Chapter 1
Modeling Market Returns

In an environment subject to risk and randomness, any result or prediction has to rely on a given statistical model. This chapter reviews basic Gaussian modeling for the returns risky assets using Brownian motion and geometric Brownian motion. We also include a statistical benchmarking of such Gaussian-based models to actual market returns.

1.1 Random Walks

Our approach to the modeling of the price of risky assets is based on the use of random walks indexed by discrete or continuous time. We consider the random walk $(S_n)_{n\geqslant 0}$, also called the *Bernoulli* random walk, and defined by letting $S_0 := 0$ and

$$S_n := \sum_{k=1}^{n} X_k = X_1 + \cdots + X_n, \qquad n \geqslant 1,$$

where the increments $(X_k)_{k\geqslant 1}$ form a sequence of *independent and identically distributed* (*i.i.d.*) Bernoulli random variables, with distribution

$$\begin{cases} \mathbb{P}(X_k = +1) = p, \\ \mathbb{P}(X_k = -1) = q, \qquad k \geqslant 1, \end{cases}$$

with $p+q=1$. In other words, the random walk $(S_n)_{n\geqslant 0}$ can only evolve by going up or down by one unit within the finite state space $\{0,1,\ldots,S\}$. We have

$$\mathbb{P}(S_{n+1}=k+1 \mid S_n=k)=p \quad \text{and} \quad \mathbb{P}(S_{n+1}=k-1 \mid S_n=k)=q,$$

$k\in\mathbb{Z}$.

Proposition 1.1. *The mean value of S_n can be expressed as*

$$\mathbb{E}[S_n \mid S_0=0]=\mathbb{E}\left[\sum_{k=1}^{n} X_k\right]=\sum_{k=1}^{n}\mathbb{E}\left[X_k\right]=(2p-1)n=(p-q)n,$$

and its variance can be computed as

$$\mathrm{Var}\left[S_n \mid S_0=0\right]=\mathrm{Var}\left[\sum_{k=1}^{n} X_k\right]=\sum_{k=1}^{n}\mathrm{Var}\left[X_k\right]=4npq.$$

Figure 1.1 enumerates the $120=\binom{10}{7}=\binom{10}{3}$ possible paths corresponding to $n=5$ and $k=2$, which all have the same probability $p^{n+k}q^{n-k}=p^7q^3$ of occurring.

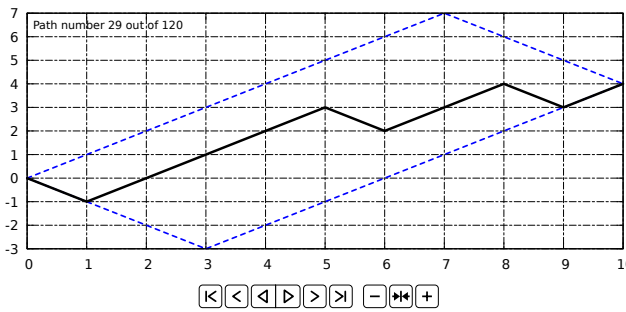


Fig. 1.1: Graph of $120=\binom{10}{7}=\binom{10}{3}$ paths with $n=5$ and $k=2$.*

Proposition 1.2. *The probability distribution of S_{2n}, $n\geqslant 1$, is given by*

* Animated figure (works in Acrobat Reader).

This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html

$$\mathbb{P}(S_{2n} = 2k \mid S_0 = 0) = \binom{2n}{n+k} p^{n+k} q^{n-k}, \quad -n \leqslant k \leqslant n. \tag{1.1}$$

In addition, we note that in an even number of time steps, $(S_n)_{n\in\mathbb{N}}$ can only reach an even state in $\mathbb{Z}$ after starting from 0.

Similarly, in an odd number of time steps, $(S_n)_{n\in\mathbb{N}}$ can only reach an odd state in $\mathbb{Z}$ after starting from 0.

Brownian Motion

The modeling of random assets in finance is mainly based on stochastic processes, which are families $(X_t)_{t\in I}$ of random variables indexed by a time interval I. Brownian motion is a fundamental example of a stochastic process.

Brown (1828) observed the movement of pollen particles as described in "A brief account of microscopical observations made in the months of June, July and August, 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies." Phil. Mag. 4, 161-173, 1828.

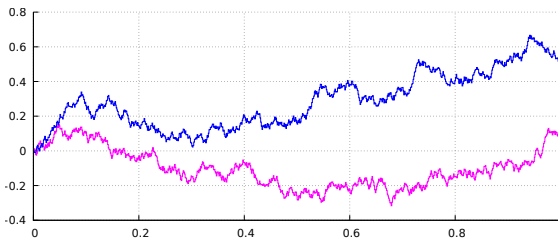


Fig. 1.2: Two sample paths of one-dimensional Brownian motion.

Einstein (1905) received his 1921 Nobel Prize in part for investigations on the theory of Brownian motion: "... in 1905 Einstein founded a kinetic theory to account for this movement", presentation speech by S. Arrhenius, Chairman of the Nobel Committee, Dec. 10, 1922.

Bachelier (1900) used Brownian motion for the modeling of stock prices in his PhD thesis "Théorie de la spéculation", Annales Scientifiques de l'Ecole Normale Supérieure 3 (17): 21-86, 1900.

Wiener (1923) is credited, among other fundamental contributions, for the mathematical foundation of Brownian motion, published in 1923. In particular he constructed the Wiener space and Wiener measure on $\mathcal{C}_0([0,1])$ (the space of continuous functions from $[0,1]$ to $\mathbb{R}$ vanishing at 0).

