Forecasting Time Series using a Geometrical Brownian Model

Higher Diploma in Science in Data Analytics

Anastasia Kasara
Dublin Business School

September 4, 2018

Contents

1 Introduction

Data are generated continuously in time in many different areas of the modern society activities, eg. business, economics, medicine, biology and in science in general, e.g. physical observations (temperature, pressure, ...), stock-prices, virus population, sensor readings, etc. These data are mainly digitally stored and are available to further investigation and analysis. Data may possess temporal patterns such increasing or decreasing trends, periodical or be aperiodic, seasonal fluctuations, or be chaotic. Their study and use toward prediction forecasting is the subject of time-series modeling analysis.

Time series modeling focuses on the colection and analysis of past observational data with the objective to develop a mathematical/computational framework capable to provide future values for the series, namely to *forecast*. Given the importance of forecasting in a vast number of practical fields such as economics, business, science, engineering, biology, development of the model (fitting of the underlying time series) a number of mathematical models have been developed with a large variety of the sophistication level. Undoubtely, a successful forecasting depends critically on the appropriate model fitting, which in turn depends also on the nature of the time-series data available (e.g. size, source, ..). So, in brief, prediction *accuracy* is the number one goal of time-series modelling. A second one, certainly less critical in many cases, is the *efficiency* performance of the model. Obviously, it is pointless to develop a model with 100% accuracy prediction for the next's day weather forecast when it takes 3 days for the calculations. This aspect of the time-series modeling (efficiency) will not be touched in this project and the focus will be restricted to the *accuracy* aspect only.

ARIMA models. With regards to accuracy prediction requirement of the most frequently used stochastic time series models are the Autoregressive (AR) and Moving Average (MA) models and their combined use Autoregressive Moving Average (ARMA) and Autoregressive Integration Moving Average (ARIMA) models [1]. *Main assumptions to implement these models is that the time series are linear in time and random fluctuations follow the normal distribution.* ARIMA models have subclasses of other models, such as the Autoregressive (AR). In the cases where inherent (deterministic) structure is present the above models can be customized to include a deterministic time-dependence as proposed by Box-Jenkins e.g. seasonal ARIMA (SARIMA) [1]. The popularity of these (linear) modeling approaches is attributed to their implementation simplicity as well as to their straightforward manner to extract useful information from the data time series. Note that, inevitably, the working mathematics of these modesl aren't highly demanding which entails to a smooth and quick learning curve. Consequently, these models are the first to encounter when one first copes with time-series analysis.

Having said the above, one should be cautious and not fooled by their popularity. There are many cases where the severe limitation of the pre-assumed linear time-dependence modelling is completely inadequate for forecasting. Obviously, the only route left is to abandon this linearity and consider models that allow non-linear time-dependences. Needless to say, that the price to pay is that both from the conceptual and mathematical point of view

these models are not as simple as the ARIMA models. Of course, this has consequences to the implementation aspects of the modeling.

Brownian Motion modelling The present project investigates the so-called Geometric Brownian Model (GBM) to analyze time-series data. This model owes its name from the Scottish botanist Robert Brown who was the first to observe the irregular motion of air-dust particles back in 1827 (Brownian motion)[2]. Historically, this model was first developed from Bachelier (1900) and independently from Einstein (1905) for the explanation of the molecular Brownian motion. The GBM model has found applications in a diverse range of domains, eg. physics, economics, biology, environmental, etc.. [4, 5]. More details on the particular mathematical approach used are given below.

In chapter 1 the underlying principles of the Brownian model are discussed by working the case of the *simple random walk model*. The general case of the BM models is presented and then two of its special cases (arithmetic BM (ABM) and geometric BM (GBM), treated in this project, are also discussed. Finally in chapter 2 applications of the GBM for forecasting share price values are demonstrated. In chapter 3 have relegated material that is necessary for the present project but do not represent its core. This material is some backround knowledge of statistical quantities as well as some derivations, mainly for completeness reasons. Finally in chapter 4 the source codes used for the calculation of the p lots are included. The main programming language is the freely available R language, although some of the code is written with the (also freely available) Python programming language.

Chapter 1

Theory of Brownian motion model

1 Random walk and Brownian motion

A standard and very fundamental starting point for the understanding of Brownian motion is the so-called *random walk* model [3]. For simplicity one may consider the case where a particular object can move in discrete steps of random size, s_i along an one-dimensional line, characterized by the positions $x_i, i = 1, 2, ...,$. Each jump (event) is then associated with its starting position, x_i, and the 'jump' duration, t_i. Assuming that the object is initially x_0 is at position at x_0 for the first steps, $s_1, s_2, s_3, ..$, occuring at times $t_1, t_2, t_3, ...$, respectively, we have for the corresponding positions, $x_1, x_2, x_3, ...,$

$$
\begin{aligned}
&x_1 = x_0 + s_1, && \text{position at } t_1 \text{ following random jump } s_1\\
&x_2 = x_0 + s_1 + s_2, && \text{position at } t_2 \text{ following random jumps } s_1.s_2\\
&x_3 = x_0 + s_1 + s_2 + s_3, && \text{position at } t_3 \text{ following random jump } s_1, s_2, s_3\\
&\quad .... &&
\end{aligned}
$$

So, generally the position of the object at time t_n is a random quantity, x_n, equal to:

$$x_n = x_0 + s_0 + s_1 + \cdots + s_n = x_0 + \sum_{i=1}^{n} s_i, \qquad i = 1, 2, ... \tag{1.1}$$

The object's position at time n, is a random quantity because it consists of a sum of a random quantities, $s_i,\ i = 1, 2, ...$ An alternative, but equivalent, expression for this kind of motion can be obtained as follows: Consider the expression (1.1) at times t_i and t_{i-1} and take the difference, $x_i - x_{i-1}$. Then is readily found that,

$$x_i = x_{i-1} + s_i, \qquad i = 1, 2, ..., n \tag{1.2}$$

Note, that the latter form has the familiar form of a discrete time-series sequence. The last bit of information that is required for the complete description of the above model is to define the probability distribution, $P(s_i)$, for the possible values of the steps, $s_i,\ i = 1, 2, ..$ (s_i can be either negative or positive of any size). *The specification of $P(s_i)$, for all i, defines a particular specialization of what is known as random walk motion.*

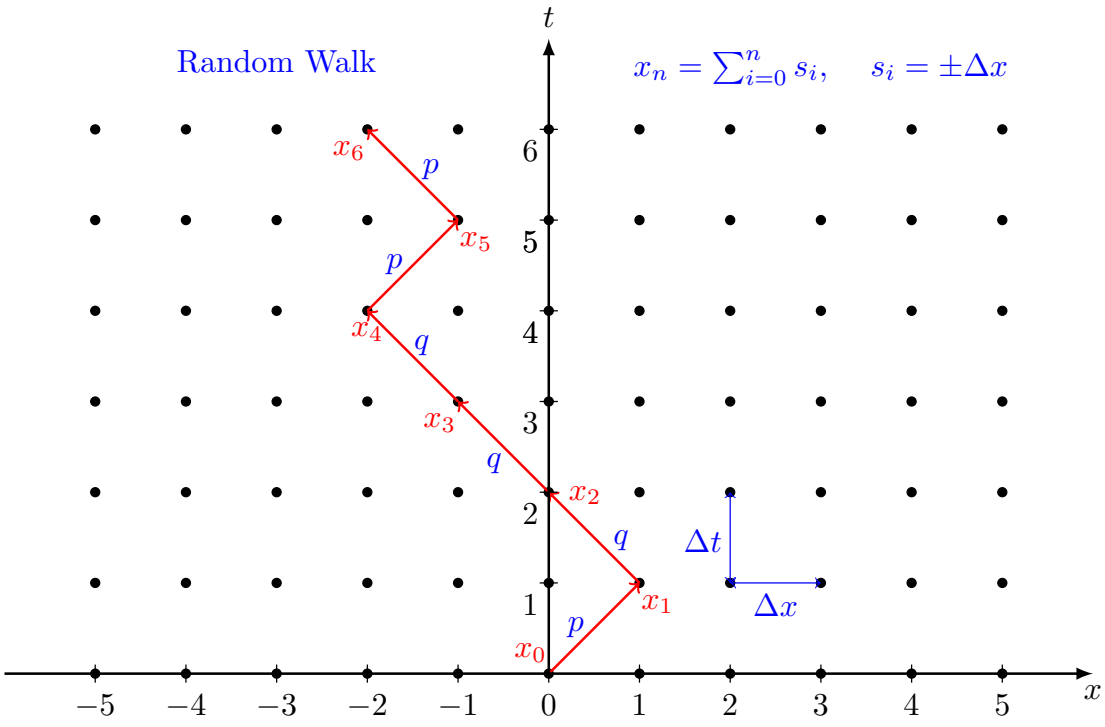


Figure 1.1: Discrete random walk sketch. Probability that the object will jump to its right a distance Δx is p while it is $q = 1 - p$ for a jump to its left. Duration of the jump is Δt. After n steps the time elapsed is $t = n\Delta t$ and the position x_n is the sum of the individual jumps. Here $n = 6$. x_n is a random variable with Binomial distribution.

Simple random walk. The *simple random walk* is defined as this random walk with Benoulli probability distribution, $P(s_i)$ as,

$$P(s_i) = \begin{cases} p, & s_i = \Delta x \\ 1 - p, & s_i = -\Delta x \end{cases} \qquad i = 1, 2, ...$$

Let's now assume the object has performed n jumps, where n_+ to its right and n_- to its left, so that $n_+ + n_- = n$. Let's now assume that the position of the object at the end of the n jumps is $x_k = k\Delta x$ (for example in figure (1.1) $n = 6$, $k = -2$, $n_+ = 2$ and $n_- = 4$). Then

$$k\Delta x = n_+\Delta x - n_-\Delta x \quad \longrightarrow \quad k = n_+ - n_-.$$

Standard argumentation about the probability of n repeated Bernoulli trials jumps) to occur n_+ times to the right with probability p in any order results to the Binomial distri-

bution; The probability that $x_n = k\Delta x$ is given by,

$$Prob(x_n = k\Delta x) = \binom{n}{n_1} p^{n_1} q^{n_2}$$

However, $n_\pm$ can be written in terms of n, k,

$$n_+ = \frac{n+k}{2} \qquad n_- = \frac{n-k}{2}.$$

Replacing these values and since $p = 1 - q$ we finally obtain:

$$Prob(x_n = k\Delta x) = \tag{1.3}$$

$$B(\frac{n+k}{2}, p, n) = \binom{n}{\frac{n+k}{2}} p^{\frac{n+k}{2}} (1-p)^{\frac{n-k}{2}}, \tag{1.4}$$

$$k = -n, -n+2, ..., n-2, n.$$

The above central result is the starting point to relate the *Random Walk* process with the *Brownian motion*. To this end the number of jumps will be taken very large while its duration Δt and displacement size Δx will be assumed very small so that the fundamental Central Limit Theorem (CLT) of probability becomes applicable.

Multidimensional random walk. A comment may be is appropriate here. The original Brownian motion (as observed by R. Brown) takes place in the three-dimensional space. By allowing the jumps to take place not along a straight line only but on a plane or or in a 3-dimension space one can similarly study a 2-D or 3-D random walk. For the purposes of the present project there is no need to consider these more complicated cases.

1.1 Brownian motion derived from simple random walk

The simplest Brownian motion can be derived as limiting case of the simple random walk ($p = q = 1/2$) where $s_i = \pm\Delta x$. More specifically, a constant time step is assumed, $\tau = t_{i+1} - t_i, i = 1, 2, 3, ..$ with the associated jump steps, s_i, to take values either Δx and $-\Delta x$ with equal probability:

$$P(s_i) = \begin{cases} \frac{1}{2}, & s_i = \Delta x \\ \frac{1}{2}, & s_i = -\Delta x \end{cases} \qquad i = 1, 2, ..., n.$$

Without loss of generality, for convenience, we'll assume $x_0 = 0$. Then, accordingly, the mean value and the standard deviation of x_n (1.1) as,

$$E[x_n] = E[\sum_{i=1}^{n} s_i] = \sum_{i=1}^{n} E[s_i] = \sum_{i=1}^{n} [P(s_i = \Delta x)\Delta x + P(s_i = -\Delta x)(-\Delta x)] =$$
$$= \sum_{i=1}^{n} \left[\frac{1}{2}\Delta x + \frac{1}{2}(-\Delta x)\right] = 0 \tag{1.5}$$

where the definition of the mean value (in terms of the probability distribution) was used (3.5). Since the variance of x_n is given by $V[x_n] = E[x_n^2] - (E[x_n])^2 = E[x_n^2] - 0$[1], we have,

$$V[x_n] = E[x_n^2] = E[(\sum_{i=1}^{n} s_i)^2] = E[(s_1 + s_2 + ... + s_n)(s_1 + s_2 + ... + s_n)] =$$
$$= E[\sum_{i}^{n} s_i^2 + \sum_{i}\sum_{j\neq i} s_i s_j)] = \sum_{i}^{n} E[s_i^2] + \sum_{i,j\neq j} E[s_i s_j]$$
$$= n\left[\frac{1}{2}\Delta x^2 + \frac{1}{2}(-\Delta x)^2\right] - n\left[\frac{1}{4}(\Delta x)(\Delta x) + \frac{1}{4}(-\Delta x)(\Delta x) + \frac{1}{4}(\Delta x)(-\Delta x) + \frac{1}{4}(\Delta x)(\Delta x)\right]$$
$$= n\Delta x^2 - 0.$$

So we finally have:

$$E[x_n] = 0 \qquad V[x_n] = n\Delta x^2. \tag{1.6}$$

Finally the probability distribution is given by (1.4) for $p = q = 1/2$ as,

$$Prob(x_n = k\Delta x) = \binom{n}{\frac{n+k}{2}} (\frac{1}{2})^n, \qquad k = -n, -n+2, ..., n-2, n$$

The above considerations say that the average value of x_n will be zero while the standard deviation will be equal to $n\Delta x^2$ (so proportional to the number of jumps). Now by invoking the Central Limit Theorem which states that any distribution made of a large number of independent arbitrary distributions will eventually end up to a normal distribution with the same mean and standard deviation. In our case, the random variable under question is the x_n, made up by the (independent) jumps, s_i, as expressed in (1.1). From the above considerations about the mean and variance of x_n one can conclude that in the limit of large number of jumps (steps), the time series sequence approaches the equivalent *normal distribution*, $N(x; 0, n\Delta x^2)$ according to the central limit theorem:

$$\lim_{n>>1} Prob(x_n = k\Delta x) \longrightarrow Prob(x < \tilde{x} < x + \Delta x) = N(x; 0, n\Delta x^2) \tag{1.7}$$

where with $\tilde{x}$ is denoted the *continuous* random variable instead of the *discrete* random variable x_n (see for example figure (1.2)). The interesting observation here, is that the average distance (from the initial position) folllowing n jumps is proportional to $\sim \sqrt{n}$ rather than $\sim n$.

