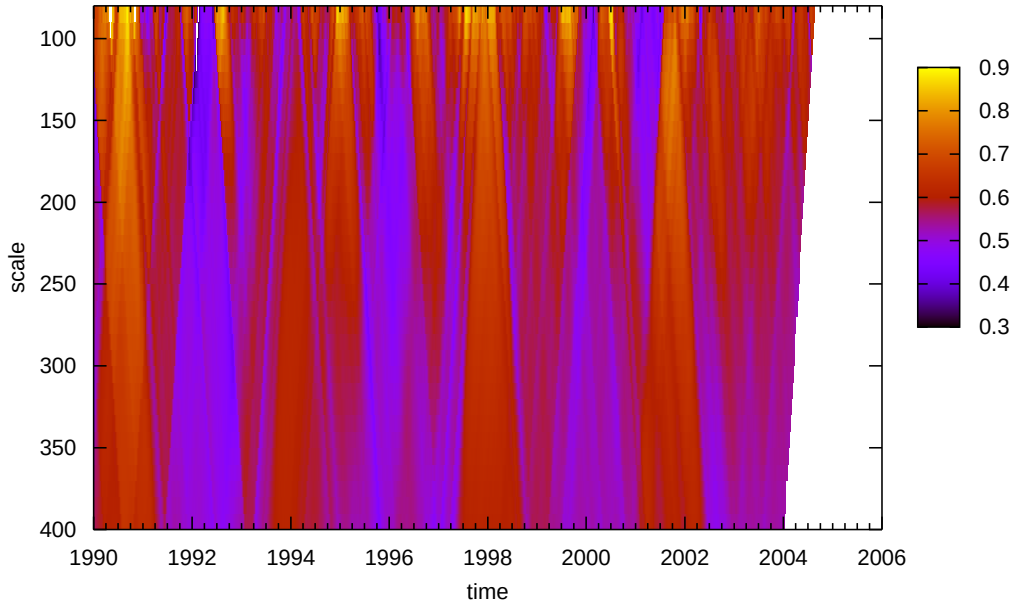




Singapore (Straits Times)

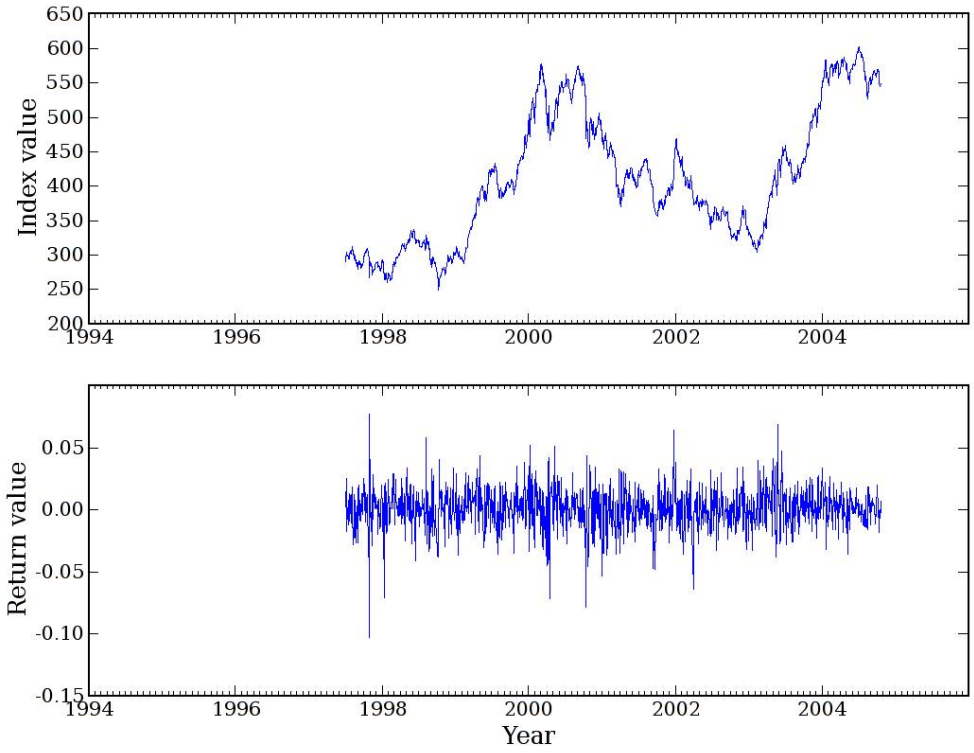
Lévy:$(\alpha, \beta) = (1.622, -0.008)$

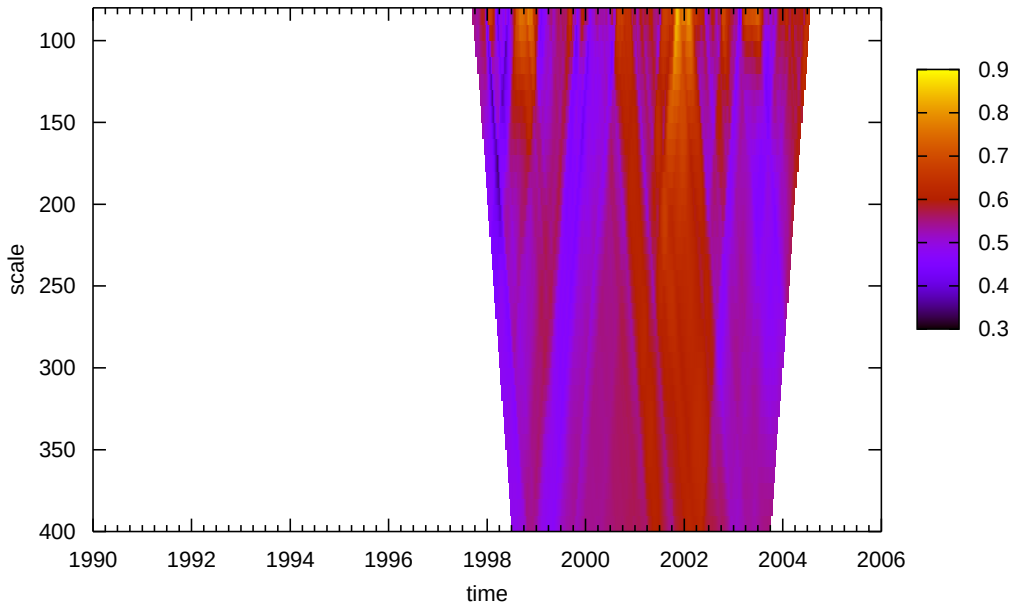




Israel (TA-100)

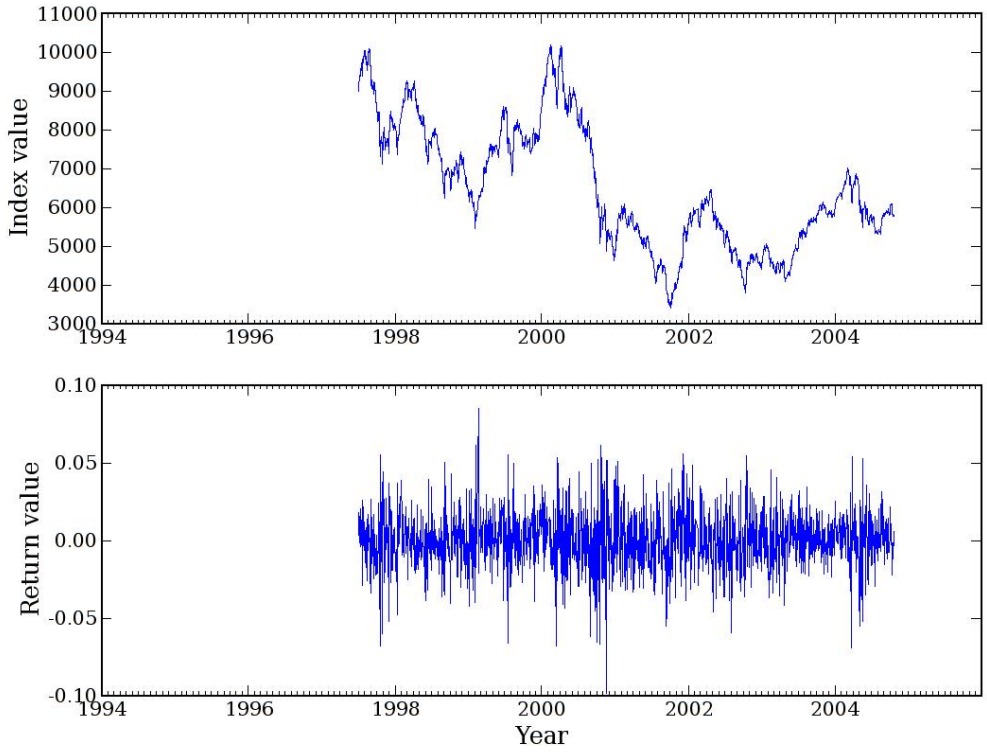
Lévy: $(\alpha, \beta) = (1.784, 0.017)$

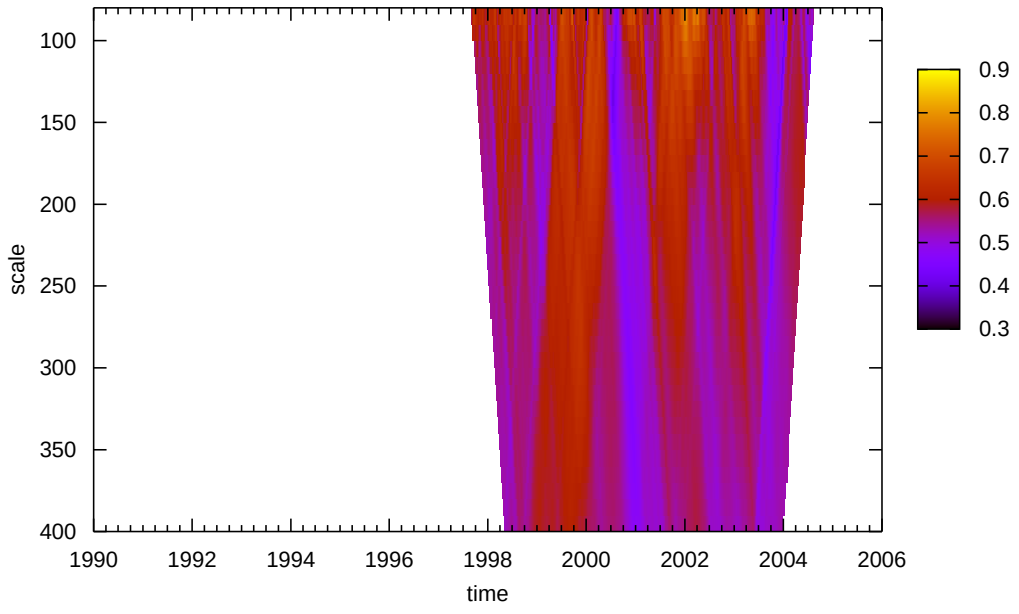




Taiwan (Taiwan Weighted)