Itô (1944) constructed the Itô integral with respect to Brownian motion, and the stochastic calculus with respect to Brownian motion, which laid the foundation for the development of calculus for random processes, see Itô (1951) "On stochastic differential equations", *in* Memoirs of the American Mathematical Society.

Definition 1.3. *The standard Brownian motion is a stochastic process* $(B_t)_{t\in\mathbb{R}_+}$ *such that*

1. $B_0 = 0$ *almost surely,*
2. *The sample paths* $t \mapsto B_t$ *are (almost surely) continuous.*
3. *For any finite sequence of times* $t_0 < t_1 < \cdots < t_n$*, the increments*

$$B_{t_1} - B_{t_0}, B_{t_2} - B_{t_1}, \ldots, B_{t_n} - B_{t_{n-1}}$$

are independent.

4. *For any times* $0 \leqslant s < t$*,* $B_t - B_s$ *is normally distributed with mean zero and variance* $t - s$*.*

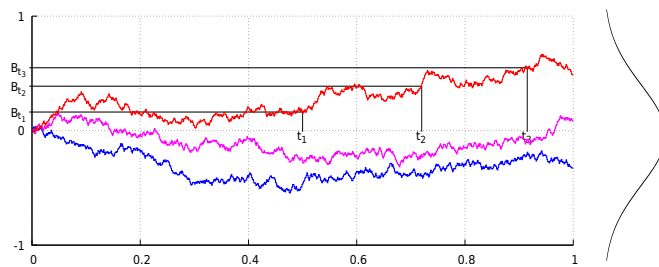


Fig. 1.3: Sample paths of a one-dimensional Brownian motion.

See e.g. Chapter 1 of Revuz and Yor (1994) and Theorem 10.28 in Folland (1999) for proofs of existence of Brownian motion as a stochastic process $(B_t)_{t\in\mathbb{R}_+}$ satisfying the Conditions *1-4* of Definition 1.3.

This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html

Approximating Brownian motion as a random walk

We will informally regard Brownian motion as a random walk over infinitesimal time intervals of length Δt, whose increments

$$\Delta B_t := B_{t+\Delta t} - B_t \simeq \mathcal{N}(0, \Delta t)$$

over the time interval $[t, t+\Delta t]$ will be approximated by the Bernoulli random variable

$$\Delta B_t \approx \pm\sqrt{\Delta t} \tag{1.2}$$

with equal probabilities $(1/2, 1/2)$, hence

$$\mathbb{E}[\Delta B_t] = \frac{1}{2}\sqrt{\Delta t} - \frac{1}{2}\sqrt{\Delta t} = 0,$$

and

$$\mathrm{Var}[\Delta B_t] = \mathbb{E}\left[(\Delta B_t)^2\right] = \frac{1}{2}\Delta t + \frac{1}{2}\Delta t = \Delta t.$$

Figure 1.4 presents a simulation of Brownian motion as a random walk with $\Delta t = 0.1$.

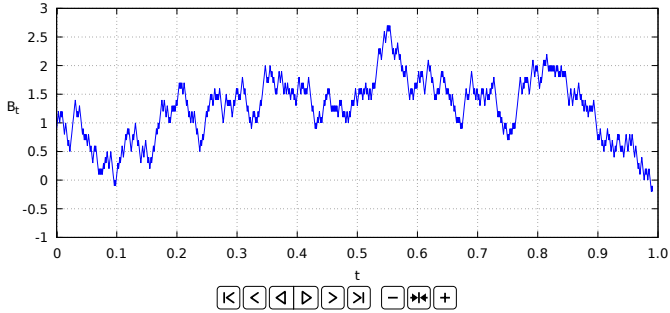


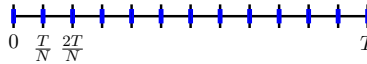
Fig. 1.4: Construction of Brownian motion as a random walk.*

In order to recover the Gaussian distribution property of the random variable B_T, we can split the time interval $[0, T]$ into N subintervals

$$(t_{k-1}, t_k], \qquad k = 1, 2, \ldots, N,$$

* The animation works in Acrobat Reader on the entire pdf file.

of same length $\Delta t = T/N$, where $t_k = kT/N$, $k = 0, 1, \ldots N$, and N is "large".



Letting

$$X_k := \pm\sqrt{T} = \pm\sqrt{N\Delta t}, \qquad k = 1, 2, \ldots N,$$

denote a sequence of symmetric Bernoulli random variables taking values in $\{-\sqrt{T}, \sqrt{T}\}$, with equal probabilities $(1/2, 1/2)$ and

$$\Delta B_{t_k} := \pm\sqrt{\Delta t} = \pm\sqrt{\frac{T}{N}} = \frac{X_k}{\sqrt{N}}, \qquad k = 1, 2, \ldots N,$$

we can write

$$B_T \simeq \sum_{k=1}^{N} \Delta B_{t_k} = \frac{X_1 + X_2 + \cdots + X_N}{\sqrt{N}},$$

hence by the central limit theorem we recover the fact that B_T has the centered Gaussian distribution $\mathcal{N}(0, T)$ with variance T, cf. point 4 of the above Definition 1.3 of Brownian motion, and the illustration given in Figure 1.5.

Remark 1.4.

i) The choice of the square root in (1.2) is in fact not fortuitous. Indeed, any choice of $\pm(\Delta t)^\alpha$ with a power $\alpha > 1/2$ would lead to explosion of the process as dt tends to zero, whereas a power $\alpha \in (0, 1/2)$ would lead to a vanishing process, as can be checked from the R code [*] *below.*

ii) According to this representation, the paths of Brownian motion are not differentiable, although they are continuous by Property 2, as we have

$$\frac{\Delta B_t}{\Delta t} \simeq \frac{\pm\sqrt{\Delta t}}{\Delta t} = \pm\frac{1}{\sqrt{\Delta t}} \simeq \pm\infty, \tag{1.3}$$

see e.g. Theorem 10.3 page 153 of Schilling and Partzsch (2014).