[1] $Var \equiv V$ for convenience

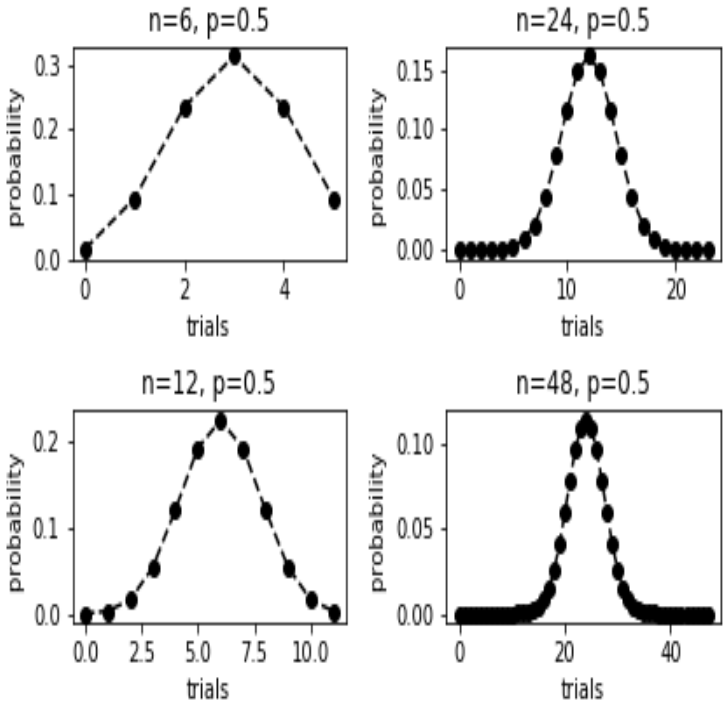


Figure 1.2: This plot, in agreement with the CLT, shows how the Binomial distribution resembles the Gaussian distribution towards larger n. Here $p = q = 1/2$.

Continuous time limit and Brownian motion. The corresponding continuous time limit of the above process is known as *Brownian motion* or *Wiener process*. This limiting case is achieved as follows: starting at time $t_0 = 0$, after n jumps the time, $\Delta t = t - t_0$, is,

$$\Delta t = t - t_0 = n(t_{i+1} - t_i) = n\tau \quad \longrightarrow \quad n = \frac{\Delta t}{\tau}.$$

By reversing the standard discretization procedure of a continuous variable (here is the time) we require the number of steps to approach infinite ($n \to \infty$) with the time step, τ, decreasing accordingly as $\tau \to 0$. The rate of growth for n and decrease of τ should be such that Δt remains always finite $\Delta t < \infty$. This means, that (1.7) should replaced by,

$$\lim_{(n\to\infty,\tau\to 0)} Prob(x_n = n\Delta x) \quad \longrightarrow Prob(x < \tilde{x} < x + \Delta x) = N(x; 0, \frac{\Delta x^2}{\tau}\Delta t)$$

Now since $\tau \to 0$ the value of Δx (the size of jumps or steps) should decrease accordingly in a way such that $\Delta x^2/\tau$ its limit is finite. Eventually we result to the following limit,

$$Prob(x < \tilde{x} < x + \Delta x) = N(x; 0, \sigma^2 \Delta t), \qquad x_n \to \tilde{x}, \qquad \lim_{(\Delta \to 0, \tau \to 0)} \frac{\Delta x^2}{\tau} \to \sigma. \tag{1.8}$$

Wiener process Conventionally, the time-dependent random quantity distributed according to the standard normal distribution with variance $\sigma = t$, denoted as $B(x;t)$, is called *Wiener* process[2]. The *Wiener*-process is fully defined by the following three basic properties:

- $B(x;0) = 0$
- **Independent increments:** increments of $dB(x;t_i) = B(x;t_i) - B(x;t_i + \Delta t), i = 1, 2, ...$ are independent each other.
- **Gaussian property:** Increments are distributed normally:

$$dB(x;t) = B(x;t) - B(x;t') \sim N(x; 0, dt), \qquad dt = t - t' > 0$$

The mathematical function, named Wiener-process is a function where is *everywhere continuous but nowhere differentiable*. The rigorous definition (and study) of the Wiener-process and its properties were first presented by Norbert Wiener at 1923. For the Brownian motion (or Wiener-process) the following statistical properties hold:

$$E[B(x;t)] = 0, \tag{1.9}$$
$$V[B(x;t)] = t \tag{1.10}$$
$$E[B(x;t)B(x;t')] = \min(t, t'), \tag{1.11}$$
$$E[B(x;t) - B(x;t')] = \sqrt{t - t'}. \tag{1.12}$$

In addition the Wiener - increments, $dB(x;t) = B(x;t) - B(x,t')$ are normally distributed with mean and variance as,

$$E[dB(x;t)] = 0, \qquad V[dB(x;t)] = dt \tag{1.13}$$

Brownian motion as a Wiener process. Now we are at the position to relate the limiting form of a random walk (Brownian motion) with the Wiener process. Since we have chosen as initial value for x_0 (not necessarily zero) the time-series sequence, $x_n(t_n)$

[2] In the literature the symbol $W(x;t)$ is also used (from N. Wiener)

of random walk, actually is the random difference between the current value x_n and the initial value $x_0 = x(t_0)$. This difference, denoted as, $dx_n = x_n - x_0$ takes its limiting form as dx. So actually $x \to dx = x - x_0$

$$x_n \to dx_n(t_n) = x_n - x_0 \quad \longrightarrow \quad dx(t) = x - x_0.$$

We know that after infinitely many steps, this difference distributes as the *normal distribution* as[3] , with variance σdt. But this last property is possesed by the *increments* of the Wiener process, $dB(x,t)$ by definition (variance equal dt). The only difference is the proportional factor σ. From the properties of the normal distribution it can be shown that we can relate the BM with Wiener process as,

$$dx(t) = N(x; 0, \sigma^2 dt) = \sigma dB(x;t), . \tag{1.14}$$

The above relation denotes the fact that Brownian-governed process represented by the random variable dx at time t has the same statistical properties as the quantity $\sigma dB(x;t)$.

1.2 Brownian motion with drift

The BM as derived above can be generalized to study the type of the BM process had we started off not from the symmetric random walk ($p = q = 1/2$) but for $p \neq q$. Generally these cases will result to the inclusion of a non-vanishing drift in (1.14) The straightforward interpretation of such assumption is that at each time, when at position x_i, there is an uneven probability for jumps between the two positions, $x_i + \Delta x$ or $x_i - \Delta x$. For example for $p > 1/2$ there is a tendency for jumps toward $x_i + \Delta x$ relative to the opposite direction, $x_i - \Delta x$ [see figure (1.3)]. In other words, in reference to (1.4) final probability for x_n, after many jumps we will have $n_+ > n_-$. We then say that the jump process is biased with some non-zero *drift*. Still, it is a random walk process but with different mean and standard deviation values:

$$E[x_n] = (p - q)n\Delta x, \qquad V[x_n] = 4pqn\Delta x^2. \tag{1.15}$$

The latter expressions follows from the properties of the Binomial distribution. It is easy to confirm that fort $p = q = 1/2$ we obtain the results for the symmetric random walk as in (1.6). From the above relations, we see for example that if $p > 1/2$ then generally the mean value will have positive value. Also the standard deviation will be smaller relative to the standard deviation of the simple random walk, where $p = 1/2$ [see figure (1.4)].

Had we followed the same lines of thinking as in the *simple random walk* we then would have ended up to the conclusion that the (random) distance, x_n, at time t_n, for very large n, will approach a Gaussian law distribution for its value, according to the central limit

[3] For convenience, from now on the dependence on x from the argument of $B(x;t)$ will be omitted and only implied $B(t) \equiv B(x;t)$

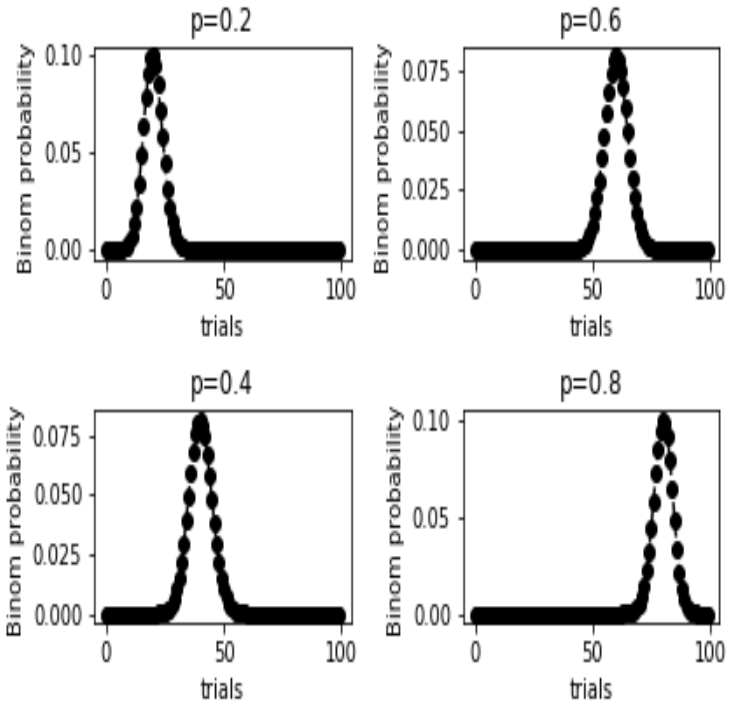


Figure 1.3: Binomial distribution $B(k, n, p)$ for $n = 100$ trials for four values p. The center of the peak moves from left ($p = 0.2$) to right ($p = 0.8$).

theorem (CLT) [see chapter 3 and (3.39)]. In the limiting case of infinitely number of jumps ($n \to \infty$) of infinitesimally small steps ($\Delta x \to 0$) and of infinitely short duration $\tau \to 0$ the random distance $dx(t) = x(t) - x(t_0)$ approaches again the Gaussian distribution as,

$$dx_n \to dx(t) = N(\mu \Delta t, \sigma^2 \Delta t) \tag{1.16}$$

The above form can be rewritten by using the transformation properties of the normal distribution (3.28) to obtain,

$$dx(t) = \mu dt + \sigma \sqrt{dt} \tilde{N} = \mu dt + \sigma dB(t) \tag{1.17}$$

2 Generalized Brownian model

The last step is to obtain the most general form of a Brownian motion process is to assume that the values, μ and σ are not constants, but depend (a) on time, t, and (b) on the process itself, $x(t)$.

$$\mu = \mu(x(t), t), \qquad \sigma = \sigma(x(t), t).$$

This generalization introduced in the literature by Itô resulting to what is known as *Itô stochastic process* [4]. For example, in the context of the 1-dimensional discrete jumps,

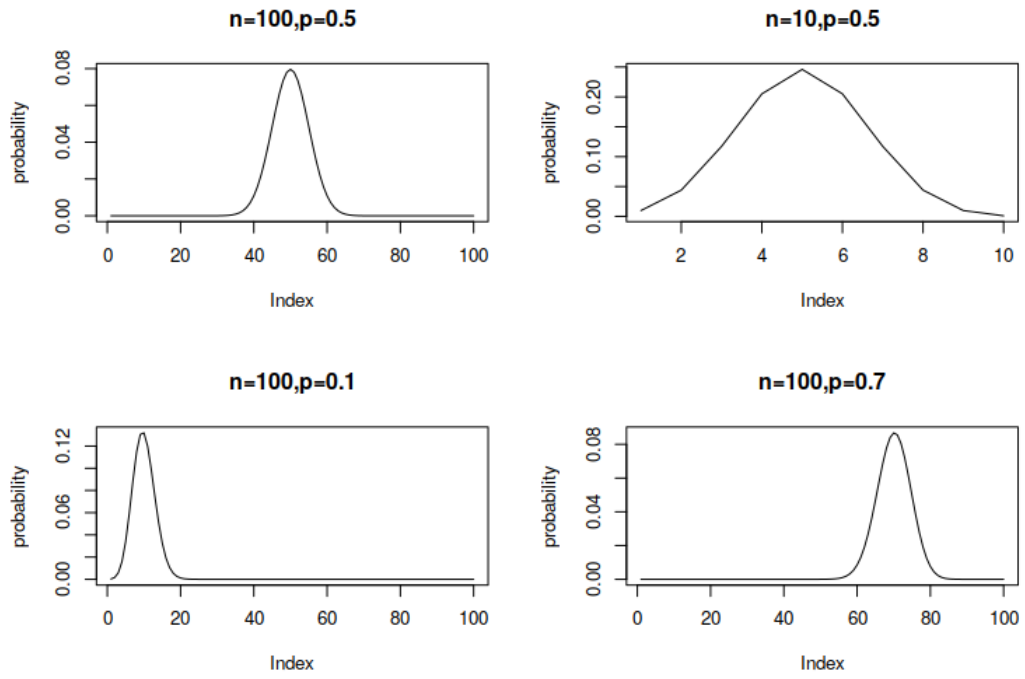


Figure 1.4: Plots of binomial distribution for different, with (bottom plots) and without drift ($p = 1/2$) (top plots). In the bottom figures the effect of drift is clearly evident as the peaks of the distribution are shifted away from $n = 50$ where the non-drift plot for $n = 100$ (top left plot) is centered. Comparison of the non-drift plots for $n = 10, p = 1/2$ and $n = 100, p = 1/2$ also shows the resemblance of the binomial distribution to the normal distribution one for the larger n.

discussed so far, [see (1.2)], this generalization would require $s_n = s_n(x_n, t_n)$. Therefore, the most general form for a random quantity, $x(t)$, described via the Brownian motion modelling is given by,

$$dx(t) = \mu(x(t), t)dt + \sigma(x(t), t)dB(t). \tag{1.18}$$

Within this generalization $\mu(x(t), t)$ and $\sigma(x(t), t)$ do not generally coincide with the *mean* and the *standard deviation* of the process (unless they are constants as will be shown later (see section on arithmetic BM later on).