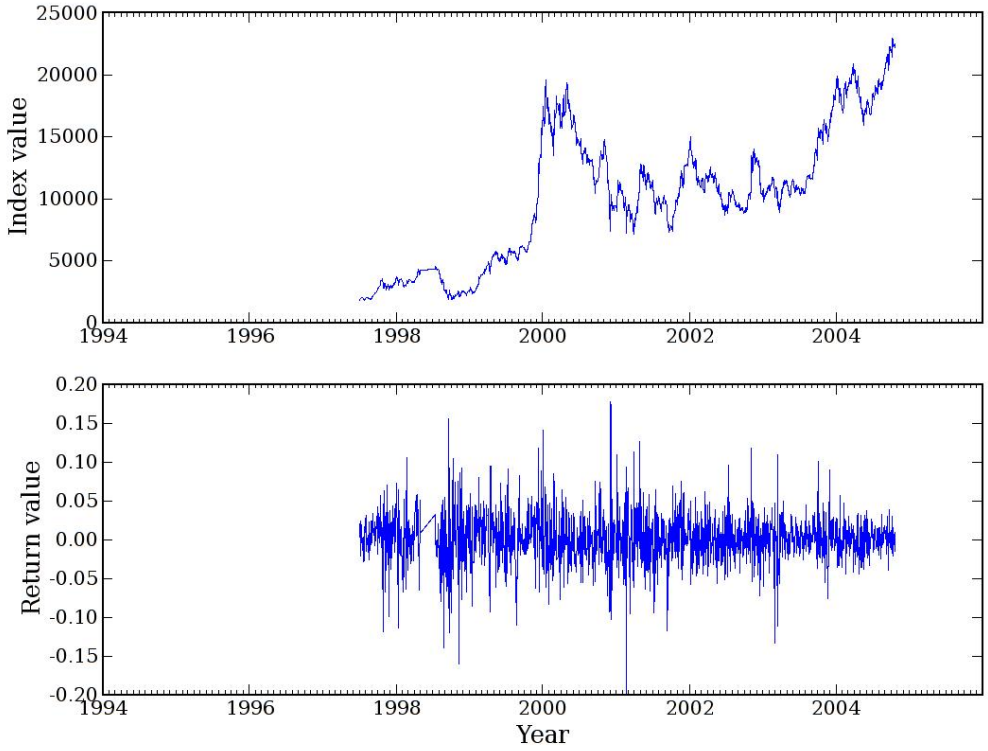
Lévy:$(\alpha, \beta) = (1.796, 0.001)$

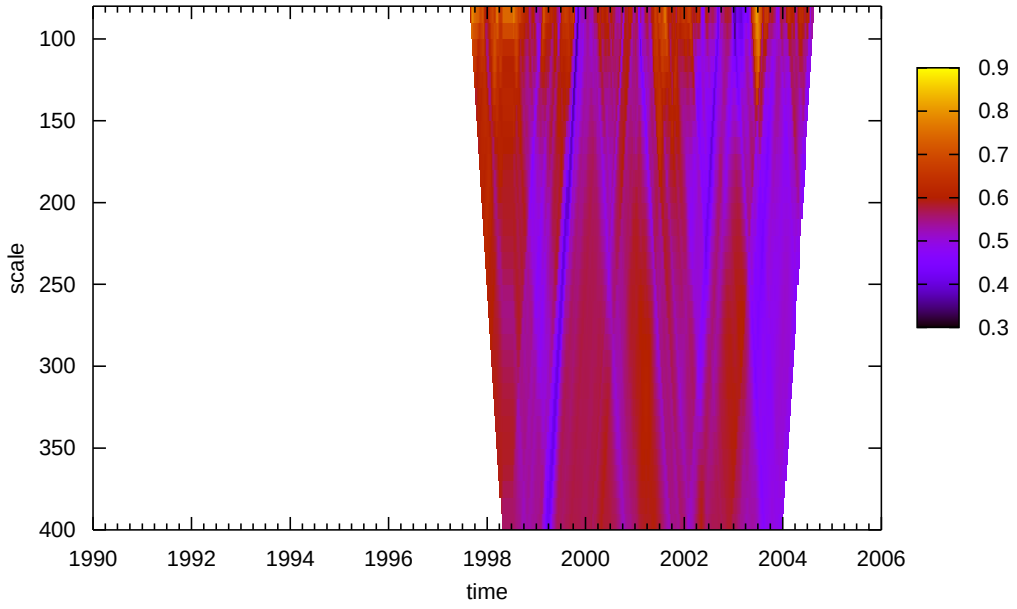




Turkey (ISE National-100)

Lévy:$(\alpha, \beta) = (1.700, -0.000)$





B. Stable Distributions

"*Three Rings for the Elven-kings under the sky,*
Seven for the Dwarf-lords in their halls of stone,
Nine for Mortal Men doomed to die,
One for the Dark Lord on his dark throne
In the Land of Mordor where the Shadows lie.
One Ring to rule them all, One Ring to find them,
One Ring to bring them all and in the darkness bind them
In the Land of Mordor where the Shadows lie." - J. R. R. Tolkien (The Lord of the Rings)

B.1. Statistical Distributions

The following statistical distributions are used in Section 2.8.

Example B.1.1. Gamma

$$\Gamma(x) = \int_0^{+\infty} t^{z-1} e^{-t} dt$$

For $n \in \mathbb{N}$, $\Gamma(n) = (n-1)!$

As we can see in Figure B.1 this function has discontinuities at negative integers where the function changes its sign.

Example B.1.2. Gaussian

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-\mu)^2}{2\sigma^2}), \qquad -\infty < x < \infty.$$

Example B.1.3. Cauchy

$$f(x) = \frac{1}{\pi} \frac{\gamma}{\gamma^2 + (x-\delta)^2}, \qquad -\infty < x < \infty.$$

Example B.1.4. Lévy distribution

$$f(x) = \sqrt{\frac{\gamma}{2\pi}} \frac{1}{(x-\delta)^{3/2}} \exp(-\frac{\gamma}{2(x-\delta)}), \qquad \delta < x < \infty$$

In Figure B.2 we can see the difference between the three families of stable distributions. For easier comparison the location and the scale parameters are the same.

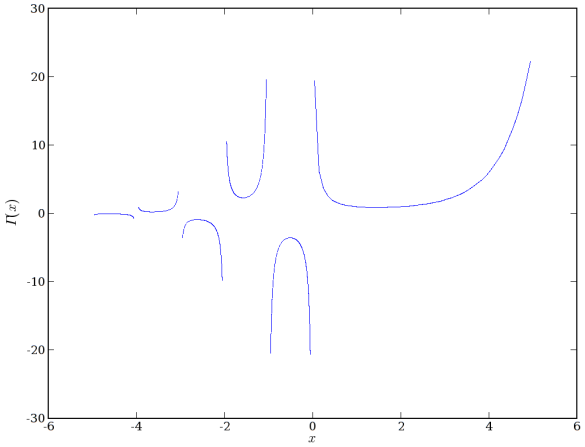


Figure B.1.: Gamma function

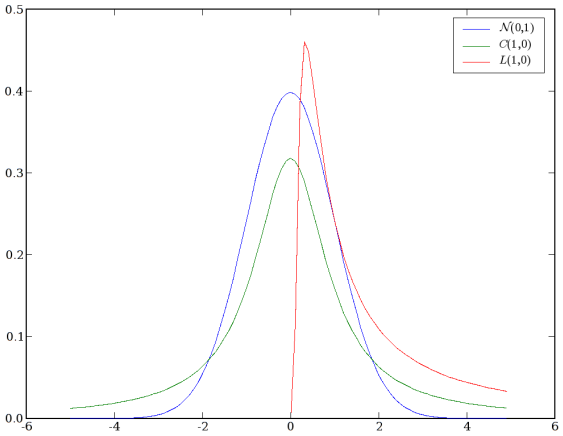


Figure B.2.: Compararison between the three stable distributions with closed formula

B.2. Stable distributions parametrisation

To uniquely identify a stable distribution we need four parameters:

- α *index of stability* or *characteristic exponent*, $0 < \alpha \leq 2$
- β skewness parameter, $-1 \leq \beta \leq 1$
- γ scale parameter, $\gamma > 0$
- δ location parameter, $\delta \in \mathbb{R}$.

There are several notations used, the traditional is $\mathbf{S}_\alpha(\gamma, \beta, \delta)$, while here we will use $\mathbf{S}(\alpha, \beta, \gamma, \delta; k)$, k is the kind of parametrisation.

B.2.1. Comparison between parametrisation

There are two reasons for the existence of different parametrisation: historical evolution and its use in different areas.

Parametrisation $k = 0$

Definition B.2.1. A random variable X is $\mathbf{S}(\alpha, \beta, \gamma, \delta; 0)$ if

$$X \stackrel{d}{=} \begin{cases} \gamma(Z - \beta \tan \frac{\pi\alpha}{2}) + \delta & \alpha \neq 1 \\ \gamma Z + \delta & \alpha = 1 \end{cases}$$

where $Z = Z(\alpha, \beta)$, (defined in relation 2.77).

This parametrisation is used in numerical work or fit data (statistical inference). It has the simplest form of the characteristic function that is continuous in all parameters and lets α and β to determine the shape of the distribution, while γ and δ determine the scale and location in the standard way. If $X \sim \mathbf{S}(\alpha, \beta, \gamma, \delta; 0)$ then $\frac{(X-\delta)}{\gamma} \sim \mathbf{S}(\alpha, \beta, 1, 0; 0)$.

Parametrisation $k = 1$

Definition B.2.2. A random variable X is $\mathbf{S}(\alpha, \beta, \gamma, \delta; 1)$ if

$$X \stackrel{d}{=} \begin{cases} \gamma Z + \delta & \alpha \neq 1 \\ \gamma Z + (\delta + \beta \frac{2}{\pi} \gamma \log \gamma) & \alpha = 1 \end{cases}$$

where $Z = Z(\alpha, \beta)$.

This parametrisation has the simple algebraic properties of distribution and the simple form of the characteristic function. This parametrisation is the most used when studying distribution properties.

The distribution is standardised when $\gamma = 1$, $\delta = 0$. We use the shorthand notation $\mathbf{S}(\alpha, \beta; 0)$ and $\mathbf{S}(\alpha, \beta; 1)$ respectively. We can relate the different parametrisations as

$$Z(\alpha, \beta) = \mathbf{S}(\alpha, \beta, 1, -\beta \tan \frac{\pi\alpha}{2}; 0) = \mathbf{S}(\alpha, \beta, 1, 0; 1).$$

Notice that parametrisation 0 and 1 are equal if $\beta = 0$, i.e if the distribution is symmetrical.