[*] Download the corresponding **IPython notebook** that can be run **here** or **here**.

This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html

```
nsim=100; N=1000; t <- 0:N; dt <- 1.0/N; dev.new(width=16,height=7); # Using Bernoulli
      samples
X <- matrix((dt)^0.5*(rbinom( nsim * N, 1, 0.5)-0.5)*2, nsim, N)
X <- cbind(rep(0, nsim), t(apply(X, 1, cumsum))); H<-hist(X[,N],plot=FALSE);
layout(matrix(c(1,2), nrow =1, byrow = TRUE));par(mar=c(2,2,2,0), oma = c(2, 2, 2, 2))
plot(t*dt, X[1, ], xlab = "", ylab = "", type = "l", ylim = c(-2, 2), col = 0,xaxs='i',las=1,
      cex.axis=1.6)
for (i in 1:nsim){lines(t*dt, X[i, ], type = "l", ylim = c(-2, 2), col = i)}
lines(t*dt,sqrt(t*dt),lty=1,col="red",lwd=3);lines(t*dt,-sqrt(t*dt), lty=1, col="red",lwd=3)
lines(t*dt,0*t, lty=1, col="black",lwd=2)
for (i in 1:nsim){points(0.999, X[i,N], pch=1, lwd = 5, col = i)}
x <- seq(-2,2, length=100); px <- dnorm(x);par(mar = c(2,2,2,2))
plot(NULL , xlab="", ylab="", xlim = c(0, max(px,H$density)), ylim = c(-2,2),axes=F)
rect(0, H$breaks[1:(length(H$breaks) - 1)], col=rainbow(20,start=0.08,end=0.6), H$density,
      H$breaks[2:length(H$breaks)]); lines(px,x, lty=1, col="black",lwd=2)
```

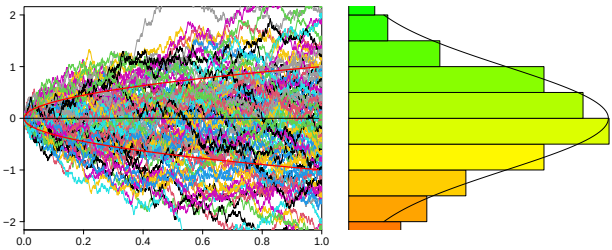


Fig. 1.5: Statistics of one-dimensional Brownian paths *vs.* Gaussian distribution.

The following R code* plots the 72 yearly return graphs of the S&P 500 index from 1950 to 2022, see Figure 1.6. The histogram of year-end returns is then fitted to a normalized Gaussian probability density function.

* Download the corresponding **IPython notebook** that can be run **here** or **here**.

```
library(quantmod); getSymbols("^GSPC",from="1950-01-01",to="2022-12-31",src="yahoo")
stock<-Cl(`GSPC`); s=0;y=0;j=0;count=0;N=240;nsim=72; X = matrix(0, nsim, N)
for (i in 1:nrow(GSPC)){if (s==0 && grepl('-01-0',index(stock[i]))) {if (count==0 || X[y,N]>0)
      {y=y+1;j=1;s=1;count=count+1;}}
if (j<=N) {X[y,j]=as.numeric(stock[i]);};if (grepl('-02-0',index(stock[i]))) {s=0;};j=j+1;}
t <- 0:(N-1); dt <- 1.0/N; dev.new(width=16,height=7);
layout(matrix(c(1,2), nrow =1, byrow = TRUE));par(mar=c(2,2,2,0), oma = c(2, 2, 2, 2))
m=mean(X[,N]/X[,1]-1);sigma=sd(X[,N]/X[,1]-1)
plot(t*dt, X[1,]/X[1,1]-1-m*t*dt, xlab = "", ylab = "", type = "l", ylim = c(-0.5, 0.5), col = 0,
      xaxs='i',las=1, cex.axis=1.6)
for (i in 1:nsim){lines(t*dt, X[i,]/X[i,1]-1-m*t*dt, type = "l", col = i)}
lines(t*dt,sigma*sqrt(t*dt),lty=1,col="red",lwd=3);lines(t*dt,-sigma*sqrt(t*dt), lty=1,
      col="red",lwd=3)
lines(t*dt,0*t, lty=1, col="black",lwd=2)
for (i in 1:nsim){points(0.999, X[i,N]/X[i,1]-1-m*N*dt, pch=1, lwd = 5, col = i)}
x <- seq(-0.5,0.5, length=100); px <- dnorm(x,0,sigma);par(mar = c(2,2,2,2))
H<-hist(X[,N]/X[,1]-1-m*N*dt,plot=FALSE);
plot(NULL , xlab="", ylab="", xlim = c(0, max(px,H$density)), ylim = c(-0.5,0.5),axes=F)
rect(0, H$breaks[1:(length(H$breaks) - 1)], col=rainbow(20,start=0.08,end=0.6), H$density,
H$breaks[2:length(H$breaks)]); lines(px,x, lty=1, col="black",lwd=2)
```

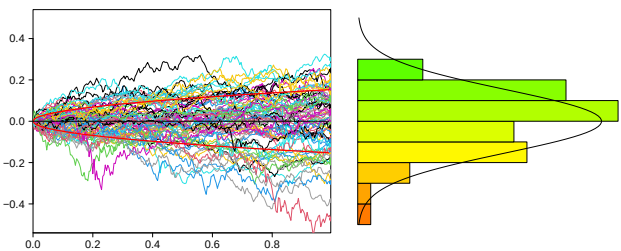


Fig. 1.6: Statistics of 72 S&P 500 yearly return graphs from 1950 to 2022.