The above stochastic differential equation (SDE) represents the main working equation of the present project upon which we'll base our model specializations. Since the behaviour of the normal distribution, $\tilde{N}(t) = N(t; 0, 1)$ is known, the starting point for a particular model is to specify $\mu(x(t), t)$ and the $\sigma(x(t), t)$ of the random quantity. More generally,

equations (1.16) and (1.18) are the starting point to model particular random time series with the form of μ and σ specified from the partical features of the quantity to be simulated. While the applications are so diverse, the underlying principles and properties of the random quantities share common behaviour. The conceptual simplicity (compacted in the specification of μ and σ) with its straightforward implementation makes the BM modelling approach so appealing thus enjoying a great popularity over the years. Needless to say, that other approaches to model randomness are also available but apparently their level of sophistication requires more in-depth relevant expertise, compared to the BM approach.

Brownian motion and white noise process It is instructive to relate the Brownian process definition with another frequent used random process known in the literature as *white-noise*. Given the Brownian motion through the relation,

$$d\tilde{x}(t) = \mu dt + \sigma dB(t) = \mu dt + \sigma\sqrt{dt}N(x;0,1) = \mu dt + \sigma dt\frac{\tilde{N}}{\sqrt{dt}}$$

where $\hat{N} = N(x;0,1)$ the stndard normal distribution as usual. The above form can be cast to the familiar deterministic for differential equations by introducing the *white noise* random process as below,

$$W(x;t) = \lim_{\Delta t\to 0}\frac{\tilde{N}}{\sqrt{\Delta t}} = \frac{N(x;0,1)}{\sqrt{dt}} = N(x;0,1/dt), \quad \text{or formally} \quad W(x;t) = \frac{dB}{dt} \tag{1.19}$$

where the last term was obtained by recalling the transformation rule (3.27) for $\mu = 0$ and $\sigma = 1/\sqrt{dt}$. Having defined the white noise process, $W(t)$, we can rewrite (1.17) as,

$$d\tilde{x}(t) = \mu dt + \sigma dt\frac{\tilde{N}}{\sqrt{dt}} = \mu dt + \sigma W(t)dt.$$

Formally this reminds the deterministic differential equations since we can bring dt in the LHS of the above equation to obtain:

$$\dot{\tilde{x}} = \mu(\tilde{x}(t),t) + \sigma(\tilde{x}(t),t)W(t). \tag{1.20}$$

Itô lemma (formula). Brownian model has very special mathematical properties which leads to a reconsideration of the conventional differential calculus rules for deterministic functions. Therefore, each time a stochastic process model is based on the fundamental Brownian (or Wiener) model the respective differential calculus should follow the Itô lemma. A very brief but practical exposition is given below. Let's assume a random process $\tilde{x}(t)$ satisfying (1.18). Assuming another process $\tilde{y}(\tilde{x},t)$, Itô lemma states that $y(x(t),t)$ is a random process as well, satisfying:

$$d\tilde{y} = \left[\frac{\partial \tilde{y}}{\partial t} + \frac{\partial \tilde{y}}{\partial x}\mu(x,t) + \frac{1}{2}\frac{\partial^2 \tilde{y}}{\partial x^2}\sigma^2(x,t)\right] dt + \left[\sigma(x,t)\frac{\partial \tilde{y}}{\partial x}\right] dB(t). \tag{1.21}$$

In the special case where $\tilde{x}(t)$ is the Brownian process itself, $\tilde{x}(t) = B(t)$ [obtained from (1.18) with $\mu = 0, \sigma = 1$] then the above expression specializes to,

$$d\tilde{y} = \left[\frac{\partial \tilde{y}}{\partial t} + \frac{1}{2}\frac{\partial^2 \tilde{y}}{\partial x^2}\right] dt + \left[\frac{\partial \tilde{y}}{\partial x}\right] dB(t). \tag{1.22}$$

3 Arithmetic BM

This BM model is specified by assuming constant mean and variation values for (1.18):

$$\mu(x(t),t) = \mu, \qquad \sigma(x(t),t) = \sigma, \quad \longrightarrow \quad dx(t) = \mu dt + \sigma dB(t). \tag{1.23}$$

In this case, the resulting random quantity can be found by direct integration to give,

$$\int_0^t dx(t') = \int_0^t \mu dt' + \int_0^t \sigma dB(t') \qquad [\mu = cnst, \quad \sigma = cnst]$$
$$\Longrightarrow \quad x(t) - x(0) = \mu(t-0) + \sigma[B(t) - B(0)].$$

Then one ends up to the following exact solution:

$$x(t) = x(0) + \mu t + \sigma B(t), \tag{1.24}$$

since $B(0) = 0$. It is now straightforward, to show that the mean and the variance of a random quantity represented by this model is given by,

$$E[x(t)] = x(0) + \mu t, \tag{1.25}$$
$$V[x(t)] = \sigma^2 t. \tag{1.26}$$

where the statistical properties of $B(t)$, Eqs. (1.10) and (1.11), were used for the derivation.

In plain language, the ABM models the time evolution of the random quantity to depend on (a) one fully deterministic (drift) represented by μ which now coincide with the mean value and (b) a random part with values randomly distributed arround zero, according to the normal distribution and with standard deviation specified by $\sigma\sqrt{dt}$. The above results for the ABM shows that both the mean and the variance for the random quantity grow *linearly* with time. Obviously, as there is nothing that prevents the random quantity to take on negative values it makes this model unsuitable for random quantities that take only positive values. One can model, free of this feature, is the so-called Geometric Brownian Motion (GBM) discussed next.

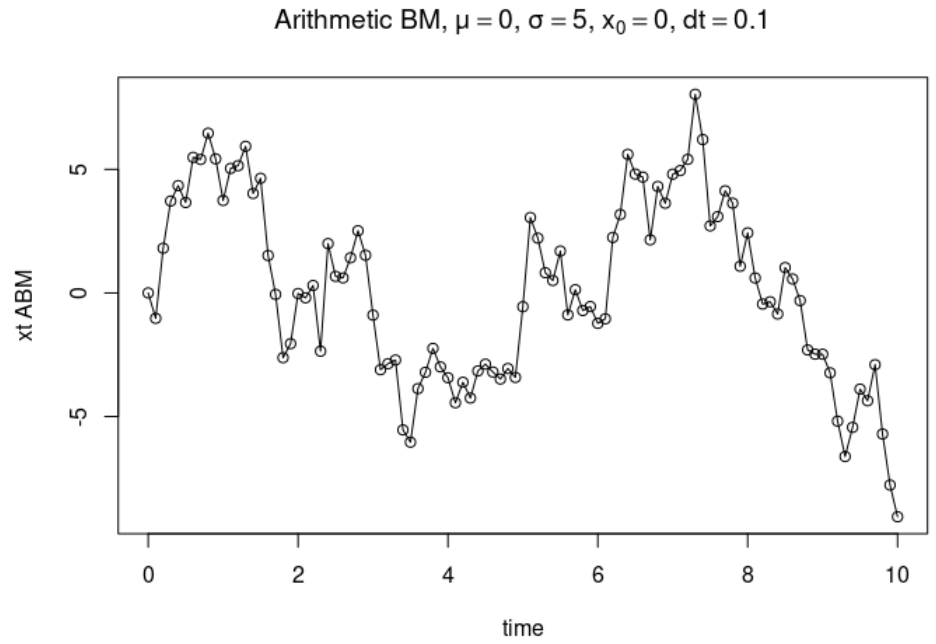


Figure 1.5: Plot of arithmetic BM model time series for mean value (driftless) $\mu = 0$, standard deviation $\sigma = 5$. The random variable starts off from zero $x_0 = 0$. The time step is $dt = 0.1$. In the plot the time points are shown.

4 Geometric Brownian motion

In order to conform with the notation usually adopted in the financial literature a change of notation from $\tilde{x}(t)$ to $S(t)$ is followed. In the context of a financial market trading, $S(t)$ represents the price of a share in the stock market. It is of fundamental importance in the context of commodities pricing (stocks, electricity prices) that it is the percentage change of the prices that is modeled as arithmetic Bronwian motion, namely $dx = dS((t)/S(t)$ [4] and NOT the share price, $S(t)$, itself. Then the time-series stock price can be considered as a BM model specified by assuming for (1.18):

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dB(t) \sim \ N(\mu dt,\ \sigma^2 dt), \tag{1.27}$$

where the last equality is derived by the properties of normal distribution (3.28) and the BM model (1.16) and (1.17). The particular random process is known as *geometric Brownian motion* (GBM) for the share price, $S(t)$. The resulting Brownian model, for constant μ (drift) and σ (volatility) represents an ABM model as described by (1.23). An analytical solution exists for the GBM model as below:

$$S(t) = S(t')e^{(\mu-\sigma^2/2)(t-t')+\sigma(B(t)-B(t'))}, \tag{1.28}$$

where $S(t)$ is the expression for the (random) value of the share price at the elapsed time $t - t'$. So this expression gives $S(t)$ provided it is known at some time t' earlier. Note that in the case where $t' = 0$ we have by definition $B(0) = 0$ and then the above equation simplifies to:

$$S(t) = S_0 e^{(\mu-\sigma^2/2)t+\sigma B(t)}, \tag{1.29}$$

with $S_0 = S(0)$ the initial share price.

4.1 Solving the GBM using the Taylor theorem

At this point the GBM was derived as the Brownian process with the mean and the standard deviation are proportionally dependent on the current value, t, of the process as:

$$\mu(x,t) = \mu S(t), \qquad \sigma(t) = \sigma S(x,t) \tag{1.30}$$

This way one is ended up to the expression (1.27). If, x, had not been random variable ($\sigma = 0$) this equation could be integrated easily to give,

$$\frac{d}{dt}(\ln S(t)) = \frac{1}{S(t)}\frac{dS(t)}{dt} = \mu \quad \Longrightarrow S(t) = S(t_0)e^{\mu(t-t_0)}.$$

But this is not the case here. $B(t)$ is a random variable and therefore $x(t)$ is a random variable as well. The Brownian process is a very special mathematical object and the particular dependence on time gives rise to a different result. If we define $G(t) = \ln S(t)$ the objective here is to find the time derivative of $G(t)$, which by definition is:

$$\frac{dG(t)}{dt} = \lim_{dt\to 0}\frac{G(t+dt) - G(t)}{dt}$$

Use of the chain rule (3.41) and the Taylor theorem (3.40) for functions expansions $G = \ln S(t)$,

$$\begin{aligned}
G(t+dt) &= G(t) + dt\frac{dG}{dt} + \frac{1}{2}dt^2\frac{d^2G}{dt^2} + \cdots \qquad \textit{(Taylor theorem)} \\
&= G(t) + dt\frac{1}{S}\frac{dS}{dt} + dt^2\frac{1}{2}\frac{d}{dt}[\frac{1}{S}\frac{dS}{dt}] + \cdots \qquad \textit{(chain rule)} \\
&= G(t) + dt\frac{1}{S}\frac{dS}{dt} + \frac{1}{2}dt^2[(-\frac{1}{S^2}\frac{dS}{dt})\frac{dS}{dt} + \frac{1}{S}\frac{d^2S}{dt^2}] + \cdots \\
&= G(t) + dt\frac{1}{S}\frac{dS}{dt} + \frac{1}{2}dt^2[(-\frac{1}{S^2}(\frac{dS}{dt})^2 + \frac{1}{S}\frac{d^2S}{dt^2}] + \cdots \\
&= G(t) + \frac{dS}{S} - \frac{1}{2}\frac{dS^2}{S^2} + \frac{1}{S}dt^2\frac{d^2S}{dt^2} + \cdots
\end{aligned}$$

All terms of the Taylor expansion of order $\sim dt^3$ are denoted by $\cdots$. Reordering the Taylor expansion and taking the limit $dt \to 0$ (only terms of the order dt are kept) to obtain:

$$\begin{aligned} dG(t) &= \frac{dS}{S} - \frac{1}{2}(\frac{dS}{S})^2 = \mu dt + \sigma dB(t) - \frac{1}{2}\left[\mu dt + \sigma dB(t)\right]^2 \\ &= \mu dt + \sigma dB(t) - \frac{1}{2}\left[\mu^2 dt^2 + \mu\sigma dt dB(t) + \sigma^2 dB(t)^2\right] \end{aligned}$$

The term with dt^2 should be ignored. What it remains now is to see the order of magnitude for the remaining terms. This means that we should evaluate the order for $dB(t)$ and $dB^2(t)$. From the statistical properties of the Brownian process for $dt = t - t'$ [from (1.12)] we have:

$$E[dB] = \sqrt{dt}$$

and from the definition of the Brownian process where increments are independent each other [also see (1.11)] we have,

$$E[dB^2] = E[dB(x;t)]E[dB(x;t)] = \sqrt{dt}\sqrt{dt} = dt.$$

Therefore $dtdB(t) \sim dt\sqrt{dt} \sim dt^{3/2}$ and $dB(t)^2 \sim dt$ we keep only the latter term to finally obtain:

$$dG(t) = (\mu - \frac{1}{2}\sigma^2)dt + \sigma dB(t). \tag{1.31}$$

This is easily integrated to obtain the claimed analytical expression for the Brownian process of (1.28). Since we have the expression for the infinitesimal change of $G(t)$ the expression for larger time interval follows if we define $\Delta t = t - t_0$ and replace $dG(t) = G(t) - G(t_0) = \ln S(t) - \ln S(t_0)$,

$$\ln S(t) = \ln S_0 + (\mu - \sigma^2/2)\Delta t + \sigma B(t) = \ln S_0 + (\mu - \sigma^2/2)\Delta t + \sigma\sqrt{\Delta t}\hat{N}., \tag{1.32}$$

where $S(t_0) = S_0$. Then by using the *normal distribution* transformation property (3.27) we end to the following important relation for the stock price distribution:

$$G(t) = \ln S(t) = N(\ln S_0 + (\mu - \frac{\sigma^2}{2})\Delta t, \sigma^2\Delta t), \tag{1.33}$$

which says that the stock value, $S(t)$ is *log-normally* distributed rather than *normally distributed* (3.29). It is now straightforward to see that the expectation values for $G(t) = \ln S(t)$ are,

$$E[G(t)] = \ln S_0 + (\mu - \frac{\sigma^2}{2})\Delta t, \tag{1.34}$$

$$V[G(t)] = \sigma^2\Delta t \tag{1.35}$$

4.2 Solving the GBM using Ito's lemma.

Solution of the GBM model can also be achieved by using Ito's lemma (1.21) if we set $\tilde{y} = G(t) = \ln S(t)$. In this case we have,

$$\frac{\partial \tilde{y}}{\partial t} = 0, \qquad \frac{\partial \tilde{y}}{\partial x} = \frac{1}{S} \quad \frac{\partial^2 \tilde{y}}{\partial x^2}\sigma^2(x,t) = -\frac{1}{S^2}$$

The above results together with the GBM expressions for $\mu(x,t)$ and $\sigma(x,t)$ [see (1.30)] are substituded in (1.21) to obtain:

$$dG = \left[0 + (\frac{1}{S})\mu S + \frac{1}{2}(\frac{-1}{S^2})(\sigma S)^2\right] dt + (\sigma S)(\frac{1}{S})dB(t) = (\mu - \frac{1}{2}\sigma^2)dt + \sigma dB(t),$$

in full agreement with (1.31) derived previously.