Parametrisation $k = 2$

There is a third parametrisation that is used to study of analytical properties of strictly stable laws. It distincts from the previous since the location parameter is at the mode, the scale parameter agrees with the standard scale parameter in the Gaussian and Cauchy cases. Technically it is more cumbersome but also more intuitive for applications.

B.2.2. Densities and distribution functions

There are no explicit formulae for general stable densities.

Theorem B.2.3. *All (non-degenerate) stable distributions are continuous distributions with an infinitely differentiable density.*

In what follows we use the following notation:

$f(x|\alpha, \beta, \gamma, \delta; k)$ probability density function,

$F(x|\alpha, \beta, \gamma, \delta; 0)$ cumulative density function.

Lemma B.2.4. *The support of a stable distribution in the different parametrisation is*

$$supp\, f(x|\alpha, \beta, \gamma, \delta; 0) = \begin{cases} [\delta - \gamma \tan \frac{\pi\alpha}{2}, +\infty) & \alpha < 1 \text{ and } \beta = 1 \\ (-\infty, \delta + \gamma \tan \frac{\pi\alpha}{2}) & \alpha < 1 \text{ and } \beta = -1 \\ (-\infty, +\infty) & \text{otherwise} \end{cases}$$

$$supp\, f(x|\alpha, \beta, \gamma, \delta; 1) = \begin{cases} [\delta, +\infty) & \alpha < 1 \text{ and } \beta = 1 \\ (-\infty, \delta) & \alpha < 1 \text{ and } \beta = -1 \\ (-\infty, +\infty) & \text{otherwise} \end{cases}$$

Proposition B.2.5. ***Reflection property***. *For any α and β, $Z \sim \mathbf{S}(\alpha, \beta; k)$, $k = 0, 1, 2$*

$$Z(\alpha, -\beta) \stackrel{d}{=} -Z(\alpha, \beta).$$

Therefore the density and distribution function of $Z(\alpha, \beta)$ satisfy $f(x|\alpha, \beta; k) = f(-x|\alpha, -\beta; k)$ and $F(x|\alpha, \beta; k) = 1 - F(-x|\alpha, -\beta; k)$.

Proposition B.2.6. *When $1 < \alpha \leq 2$, the mean of $X \sim \mathbf{S}(\alpha, \beta, \gamma, \delta_k; k)$, for $k = 0, 1, 2$, is*

$$\mu = E[X] = \delta_1 = \delta_0 - \beta\gamma_0 \tan\frac{\pi\alpha}{2}.$$

C. Software

> "*Ken Thompson, co-inventor of Unix, is reported to have uttered the epigram "When in doubt, use brute force". He probably intended this as "a ha ha only serious", but the original Unix kernel's preference for simple, robust, and portable algorithms over brittle "smart" ones does seem to have been a significant factor in the success of that OS. Like so many other tradeoffs in software design, the choice between brute force and complex, finely-tuned cleverness is often a difficult one that requires both engineering savvy and delicate esthetic judgment.*" - Eric S. Raymond (The Jargon File)

The author has developed a Python module for analysing the financial time series, as follows.

One of the approaches of this module was to hide some of the low levels details within this implementation. This allowed us to change the internal implementation several times while retaining the client code unmodified due to the stable API (application program interface).

Listing C.1: Module information

```
1  # Copyright (C) 2005-2006 José Matos <jamatos@fc.up.pt>
2  #
3  # This program is free software; you can redistribute it and/or
4  # modify it under the terms of the GNU General Public License
5  # as published by the Free Software Foundation; either version 2
6  # of the License, or (at your option) any later version.
7  #
8  # This program is distributed in the hope that it will be useful,
9  # but WITHOUT ANY WARRANTY; without even the implied warranty of
10 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
11 # GNU General Public License for more details.
12 #
13 # You should have received a copy of the GNU General Public License
14 # along with this program; if not, write to the Free Software
15 # Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA
      02111-1307, USA.
16
17 __all__ = ["data", "tools", "correlation", "dfa", "entropy", "
      multifractal", "rmd"]
```

Listing C.2: TimeSeries: correlation

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # Copyright (C) 2006 José Matos <jamatos@fc.up.pt>
4  #
5  # This program is free software; you can redistribute it and/or
```

```
6  # modify it under the terms of the GNU General Public License
7  # as published by the Free Software Foundation; either version 2
8  # of the License, or (at your option) any later version.
9  #
10 # This program is distributed in the hope that it will be useful,
11 # but WITHOUT ANY WARRANTY; without even the implied warranty of
12 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 # GNU General Public License for more details.
14 #
15 # You should have received a copy of the GNU General Public License
16 # along with this program; if not, write to the Free Software
17 # Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA
       02111-1307, USA.
18
19 # José Matos
20 # FCUP - Porto
21 # 2006/05/12
22
23 import numpy as N
24 from numpy.linalg import Heigenvalues
25 import datetime
26
27 # Todo:
28 # * Use an helper function to say what are the possible dates for a
       given set of markets
29
30 def cor_pair(period, m1, m2, R, T):
31     wsum, csum, samples = .0, .0, 0
32     for i in range(T+1):
33         w = R**-(T-i)
34         time_step = period[i][1]
35         if m1 in time_step and m2 in time_step:
36             wsum += w
37             csum += w*time_step[m1]*time_step[m2]
38             samples += 1
39
40     if samples:
41         return csum/wsum
42     else:
43         #print period[0][0], m1, m2
44         return .0
45
46
47 def correlation_matrix(period, markets, R, T):
48     lm = len(markets)
49     cor = N.zeros((lm,lm),N.Float)
50     for i in range(lm):
51         for j in range(i):
52             cor[i,j] = cor_pair(period, markets[i],markets[j], R, T)
53             cor[j,i] = cor[i,j]
54         cor[i,i] = cor_pair(period, markets[i],markets[i], R, T)
55
56     return cor
57
58
59 def correlation(series_collection, markets, R= 0.9, T= 20):
60
61     vals = sorted(series_collection.items())
62     # initial seed
63     period = (None,) + tuple(vals[:T])
64
65     eigen = []
66     date = []
67     for d in vals[T:]:
68         period = period[1:] + (d, )
```

```
69          date.append(d[0])
70          cor = correlation_matrix(period, markets, R, T)
71          eigen.append(sorted(Heigenvalues(cor), reverse=True))
72
73      return date, N.array(eigen)
```

Listing C.3: TimeSeries: data

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # Copyright (C) 2004-2006 José Matos <jamatos@fc.up.pt>
4  #
5  # This program is free software; you can redistribute it and/or
6  # modify it under the terms of the GNU General Public License
7  # as published by the Free Software Foundation; either version 2
8  # of the License, or (at your option) any later version.
9  #
10 # This program is distributed in the hope that it will be useful,
11 # but WITHOUT ANY WARRANTY; without even the implied warranty of
12 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 # GNU General Public License for more details.
14 #
15 # You should have received a copy of the GNU General Public License
16 # along with this program; if not, write to the Free Software
17 # Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA
       02111-1307, USA.
18
19 # José Matos
20 # School of Computing, DCU
21 # 2004/10/08
22
23 """ The goal of this module is centralize data access trough a single
24 point, and hide the implementation details in a single place, in turn
25 this avoids the different copies available everywhere otherwise."""
26
27 import os
28 from os import path
29 import sys
30 import numpy as N
31 from glob import glob
32 import tools
33
34 ##
35 # Private part: implementation, the details are subject to change
36 #
37 # To make this process more general we could read this location from a
38 # hidden file in $HOME, for the moment this is fix
39
40 __data_dir = "%s/research/econophysics/data/indices/" % os.getenv('HOME
       ')
41 __doc_dir = "%s/research/econophysics/data/doc/" % os.getenv('HOME')
42
43 def __read_from(name, col, sep=","):
44     filename = __data_dir + name + '.csv'
45     return [float(line[:-1].split(sep)[col]) for line in
46             open(filename) if line[:1] != "#"]
47
48
49 # We need some way of getting automatically the information from the
50 # different styles from the data files. Perhaps placing this in the
51 # header in the top of file.
52 def __read_dates_from(name):
53     "Returns the value at closing from market _name_"
54     filename = __data_dir + name + '.csv'
```

```
55      return [(int(line[:4]), int(line[4:6]), int(line[6:8]))
56              for line in open(filename) if line[:1] != "#"]
57
58  ##
59  # Public part
60  #
61  def read(name):
62      "This should take care of all the details related with data."
63      return N.array(__read_from( name, 6, ','), N.Float)
64
65
66  def read_dates(name):
67      "This return the available dates for each market"
68      return __read_dates_from( name)
69
70
71  def markets():
72      "Returns a list of the available markets with data."
73      names = sorted(glob(__data_dir + '*.csv'))
74      return [path.basename(name).replace(".csv","") for name in names]
75
76
77  def get_markets_info():
78      "Return a dictionary of markets information indexed by the market
            name"
79      info = {}
80      field = ["tick","name","country","location","state"]
81      filename = __doc_dir + "data.csv"
82      vals = [line[:-1].split(',') for line in open(filename) if line[:1]
             != "#"]
83      for line in vals:
84          market = line[0].lower().replace('^','')
85          part = {}
86          for i, name in enumerate(field):
87              part[name] = line[i].replace('"','')
88          part["market"] = market
89          info[market] = part
90
91      return info
92
93
94  def interesting_markets():
95      """ Returns a list of markets with interesting
96      properties, it usual is a placeholder to be later replaced by
97      markets."""
98      return 'bvsp', 'ftse', 'isct', 'iseq', 'isci', 'n225', 'psi20', '
            gspc'
99
100
101 def close_series(name):
102     """ Returns a list whose members are a tuple with the date and a
            value,
103     this time series is sorted."""
104     return sorted(zip(read_dates(name), read(name)))
105
106
107 def close_returns(name):
108     """ Returns a list whose members are a tuple with the date and a
109     return value, this time series is sorted."""
110     return sorted(zip(read_dates(name)[1:], tools.returns(read(name))))
111
112
113 def get_returns_collection(markets, begin = None, end = None):
114     # value of returns
115     rets = {}
```

```
116
117     for m in markets:
118         for date, val in close_returns(m):
119             if begin and date < begin:
120                 continue
121             if end and date > end:
122                 continue
123             if date in rets:
124                 rets[date][m] = val
125             else:
126                 rets[date] = {m: val}
127
128     return rets
129
130
131 def cutter(x, begin= None, end= None):
132     """ Returns a list of values that follow inside of the given
        interval."""
133
134     if begin:
135         for date, value in x:
136             if date >= begin:
137                 y = (date, value)
138
139     else:
140         if not end:
141             return x
```