1.2 Geometric Brownian Motion

The market returns of an asset price process $(S_t)_{t\in\mathbb{R}_+}$ over time can be estimated in various ways.

Definition 1.5. *1. Standard returns are defined as*

$$\frac{\Delta S_t}{S_t} := \frac{S_{t+\Delta t} - S_t}{S_t}. \tag{1.4}$$

2. Log-returns are defined as

$$\Delta \log S_t = \log S_{t+\Delta t} - \log S_t = \log \frac{S_{t+\Delta t}}{S_t} := \log\left(1 + \frac{\Delta S_t}{S_t}\right), \quad t \geqslant 0. \tag{1.5}$$

This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html

Estimating market returns with R

The R package quantmod can be used to fetch financial data from various sources such as Yahoo! Finance or the Federal Reserve Bank of St. Louis (FRED). It can be installed and run via the following command.

```
install.packages("quantmod")
library(quantmod)
getSymbols("DEXJPUS",src="FRED") # Japan/U.S. Foreign Exchange Rate
getSymbols("CPIAUCNS",src='FRED') # Consumer Price Index
getSymbols("GOOG",src="yahoo") # Google Stock Price
```

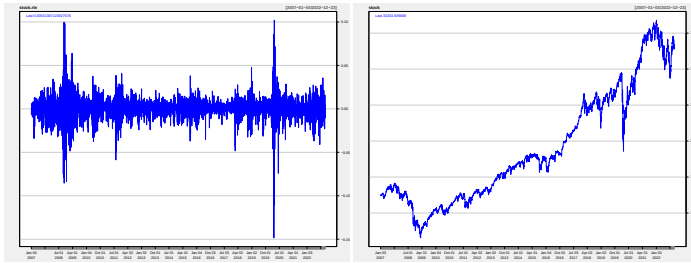
The package yfinance can be similarly used in Python.

```
import pandas as pd
import yfinance as yf
# Google Stock Price
GOOG = yf.download("GOOG", start="2022-01-01", end="2023-12-31")
```

The following R script allows us to fetch market price data using the quantmod package. Market returns can be estimated using either the standard returns $\Delta S_t/S_t$ or the log-returns $\Delta \log S_t$, $t \geqslant 0$, which can be computed by the command diff(log(stock)), here with $dt = 1/365$.

```
getSymbols("^DJI",from="2007-01-03",to=Sys.Date(),src="yahoo")
stock=Ad(`DJI`)
chartSeries(stock,up.col="blue",theme="white")
stock.logrtn=diff(log(stock)); # log returns
stock.rtn=(stock-lag(stock))/lag(stock); # standard returns
chartSeries(stock.rtn,up.col="blue",theme="white")
n = length(!is.na(stock.rtn))
```

The adjusted close price Ad() is the closing price computed after adjustments for applicable splits and dividend distributions.



(a) Stock returns. (b) Cumulative stock returns.

Fig. 1.7: Returns *vs.* cumulative returns.

The package yfinance can be similarly used in Python.

```
import numpy as np; import pandas as pd; import yfinance as yf
import matplotlib.pyplot as plt # Get stock data
stock_data = yf.download("^DJI", start="2007-01-03", end=pd.to_datetime("today"))
stock = stock_data["Adj Close"] # Extract adjusted close prices
stock_logrtn = stock.apply(lambda x: np.log(x)).diff() # Calculate log returns
stock_rtn = (stock - stock.shift(1)) / stock.shift(1) # Calculate standard returns
plt.subplot(1, 2, 1); stock_rtn.plot(color="blue");plt.title("Returns Chart");
plt.subplot(1, 2, 2); stock.plot(color="blue"); plt.title("Stock Chart"); # Plot stock chart
plt.tight_layout(); plt.show() # Plot returns chart
n = stock_rtn.count() # Calculate the number of non-NaN returns
```

Proposition 1.6. *The sequence* $(S_{t_k})_{k=1,\ldots,N}$ *of market prices can be recovered from the sequence of standard returns*

$$\frac{\Delta S_{t_0}}{S_{t_0}} = \frac{S_{t_1} - S_{t_0}}{S_{t_0}}, \ldots, \frac{\Delta S_{t_{N-1}}}{S_{t_{N-1}}} = \frac{S_{t_N} - S_{t_{N-1}}}{S_{t_{N-1}}}$$

from the telescoping product identity

$$S_{t_n} = S_0 \prod_{k=1}^{n} \frac{S_{t_k}}{S_{t_{k-1}}} = S_0 \prod_{k=1}^{n} \left(1 + \frac{\Delta S_{t_{k-1}}}{S_{t_{k-1}}}\right), \quad 1 \leqslant n \leqslant N. \tag{1.6}$$

Using log-returns, we have similarly

$$S_{t_n} = S_0 \exp\left(\sum_{k=1}^{n} \Delta \log S_{t_k}\right) = S_0 \prod_{k=1}^{n} \mathrm{e}^{\Delta \log S_{t_k}}, \qquad 1 \leqslant n \leqslant N.$$

The next R code recovers cumulative returns from standard market returns, according to (1.6).