Chapter 2

Applications of BM models

In the below few applications of the BM are shown. First a brief discussion of the numerical approximation of the GBM is presented since it is fundamental for any application of this model to real life applications, as for example share price stock-market simulation. As a second application, given the analytical solution that the analytical solution for the GBM is known it is exploited in order to estimate confidence intervals for BM forecastings, without the need to perform Monte-Carlo runs. Finally, the third subject discussed here refers to the development of a Web application where Monte-Carlo runs for GBM are quickly produced. The GBM curves are produced using either user inputs (mean value and standard deviation) or based on financial data for stock prices retrieved at real time.

1 Generating ABM and GBM time series

To generate the time-series (in the present work) we assume the process takes place in a time interval $[t_0, t_n]$ while the forecasting time interval is beyond in time $[t_{n+1}, t_{n+k}]$. For convenience we set $t_0 = T_0, t_n = T_1$ and $t_{n+k} = T_2$. These time intervals are equidistantly discretized as $\mathbf{t} = [t_0, t_1, t_2, ...t_n, t_{n+1}, ..., t_k]$, with $dt = t_{i+1} - t_i,\ i = 1, 2, .., n + k$. Where $\mathbf{t}$ denotes the time grid.

1.1 Arithmetic BM time-series simulation

ABM is ideal to illustrate the methods used to generate BM random time series. To this end we can use either the differential form (1.23) or its analytical expression (1.24), which is the solution of (1.23) .

Use of (1.23) Since $dx = x(t + dt) - x(t)$ this expression is written as, $x(t + dt) = x(t) + x(0) + \mu dt + \sigma B(dt)$, since $dB(t) = B(t + dt) - B(t) = B(dt)$. For a discretized time interval one sets $x(t + dt) = xt_{i+1} = x_{i+1}$, $x(t) = x_i$ and $dB(t) = \sqrt{dt}N(0, 1)$ to write,

$$x_{i+1} = x_i + \mu dt + \sigma\sqrt{dt}N(0, 1), \quad i = 0, 1, .., n$$

with $N(0,1)$ as usual the standard normal distribution. This says that the value x_{i+1} is obtained by the knowledge of the value at the previous time step x_i plus the random term,

$$R_i = \mu dt + \sigma\sqrt{dt}\hat{N}, \qquad i = 1, 2, \cdots, n \tag{2.1}$$

It is not difficult to show that the above expression is rewritten as,

$$x_i = x_0 + R_1 + R_2 + \cdots + R_i = x_0 + \sum_{j=1}^{i} R_i \tag{2.2}$$

So to generate the ABM from $[t_0, t_n]$ we need (a) to generate all the n random terms $R_i, i = 1, 2, .., n$ and (b) to obtain the random x_i we need to sum up over all terms up to i-term. So a cumulative sum is required for the efficient generation of the ABM random path.

An example of R code is given below for illustrative purposes:

```
#ABM
# input
mu    = 0                                  # drift
   sd = 5                                  # standard deviation
    T = 10                                  # total time
   n  = 100                                 # number of discrete time points
   x0 = 0.0                                 # initial value
# numerical
   dt = T/n                                 # time step
   t  = seq(0,T,by=dt)                      # time grid [t_0,t_1,...,t_n]
#
   R = mu*dt + sd* sqrt(dt) * rnorm(n,0,1)  # random term i =1,2,..,n
   x = c( x0, R )                           # prepare for ABM, i=1,2,...,n
  xt = cumsum(x)                            # cumulative sum
plot(t,xt,type='l',xlab="time",ylab="xt ABM")          # plot
```

In figure (1.5) it is shown one realization of ABM produced using the above numerical scheme.

Use of (1.24) In the particular case of ABM there is no any difference in using either (1.24) or (1.23). Straightforward application of the discretization time grid and use of (1.24 ends to the exact same numerical scheme derived previously. For this reason there is no any advantage in using the analytical expression for the ABM.

1.2 Geometric BM time-series simulation

In contrast with the ABM process, as we'll see next, the two approaches for the generation of GBM time-series differ each other. The differential definition (1.27) and the explicit solution (1.28) approaches coincide only when the time step becomes sufficiently small. In principle the use of the analytical solution is the most accurate.

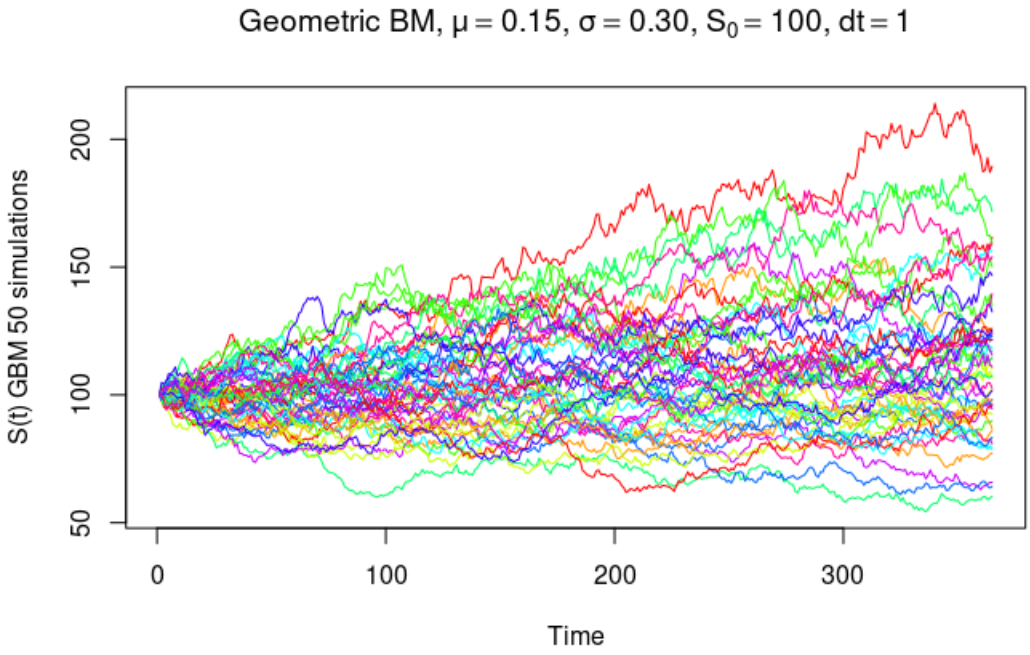


Figure 2.1: Plots of geometric BM model time series for mean value $\mu = 0.15$, standard deviation $\sigma = 0.3$ and $S_0 = 0$. The time step is $dt = 1$. The total plot time is $T = 365$, so each time point represents a day.

Use of (1.27). From this equation we have,

$$dS(t) = \mu S(t)dt + \sigma S(t)dB(t) \quad \longrightarrow S(t+dt) - S(t) = \mu S(t)dt + \sigma S(t)dB(t)$$
$$S(t+dt) = S(t) + \mu S(t)dt + \sigma S(t)dB(t).$$

The above relation when applied for the discretized time-grid is written as,

$$S_{i+1} = S_i + \mu S_i dt + \sigma S_i \sqrt{dt} N(0,1). \qquad i = 0, 1, .., n, \tag{2.3}$$

where $S_i \equiv S(t_i)$. Therefore the value of S_{i+1} it is found by simply adding in the value of S_i the term $\mu S_i dt + \sigma S_i \sqrt{dt} N(0,1)$ (which now explicitly depends on S_i). This means that we end up to a time series for the GBM same as that of the ABM (2.2), but now the random term is different:

$$R_i' = \mu S_i dt + \sigma S_i \sqrt{dt} \hat{N} \tag{2.4}$$

The numerical implementation for the generation of the GBM time-series differs only at this point and as such it is straightforward to generate the GBM based on the simulation scheme of the ABM time-series. Again a cumulative sum of the random term is involved.

Use of (1.28). Thinking along similar lines how to use (1.28) it is quite straightforward to follow; set $t' = t - dt$,

$$\begin{aligned} S(t) &= S(t-dt)e^{\mu-\sigma^2/2)dt+\sigma(B(t)-B(t-dt))} = S(t-dt)e^{\mu-\sigma^2/2)dt+\sigma B(dt)} \\ &= S(t-dt)e^{\mu-\sigma^2/2)dt+\sigma\sqrt{dt}N(0,1)} \end{aligned}$$

where again the basic property of the BM $dB(t) = B(t) - B(t') = B(dt) = N(0, dt)$ was exploited.

Implementing a time grid this expression is written as,

$$S_{i+1} = S_i e^{(\mu-\frac{\sigma^2}{2})dt+\sigma\sqrt{dt}N(0,1)} = S_{i-1}e^{R_i}, \qquad i = 0, 1, .., n, \tag{2.5}$$

where $S(t) = S(t_{i+1}) = S_{+1i}$ and $S(t - dt) = SS(t_i) = S_i$. In the present case the value of S_{i+1} can be found again through the previously obtained one S_i but now multiplied by an exponential (random) factor dependent on dt, μ, σ and $N(0, 1)$. In contrast to the previous approach where we had a cumulative sum to evaluate here we end up to a cumulative product to evaluate. It is easily obtained the following expression for the time-series generation:

$$S_i = S_0 e^{R_1} e^{R_2} \cdots e^{R_{i-1}} = S_0 \Pi_{j=1}^{j=i-1} e^{R_j} = S_0 e^{\sum_{j=1}^{i-1} R_j}, \quad i = 1, 2, ..., n \tag{2.6}$$

Therefore the GBM time series in this case can be generated either as a cumulative product of terms e_i^R or as cumulative sum of terms R_i for $i = 1, 2, ..., n$. An example of such calculation it is shown in figure (2.1).

```
set.seed(154)
nsim   <-  50          # number of random paths
S0     <- 100
mu     <- 0.15
sd     <- 0.30
T      <- 365
gbm    <-  matrix(ncol = nsim,  nrow = t)
 dt    <- 1 / T

for (s in 1:nsim) {    # simulation paths loop
         gbm[1, s] <- S0
  for (j in 2:T) {       # time grid loop
    N <- rnorm(T)
    R <- (mu - sd*sd /2) * dt + sd * N[j] * sqrt(dt)
          gbm[j, s] <- exp(R)
  }
}
#cumulative product      S[i] = So * Prod(e^R}
gbm <- apply(gbm, 2, cumprod)

ts.plot(gbm, gpars = list(col=rainbow(10)))  # plot
```

For both approaches knowledge of dt, μ, σ and n (total time interval is $T = ndt$ is sufficient to generate the GBM. In addition, note that the two approaches coincide for sufficiently small dt. Normally, the second approach is generally the most accurate one and it is the one that is preferred here. To see it consider at the moment

2 Simple GBM forecasting model of stock price

(R code scripts are included in the appendix : simple-GBM-forecast.r)

Having obtained some confidence about simulationg ABM and GBM the next step is to apply the models in realistic problems such as the stock price forecasting. Needless to say that the present work does not aim to investigate whether GBM is a reliable to model stock prices. The purpose here is to illustrate a route of potential usefulness of the model. To this end historical data from real stock prices (available on the Web) will be used to estimate the necessary variables for forecasting, namely the drift rate and the standard deviation (or equivalently the variance). Based on these calculations a GBM forecasting is performed for a time period beyond the available historical data. The mean values and the associated confidence intervals of the stock prices as a function of time are forecasted. The steps followed are presented below:

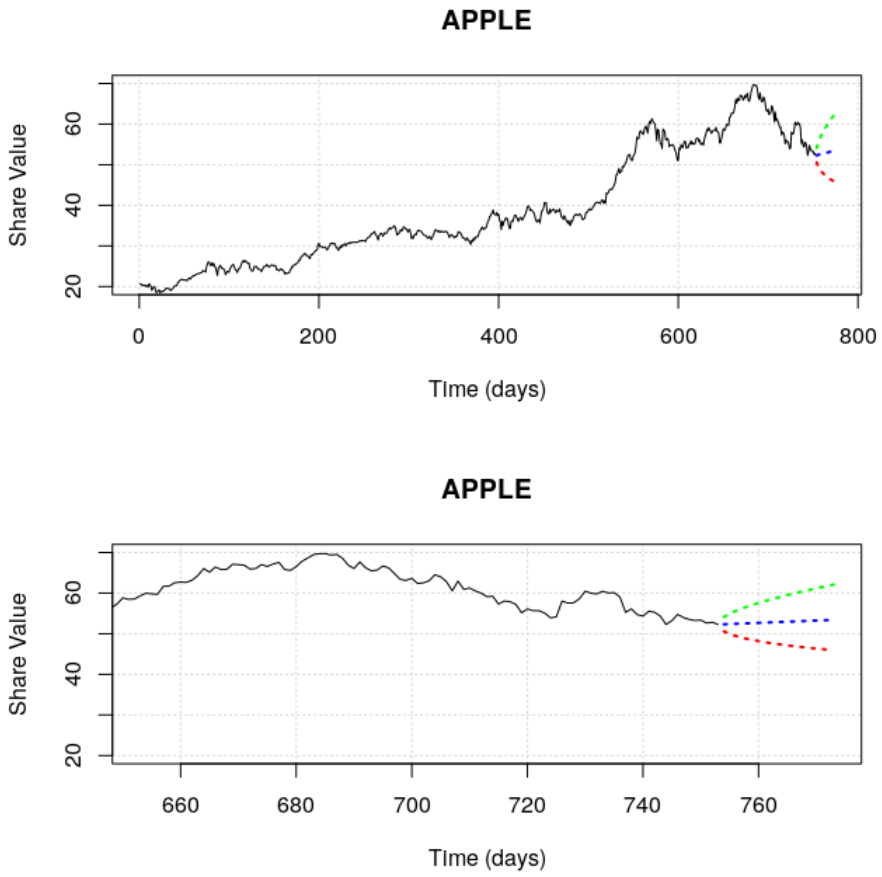


Figure 2.2: Estimated forecast *Apple'*s share price for the last 20 days of the shown period (01/01/2010 - 20/01/2013). *Top graph*: Historical data available from (t=0) to 31/12/2012 (t=753). Confidence intervals are evaluated for 95% probability that share prices will lie within the green (upper value) and red (low value) lines. Blue line indicate the mean value time evolution of the share price. *Bottom figure:* A zoomed version of the top graph.

1. Based on historical time-series data, of size n, of equidistant time points, $t_i, i = 1, 2, ..., n$, using expressions (3.32) the sample mean, m_n and variance, s_n^2 can be calculated as:
$$m_n = \frac{\sum_{i=1}^{n} x_i}{n}, \qquad s_n^2 = \frac{\sum_i (x_i - m_n)^2}{n-1}. \tag{2.7}$$
In the above expression, i may represent days, week, months or any other time period is desirable (for the shown example, just following, i represents days). .

2. Using the calculated sample values m and s^2 we calculate the corresponding

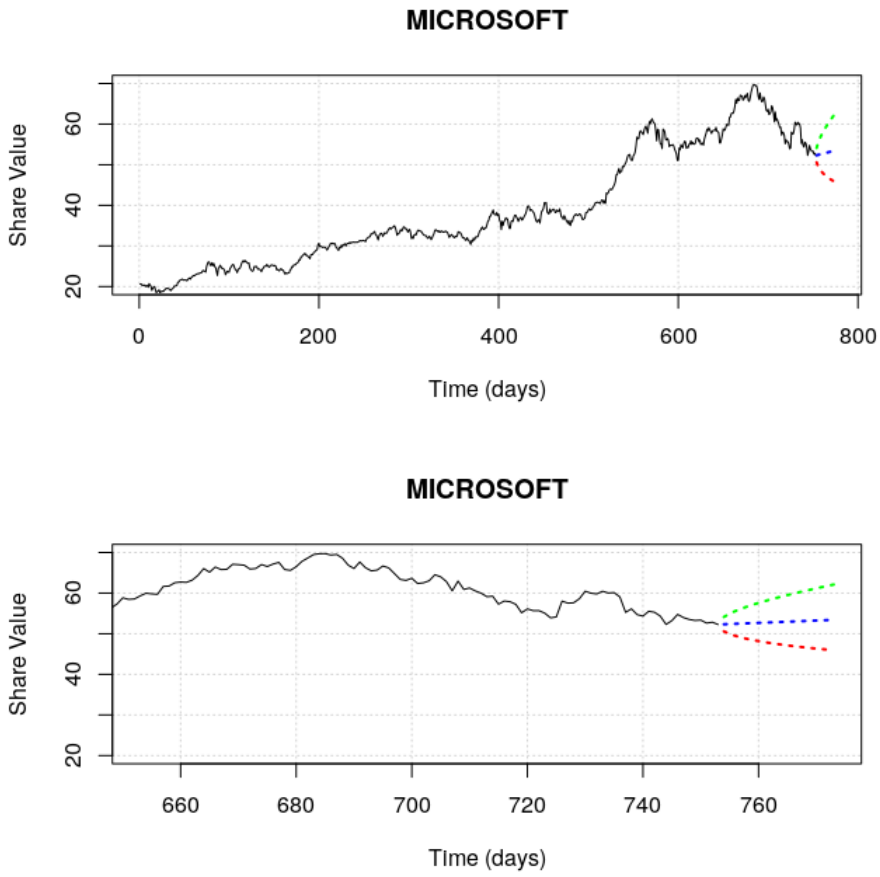


Figure 2.3: Historical data as in figure (2.2) for *MicroSoft*.

confidence intervals, $m_- < m_n < m_+$ and $s_-^2 < s_n^2 < s_+^2$. For this calculation, equations (3.34) and (3.36) are calculated by specifying the value of α and n to have the desired CIs. For example for 95% CI, $\alpha = 0.5$:

$$m_\pm(0.5) = m_n \pm |z_{0.025}|\frac{s_n}{\sqrt{n}} \simeq m_n \pm 1.96\frac{s_n}{\sqrt{n}}$$

$$s_\pm^2(0.5) = s_n^2(1 \pm |z_{0.025}|\sqrt{\frac{2}{n}}) \simeq s_n^2(1 \pm 1.96\sqrt{\frac{2}{n}})$$

3. Next a future time intervals $[t_1, t_2]$ is defined and expectation values for the share price, based on the geometric Brownian model, (1.28) can be calculated.

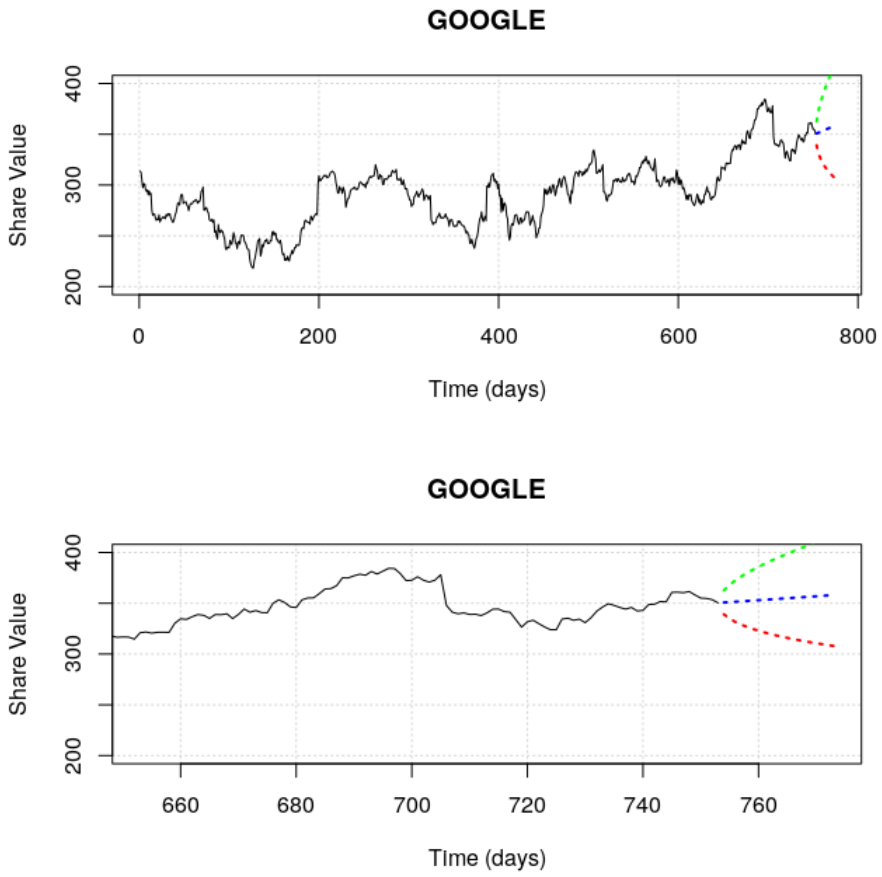


Figure 2.4: Historical data as in figure (2.2) for *Google*.

Stock share price values The above procedure was followed for the plots (2.2) (2.3) and (2.4), corresponding to 3 years share prices from 01/01/2010-31/12/2012. So $t_0 = 1$ corresponds to 01/01/2010 while t_1 corresponds to 31/12/2012. In the figures t covers the period from 01/01/2013 to 20/01/2013. The code used for the estimation is the R programming language while the 'quantmod' financial package was used to download the data from the 'yahoo' website.

At this stage of the project no attempt is made to discuss the results, as the main focus was to develop the theoretical background and the associated implementation code. It is also known that BM modelling for stock share prices is considered unrealistic for various reasons. In addition methods of calculations of mean and variance based on historical data for forecasting use is highly debatable nowadays. As certainly more sophisticated approaches are in the literature, here only a very simple (and straightforward) method is used.

One variation of the above method of calculation of the mean and the variance is to

use different time periods. A longer one for the *mean* and a much shorter for the *variance*.

3 A GBM model Web application for stock prices

(R code scripts are included in the appendix : ui.r and server.R)

The third application is about the development of a Web application where (a) multiple plots of GBM paths can be plotted with inputs provided from the user and (b) forecasting GBM paths are plotted based on historical data retrieved at real time from the Web. The web application is based on a free R package named *Shiny*. Detailed information about Web applications developed through this package can be found in https://shiny.rstudio.com/.

In this application (1.28) is modeled. The procedure to build the reactive web model is briefly presented below [see figures (2.5) and (2.6)]. The former figure refers to a GBM calculation with user defined inputs while the latter to a calculation based on historical data (in this particular case from APPLE). So for the web application we need to do the followings:

1. First we create a directory (folder) e.g. *'shiny-gbm'*
2. Two R scripts are need to placed in this folder, named: *'ui.r'* and '*server.r*'

   - **ui.r:** defines the User Interface for the Web application

   - **server.r:** essentially it contains the main code. This code runs each time the user alters the input values (read by the 'ui.r' script) and clicks a 'submit' button .
3. The Web application can be directly run from inside the Rstudio API and hosted from the system's default browser.

Based on the developed Web application the following 4 forecast time-series for the APPLE, GOOGLE, MICROSOFT and TOYOTA international companies (all well known) are produced. In the associated captions some relevant information is given. As it is not the purpose of this project to fully analyze the behaviour of the stock prices based on the GBM but rather the possibility to use available online data as inputs for the GBM and demonstration of some of the computational tools available to popular data-analysis programming languages (e.g. R or python) I'll not elaborate on these plots.

This Web application will be hosted at the moment at ' https://github.com/anastasiakasara' with the hope that its functionality will be enriched constantly during the following years as more experience is gained.

GBM Monte Carlo Simulation: Kasara Anastasia DBS Higher Diploma in Science Data Analytics

Initial Stock Price
100

Drift Rate (%):
0

Annual Standard Deviation (%)
1

Confidence Interval (%)
95

Number of Simulations
1

Forecast days:
365

Set seed?

Select number of seed
1

Use historical data?

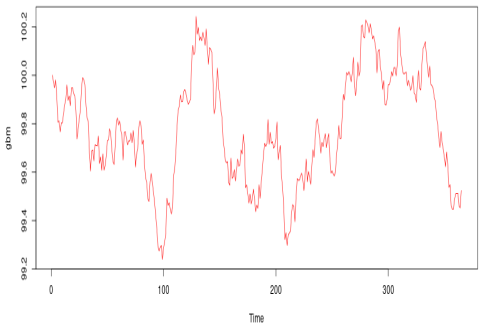
Company's stock market identifier

Start day of historical data:

End day of historical data:

past days plotted
200

Submit



$$S(t) = S(t_0)e^{\left(\mu - \frac{\sigma^2}{2}\right)(t-t_0)+\sigma}$$

Inputs:

1. Initial Stock Price is the current price of the stock;
2. Drift rate is the expected rate of return;
3. Annual Standard Deviation is the volatility of the stock price;
4. Confidence Interval for the plot output;
5. Number of Simulation represents how many simulation of stock price you want to display;
6. Forecast days

check box: mark to set the seed to a fixed value

: unmarked the seed number takes a random value

7. Fix value of seed

check box: mark to use historical data

: unmarked only user input GBM is possible

8. Company's stockmarket keyword, example GOOGL, MSFT, AAPL, TYO
9. start day of historical data: t0
10. end day of historical data: t1
11. plot days of historical data

Figure 2.5: Reactive web application for the simulation of multiply GBM paths based either on user defined inputs or on historical data retrieved from the Web. The freely available *Shiny* R- package was used for the application.