Listing C.4: TimeSeries: DFA

```
1  # Copyright (C) 2005-2006 José Matos <jamatos@fc.up.pt>
2  #
3  # This program is free software; you can redistribute it and/or
4  # modify it under the terms of the GNU General Public License
5  # as published by the Free Software Foundation; either version 2
6  # of the License, or (at your option) any later version.
7  #
8  # This program is distributed in the hope that it will be useful,
9  # but WITHOUT ANY WARRANTY; without even the implied warranty of
10 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
11 # GNU General Public License for more details.
12 #
13 # You should have received a copy of the GNU General Public License
14 # along with this program; if not, write to the Free Software
15 # Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA
     02111-1307, USA.
16
17 from tools import fit
18 import numpy as N
19 import math
20
21
22 def _interval_residual(slice):
23     return fit(N.arange(len(slice), dtype=N.Float), slice, True)[2]
24
25
26 def _F2(t, x):
27     sum = 0.0
28     for i in range(len(x)-t+1):
29         sum += _interval_residual(x[i:i+t])
30
31     return sum/(len(x)-t+1)/t
32
33
```

```
34 def sample_size(n, min_box = 0, max_box = 0):
35     if not min_box: min_box = 4
36     if not max_box: max_box = n/4
37     log_scale = 1.2
38     res = [min_box]
39     cand = min_box
40     while res[-1] < max_box:
41         tmp = cand
42         cand *= log_scale
43         while int(cand) == int(tmp):
44             cand *= log_scale
45         res.append(int(cand))
46     return res
47
48
49 def dfa(y, sizes = None):
50     n = len(y)
51     if sizes == None:
52         sizes = sample_size(n)
53
54     result = N.sqrt([_F2(i, y) for i in sizes])
55     ny = N.log(result)
56     (a,b), res, sqr2= fit(N.log(sizes), ny, True)
57     sxx, sxy, syy = res[0,0], res[1,0], res[1,1]
58     sy  = N.sum(ny)
59     syy = N.sum(ny*ny)
60     return a, math.sqrt(1. - n*sqr2/(n*syy-sy*sy))
61
62
63 if __name__ == "__main__":
64     pass
```

Listing C.5: TimeSeries: entropy

```
1  #!/usr/bin/python
2  # Copyright (C) 2005-2006 José Matos <jamatos@fc.up.pt>
3  #
4  # This program is free software; you can redistribute it and/or
5  # modify it under the terms of the GNU General Public License
6  # as published by the Free Software Foundation; either version 2
7  # of the License, or (at your option) any later version.
8  #
9  # This program is distributed in the hope that it will be useful,
10 # but WITHOUT ANY WARRANTY; without even the implied warranty of
11 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
12 # GNU General Public License for more details.
13 #
14 # You should have received a copy of the GNU General Public License
15 # along with this program; if not, write to the Free Software
16 # Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA
       02111-1307, USA.
17
18 import numpy as N
19 import math
20 import TimeSeries
21 from TimeSeries.tools import diff, basic_statistics, max, min
22 from TimeSeries import data
23
24
25 #
       ##############################################################################
26 # Helper functions
27 #
```

```
28 def _entropy(seq, word_size):
29     frequency = {}
30     size = len(seq)
31 
32     step =1
33     total = size -word_size +1
34 
35     for i in range(0, size - word_size + 1, step):
36         word = seq[i:i + word_size]
37         if word not in frequency:
38             frequency[word] = 1
39         else:
40             frequency[word] += 1
41 
42     result = 0
43     nstates = 0
44 
45     for word, value in frequency.items():
46         nstates += 1
47         p = value/float(total)
48         result -= p*math.log(p)
49 
50     return result, nstates, total
51 
52 
53 def running_entropy(seq, word_size, window_length):
54     frequency = {}
55 
56     assert window_length <= len(seq)
57 
58     for i in range( window_length - word_size + 1):
59         word = seq[i:i + word_size]
60         if word not in frequency:
61             frequency[word] = 1
62         else:
63             frequency[word] += 1
64 
65     total = window_length - word_size + 1
66 
67     # evaluate entropy for the first block
68     result = 0
69     for word, value in frequency.items():
70         p = value/float(total)
71         result -= p*math.log(p)
72 
73     running_value = [result]
74 
75     i_min = 0
76     i_max = total
77     for i in range(total, len(seq) - word_size +1):
78         word_out = seq[i_min: i_min + word_size]
79         word_in  = seq[i_max: i_max + word_size]
80 
81         if word_in != word_out:
82 
83             # before
84             p_out = frequency[word_out]/float(total)
85             result += p_out * math.log(p_out)
86 
87             if word_in in frequency:
88                 p_in = frequency[word_in]/float(total)
89                 result += p_in * math.log(p_in)
90             else:
91                 frequency[word_in] = 0
92 
```

```
93              # after
94              frequency[word_out] -= 1
95              if frequency[word_out] == 0:
96                  del frequency[word_out]
97              else:
98                  p_out = frequency[word_out]/float(total)
99                  result -= p_out * math.log(p_out)
100
101             frequency[word_in] += 1
102             p_in = frequency[word_in]/float(total)
103             result -= p_in * math.log(p_in)
104
105         running_value.append(result)
106
107         i_min += 1
108         i_max += 1
109
110     return running_value
111
112
113 def frequency_counter(seq):
114     frequency = {}
115     for item in seq:
116         if item not in frequency:
117             frequency[item] = 1
118         else:
119             frequency[item] += 1
120     return frequency.items()
121
122
123 def partition(seq, bins, min = 0, max = 0):
124     # Take a copy of the sequence, so that we don't change it.
125     tmp_seq = seq.copy()
126     if max == min:
127         max = N.maximum.reduce(seq)
128         min = N.minimum.reduce(seq)
129
130     # the delta factor is used to avoid have the maximum in a single
        category
131     max += (max -min) * 1e-8
132
133     tmp_seq -= min
134     tmp_seq *= bins/(max - min)
135
136     return tuple(tmp_seq.astype(N.Int))
137
138
139 def centre_sequence(seq, n):
140     left = n/2
141     right = n -left
142     return seq[left:-right]
143
144
145 #
    ##############################################################################

146 # Class area
147 #
148 class Entropy:
149     def __init__(self, seq, upper_lim = 30):
150         self.seq = seq
151         self.upper_lim = upper_lim
152
153
154     def shift(self, n):
```

```
155         return tuple(self.seq[-n:]) + tuple(self.seq[:-n])
156
157
158     def running_entropy(self):
159         result = []
160         for i in range(2, self.upper_lim):
161             val, nstates, total = _entropy(self.seq, i)
162             result.append([i, val, nstates, total])
163         return result
164
165
166     def __str__(self):
167         res = ""
168         for line in self.running_entropy():
169             for item in line:
170                 res += str(item) + '\t'
171             res += '\n'
172         return res
173
174
175 class Entropy_lim(Entropy):
176     def running_entropy(self):
177         result = []
178         i = 1
179         nstates, total = 0, 1
180
181         while nstates != total and i < len(self.seq):
182             i += 1
183             val, nstates, total = _entropy(self.seq, i)
184
185         if i < len(self.seq):
186             return i
187         assert 0
188
189
190 #
    ##############################################################################

191 # Test Area
192 #
193 def test_random():
194     import RandomArray
195     a = tuple(RandomArray.randint(0, 20, (10000,)))
196     print Entropy(a)
197
198
199 def test_granularity():
200     a = data.read("bvsp")
201     for i in (50, 100, 200, 400,800, 1600):
202         out = open('entropy-bvsp-%.2i.dat' % i, 'w')
203         print >> out, Entropy(partition(diff(N.log(a)), i))
204         out.close()
205
206
207 def test_saturation_limit():
208     for market in data.markets():
209         dat = data.read(market)
210         ret = diff(N.log(dat))
211         c = []
212         for res in range(3,20):
213             part = partition(ret, res)
214             c.append((res, Entropy_lim(part).running_entropy()))
215             out = open('histogram-%s-%.2i.dat' % (market, res), 'w')
216             for bin in frequency_counter(part):
217                 print >> out, '%d\t%d' % bin
```

```
218             out.close()
219             out = open('entropy-%s-%.2i.dat' % (market, res), 'w')
220             print >> out, Entropy(part)
221             out.close()
222         out = open('saturate-%s.dat' % market, 'w')
223         print >> out, c
224         out.close()
225
226
227 def test_statistics():
228     """ Basic statistics for different series."""
229     basic_stats = {}
230
231     for market in data.markets():
232         dat = data.read(market)
233         ret = diff(N.log(dat))
234         basic_stats[market] = basic_statistics(ret)
235
236     print "\t\t".join(['market', 'min', 'max', 'mean', 'median', 'std',
            'skewness', 'kurtosis'])
237     for market in data.markets():
238         print market + '\t\t',
239         for r in basic_stats[market]: print "%6lf\t" % (r,) ,
240         print
241     print "min =", min([basic_stats[market][0] for market in data.
            markets()])
242     print "max =", max([basic_stats[market][1] for market in data.
            markets()])
243
244
245 def test_partition():
246     a = N.arange(20., typecode = N.Float)
247     print partition(a, 10, -10, 30)
248
249
250 def test_uniffied_partition():
251     basic_stats = {}
252     ret = {}
253     for market in data.markets():
254         dat = data.read(market)
255         ret[market] = diff(N.log(dat))
256         basic_stats[market] = basic_statistics(ret[market])
257
258     low  = min([basic_stats[market][0] for market in data.markets()])
259     high = max([basic_stats[market][1] for market in data.markets()])
260
261     for market in data.markets():
262         for res in range(5,51, 5):
263             out = open('entropy-%s-%.2i.dat' % (market, res), 'w')
264             print >> out, Entropy(partition(ret[market], res))
265             out.close()
266
267
268 def test_running_entropy():
269     for market in data.markets():
270         dat = data.read(market)
271         ret = diff(N.log(dat))
272         date = data.read_dates(market)
273
274         part = partition(ret, 50)
275         ls = []
276         window = 100
277         for ws in range(2,10):
278             map = {}
279             for u,v in zip(centre_sequence(date, window),
```

```
                running_entropy(part, ws, window)):
280                 map[u] = v
281             ls.append(map)
282
283         date.reverse()
284         dict_merge(centre_sequence(date, window), ls, open("%s-%d.%d.
                dat" %(market, 50, window), 'w'))
285
286
287 def dict_merge(keys, maps, file):
288     for day in keys:
289         print >> file, "%d/%.2d/%.2d" % day,
290         for map in maps:
291             if day in map:
292                 print >> file, map[day],
293             else:
294                 print >> file, '?',
295         print >> file
296
297
298 def test_running_entropy_modulus():
299     for market in data.markets():
300         dat = data.read(market)
301         ret = N.fabs(diff(N.log(dat)))
302         date = data.read_dates(market)
303
304         part = partition(ret, 50)
305         ls = []
306         window = 100
307         for ws in range(2,10):
308             map = {}
309             for u,v in zip(centre_sequence(date, window),
                    running_entropy(part, ws, window)):
310                 map[u] = v
311             ls.append(map)
312
313         date.reverse()
314         dict_merge(centre_sequence(date, window), ls, open("abs-%s-%d.%
                d.dat" %(market, 50, window), 'w'))
315
316
317 def test_centre_sequence():
318     a = range(10)
319     print centre_sequence(a, 2)
```