This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html

```
stock<-stock[!is.na(stock.rtn)];stock.rtn<-stock.rtn[!is.na(stock.rtn)]
times=index(stock);dev.new(width=16,height=7);par(mfrow=c(1,2))
plot(times,stock.rtn,pch=19,xaxs="i",yaxs="i",cex=0.03,col="blue", ylab="", xlab="", main =
      'Asset returns', las=1, cex.lab=1.8, cex.axis=1.8, lwd=3)
segments(x0 = times, x1 = times, y0 = 0, y1 = stock.rtn,col="blue")
plot(times,100 * cumprod(1 + as.numeric(stock.rtn)),type='l',col='black',main = "Asset
      prices",ylab="", cex=0.1,cex.axis=1,las=1)
```

The package yfinance can be similarly used in Python.

```
# Remove NaN values from stock and stock_rtn
stock = stock[~np.isnan(stock_rtn)]; stock_rtn = stock_rtn[~np.isnan(stock_rtn)]
times = stock.index; plt.subplot(1, 2, 1); plt.plot(times, stock_rtn, color='blue')
plt.xlabel('Time'); plt.ylabel('Asset Returns')
plt.title('Asset Returns'); plt.xticks(rotation=90)
plt.grid(True) # Plot asset returns
plt.subplot(1, 2, 2); cumulative_returns = (1 + stock_rtn.astype(float)).cumprod() * 100
plt.plot(times, cumulative_returns, color='black'); plt.xlabel('Time')
plt.ylabel('Asset Prices'); plt.title('Asset Prices'); plt.xticks(rotation=90); plt.grid(True)
plt.tight_layout(); plt.show(); # Plot asset prices
```

Geometric Brownian Motion

Samuelson (1965) rediscovered Bachelier's ideas and proposed geometric Brownian motion as a model for stock prices. In an interview he stated "In the early 1950s I was able to locate by chance this unknown Bachelier (1900) book, rotting in the library of the University of Paris, and when I opened it up it was as if a whole new world was laid out before me." We refer to "Rational theory of warrant pricing" by Paul Samuelson, *Industrial Management Review*, p. 13-32, 1965.

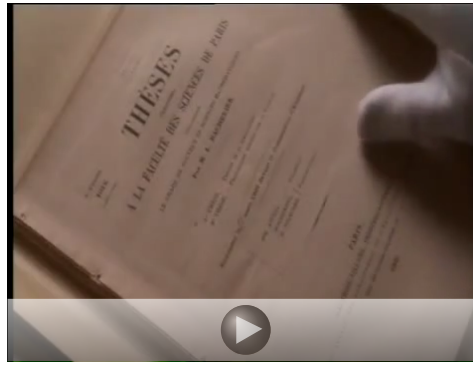


Fig. 1.8: Clark (2000) "As if a whole new world was laid out before me."*

The evolution of a riskless bank account value $(A_t)_{t\in\mathbb{R}_+}$ is constructed from standard returns, defined as follows:

$$\frac{A_{t+\Delta t}-A_t}{A_t}=r\Delta t, \quad i.e \quad \frac{\Delta A_t}{A_t}=r\Delta t.$$

This equation can be regarded as a discretization of the differential equation

$$A'_t=\frac{dA_t}{dt}=rA_t, \quad t\geqslant 0,$$

which has the solution

$$A_t=A_0\mathrm{e}^{rt}, \qquad t\geqslant 0, \tag{1.7}$$

where $r>0$ is the risk-free interest rate.†

In what follows, we will model the risky asset price process $(S_t)_{t\in\mathbb{R}_+}$ using standard returns, from the equation

$$\frac{S_{t+\Delta t}-S_t}{S_t}=\frac{\Delta S_t}{S_t}=\mu\Delta t+\sigma\Delta B_t, \qquad t\geqslant 0, \tag{1.8}$$

* Click on the figure to play the video (works in Acrobat Reader on the entire pdf file).

† "Anyone who believes exponential growth can go on forever in a finite world is either a madman or an economist", K. E. Boulding (1973), page 248.

which can be solved numerically according to the following R and Python codes.

```
N=2000; t <- 0:N; dt <- 1.0/N;mu=0.5; sigma=0.2; nsim <- 10; S <- matrix(0, nsim, N+1)
Z <- matrix(rnorm(nsim*N,mean=0,sd=sqrt(dt)), nsim, N+1)
for (i in 1:nsim){S[i,1]=1.0;
for (j in 1:N+1){S[i,j]=S[i,j-1]*(1 + mu*dt+sigma*Z[i,j])}}
plot(t*dt, rep(0, N+1), xlab = "Time", ylab = "Geometric Brownian motion", lwd=2, ylim =
      c(min(S),max(S)), type = "l", col = 0,las=1, cex.axis=1.5,cex.lab=1.5, xaxs='i', yaxs='i')
for (i in 1:nsim){lines(t*dt, S[i, ], lwd=2, type = "l", col = i)}
```

```
import numpy as np; import matplotlib.pyplot as plt
%matplotlib
N = 2000; t = np.arange(N+1); dt = 1.0 / N; mu = 0.5; sigma = 0.2; nsim = 10
X = np.zeros((nsim, N+1)); Z = np.random.normal(0, np.sqrt(dt), (nsim, N+1))
for i in range(nsim):
    X[i, 0] = 1.0
    for j in range(1, N+1):
        X[i, j] = X[i, j-1] + mu * X[i, j-1] * dt + sigma * X[i, j-1] * Z[i, j]
plt.plot(t*dt, np.zeros(N+1), color='black', linewidth=2)
plt.xlabel('Time'); plt.ylabel('Geometric Brownian motion'); plt.xlim(0, N*dt)
plt.ylim(np.min(X), np.max(X)); plt.xticks(fontsize=12); plt.yticks(fontsize=12)
for i in range(nsim):
    plt.plot(t*dt, X[i, :], linewidth=2)
plt.show()
```

The following proposition gives the explicit solution to (1.8) regarded as a discretization of the stochastic differential Equation (1.9) below.