4 Conclusions

At this stage the Geometrical Brownian Motion has been presented and used on available financial time-series data of stock market. This model approach goes beyond the traditional

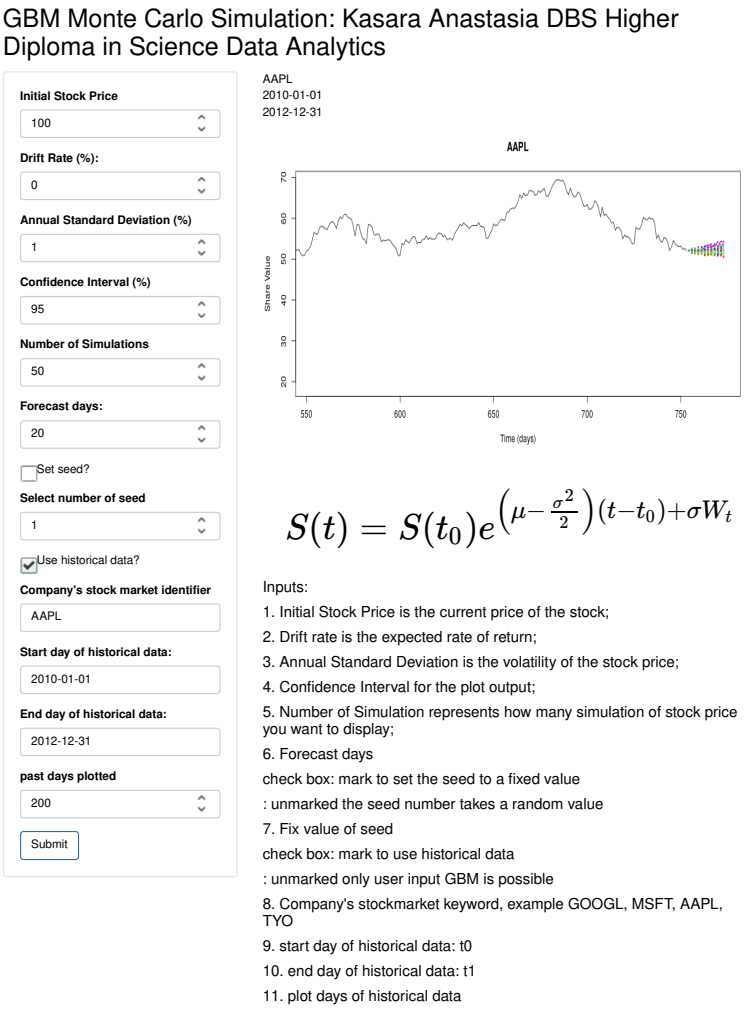


Figure 2.6: Reactive web application User Interface for the simulation of multiply GBM paths on APPLE's historical data retrieved from the Web . Forecast is for 20 days.

linear ARIMA models and has a diverse range of application domains, from Business and finance to biology and medicine and physics. During this project there was the need to review once again the theoretical basis of probability and statitics most necessary for this

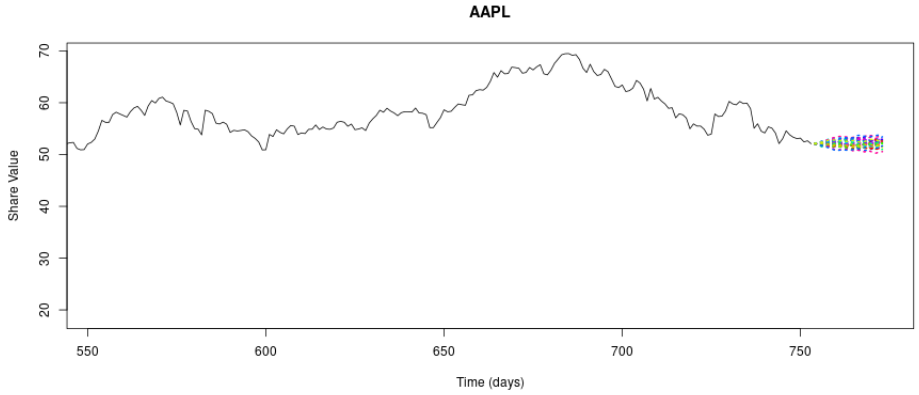


Figure 2.7: Calculated 20 days forecast for APPLE's share price. The day 753 corresponds to 31/12/2012. The historical data prices for the past (past of 31/12/2012) 200 days are shown as well. The number of the GBM curves are 44. Mean value and standard deviation was calculated for the period (01/01/2010 - 31/12/2012)

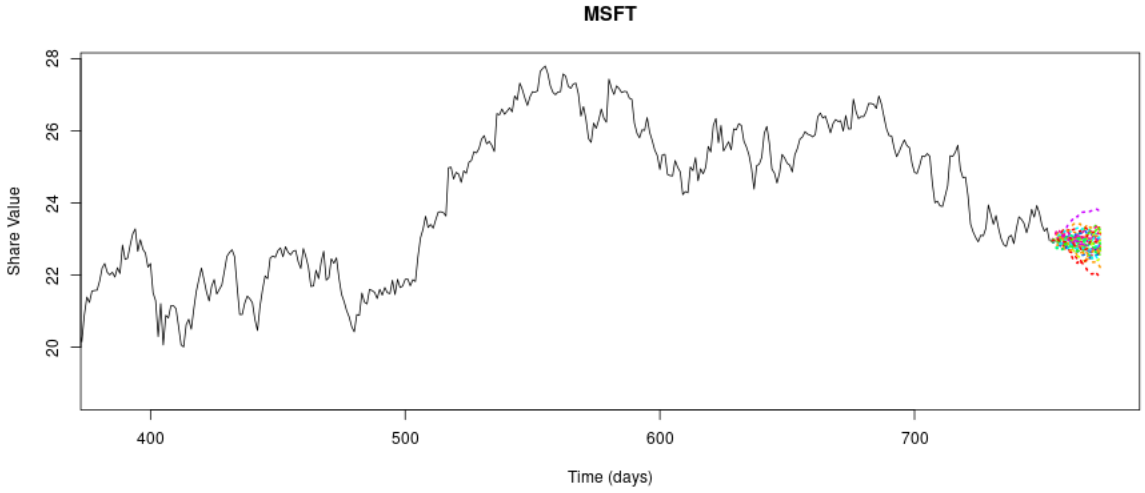


Figure 2.8: Same as of fig (2.7) for MICROSOFT.

work to be self-contained as possible. The use of the publicly (and freely) available R (mostly) and Python is used as the primary programming language. In addition, a web application was developed through the Rstudio/Shiny API. It is hoped this particular Web application to be worked out during the following years to come so that its functionality to include other stock market models than the simple GBM studied here.

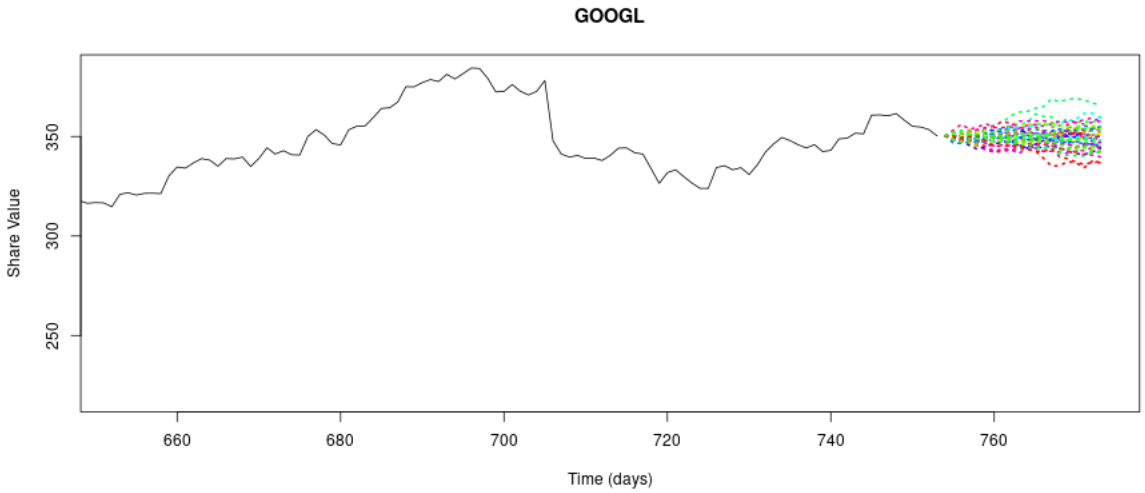


Figure 2.9: Same as of fig (2.7) for GOOGLE.

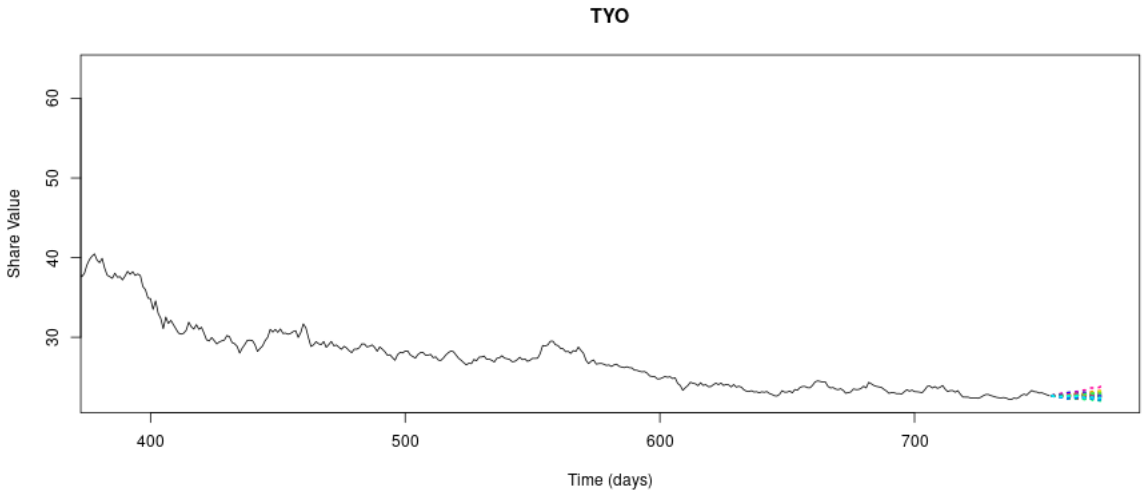


Figure 2.10: Same as of fig (2.7) for TOYOTA.

Chapter 3

Basics of probability theory and statistics

1 Elements of Statistics

1.1 Discrete valued random variables

Assume a random variable, $\tilde{X}$, taking a set (S) of possible values, $x_i, i = 1, 2, .., n$. Next we assume that each of the probabilities that the variable, $\tilde{X}$, will take each of these value is, $P_i, i = 1, 2, ..., N$, namely,

$$P_i = P(X = x_i), \qquad \text{Probability that } X = x_i. \tag{3.1}$$

Then,

$$\sum_{i=1}^{N} P_i = 1, \qquad 0 < P_i < 1, \quad i = 1, 2, .., \tag{3.2}$$

A straightforward and the simplest definition of the probability, p_i, is given by using the occuring frequency, n_i:

$$P_i = P(x_i) = \frac{\#\text{ of occurences of } i \text{ event } (x_i)}{\#\text{ of trials}} = \lim_{n>>1} \frac{n_i}{N}. \tag{3.3}$$

From the above definition and the probability properties (3.2) it is immediately concluded that

$$\sum_{i}^{N} n_i = N. \tag{3.4}$$

Amongst the most important (and useful) parameters characterizing a particular probability distribution, (P_i), are its *mean* and *variance*. The mean (conventionally denoted as μ) is defined as,

$$\mu = E[\tilde{X}] = \frac{\sum_{i=1}^{N} n_i x_i}{\sum_{i}^{N} n_i} = \frac{\sum_{i=1}^{N} n_i x_i}{N} = \sum_{i=1}^{N} (\frac{n_i}{N}) x_i = \sum_{i=1}^{N} P(x_i) x_i, \tag{3.5}$$

where relations (3.4) and (3.3) were used. The variance (or *dispersion*), denoted by σ^2), is defined as the following average:

$$\sigma^2 = V[\tilde{X}] = E[(\tilde{X} - \mu)^2] = \frac{\sum_{i=1}^{N} n_i (x_i - \mu)^2}{\sum_i^N n_i} = ... = \sum_{i=1}^{N} P(x_i)(x_i - \mu)^2 \tag{3.6}$$

The square root of the variance, σ, is called the *standard deviation* of $\tilde{X}$. Using the definition (3.15) for the variance it is easy to show the following handy formula:

$$\sigma^2 = E[\tilde{X}^2] - E[\tilde{X}]^2. \tag{3.7}$$

The average $E[\tilde{X}^2]$ is easily calculated since in general the average of a random function, $f(\tilde{X})$, is defined as,

$$E[f(\tilde{X})] = \sum_{i=1}^{N} P(x_i) f(x_i). \tag{3.8}$$

In the particular case where, $f(\tilde{X}) = \tilde{X}^2$ we end up to

$$E[\tilde{X}^2] = \sum_{i=1}^{N} P(x_i) x_i^2. \tag{3.9}$$

In particular, the quantites $E[\tilde{X}^k]$ are called *moments*. So the mean values is the first moment, $E[\tilde{X}^2]$ is the second moment and so on.:

$$\mu_k = E[\tilde{X}^k] = \sum_{i=1}^{N} P(x_i) x_i^k. \tag{3.10}$$

1.2 Continuously valued random variables.

Without going to details the above definitions and concepts can be generalized in the case that the variables can take continuously any value within a predefined interval $[a, b]$ (very often this is the complete real number axis $-\infty$ to ∞). So in the below the above

expressions are generalized as,

$$P(x)dx \qquad \text{Probability that } X \text{ will take value between } x \text{ and } x+dx, \tag{3.11}$$

$$\int_a^b dxP(x) = 1, \qquad 0 < P(x) < 1, \quad i = 1, 2, .., \tag{3.12}$$

$$P(a < x < b) = \int_a^c dxP(x), \qquad \text{Probability that } a \leq x \leq c., \tag{3.13}$$

$$\mu = E[\tilde{X}] = \int_a^b dxP(x)x, \tag{3.14}$$

$$\sigma^2 = V[\tilde{X}] = \int_a^b dxP(x)(x-\mu)^2 \tag{3.15}$$

$$E[f(\tilde{X})] = \int_a^b dxP(x)f(x). \tag{3.16}$$

$$\mu_k = E[\tilde{X}^k] = \int_a^b dxP(x)x^k. \tag{3.17}$$

2 Bernouli trials and Binomial distribution

Very fundamental discrete valued distribution is the Binomial distribution. Generally, one assumes by that a trial has two outcomes (known as *Bernoulli trials*), with propability $P()p$ (outcome '1') and q (outcome '2'). Necessarily, by definition, $p+q=1$ for each trial. Then the following question can be posed:

After, n trials, what is the probability that the outcome '1' will have occured k times?