Listing C.6: TimeSeries: multifractal

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3  # Copyright (C) 2005-2006 José Matos <jamatos@fc.up.pt>
4  #
5  # This program is free software; you can redistribute it and/or
6  # modify it under the terms of the GNU General Public License
7  # as published by the Free Software Foundation; either version 2
8  # of the License, or (at your option) any later version.
9  #
10 # This program is distributed in the hope that it will be useful,
11 # but WITHOUT ANY WARRANTY; without even the implied warranty of
12 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 # GNU General Public License for more details.
14 #
15 # You should have received a copy of the GNU General Public License
16 # along with this program; if not, write to the Free Software
17 # Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA
```

```
    02111-1307, USA.
18
19  # José Matos
20  # FCUP - Porto
21  # 2005/06/20
22
23  import sys
24  import csv
25  import math
26  import pylab
27  import numpy as N
28  import datetime
29  from tools import fit
30
31  def scale(T):
32      rs = [2]
33      t, t_old = 2, 1
34      mult = math.sqrt(2)
35
36      while t < T:
37          if t_old != int(t):
38              rs.append(int(t))
39          t_old = int(t)
40          t *= mult
41
42      return rs
43
44
45  def mom_diff(q, d, x):
46      T = len(x)
47      s_q = .0
48      for t in range(0,T-d):
49          s_q += abs(x[t+d] - x[t])**q
50
51      return s_q / (T-d)
52
53
54  def mom(q, x):
55      T = len(x)
56      m_q = .0
57      for t in x:
58          m_q += abs(t)**q
59
60      return m_q / T
61
62
63  def general_H(x, Q):
64      T = len(x)
65      D = scale(T/4)
66
67      mm = {}
68      xx = {}
69
70      for q in Q:
71          mm[q] = mom(q, x)
72          xx[q] = []
73
74      for d in D:
75          for q in Q:
76              xx[q].append(mom_diff(q, d, x) / mm[q])
77
78      H = [fit(N.log(D), N.log(xx[q])/q)[0] for q in Q]
79
80      # evaluate the regression coefficient
81      if False:
```

```
82          r = []
83          for q in Q:
84              params = fit(N.log(D), N.log(xx[q]))
85              sse = params[5]
86              sst = params[4]
87              r.append(1.-sse/sst)
88
89      return H
90
91
92  def av_dev(x, d, l, ld):
93      sum = 0.0
94      for i in range(l, ld):
95          sum += abs(x[i+d]-x[i])
96
97      return sum
98
99
100 def general_D(x, Q):
101     T = len(x)
102     D = [d for d in scale(T) if 2*d < T]
103
104     mr = {}
105     mlr = {}
106     xx = {}
107
108     for q in Q:
109         xx[q] = []
110
111     for d in D:
112         L = range(0,T-2*d)
113
114         #mlr[0] = av_dev(x, d, 0, d)
115         for l in L:
116             mlr[l] = av_dev(x, d, l, l+d)/d
117             #mlr[l] = mlr[l-1] - abs(x[l+d-1] - x[l-1]) + abs(x[l+2*d
                -1] - x[l+d-1])
118
119         sum = 0
120         for l in L:
121             sum += mlr[l]
122
123         mr[d] = sum/len(L)
124
125         for q in Q:
126             sum = 0.0
127             for l in L:
128                 sum += (mlr[l])**q
129             sum /= len(L)
130
131             xx[q].append(sum / mr[d]**q)
132
133     return N.array([-fit(N.log(D), N.log(xx[q]))[0] for q in Q])
```

Listing C.7: TimeSeries: tools

```
1  #!/usr/bin/python
2  # Copyright (C) 2005-2006 José Matos <jamatos@fc.up.pt>
3  #
4  # This program is free software; you can redistribute it and/or
5  # modify it under the terms of the GNU General Public License
6  # as published by the Free Software Foundation; either version 2
7  # of the License, or (at your option) any later version.
8  #
```

```
9  # This program is distributed in the hope that it will be useful,
10 # but WITHOUT ANY WARRANTY; without even the implied warranty of
11 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
12 # GNU General Public License for more details.
13 #
14 # You should have received a copy of the GNU General Public License
15 # along with this program; if not, write to the Free Software
16 # Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA
       02111-1307, USA.
17
18 import sys
19
20 import numpy as N
21 from numpy.linalg import singular_value_decomposition
22
23 import Scientific
24 import Scientific.IO
25 import Scientific.IO.ArrayIO
26 import Scientific.Statistics
27 import Scientific.Statistics.Histogram
28
29 import math
30
31 import pylab
32 import datetime
33 from scipy.optimize import leastsq
34
35 """
36 This package wants to ensure that all the operations returns a numeric.
       array
37 """
38
39 read_array = Scientific.IO.ArrayIO.readFloatArray
40 write_array = Scientific.IO.ArrayIO.writeArray
41 mean = Scientific.Statistics.mean
42 variance = Scientific.Statistics.variance
43 std = Scientific.Statistics.standardDeviation
44 median = Scientific.Statistics.median
45 skewness = Scientific.Statistics.skewness
46 kurtosis = Scientific.Statistics.kurtosis
47 correlation = Scientific.Statistics.correlation
48 max = pylab.max
49 min = pylab.min
50 fft = N.fft
51
52
53 def date2array(dates):
54     return N.array([pylab.date2num(datetime.date(*day)) for day in
           dates])
55
56
57 def get_date_value(dts):
58     dates, values = [], []
59     for day, value in dts:
60         dates.append(pylab.date2num(datetime.date(*day)))
61         values.append(value)
62
63     return N.array(dates), N.array(values)
64
65
66 def diff(ts):
67     "Returns the discrete derivative."
68     return N.array(ts[1:] - ts[:-1])
69
70
```

```
71  def returns(ts):
72      "Returns the returns, difference of log"
73      return diff(N.log(ts))
74
75
76  def poincare(ts):
77      "Returns the pair (x[t-1],x[t]) for each t in the time series x."
78      return N.array(zip(ts[1:],ts[:-1]))
79
80
81  # This is the same as abs(fft(self))
82  def power(ts):
83      "Returns the square of the module."
84      return (N.conjugate(ts)*ts).real
85
86
87  def svd(ts, n= 25):
88      "Returns the result of the singular value decomposition."
89      comp = [ ts[i+j] for j in range(len(self)-n)  for i in range(n)]
90      comp = N.array(comp)
91      comp = N.reshape(comp, (-1, n))
92      return singular_value_decomposition(comp)
93
94
95  def dispersion(ts):
96      "Returns the pair (x[t],x[t]-x[t-1]) for each t in the times series
            x."
97      return N.array(zip(ts[1:],diff(ts)))
98
99
100 def wrap(ts):
101     "Returns the pair (t,x[t]) for each t in the time series x."
102     return N.array(zip(range(len(ts)), ts))
103
104
105 def histogram(ts, n):
106     "Returns the histogram with n beans for the time series ts."
107     return N.array(pylab.hist(ts, n))
108
109
110 def running_avg_std(val, win = 100):
111     # Evaluate mean and standard deviation for a window of lenght win
112     mean = []
113     std  = []
114     vque = []
115     sum  = 0
116     sum2 = 0
117
118     for x in val[:win]:
119         sum += x
120         sum2 += x*x
121         vque.insert(0,x)
122
123     for x in val[win:]:
124         prev = vque.pop()
125         vque.insert(0,x)
126         sum += x - prev
127         sum2 += x*x - prev*prev
128         mean.append(sum / win)
129         std.append(math.sqrt(sum2/win - sum*sum/(win*win)))
130
131     return N.array(zip(val[win/2:],mean,std))
132
133
134 def trend_cycle(table):
```

```
135     ref_cycle = []
136
137     prev_val = table[0][1] #previous value
138     cum_dif  = prev_val    #cumulative difference
139     cycle_len = 1          #cycle lenght
140
141     for val, dif in table[1:]:
142         #if the trend changes then save and reset counters
143         if prev_val * dif < 0:
144             if cum_dif > 0: sign = 1
145             elif cum_dif < 0: sign = -1
146             else: sign = 0
147             ref_cycle.append((val, cum_dif, cycle_len * sign))
148             cycle_len = 0
149             cum_dif = 0.
150
151         cum_dif += dif
152         cycle_len += 1
153         prev_val = dif
154
155     return N.array(ref_cycle)
156
157
158 def cond_variance(vec_diff, n_bins = 20):
159     diff = N.sort(vec_diff, 0)
160     x_min = diff[0][0]
161     x_max = diff[-1][0]
162     delta_bin = ( x_max - x_min )/ n_bins
163
164     bin = 1
165     num = 1
166     sum = diff[0][1]**2
167
168     var = []
169     for x,y in diff[1:]:
170         if x > x_min + bin * delta_bin:
171             #print bin, num, sum/num
172             var.append(sum/num)
173             bin += 1
174             num = 1
175             sum = y*y
176         else:
177             num += 1
178             sum += y*y
179
180     # Note xrange is a built-in function, probably a good candidate
181     # for this calculation would be the avereage point of the interval
182     xrange=[x_min + (i + 0.5)* delta_bin for i in range(n_bins)]
183
184     var.append(sum/num)
185     #print bin, num, sum/num
186     return N.array(zip(xrange,var))
187
188
189 def basic_statistics(vec):
190     """ Return some basic statistics for a given series:
191     min, max, mean, median, std, skewness, kurtosis."""
192     return [measure(vec) for measure in (min, max, mean, median, std,
            skewness, kurtosis)]
193
194
195 def pprint(tseries, dformat= "%d%.2d%.2d", vformat= "%.2lf", ofs= sys.
        stdout):
196     oformat = "%s\t%s" % (dformat, vformat)
197     for day,value in tseries:
```

```
198         values = day + (value,)
199         print >> ofs, oformat % values
200
201
202 def gnuplot_convert_matrix(ifs, ofs, initial_value, step, delimiter ='\
    t'):
203     for line in ifs:
204         if not line:
205             break
206
207         if line[0] == '#':
208             continue
209
210         val= line[:-1].split(delimiter)
211
212         date = val[0]
213         scale = initial_value
214         for i in val[1:]:
215             print >> ofs, "%s\t%d\t%s" % (date, scale, i)
216             scale += step
217
218         print >> ofs
219
220
221 def fit(x, y, residuals = False):
222     p0 = 0., 0.
223     plsq = leastsq(lambda p,u,v: u - p[0]*v -p[1], p0, args=(y,x),
        full_output= 1)
224
225     p = plsq[0]
226     if not residuals:
227         return p
228
229     res = y - p[0]*x -p[1]
230     return  plsq[:2] + (sum(res*res),)
```

For TSDFA due to the huge number of calculations involved it is necessary to use C code to speed up results. The author used the C code from the original authors Peng et al. [1994]. The differences between this version and the original can be summarised as:

- The code coming from Numerical Recipes was factored to the header file, `nr.h`. This code has a license that makes it free software.
- The references to polifit routines were changed to gsl. This code has a license that explicitly does not allow distribution. With this change the code is fully redistributable.

Listing C.8: DFA: C code