Proposition 1.7 (Geometric Brownian motion)**.** *The solution of the stochastic differential equation*

$$dS_t = \mu S_t dt + \sigma S_t dB_t \tag{1.9}$$

is given by

$$S_t = S_0 \exp\left(\sigma B_t + \left(\mu - \frac{1}{2}\sigma^2\right) t\right), \qquad t \geqslant 0. \tag{1.10}$$

Proof. Using (1.8), the log-returns (1.5) of an asset priced $(S_t)_{t\in\mathbb{R}_+}$ satisfy

$$\frac{dS_t}{S_t} = \frac{S_{t+dt} - S_t}{S_t} = \mu dt + \sigma dB_t.$$

Hence, using the second order approximation $\log(1+x) \simeq x - x^2/2$ as x tends to zero, we have

$$\begin{aligned} d\log S_t &\simeq \log S_{t+dt} - \log S_t \\ &= \log \frac{S_{t+dt}}{S_t} \end{aligned}$$

$$
\begin{aligned}
&= \log\left(1+\frac{S_{t+dt}-S_t}{S_t}\right)\\
&= \log\left(1+\frac{dS_t}{S_t}\right)\\
&= \log(1+\mu dt+\sigma dB_t)\\
&= \mu dt+\sigma dB_t-\frac{1}{2}(\mu dt+\sigma dB_t)^2\\
&\simeq \mu dt+\sigma dB_t-\frac{\mu^2}{2}(dt)^2-\mu\sigma dB_t\bullet dt-\frac{\sigma^2}{2}(dB_t)^2\\
&\simeq \mu dt-\frac{\sigma^2}{2}dt+\sigma dB_t, \qquad t\geqslant 0.
\end{aligned}
$$

By integration over the time interval $[0,t]$ we find

$$
\begin{aligned}
\log S_t &= \log S_0+\int_0^t d\log S_s\\
&= \int_0^t \mu ds-\frac{\sigma^2}{2}\int_0^t ds+\sigma\int_0^t dB_s\\
&= \mu t-\frac{\sigma^2 t}{2}+\sigma B_t,
\end{aligned}
$$

which yields (1.10) after exponentiation. □

The next Figure 1.9 presents an illustration of the geometric Brownian process of Proposition 1.7 according to the following R and Python codes.

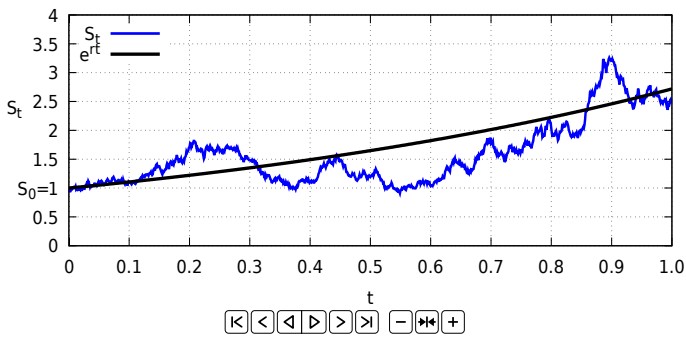


Fig. 1.9: Geometric Brownian motion started at $S_0=1$, with $\mu=r=1$ and $\sigma^2=0.5$.*

* The animation works in Acrobat Reader on the entire pdf file.

This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html

```r
N=1000; t <- 0:N; dt <- 1.0/N; sigma=0.2; mu=0.5
Z <- rnorm(N,mean=0,sd=sqrt(dt));
plot(t*dt, exp(mu*t*dt), xlab = "time", ylab = "Geometric Brownian motion", type = "l", ylim
    = c(0.75, 2), col = 1,lwd=3)
lines(t*dt, exp(sigma*c(0,cumsum(Z))+mu*t*dt-sigma*sigma*t*dt/2),xlab = "time",type =
    "l",ylim = c(0, 4), col = 'blue', xaxs='i', yaxs='i')
```

```python
import numpy as np; import matplotlib.pyplot as plt
%matplotlib
N = 1000; t = np.arange(0, N+1); dt = 1.0 / N; sigma = 0.2; mu = 0.5
Z = np.random.normal(0, np.sqrt(dt), N)
GBM = np.exp(sigma * np.concatenate(([0], np.cumsum(Z))) + mu * t * dt - (sigma**2) * t *
    dt / 2)
plt.plot(t*dt, np.exp(mu * t * dt), color='blue', linewidth=3, label='Expected GBM')
plt.plot(t*dt, GBM, color='red', linewidth=1.5, label='Simulated GBM')
plt.xlabel('Time'); plt.ylabel('Geometric Brownian motion')
plt.xlim(0, N*dt); plt.ylim(0.75, 2); plt.legend(); plt.grid(True); plt.show()
```