Obviously, outcome '1' can occur either 0, or 1, 2, .. or n times. So $k = 0, 1, 2, .., N$. The answer to the above question is given by the *binomial probability distribution* over the k numbers as,

$$P_B(k; n, p) = \binom{n}{k} p^k q^{n-k} = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}. \tag{3.18}$$

The binomial probability does not take into account the occuring order of the events. If one asks for the probability that out of n trials the first k are successes and the last $n-k$ failures then, we must ignore the binomial factor and take:

$$P_B(k; n, p) = p^k q^{n-k}. \tag{3.19}$$

Thus the meaning of the binomial factor is the number of ways that k events may occur out of n trials.

For clarity we can take as an example the case of a coin toss where outcome '1' is *head* and outcome '2' is *tail* with (equal) probability ($p = q = 1/2$). Then the binomial distribution gives immediately the probability that 'heads' will appear e.g. say 3 times after 10 trials, as $P_B(3, 10, 1/2) = (10!/3!(7!)) \times (1/2)^{10} = 8 \times 9 \times 10/(2 \times 3) \times (1/2)^{10} = 0.1171875$.

Within our context, let's define the discrete random variable, $\tilde{X}$, will take only two values $\tilde{X} = 0$ or $\tilde{X} = 1$ with probability p and $1 - p$ respectively. In addition we define another discrete variable $\tilde{Y}$ to be the sum of the independent trials for $\tilde{X}$,

$$\tilde{Y} = \tilde{X}_1 + \tilde{X}_2 + ...\tilde{X}_n = \sum_i \tilde{X}_i,$$

where $\tilde{X}_i$ represents the trial i. Based on this definition the probability for $\tilde{Y}$ to obtain some particular value (between 0 and n) is given by the binomial distribution in (3.18) as $P_B(\tilde{Y} = k; n, p)$.

3 Normal distribution

Assume a random quantity, Q, taking values in the real axis $-\infty < x < \infty$. Amongst the most important probability distributions for such random variable is the so-called *normal* (Gaussian) distribution, $N(\mu, \sigma)$, is the normal distribution, given by,

$$N(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \qquad N(x)dx = Prob(Q \text{ will take a value in } (x, x + dx) \tag{3.20}$$

The above distributions gives the probability that Q will take a value between $(x, x+dx)$. A related useful (and equivalent) distribution is the cumulative normal distribution, defined by:

$$P(x; \mu, \sigma) = \int_{-\infty}^{x} dx' N(x'; \mu, \sigma), \quad N(x)dx = Prob(Q \text{ will take a value in } (-\infty, x] \tag{3.21}$$

which gives the probabilty that the random variable Q will take a value less than x, $P(x) = Prob(Q < x)$. In passing, note that $N(x) = dP(x)/dx$, justifying then the name probability distribution *density* for $N(x)$ and *cumulative* probability for $P(x)$.

Confidence interval (CI) One more useful quantity that gives the probability that Q will take values in $[a, b], a < b$ in terms of $N(x)$, $F(x)$ and μ and σ is given below:

$$P(a < x < b) = P(b) - P(a) = \int_a^b dx N(x; \mu, \sigma) \tag{3.22}$$

This way we can always confidence intervals (CI$(1-\alpha)\%$) defined by the boundary values, $-z_\alpha, z_\alpha$ where the probability that the Q will take a value in between with probability equal to $1 - \alpha$:

$$1 - \alpha = P(-z_\alpha \leq Q \leq z_\alpha) = \int_{-z_\alpha}^{z_\alpha} dx N(x; \mu, \sigma) = P(z_\alpha) - P(-z_\alpha) \tag{3.23}$$

For $\alpha = 0.05$ we have,

$$\text{CI}(95\%) \qquad \alpha = 0.05 \qquad \longrightarrow \qquad z_\alpha = 1.959964\,\sigma \tag{3.24}$$

Moments Quantities of central importance for any probability distribution are the moments, $E_n = E[x^n]$, $n = 1, 2, ...$ For the normal distribution we have for the *central moments*,

$$E[(x-\mu)^n] = \begin{cases} 0, & n = 2k+1 \\ \sigma^n (n-1)!! & n = 2k \end{cases} \tag{3.25}$$

with $(n-1)!! = 1 \times 3 \times 5 \cdots \times (n-3) \times (n-1)$. Based on this formula one can find for the mean (for $n = 1$) and the variance (for $n = 2$) of this quantity is,

$$E[x] = \mu, \qquad V[x] = \sigma^2. \tag{3.26}$$

As it is obvious from the above expressions the statistical properties of the random variable, x, are fully determined by knowning those two values (mean and standard deviation).

Transformation properties of Normal distribution. Let's as consider the random variable $\tilde{x}$ with probability density function $N(x; \mu, \sigma^2)$. The statistical properties of this variable is fully determined by knowledge of μ, σ^2 and of course from its (normal distribution) probability density function (pdf). By definition, we have,

$$E[x] = \mu \qquad V[x] = \sigma^2$$

Assume now the random variable $\tilde{y} = a\tilde{x} + b$. The question is that by knowning the pdf of $\tilde{x}$ what is the pdf of $\tilde{y}$. it can be proven that the values of $\tilde{y}$ are normally distributed with mean equal to $a\mu + b$ and variance, $b^2\sigma^2$. Formally, this is expressed as,

$$\begin{aligned} \tilde{x} = N(x; \mu, \sigma^2) \quad \overset{\tilde{y}=a\tilde{x}+b}{\longrightarrow} \quad & \tilde{y} = N(y; a\mu + b, b^2\sigma^2) \\ & E[y] = a\mu + b \\ & V[y] = b^2\sigma^2 \end{aligned} \tag{3.27}$$

A special case of the above property is to assume a random quantity distributed as $N(x; 0, 1)$ and then take the following linear transformation of it $\tilde{y} = \sigma\tilde{x} + \mu$. Then, we have from the above for $a = 0, b = 1$:

$$\tilde{x} = N(x; 0, 1) = N(\frac{y-\mu}{\sigma}; 0, 1) \quad \overset{\tilde{y}=\sigma\tilde{x}+\mu}{\longrightarrow} \quad \tilde{y} = N(y; \mu, \sigma^2) \tag{3.28}$$

with $\tilde{N}(x) = N(x; 0, 1)$.

3.1 Lognormal distribution

Another important distribution, especially in the field of financial stock-market, is the so-called *lognormal* distribution, $L(y; \mu, \sigma)$, given by,

$$L(y; \mu, \sigma) = \frac{1}{y\sigma\sqrt{2\pi}} e^{-\frac{(\ln y - \mu)^2}{2\sigma^2}} \tag{3.29}$$

The above distribution gives the probability that a particular random quantity, $\tilde{y}$, will take a value between $(y, y+dy)$. A useful alternative representation for $\tilde{y}$ is given in terms of a normally distributed variable, $\tilde{x}$:

$$\tilde{y}(\tilde{x}) = e^{\alpha+\beta\tilde{N}(x)}, \tag{3.30}$$

It can be shown that the log-normal moments are given as,

$$E[Y^n] = e^{n\alpha+n\frac{\beta^2}{2}}. \tag{3.31}$$

Given the above expression, the mean and the variances are readily calculated as,

$$E[Y] = e^{\alpha+\frac{\beta^2}{2}}$$
$$V[Y] = e^{2\alpha+\beta^2}(e^{2\beta^2}-1)$$

4 Confidence intervals (CI) for mean and variance

Let's a sample with $x_1, x_2, ..., x_n$ normally distributed values, each of them distibuted as $N(\mu, \sigma^2)$, with μ and σ^2 with *unknown*:

$$E[x_1] = E[x_2] = ... = E[X_n] = \mu, \qquad V[x_1] = V[x_2] = ... = V[x_n] = \sigma^2$$

Let's now denote as, m_n and s_n^2, the following quantities:

$$m_n = \frac{\sum_i^n x_i}{n}, \qquad s_n^2 = \frac{\sum_i (x_i - m)^2}{n-1} \tag{3.32}$$

When estimating mean and variance values for a random quantity from available data it is expected that we actually calculate the *sample mean*, m_n and *variance*, s_n^2, values, rather than their actual values, μ and σ. When the sample size is infinitely large we would expect that:

$$\lim_{n\to\infty} m_n \longrightarrow \mu, \qquad \lim_{n\to\infty} s_n^2 \longrightarrow \sigma^2.$$

Since the sample size is generally restricted it is expected that the estimated m and s are known with some uncertainty.

Especially for the normal distribution the uncertainty in the calculation of these two values can be considered independently each other. The existence of 'uncertainty' essentially means that these values, themselves, are now random quantities, rather than sure ones. So both have an associated probability distribution for the possible values that can take, and the following properties hold,

$$E[m_n] = \mu \qquad E[s_n^2] = \sigma^2,$$

which say that if one take many different samples of size n and calculate the mean value of m_n and s_n^2 then they should be equal to μ and σ^2.

Below, without proof, the CI intervals are provided for these two quantities:

Sample mean value and CI range The sample mean value $\bar{\mu}$,

$$\bar{m} = \frac{m_n - \mu}{s_n/\sqrt{n}} \qquad \Longleftrightarrow \quad f(t) = \frac{\Gamma((n+1)/2)}{\sqrt{n\pi}\Gamma(n/2)}(1+t^2/n)^{-(n+1)/2} \tag{3.33}$$

follows the so-called *Student's t−distribution*, where m_n and s_n are the sample's mean and variance values. The CI range for this random variable is given by

$$\begin{aligned} &100(1-\alpha)\% \text{ CI for mean sample} \\ m_+(\alpha) &= m_n \pm f(\alpha, n-1)\frac{s_n}{\sqrt{n}} \simeq m_n + |z_{a/2}|\frac{s_n}{\sqrt{n}} \\ m_+(\alpha) &= m_n \pm f(\alpha, n-1)\frac{s_n}{\sqrt{n}} \simeq m_n - |z_{a/2}|\frac{s_n}{\sqrt{n}} \end{aligned} \tag{3.34}$$

Sample variance value and CI range The sample variance value $\bar{s}$,

$$\bar{s}^2 = \frac{(n-1)s_n^2}{\sigma^2} = \frac{\sum_i((x_i-m)^2}{\sigma^2}, \qquad \Longleftrightarrow \quad \chi^2_k(x) = \frac{x^{k/2-1}e^{-x/2}}{2^{k/2}\Gamma(k/2)} \tag{3.35}$$

with $\Gamma(k/2)$ the Γ function. The probability distribution $\chi_k(x)$ is the so-called *chi-squared distribution*. In the particular case we have $k = n-1$ degrees of freedom (df). The CI range for this random variable is given by,

$$\begin{aligned} &100(1-\alpha)\% \text{ CI for variance sample} \\ \bar{s}^2_-(\alpha) &= s_n^2\frac{(n-1)}{\chi^2_{n-1}(\alpha/2)} \simeq s_n^2(1+|z_{a/2}|\sqrt{\frac{2}{n}}) \\ \bar{s}^2_+(\alpha)) &= s_n^2\frac{(n-1)}{\chi^2_{n-1}(1-\alpha/2)} \simeq s_n^2(1-|z_{a/2}|\sqrt{\frac{2}{n}}) \end{aligned} \tag{3.36}$$

where $\bar{s}^2_\pm$ denote the lower (-) and the upper (+) CI limit, respectively. Note that for $\alpha = 0.05$ we have $|z_{a/2}| = 1.959964$.

5 Central Limit Theorem

Consider n random variables, $x_i,\ i = 1, 2, .., n$ with

$$E[x_i] = \mu_i, \qquad V[x_i] = \sigma_i^2,$$

where μ_i and σ_i^2 are the corresponding means and variances. There is one theorem of fundamental importance in the theory of statistical analysis (and in the probability theory in general) for the case where the number of random variables, n, becomes very large.

Random sums Very often sums of such random quantities are of great importance[1]. The Central Limit Theorem (CLT), expressed below provides a quantitative formula to approximate the probabilty distribution of such sums.

Consider the *sum* function of these random variables, $\tilde{x}$. It is easily shown that for the sum variable,

$$\tilde{x} = \sum_{i=1}^{n} \tilde{x}_i, \tag{3.37}$$

the mean value and its variance are given by,

$$\mu = E[\tilde{x}] = \sum_i \mu_i \qquad \sigma^2 = V[\tilde{x}] = \sum_i \sigma_i^2. \tag{3.38}$$

Then, the CLT theorem states that:

CLT for random sums: *If the random variables take real values, then the probability distribution $P(\tilde{x})$ of $\tilde{x}$ for large n approaches the normal distribution with mean μ and variance σ^2:*

$$P(\tilde{x}) \longrightarrow N(x; \mu, \sigma^2), \qquad n \to \infty \tag{3.39}$$

In practice, n need not be too large for the probability distribution of the sum to approach the normal distribution. Also the individual random variables, x_i neither need to be normally disributed or independent each other.

5.1 Differential calculus formulas

Taylor theorem One of the most fundamental theorems of differential calculus is the Taylor theorem which for a single-variable function relates the value of the function at a shifted value of its argument with its derivatives:

$$f(x+dx) = f(x) + dx f'(x) + \frac{1}{2!} dx^2 f''(x) + ... = \sum_n \frac{f^{(n)}(x)}{n!} dx^n. \tag{3.40}$$

where $n! = 1 \cdot 2 \cdot ... \cdot n$ and $f^{(n)}(x)$ is the $n-$th order derivative, with $f^{(0)}(x) = f(x)$.