```
1 /* file: dfa.c     J. Mietus, C-K Peng, and G. Moody      8 February 2001
2                    Last revised:                        14 November 2001
                           v4.8
3                    José Matos                            5 May 2003
                             v5.0
4
5 ---------------------------------------------------------------------------

6 dfa: Detrended Fluctuation Analysis (translated from C-K Peng's Fortran
       code)
```

```
7  Copyright (C) 2001 Joe Mietus, C-K Peng, and George B. Moody
8
9  This program is free software; you can redistribute it and/or modify it
       under
10 the terms of the GNU General Public License as published by the Free
       Software
11 Foundation; either version 2 of the License, or (at your option) any
       later
12 version.
13
14 This program is distributed in the hope that it will be useful, but
       WITHOUT ANY
15 WARRANTY; without even the implied warranty of MERCHANTABILITY or
       FITNESS FOR A
16 PARTICULAR PURPOSE.  See the GNU General Public License for more
       details.
17
18 You should have received a copy of the GNU General Public License along
       with
19 this program; if not, write to the Free Software Foundation, Inc., 59
       Temple
20 Place - Suite 330, Boston, MA 02111-1307, USA.
21
22 You may contact the authors by e-mail (peng@physionet.org) or postal
       mail
23 (Beth Israel Deaconess Medical Center, Room KS-B26, 330 Brookline Ave.,
       Boston,
24 MA 02215 USA).  For updates to this software, please visit PhysioNet
25 (http://www.physionet.org/).
26 ------------------------------------------------------------------------------

27
28 This method was first proposed in:
29   Peng C-K, Buldyrev SV, Havlin S, Simons M, Stanley HE, Goldberger AL.
         Mosaic
30   organization of DNA nucleotides. Phys Rev E 1994;49:1685-1689.  [
         Available
31   on-line at http://prola.aps.org/abstract/PRE/v49/i2/p1685_1]
32
33 A detailed description of the algorithm and its application to
       physiologic
34 signals can be found in:
35   Peng C-K, Havlin S, Stanley HE, Goldberger AL. Quantification of
         scaling
36   exponents and crossover phenomena in nonstationary heartbeat time
         series.
37   Chaos 1995;5:82-87. [Abstract online at http://www.ncbi.nlm.nih.gov/
         entrez/-
38    query.fcgi?cmd=Retrieve&db=PubMed&list_uids=11538314&dopt=Abstract]
39
40 If you use this program in support of published research, please
       include a
41 citation of at least one of the two references above, as well as the
       standard
42 citation for PhysioNet:
43   Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG
         ,
44   Mietus JE, Moody GB, Peng CK, Stanley HE.  PhysioBank, PhysioToolkit,
         and
45   Physionet: Components of a New Research Resource for Complex
         Physiologic
46   Signals. Circulation 101(23):e215-e220 [Circulation Electronic Pages;
47   http://circ.ahajournals.org/cgi/content/full/101/23/e215]; 2000 (June
         13).
48 */
```

```
49
50 #include "nr.h"
51 #include <gsl/gsl_fit.h>
52 #include <stdlib.h>
53 #include <stdio.h>
54 #include <math.h>
55
56 /* Function prototypes. */
57 double * input(int *npts);
58 int* rscale(int minbox, int maxbox, double boxratio, int *rs, int *
       rslen);
59 void dfa(double *seq, int npts, int sw, int minbox, int maxbox);
60 void help(char *pname);
61
62 int main(int argc, char **argv)
63 {
64     int sw = 0;
65     int minbox = 0L, maxbox = 0L, npts;
66     int win_len = 50;
67
68     /* Read and interpret the command line. */
69     char * pname = argv[0]; /* this program's name (for use in error
           messages) */
70     for (int i = 1; i < argc && *argv[i] == '-'; i++) {
71       switch(argv[i][1]) {
72         case 'l':       /* set minbox (the minimum box size) */
73           minbox = atoi(argv[++i]); break;
74         case 'u':       /* set maxbox (the maximum box size) */
75           maxbox = atoi(argv[++i]); break;
76         case 'w':       /* set window size */
77           win_len = atoi(argv[++i]); break;
78         case 's':       /* enable sliding window mode */
79           sw = 1; break;
80         case 'h':       /* print usage information and quit */
81         default:
82           help(pname);
83           exit(1);
84       }
85     }
86
87     double *seq = input( &npts);        /* input data buffer */
88
89     /* Set minimum and maximum box sizes. */
90     if (minbox < 4) minbox = 4;
91     if (maxbox == 0 || maxbox > npts/4) maxbox = npts/4;
92     if (minbox > maxbox) {
93         printf("number of points = %d\nminbox = %d\nmaxbox = %d\n",
               npts, minbox, maxbox);
94         error("Invalid ranges: minimum box larger than maximum box");
95     }
96
97     /* Measure the fluctuations of the detrended input data at each box
            size
98        using the DFA algorithm; fill mse[] with these results. */
99     for (int i=0; i < npts - win_len ; ++i)
100      dfa(seq + i, win_len, sw, minbox, win_len/4);
101
102    //dfa(seq, npts, sw, minbox, maxbox);
103
104    free(seq);                  /* allocated by input() */
105    exit(0);
106 }
107
108 /* Read input data, allocating and filling an array whose pointer is
       returned.
```

```
109     The number of points read is passed in n_pts.
110
111     This function allows the input buffer to grow as large as necessary, up to
112     the available memory (assuming that a int is large enough to address
113     any memory location). */
114 double *input(int *n_pts)
115 {
116     double *seq = 0;
117     int maxdat = 0L;
118     double y;
119     int npts = 0L;
120
121     while (scanf("%lf", &y) == 1) {
122         if (npts >= maxdat) {
123             double *s;
124             maxdat += 50000;    /* allow the input buffer to grow (the
125                                    increment is arbitrary) */
126             if ((s = realloc(seq, maxdat * sizeof(double))) == NULL) {
127                 fprintf(stderr,
128                       "dfa: insufficient memory, truncating input at
                             row %d\n",
129                        npts);
130                 break;
131             }
132             seq = s;
133         }
134         seq[npts++] =  y;
135     }
136
137     if (npts < 1) error("no data read");
138     *n_pts = npts;
139     return seq;
140 }
141
142 /* rscale() allocates and fills rs[], the array of box sizes used by dfa()
143     below.  The box sizes range from (exactly) minbox to (approximately) maxbox,
144     and are arranged in a geometric series such that the ratio between
145     consecutive box sizes is (approximately) boxratio.  The return value is
146     the number of box sizes in rs[].
147 */
148 int* rscale(int minbox, int maxbox, double boxratio, int *nr, int *
        rslen)
149 {
150     int ir, n;
151     int rw;
152     int *rs;
153
154     /* Determine how many scales are needed. */
155     *rslen = log10(maxbox / minbox) / log10(boxratio) + 1;
156     rs = ivector(0, *rslen -1);
157     for (ir = 0, n = 1, rs[0] = minbox; n < *rslen && rs[n-1] < maxbox;
            ir++)
158       if ((rw = minbox * pow(boxratio, ir) + 0.5) > rs[n-1])
159             rs[n++] = rw;
160     if (rs[--n] > maxbox) --n;
161     *nr = n;
162     return rs;
163 }
164
165 void fit_linear (double *xv, double *yv, int nr, double *c0, double *c1
        , double *cov00, double *cov01, double *cov11, double * chisq)
```

```
166 {
167     double s1 = 0 ,sx = 0, sxy = 0,sxx = 0,sy = 0, syy = 0;
168
169     for(int i=0;i < nr; ++i) {
170         s1   += 1;
171         sx  += xv[i];
172         sxx += xv[i]* xv[i];
173         sxy += xv[i]* yv[i];
174         syy += yv[i]* yv[i];
175         sy  += yv[i];
176     }
177
178     double delta = sxx * s1 - sx * sx;
179     *c1 = (sxy * s1 - sx * sy ) / delta;
180     *c0 = (sxx * sy - sx * sxy) / delta;
181     *cov00 = (sxx - sx * sx / s1) / s1;
182     *cov01 = (sxy - sx * sy / s1) / s1;
183     *cov11 = (syy - sy * sy / s1) / s1;
184
185     *chisq = 0.0;
186     for(int i=0;i < nr; ++i) {
187         double d = yv[i] - *c1 * xv[i] - *c0;
188         *chisq += d*d;
189     }
190
191 }
192
193 /* Detrended fluctuation analysis
194     seq:        input data array
195     npts:       number of input points
196     rs:         array of box sizes (uniformly distributed on log scale)
197     nr:         number of entries in rs[] and mse[]
198     sw:         mode (0: non-overlapping windows, 1: sliding window)
199    This function returns the mean squared fluctuations in mse[].
200 */
201 void dfa(double *seq, int npts, int sw,int minbox, int maxbox)
202 {
203     int boxsize, inc, j;
204     int nr;             /* number of box sizes */
205     double stat;
206
207     int *rs = 0;        /* box size array; allocated and filled by
        rscale() */
208     int rslen;          /* length of rs[] */
209     double c0, c1, cov00, cov01, cov11, chisq;
210
211     /* Allocate and fill the box size array rs[].  rscale's third
        argument
212        specifies that the ratio between successive box sizes is 2^(1/8)
           . */
213     rs = rscale(minbox, maxbox, pow(2.0, 1.0/8.0), &nr, &rslen);
214
215     /* Allocate memory for dfa() and the functions it calls. */
216     double *mse = vector(0, nr-1); /* fluctuation array; */
217
218     double *x = vector(0,npts -1);
219     for(int i=0; i< npts; ++i)
220       x[i] = i;
221
222     for (int i = 0; i < nr; i++) {
223         boxsize = rs[i];
224         //printf("boxsize[%i]=%i\n",i,boxsize);
225         if (sw) { inc = 1; stat = (int)(npts - boxsize + 1) * boxsize;
              }
226         else { inc = boxsize; stat = (int)(npts / boxsize) * boxsize; }
```