Multiple sample paths of the solution (1.10) to (1.9) can also be simulated by the following R and Python codes.

```r
N=2000; t <- 0:N; dt <- 1.0/N; mu=0.5;sigma=0.2; nsim <- 10; par(oma=c(0,1,0,0))
X <- matrix(rnorm(nsim*N,mean=0,sd=sqrt(dt)), nsim, N)
X <- cbind(rep(0, nsim), t(apply(X, 1, cumsum)))
for (i in 1:nsim){X[i,] <- exp(mu*t*dt+sigma*X[i,]-sigma*sigma*t*dt/2)}
plot(t*dt, rep(0, N+1), xlab = "Time", ylab = "Geometric Brownian motion", lwd=2, ylim =
    c(min(X),max(X)), type = "l", col = 0,las=1,cex.axis=1.5,cex.lab=1.6, xaxs='i', yaxs='i')
for (i in 1:nsim){lines(t*dt, X[i, ], lwd=2, type = "l", col = i)}
```

```python
import numpy as np; import matplotlib.pyplot as plt
%matplotlib
N = 2000; t = np.arange(0, N+1); dt = 1.0 / N; mu = 0.5; sigma = 0.2; nsim = 10
X = np.random.normal(0, np.sqrt(dt), (nsim, N))
X = np.concatenate((np.zeros((nsim, 1)), np.cumsum(X, axis=1)), axis=1)
for i in range(nsim): X[i, :] = np.exp(mu * t * dt + sigma * X[i, :] - (sigma**2) * t * dt / 2)
plt.plot(t*dt, np.zeros(N+1), color='black', linewidth=1)
plt.xlabel('Time'); plt.ylabel('Geometric Brownian motion'); plt.xlim(0, N*dt)
plt.ylim(np.min(X), np.max(X)); plt.xticks(fontsize=12); plt.yticks(fontsize=12)
for i in range(nsim): plt.plot(t*dt, X[i, :], linewidth=1)
plt.show()
```

Figure 1.10 presents sample paths of geometric Brownian motion.

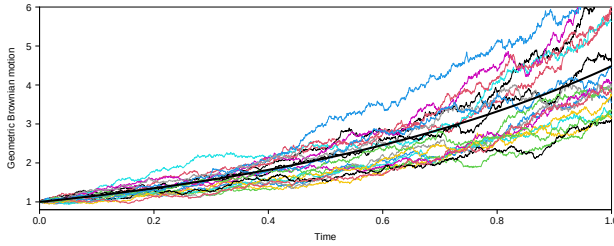


Fig. 1.10: Ten sample paths of geometric Brownian motion $(S_t)_{t\in\mathbb{R}_+}$.

The next R and Python codes compare geometric Brownian motion simulations to asset price data.

```
library(quantmod); getSymbols("0005.HK",from="2016-02-15",to="2017-05-11",src="yahoo")
Marketprices<-Ad(`0005.HK`);
returns = (Marketprices-lag(Marketprices)) / Marketprices
sigma=sd(as.numeric(returns[-1])); r=mean(as.numeric(returns[-1]))
N=length(Marketprices); t <- 0:N; a=(1+r)*(1-sigma)-1;b=(1+r)*(1+sigma)-1
X <- matrix((a+b)/2+(b-a)*rnorm( N-1, 0, 1)/2, 1, N-1)
X <- as.numeric(Marketprices[1])*cbind(0,t(apply((1+X),1,cumprod))); X[,1]=Marketprices[1];
x=seq(100,100+N-1); dates <- index(Marketprices)
GBM<-xts(x =X[1,], order.by = dates); myPars <- chart_pars();myPars$cex<-1.4
myTheme <- chart_theme();myTheme$col$line.col <- "blue"; myTheme$rylab <- FALSE;
chart_Series(Marketprices,pars=myPars, theme = myTheme);
dexp<-as.numeric(Marketprices[1])*exp(r*seq(1,305)); ddexp<-xts(x =dexp, order.by = dates)
dev.new(width=16,height=8); par(mfrow=c(1,2));
add_TA(exp(log(ddexp)), on=1, col="black",layout=NULL, lwd=4 ,legend=NULL)
graph <- chart_Series(GBM,theme=myTheme,pars=myPars); myylim <- graph$get_ylim()
graph <- add_TA(exp(log(ddexp)), on=1, col="black",layout=NULL, lwd=4 ,legend=NULL)
myylim[[2]] <- structure(c(min(Marketprices),max(Marketprices)), fixed=TRUE)
graph$set_ylim(myylim); graph
```


This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html

```
import numpy as np; import pandas as pd; import yfinance as yf
from matplotlib import pyplot as plt
data = yf.download("0005.HK", start="2016-02-15", end="2017-05-11", progress=False)
Marketprices = data['Adj Close'];
returns = (Marketprices - Marketprices.shift(1)) / Marketprices
sigma = np.std(returns[1:]); r = np.mean(returns[1:]); N = len(Marketprices);
t = np.arange(0, N); a = (1 + r) * (1 - sigma) - 1; b = (1 + r) * (1 + sigma) - 1
X = np.array([(a + b) / 2 + (b - a) * np.random.normal(0, 1, N-1) / 2])
X = np.concatenate(([Marketprices[0]], Marketprices[0]*np.cumprod(1 + X)))
x = np.arange(100, 100 + N); dates = Marketprices.index; GBM = pd.Series(X, index=dates)
Y = Marketprices[0]*pd.Series(np.exp(r * np.arange(1, N+1)), index=dates)
fig = plt.figure(figsize=(16, 8)); ax1 = fig.add_subplot(121)
ax1.plot(Marketprices, color='blue', linewidth=2)
ax1.set_ylabel('Price'); ax1.set_xlabel('Date'); ylim1 = ax1.get_ylim()
ax2 = fig.add_subplot(122); ax2.plot(GBM, color='blue', linewidth=2)
ax2.plot(Y, color='black', linewidth=2)
ax2.set_ylabel('GBM'); ax2.set_xlabel('Date'); ax2.set_ylim(ylim1)
plt.tight_layout(); plt.show()
```

Figure 1.11 presents a graph of underlying asset price market data, which is compared to the geometric Brownian motion simulation of Figure 1.10 in Figure 1.12.