[1] One such example is the average of the results of a random quantity (e.g. coin toss)

Chain rule for derivatives The chain rule is used to calculate the derivative of a function dependent on another function; Given the known functions $y = y(x)$ and $G = G(y)$ we have for the first-order derivative of $G(y)$ with respect to x:

$$G(y) = F(y(x)) \quad \Longrightarrow \quad \frac{dG(y(x))}{dx} = \frac{dy(x)}{dx}\frac{dF(y)}{dy} = y'(x)F'(y) \tag{3.41}$$

Chapter 4

Source code

1 Simple GBM forecasting model of stock price

```
#simple-GBM-forecast.r
#
# Kasara Anastasia  INterim Report 09/07/2018
#
#
install.packages("quantmod")
install.packages("timeSeries")
#install.packages("Defaults")
library(timeSeries)
library(quantmod)

#GOOGL = GOOGLE
#MSFT = MICROSOFT
#TYO  = TOYOTA
#AAPL = APPLE


#
# Here insert company's keyword
#
name         = "GOOGLE"
ticker       = "GOOGL"
variable     = "Adjusted"
column_name  = paste(ticker,variable,sep=".")


print(column_name)
# downloading stock price
start="2010-01-01"
end="2012-12-31"

# returns an xts object
portfolio    <- getSymbols(Symbols = ticker,from= start,to = end,auto.assign=FALSE)
portfolio.df <- as.data.frame(portfolio)  # same data as dataframe

print(paste("Company    ticker:", ticker     ))
print(paste("Historica l   data:", column_name ))
print(paste("             start:", start       ))
print(paste("               end:", end         ))
```

```
# some descriptive statistics  (tail,summary is not working for xts)
str(portfolio)
head(portfolio)
#
tail(portfolio.df)
summary(portfolio.df)
# some charts
barChart(portfolio)
#chartSeries(portfolio)
dev.off()
# get variable to be simulated  (adjusted closed price)
Y <- coredata(portfolio[, column_name] )

size_y = length(Y)


#
t0      <- 1                                #  t0    = first day
t1      <- size_y                           #  t1    = last day
dt_01   <- c(t0:t1)                         #  dt_01 = [t0,t1]
Y01     <- Y[dt_01]                         #  y01   = y(1),y(2),...y(t1)
Y01.ret <- returns(Y1)             # historical returns = (Y(t1)-Y(t0))/Y(t0)


n = length(Y01.ret)

#
#
# STANDARD DEVIATION/VARIANCE (VOLATILITY)
#
#

s01  <-   sd(Y01.ret, na.rm = T)           # sd   (volatility)
m01  <- mean(Y01.ret, na.rm = T)           # mean value (drift)
v01  <- s01*s01                           # variance

#
# confidence intervals
#

#
# a = 0.5
#

alpha = 0.5

# sample standard deviation follows a chi-squared-distribution

vlow = (n-1)* v01 / qchisq(1-alpha/2.,  df = n-1)
vup  = (n-1)* v01 / qchisq(alpha/2,  df = n-1)

s01_low <- sqrt(vlow)
s01_up  <- sqrt(vup )

s01_95 <- c(s01_low, s01, s01_up)         # sd confidence interval


# sample mean value follows a t-distribution

m01_low <- m01 - qt(1-alpha/2, df = n-1) * s01 / sqrt(n)   # 95% confidence interval for the mean
m01_up  <- m01 + qt(1-alpha/2., df = n-1) * s01 / sqrt(n)

m01_95  <- c(m01_low, m01, m01_up)            # mean value confidence interval
```

```
print(paste("        Mean value = ", m01))
print(paste("Standard deviation = ", s01))
print(paste("          Variance = ", v01))


# Plot for the future share price

t    <- 1:20            # relative future time from t1, t1+1,...,t+20
yt   <- log(  Y01[t1] )  + (m01 - v01/2) * t  # mean value prediction based on Normal Distribution
st   <- exp(yt)
st_l <- exp(yt - 1.96 * s01* sqrt(t))
st_u <- exp(yt + 1.96 * s01* sqrt(t))




plot( Y[ t0:(t1 + max(t)) ],   # stock values
 type = 'l',                     # line plot
     xlim = c(t1-100, t1 + max(t)),
     ylim = c(200, 400),
     xlab = "Time (days)",
     main = paste(name),
     ylab = "Share Value",
     panel.first = grid())


#
# add in the plot the estimated forecast
#

xt      <- c(t1 + t)
lines(x = xt, y = st_l, lty = 'dotted', col = 'red',    lwd = 2) # lower bound stock values
lines(x = xt, y = st,   lty = 'dotted', col = 'blue',   lwd = 2) # expected stock values
lines(x = xt, y = st_u, lty = 'dotted', col = 'green',  lwd = 2) # upper bound stock values

#
###############
```

2 A GBM model Web application for stock prices

```
#ui.r
library(timeSeries)
library(quantmod)
library(shiny)

shinyUI(
  fluidPage(

    titlePanel("GBM Monte Carlo Simulation: Kasara Anastasia DBS Higher Diploma in Science Data Analytics"),

    sidebarLayout(

      sidebarPanel(

        numericInput("initPrice", "Initial Stock Price",           min = 1, value = 100 ), #1
        numericInput("drift"    , "Drift Rate (%):",               min = 0, value = 0  ),  #2
        numericInput("stdev"    , "Annual Standard Deviation (%)", min = 0, value = 1  ),  #3
        numericInput("confint"  , "Confidence Interval (%)",       min = 1, value = 95  ), #4
        numericInput("simul"    , "Number of Simulations",         min = 1, value = 1   ), #5
        numericInput("time"     , "Forecast days:",             min = 1, value = 365 ), #6
```

```
        checkboxInput("seeds"   , "Set seed?"),
        numericInput("setseed"  , "Select number of seed",         min = 1, value = 1   ), #7
        checkboxInput("hist"    , "Use historical data?"),
        textInput("name"        , "Company's stock market identifier"),                    #8
        textInput("start"       , "Start day of historical data:"),                     #9
        textInput("end"         , "End   day of historical data:"),                     #10
        numericInput("past"     , "past days plotted",             min =0, value= 200),    #11
        submitButton("Submit")
      ),

      mainPanel(
        textOutput("err"),
        textOutput("name"),
        textOutput("ti"),
        textOutput("tf"),
        plotOutput("distPlot"),
        headerPanel(withMathJax("$$S(t) = S(t_0) e^{\\left(\\mu
        - \\frac{\\sigma^2}{2}\\right)(t-t_0) + \\sigma W_t} $$")),
        h4("Inputs:"),
        h4("1. Initial Stock Price is the current price of the stock;"),
        h4("2. Drift rate is the expected rate of return;"),
        h4("3. Annual Standard Deviation is the volatility of the stock price;"),
        h4("4. Confidence Interval for the plot output;"),
        h4("5. Number of Simulation represents how many simulation of stock price you want to display;"),
        h4("6. Forecast days"),
        h4("check box: mark to set the seed to a fixed value"),
        h4("         : unmarked the seed number takes a random value"),
        h4("7. Fix value of seed"),
        h4("check box: mark to use historical data"),
        h4("         : unmarked only user input GBM is possible"),
        h4("8.  Company's stockmarket keyword, example GOOGL, MSFT, AAPL, TYO"),
        h4("9.  start day of historical data: t0"),
        h4("10. end day of historical data: t1"),
        h4("11. plot days of historical data")
      )
    )
  )
)

#server.r
library(shiny)

shinyServer(function(input, output) {

  output$name <- renderText({

#    if(input$hist == FALSE){
#            print("USER DEFINED VALUES")
#    }
#      else
    print(input$name)
#    }
    })
  output$ti   <- renderText({ print(input$start) })
  output$tf   <- renderText({ print(input$end)   })

  output$distPlot <- renderPlot({

    if (input$seeds == TRUE) {
    set.seed(input$setseed)
          }
```

```
  if(input$hist == TRUE){

     acronym = input$name    #"GOOGL"  company's keyword
       start = input$start   #"2010-01-01"
         end = input$end     #"2012-12-31"

      variable     = "Adjusted"
      column_name  = paste(acronym,variable,sep=".")
      print(column_name)

      # returns an xts object
      portfolio <- getSymbols(Symbols = acronym,from= start,to = end,auto.assign=FALSE) # xts object
      Y         <- coredata(portfolio[, column_name] ) # sumulated variable (adjusted  closed price)
      size_y    <- length(Y)
      t0      <- 1                        #  t0    = first day
      t1      <- size_y                   #  t1    = last day
      dt_01   <- c(t0:t1)                 #  dt_01 = [t0,t1]
      Y01     <- Y[dt_01]                 #  y01   = y(1),y(2),...y(t1)
      Y01.ret <- returns(Y01)             # historical returns = (Y(t1)-Y(t0))/Y(t0)
      (n = length(Y01.ret))

      # if(n<input$time){
      #   output$err   <- renderText({ print("ERROR: HISTORICAL DATA LESS THAN FORECAST DAYS. ABORT.") })
      #   stopApp(7)
      #   }

      #some info
      #str(portfolio)
      #head(portfolio)


      # STANDARD DEVIATION/VARIANCE (VOLATILITY)

      s01  <-   sd(Y01.ret, na.rm = T)           # sd   (volatility)
      m01  <- mean(Y01.ret, na.rm = T)           # mean value (drift)
      S0   <- Y01[t1]
#      v01  <- s01*s01                           # variance

      }else{
          t0    <- 1
          t1    <- t0
          m01   <- input$drift/100
          s01   <- input$stdev/100
          S0    <- input$initPrice
          acronym = "User defined Brownian motion"
  }


    total_forecast  <- input$time             #forecast time
    total_past      <- input$past
    nsim   <- input$simul
     gbm   <- matrix(ncol = nsim, nrow = total_forecast)
#     gbm_2 <- matrix(ncol = nsim, nrow = total_forecast)
     dt = 1 / total_forecast              #time-step

          for (simu in 1:nsim) {

                  gbm[1, simu]   <- S0
#                  gbm_2[1, simu] <- S0
                  ds = 0
                  for (day in 2:total_forecast) {

                          epsilon <- rnorm(total_forecast)     #N(n,0,1)
                          R = (m01 - s01*s01/2.) * dt + s01 * epsilon[day] * sqrt(dt)
```

```
                        gbm[day, simu] <- exp(R)

                }
          }
          gbm <- apply(gbm, 2, cumprod)    #cumulative product for GBM

          t2p <- t1 - total_past
          t2f <- t1 + total_forecast
          t   <- 1:total_forecast
          xt  <- c(t1 + t)
          if(input$hist == TRUE){

            ts.plot( Y[ t0:t2f],       # stock values
                  type = 'l',                          # line plot
                  xlim = c(t2p, t2f),
#                  ylim = c(200, 400),
                  xlab = "Time (days)",
                  main = paste(acronym),
                  ylab = "Share Value",
                  panel.first = grid())

            for(simu in 1:nsim){ lines(x = xt,
            y = gbm[1:total_forecast,simu], lty = 'dotted', col = sample(rainbow(10)),    lwd = 2)
            }

          }
          else{
          ts.plot(gbm, gpars = list(col=rainbow(10)))
#          lines(x = xt, y = gbm_2, lty = 'dotted', col = sample(rainbow(10)), lwd = 2)
        }
  })
})
```

3 Python code for Binomial plots

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Jul  3 18:39:11 2018

@author: natasa
"""
import matplotlib.pyplot as plt
import scipy.stats as stats
import numpy as np
fig=plt.figure()
n=100
k=np.arange (0,100)
p=0.5
binomial=stats.binom.pmf(k,n,p)
ax1=plt.subplot(221)
plt.plot(binomial,color='k',linestyle='dashed',marker='o')
plt.xlabel("steps")
plt.ylabel("probability ")
plt.title("n=100, p=0.5")


#n=100
k=np.arange (0,100)
p1=0.1
binomial=stats.binom.pmf(k,n,p1)
```

```
ax2=plt.subplot(223)
plt.plot(binomial,color='k',linestyle='dashed',marker='o')
plt.xlabel("steps")
plt.ylabel("probability ")
plt.title("n=100, p=0.1")



n3=10
k3=np.arange (0,10)
p=0.5
binomial=stats.binom.pmf(k3,n3,p)
ax3=plt.subplot(222)
plt.plot(binomial,color='k',linestyle='dashed',marker='o')
plt.xlabel("steps")
plt.ylabel("probability ")
plt.title("n=10, p=0.5")

n=100
k3=np.arange (0,100)
p4=0.7
binomial=stats.binom.pmf(k,n,p4)
ax4=plt.subplot(224)
plt.plot(binomial,color='k',linestyle='dashed',marker='o')
plt.xlabel("steps")
plt.ylabel("probability ")
plt.title("n=100, p=0.7")

plt.tight_layout()
plt.savefig("binomial.png")
```

3.1 R plots for binomial distribution (as Python)

```
setwd("~/rstudio")
par(mfrow=c(2,2))
n=100
p=0.5

p_steps<-numeric(n)
for ( i in 1:n) {
  p_steps[i]=dbinom(i,n,p)
}

rbinom(30,100,0.5)
print(p_steps[50])

plot(p_steps,type='l',main="n=100,p=0.5",ylab="probability")

################################################
n1=10
#p=0.5
p_steps_2<-numeric(n1)
for ( i in 1:n1) {
  p_steps_2[i]=dbinom(i,n1,p)
}

rbinom(30,100,0.5)
print(p_steps_2[25])
plot(p_steps_2,type='l',main="n=10,p=0.5",ylab="probability")

################################################
#n=100
```

```
p2=0.1
p_steps_3<-numeric(n)
for ( i in 1:n) {
  p_steps_3[i]=dbinom(i,n,p2)
}
print(p_steps_3[25])
plot(p_steps_3,type='l',main="n=100,p=0.1",ylab="probability")
#####################################################
p3=0.7
p_steps_4<-numeric(n)
for ( i in 1:n) {
  p_steps_4[i]=dbinom(i,n,p3)
}
print(p_steps_4[25])
plot(p_steps_4,type='l',main="n=100,p=0.7",ylab="probability")
dev.off()
```

Bibliography

[1] G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. Time-series Analysis Forecasting and Control. 1994.

[2] E. Frey and Klaus Kroy. Brownian motion: a paradigm of soft matter and biological physics. Ann. Phys., 14:20–50, 2005.

[3] Daniel Gillespie and Effrosyni Seitaridou. Simple Brownian Diffusion: An Introduction to the Standard Theoretical Models. 2013.

[4] John C. Hull. Options, Futures, and Other Derivatives. 2017.

[5] Frantisek Shanina. Essential of econophysics modelling. 2014.