```
227         for (mse[i] = 0.0, j = 0; j <= npts - boxsize; j += inc) {
228           gsl_fit_linear (x+j, 1, seq+j, 1, boxsize,
229                           &c0, &c1, &cov00, &cov01, &cov11,
230                           &chisq);
231           mse[i] += chisq;
232           //printf("%i\t%g\n",i,chisq);
233           //mse[i] += polyfit(x, seq + j, boxsize, nfit);
234         }
235         mse[i] /= stat;
236     }
237
238     double *xv = vector(0,nr-1);
239     double *yv = vector(0,nr-1);
240
241     /* Output the results. */
242     for (int i = 0; i < nr; i++) {
243       xv[i] = log10((double)rs[i]);
244       yv[i] = log10(mse[i])/2.0;
245       //printf("%g\t%g\n",xv[i],yv[i]);
246     }
247     fit_linear (xv, yv, nr, &c0, &c1, &cov00, &cov01, &cov11, &chisq);
248     printf("%g\t%g\n", c1, cov01/sqrt(cov00*cov11));
249
250     //gsl_fit_linear (xv, 1, yv, 1, nr, &c0, &c1, &cov00, &cov01, &
          cov11, &chisq);
251     //printf("%lf\t%lf\t%lf\t%lf\t%lf\t%g\t%g\n",c1,c0, cov00, cov01,
          cov11, cov01/sqrt(cov00*cov11),chisq );
252     /* Release allocated memory. */
253     free_vector(x, 0, npts -1);
254     free_ivector(rs, 0, rslen -1);      /* allocated by rscale() */
255     free_vector(mse, 0, nr -1);
256     free_vector(xv,0,nr-1);
257     free_vector(yv,0,nr-1);
258 }
259
260 static char *help_strings[] = {
261  "usage: %s [OPTIONS ...]\n",
262  "where OPTIONS may include:",
263  " -d K              detrend using a polynomial of degree K",
264  "                     (default: K=1 -- linear detrending)",
265  " -h                print this usage summary",
266   " -l MINBOX        smallest box width (default: 2K+2)",
267  " -s                sliding window DFA",
268  " -u MAXBOX         largest box width (default: NPTS/4)",
269  " -w WINSIZE        window size"
270  "The standard input should contain one column of data in text format."
       ,
271  "The standard output is two columns: log(n) and log(F) [base 10
       logarithms],",
272  "where n is the box size and F is the root mean square fluctuation.",
273 NULL
274 };
275
276 void help(char *pname)
277 {
278     int i;
279
280     (void)fprintf(stderr, help_strings[0], pname);
281     for (i = 1; help_strings[i] != NULL; i++)
282         (void)fprintf(stderr, "%s\n", help_strings[i]);
283 }
```

D. External Software

> "*Science is what we understand well enough to explain to a computer. Art is everything else we do.*" - Donald Knuth

This Appendix reflects the available software existing for the study of mathematical tools used in this work. As explained in Chapter 3 the author focuses only on free software. Even with this restriction this is a huge list with high quality software.

The different areas have been splitted in Section and each Subsection represents the main language in which the tools are implemented and are intended to be used. These methods are discussed in detail in Chapter 2.

D.1. Fourier transforms

fftw `http://www.fftw.org` is a C library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data (as well as of even/odd data, i.e. the discrete cosine/sine transforms or DCT/DST).

D.2. Wavelets

Reflecting wavelets popularity, there is a broad range of free software available for dealing with wavelet analysis.

D.2.1. R

waveslim `http://www.image.ucar.edu/staff/whitcher/` is a basic wavelet routines for time series (1D), image (2D) and array (3D) analysis.

wavetresh `http://cran.r-project.org/contrib/main/Descriptions/wavethresh.html` is a software to perform 1-d and 2-d wavelet statistics and transforms.

wavelets `http://www.atmos.washington.edu/~ealdrich/wavelets/` is a package that contains functions for computing and plotting discrete wavelet transforms (DWT) and maximal overlap discrete wavelet transforms (MODWT), as well as their inverses. Additionally, it contains functionality for computing and plotting wavelet transform filters that are used in the above decompositions as well as multiresolution analyses.

D.2.2. python

PyWavelets `http://www.pybytes.com/pywavelets/` is a Python module for calculating Simple and Inverse Discrete Wavelet Transform, as well as Wavelet Packets and Stationary Wavelet Transform.

wavelets `http://wavelets.scipy.org` is a promising package from the same creators of scipy and numpy.

D.2.3. C++

WAILI `http://www.cs.kuleuven.ac.be/~wavelets/` is a wavelet transform library.

MultiWavePack `http://python.rice.edu/MultiWavePack.html` is being developed to implement particular types of wavelet calculations. The eventual goal of this C++ code will be to automate many of the tasks involved in using wavelets for the solution of ordinary and partial differential equations. Orthogonal and bi-orthogonal cases of single wavelet and multiwavelet families are allowed, requiring little more than inclusion of the recursion coefficients in a database text file (a few examples are included already). Only a modest amount of functionality is currently offered, as reflected by pre-release numbering.

D.2.4. C

LastWave `http://www.cmap.polytechnique.fr/~bacry/LastWave/` is a signal processing oriented command language.

D.3. Fractional Brownian motion

D.3.1. R

fSeries `http://cran.at.r-project.org/src/contrib/Descriptions/fSeries.html` is an environment for teaching "Financial Engineering" and "Computational Finance" Würtz [2004]. This R package comes with 1309 R functions.

D.3.2. C

dfa `http://www.physionet.org/physiotools/dfa/` is the software from the same authors of [Peng et al., 1994] method and is used to evaluate the Hurst exponent using DFA.

D.4. Stable distributions

D.4.1. R

stable `http://popgen.unimaas.nl/~jlindsey/rcode.html` is James Lindsey's [Lindsey, 2004, Lambert and Lindsey, 1999] R package for stable distribution analysis.

D.4.2. Python

PyLevy `http://www.logarithmic.net/pfh/pylevy` is a package for calculation of Lévy stable distributions (probability density function and cumulative density function) and for fitting these distributions to data.

It operates by interpolating values from a table, as direct computation of these distributions requires a lengthy numerical integration. This interpolation scheme allows fast fitting of Levy stable distributions to data using the Maximum Likelihood technique.

Does not support α values less than 0.5.

Bibliography

Internet encyclopaedias go head to head. `http://www.nature.com/news/2005/051212/full/438900a.html`, 2005.

Rules for PSI-20 weights. `http://www.euronext.pt/bvlp/files/pubs/calcpsien.pdf`, 2003.

P. Abry and D. Veitch. Wavelet analysis of long-range-dependent traffic. *IEEE Transactions on Information Theory*, 44(1):2–15, 1998. URL `citeseer.ist.psu.edu/abry98wavelet.html`.

T. Araújo and F. Louçã. Complex behavior of stock markets: process of synchronization and desynchronization during crises. In *Perspectives on Econophysics*. Universidade de Évora - Portugal, 2006.

M. Ausloos. Financial time series and statistical mechanics. *arXiv:cond-mat/0103068*, 2001.

M. Ausloos. Econophysics of Stock and Foreign Currency Exchange Markets. *arXiv:physics/0606012*, 2006.

L. Bachelier. Théorie de la Spéculation. *Ann. Sci. Ecole Norm. S.*, III(17):21–86, 1900.

P. Ball. Culture Crash. *Nature*, 441:686–688, 2006.

J-M. Bardet, G. Lang, G. Oppenheim, A. Philippe, and M.S. Taqqu. *Generators of long-range dependent processes: a survey*, pages 579–623. Birkhäuser, 2003.

P. Barrett, J.D. Hunter, and P. Greenfield. Matplotlib - A portable Python plotting package. In *Astronomical Data Analysis Software & Systems XIV.*, 2004.

M. Bartolozzi, D. B. Leinweber, and A. W. Thomas. Scale-free avalanche dynamics in the stock market, 2006. URL `http://www.citebase.org/cgi-bin/citations?id=oai:arXiv.org:physics/0601171`.

F. Black and M. Scholes. The pricing of options and corporate liabilities. *J. Polit. Econ.*, 81:637–659, 1973.

J.P. Bouchaud and M. Potters. *Theory of Financial Risks: from Statistical Physics to Risk Management*. Cambridge University Press, Cambridge, 2001.

C. S Burrus, Ramesh A. Gopinath, and Haitao Guo. *Introduction to Wavelets and Wavelets transforms*. Prentice Hall, 1998.

L. Calvet and A. Fisher. Multifractality in asset returns: Theory and evidence. *The Review of Economics and Statistics*, 84(3):381–406, 2002.

J. Carvalho. Análise de Fourier e Ôndulas. Master's thesis, FEUP, 2000.

Y.-C. Chang and S. Chang. A fast estimation algorithm on the Hurst parameter of discrete-time fractional Brownian Motion. *IEEE Transactions on Signal Processing*, 50:554–559, 2002.

C. Chatfield. *The Analysis of Time Series: An Introduction*. Chapman & Hall, 6th edition, 2003.

Z. Chen, P.C. Ivanov, K. Hu, and H.E. Stanley. Effect of nonstationarities on detrended fluctuation analysis. *Physical Review E*, 65(041107), 2002.

R. Cont, M. Potters, and J-P. Bouchaud. Scaling in stock market data: stable laws and beyond. *arXiv: cond-mat/9705087*, 1997.

James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, 19:297–301, 1965.

P.B. DePetrillo, d.A. Speers, and U.E. Ruttiman. Determining the Hurst Exponent of Fractal Time Series and its Application to Electrocardiographic Analysis. *Computer in Biology and Medicine*, 29:393–406, 1999.

T. Di Matteo, T. Aste, and Michel M. Dacorogna. Using the scaling analysis to characterize financial markets. *Journal of Banking & Finance*, 29:827–851, 2005.

Z. Ding, C.W.J. Granger, and R. Engle. A long memory property of stock returns and a new model. *Journal of Empirical Finance*, 1:83–106, 1993.

Paul Doukhan, George Oppenheim, and Murad S. Taqqu, editors. *Theory and Applications of Long-Range Dependence*. Birkhäuser, 2003.

A. Einstein. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Ann. Phys-Berlin*, 17: 549–560, 1905.

P. Erdös and A. Rényi. On Random Graphs I. *Publicationes Mathematicae*, 6:290–297, 1959.

K.J. Falconer. *The Geometry of Fractal Sets*. Cambridge University Press, 1985.

E.F. Fama. Efficient capital markets: A review of theory and empirical work. *J. Financ.*, 25:383–417, 1970.

W. Feller. *An Introduction to Probability Theory and its Applications*. John Wiley & Sons, Inc., third edition edition, 1968.

S. Gallucio, J.P. Bouchaud, and M. Potters. Rational Decisions, Random Matrices and Sping Glasses. *Physica A*, 259:449–456, 1998.

A. Graps. An Introduction to Wavelets. *IEEE Computational Science and Engineering*, 2(2), 1995.

T.C. Halsey, M.H. Jensen, L.P. Kadanoff, I. Procaccia, and B.I. Shraimain. Fractal measures and their singularities: The characterization of strange sets. *Physical Review A*, 33:1141, 1986.

A.C. Harvey. Long memory in stochastic volatility. Research Report 10, London School of Economics, 1993.

T. Higushi. Approach to an Irregular Time Series on the Basis of the Frac. *Physica D*, pages 277–283, 1988.

K. Hu, P.C. Ivanov, Z. Chen, P. Carpena, and H.E. Stanley. Effect of Trends on Detrended Fluctuation Analysis. *Physical Review E*, 64(011114), 2001.

H.E. Hurst. Long-term storage capacity of reservoirs. *Trans. Am. Soc. Civ. Eng.*, 116: 770–808, 1951.

G. Kaiser. *A friendly guide to Wavelets*. Birkhäuser, 1994.

J.W. Kantelhardt, E. Koscielny-Bunde, H.H.A. Rego, S. Havlin, and A. Bunde. Detecting long-range correlations with detrended fluctuation analysis. *Physica A*, 295:441, 2001.

J.W. Kantelhardt, S.A. Zschiegner, E. Koscielny-Bunde, S. Havlin, A. Bunde, and H.E. Stanley. Multifractal detrended fluctuation analysis of nonstationary times series. *Physica A*, 316:87–114, 2002.

H. Kantz and T. Schreiber. *Nonlinear Time Series Analysis*. Cambridge University Press, second edition, 2004.

C.M. Kelty. Free Software/ Free Science. *First Monday*, 6(12), 2001. `http://firstmonday.org/issues/issue6_12/kelty/`.

D.E. Knuth. *The TeXbook*. Addison-Wesley, 1984.

A.N. Kolmogorov. A new invariant of transitive dynamical systems. *Dokl. Akad. Nauk. SSSR*, 119:861, 1958.

I. Koponen. Analytical approach to the problem of convergence of truncated Lévy flights towards the Gaussian stochastic process. *Physical Review Letter E*, 52:1197, 1995.

L. Laloux, P. Cizeau, J.P. Bouchaud, and M. Potters. Noise Dressing of Financial Correlation Matrices. *Physical Review Letters*, 83(7):1467–1470, 1999.

L. Laloux, P. Cizeau, and M. Potters. Random matrix theory and financial correlations. *International Journal of Theoretical and Applied Finance*, 3(3):391–397, 2000.

P. Lambert and J.K. Lindsey. Analysing financial returns using regression models based on non-symmetric stable distributions. *Applied Statistics*, 48:409–424, 1999.

L. Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, 1986.

J.K. Lindsey. *Statistical Analysis of Stochastic Processes in Time*. Number 14 in Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2004.

R. Litterman and K. Winkelmann. *Estimating Covariance Matrices*. Goldman-Sachs Risk Management Series. Goldman, Sachs and Co., 1998.

A. Lo. Long-Term memory in stock market prices. *Econometrica*, 59:1279–1313, 1991.

T. Lux. Detecting Multi-Fractal Properties in Asset Returns: An Assessment of the 'Scaling Estimator'. *International Journal of Modern Physics*, 15:481 – 491, 2004.

E. Maasoumi and J. Racine. Entropy and predictability of stock markets returns. *Journal of Econometrics*, 107:291–312, 2002.

S. G. Mallat. Multiresolution approximation and wavelet orthonormal bases of L^2. *Transactions of the American Mathematical Society*, 315:69–87, 1989a.

S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Recognition and Machine Inteligence*, 11(7):674–693, 1989b.

B.B. Mandelbrot. New anomolous multiplicative multifractals: left sided $f(\alpha)$ and the modelling of DLA. *Physica A*, 168:95–111, 1990.

B.B. Mandelbrot. The variation of certain speculative prices. *J. Bus.*, XXXVI(4):394–419, 1963.

B.B. Mandelbrot. *Statistical Models and turbulence: Possible refinements of the lognormal hypothesis concerning the distribution of energy dissipation in intermitent turbulence*. Springer Verlag (New York), 1972.

B.B. Mandelbrot. *Fractals: Form, Chance and Dimension*. W H Freeman and Co, 1977.

B.B. Mandelbrot. *The Fractal Geometry of Nature*. W H Freeman and Co, 1982.

B.B. Mandelbrot and J.W Van Ness. Fractional Brownian Motion, fractional noises and applications. *SIAM Review*, 10:422, 1968.

B.B. Mandelbrot, A.J. Fisher, and L.E. Calvet. A Multifractal Model of Asset Returns. Cowles Foundation Discussion Paper 1164, 1997. Available at SSRN: http://ssrn.com/abstract=78588.

R.N. Mantegna and H.E. Stanley. Scaling behaviour in the dynamics of an economic index. *Nature*, 376:46 – 49, 2002.

R.N. Mantegna and H.E. Stanley. *Physical Review Letter*, 73:2946, 1994.

R.N. Mantegna and H.E. Stanley. *An Introduction to Econophysics*. Cambridge University Press, Cambridge, 2000.

J.A.O. Matos and J.A.M.S. Duarte. On a conservative lava flow automaton. *International Journal of Modern Physics C*, 10(1):321–335, 1999.

J.A.O. Matos, S.M.A. Gama, H.J. Ruskin, and J.A.M.S. Duarte. An econophysics approach to the Portuguese Stock Index–PSI-20. *Physica A*, 342(3-4):665–676, 2004.

J.A.O. Matos, S.M.A. Gama, H.J. Ruskin, A. Sharkasi, and M. Crane. Correlation of worldwide markets' entropies. In *Perspectives on Econophysics*. Universidade de Évora - Portugal, 2006a.

J.A.O. Matos, S.M.A. Gama, A. Sharkasi, H.J. Ruskin, and M. Crane. Temporal and Scale DFA Applied to Stock Markets. In preparation, 2006b.

J. McCauley. Thermodynamics analogies in economics and finance: instabilities of markets. *Physica A*, 329:199–212, 2003.

G. Moody. Parallel universes: open access and open source. `http://lwn.net/Articles/172781/`, 22 February 2006a.

G. Moody. Gutenberg 2.0: the birth of open content. http://lwn.net/Articles/177602/, 29 March 2006b.

G. Moody. Learning the lesson: open content licensing. `http://lwn.net/Articles/181374/`, 26 April 2006c.

G. Moody. Open Content III: the code. `http://lwn.net/Articles/183907`, 16 May 2006d.

M.E.J. Newman. The structure and function of networks. *SIAM Review*, 45:167–256, 2003.

J.P. Nolan. *Lévy Processes: Theory and Applications*, chapter Maximum likelihood estimation of stable parameters, pages 379–400. Boston: Birkhäuser, 2001.

J.P. Nolan. Bibliography on stable distributions, processes and related topics. http://academic2.american.edu/ jpnolan/stable/StableBibliography.pdf, 2005.

J.P. Nolan. *Stable Distributions - Models for Heavy Tailed Data*. Boston: Birkhäuser, 2006.

M.F.M. Osborne. Brownian motion in the stock market. *Oper. Res.*, 7:145–173, 1959.

M.F.M. Osborne. *The Stock Market and Finance from a Physicist's Viewpoint*. Crossgar Press, 1977.

A. Papoulis. *Probability, Random Variables and Stochastic Processes*. Mc Graw Hill, 1985. ISBN 0-07-048468-6.

V. Pareto. *Cours d'Économie Politique*. 1897.

E. Parzen. *Stochastic Processes*. SIAM, 1999.

H-O. Peitgen, H. Jürgens, and D. Saupe. *Chaos and Fractals, New Frontiers of Science*. Springer-Verlag, 1992.

C.-K. Peng, S.V. Buldyrev, S. Havlin, M. Simons, H.E. Stanley, and A.L. Golderberger. On the mosaic organization of DNA sequences. *Phys. Rev. E*, 49:1685–1689, 1994.

D. Percival and A. Walden. *Wavelet methods for time series analysis*. Cambridge University Press, 2000.

V. Plerou, P. Gopikrishnan, B. Rosenow, L.A.N. Amaral, and H.E. Stanley. Universal and non-universal properties of cross correlations in financial time series. *Physical Review Letters*, 83(7):1471–1474, 1999.

V. Plerou, P. Gopikrishnan, and B. Rosenow. Collective behaviour of stock price movement: a random matrix theory approach. *Physica A*, 299:175–180, 2001.

A. Rényi. On measures of information and entropy. In *4th Berkeley Symposium on Mathematics, Statistics and Probability*, pages 547–561, 1961.

P.A. Samuelson. Mathematics of speculative prices. *SIAM Rev.*, 15:1–34, 1973.

C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

S. Sharifi, M. Crane, A. Shamaie, and H.J. Ruskin. Random matrix theory for portfolio optimization: a stability approach. *Physica A*, 335(3-4):629–643, 2004.

A. Sharkasi, M. Crane, H.J. Ruskin, and J.A.O. Matos. The Reaction of Stock Markets to Crashes and Events: A Comparison Study between Emerging and Mature Markets using Wavelet Transforms. *Physica A*, 368(2):511–521, 2006a.

A. Sharkasi, H.J. Ruskin, M. Crane, J.A.O. Matos, and S.M.A. Gama. A wavelet-based method to measure stages of stock market development. In preparation, 2006b.

M. F. Shlesinger, U. Frisch, and G. Zaslavsky, editors. *Lévy Flights and Related Phenomena in Physics*. Springer, 1995.

M.F. Shlesinger, G.M. Zaslavsky, and J. Klafter. Strange kinetics. *Nature*, 363:31–37, 1993.

A.G. Sinai. On the concept of entropy of a dynamical system. *Dokl. Akad. Nauk. SSSR*, 124:768, 1959.

R.M. Stallman. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. 2002. ISBN 1-882114-98-1.

H.E. Stanley. Econophysics: can physicists contribute to the science of economics? *Computing in Science & Engineering*, 1(1):74–77, 1999.

C. Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52:479, 1988a.

C. Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52:479, 1988b.

C. Tsallis, C. Anteneodo, L. Borland, and R. Osorio. Nonextensive statistical mechanics and economics. *Physica A*, 324:89–100, 2003.

M. Ueda and S. Loadha. Wavelets: An Elementary Introduction and Examples. Technical report, UCSC-CRL 94-47, 1995.

C. Valens. A really friendly guide to wavelets, 1999. URL `citeseer.ist.psu.edu/valens99really.html`.

W.N. Venables and B.D. Ripley. *S Programming*. Springer, 2000.

R. Vilela Mendes, T. Araújo, and F. Louçã. Reconstructing an Economic Space from a market metric. *Physica A*, 323:635–650, 2003.

T.A. Vuorenmaa. *Proceedings of SPIE: Noise and Fluctuations in Econophysics and Finance, Vol. 5848*, chapter A Wavelet Analysis of Scaling Laws and Long-Memory in Stock Market Volatility, pages 39–54. 2005.

N. Walsh and L. Muellner. *DocBook: The Definitive Guide*. O'Reilly, 1999.

D. Wilcox and T. Gebbie. On the analysis of cross-correlations in South African market data. *Physica A*, 344(1-2):294–298, 2004.

J. Willinsky. The unacknowledged convergence of open source, open access, and open science. *First Monday*, 10(8), 2005. `http://firstmonday.org/issues/issue10_8/willinsky/`.

D. Würtz. *Rmetrics: an environment for teaching financial engineering and computational finance with R.* Rmetrics, ITP, ETH Zürich, Zürich, Switzerland, 2004. `http://www.rmetrics.org`.

Y.C. Zhang. Modeling Market Mechanism with Evolutionary Games. *Europhysics News*, 29:51, 1998.