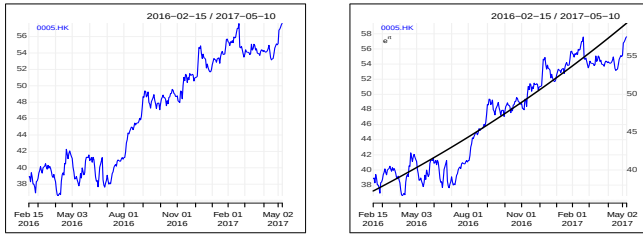


Fig. 1.11: Graph of underlying market prices.

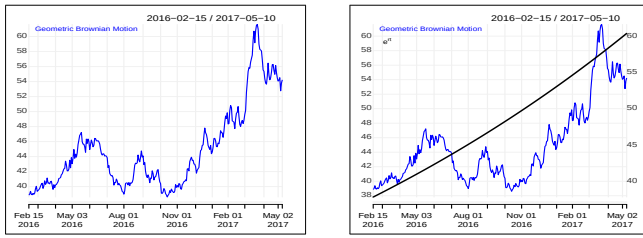


Fig. 1.12: Graph of simulated geometric Brownian motion.

This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html

The R package Sim.DiffProc can be used to estimate the coefficients of a geometric Brownian motion fitting observed market data.

```
library("Sim.DiffProc")
fx <- expression( theta[1]*x ); gx <- expression( theta[2]*x )
fitsde(data = as.ts(Marketprices), drift = fx, diffusion = gx, start = list(theta1=0.01,
       theta2=0.01),pmle="euler")
```

In the next proposition, we compute the probability distribution of geometric Brownian motion at any given time.

Proposition 1.8. *At any time $T > 0$, the random variable*

$$S_T := S_0 \mathrm{e}^{\sigma B_T + (\mu - \sigma^2/2)T}$$

has the lognormal distribution *with probability density function*

$$x \longmapsto f(x) = \frac{1}{x\sigma\sqrt{2\pi T}} \mathrm{e}^{-(-(\mu-\sigma^2/2)T + \log(x/S_0))^2/(2\sigma^2 T)}, \quad x > 0, \tag{1.11}$$

with log-variance σ^2 and log-mean $(\mu - \sigma^2/2)T + \log S_0$, see Figure 1.14.

Proof. For all $x \in \mathbb{R}$, we have

$$\begin{aligned}
\mathbb{P}(S_T \leqslant x) &= \mathbb{P}\big(S_0 \mathrm{e}^{\sigma B_T + (\mu - \sigma^2/2)T} \leqslant x\big) \\
&= \mathbb{P}\left(\sigma B_T + \left(\mu - \frac{\sigma^2}{2}\right) T \leqslant \log \frac{x}{S_0}\right) \\
&= \mathbb{P}\left(B_T \leqslant \frac{1}{\sigma}\left(\log \frac{x}{S_0} - \left(\mu - \frac{\sigma^2}{2}\right) T\right)\right) \\
&= \int_{-\infty}^{(\log(x/S_0) - (\mu - \sigma^2/2)T)/\sigma} \mathrm{e}^{-y^2/(2T)} \frac{dy}{\sqrt{2\pi T}} \\
&= \int_{-\infty}^{(\log(x/S_0) - (\mu - \sigma^2/2)T)/(\sigma\sqrt{T})} \mathrm{e}^{-z^2/2} \frac{dz}{\sqrt{2\pi}} \\
&= \Phi\left(\frac{1}{\sigma\sqrt{T}}\left(\log \frac{x}{S_0} - \left(\mu - \frac{\sigma^2}{2}\right) T\right)\right),
\end{aligned}$$

where

$$\Phi(x) := \int_{-\infty}^{x} \mathrm{e}^{-y^2/2} \frac{dy}{\sqrt{2\pi T}}, \qquad x \in \mathbb{R}, \tag{1.12}$$

denotes the standard Gaussian Cumulative Distribution Function (CDF) of a standard normal random variable $X \simeq \mathcal{N}(0,1)$, with the relation

This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html

$$\Phi(-x) = 1 - \Phi(x), \qquad x \in \mathbb{R}.$$

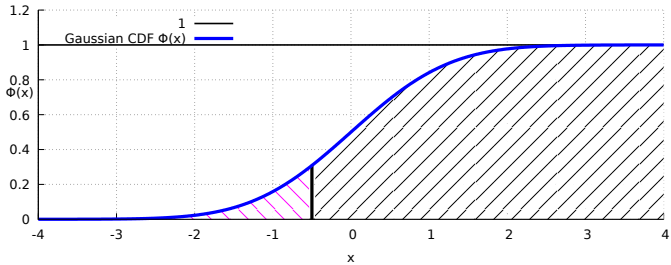


Fig. 1.13: Graph of the Gaussian Cumulative Distribution Function (CDF).

After differentiation with respect to x, we find the lognormal probability density function

$$\begin{aligned} f(x) &= \frac{\mathrm{d}\mathbb{P}(S_T \leqslant x)}{dx} \\ &= \frac{\partial}{\partial x} \int_{-\infty}^{(\log(x/S_0)-(\mu-\sigma^2/2)T)/\sigma} \mathrm{e}^{-y^2/(2T)} \frac{dy}{\sqrt{2\pi T}} \\ &= \frac{\partial}{\partial x} \Phi\left(\frac{1}{\sigma\sqrt{T}} \left(\log \frac{x}{S_0} - \left(\mu - \frac{\sigma^2}{2} \right) T \right) \right) \\ &= \frac{1}{x\sigma\sqrt{T}} \varphi\left(\frac{1}{\sigma\sqrt{T}} \left(\log \frac{x}{S_0} - \left(\mu - \frac{\sigma^2}{2} \right) T \right) \right) \\ &= \frac{1}{x\sigma\sqrt{2\pi T}} \mathrm{e}^{-(-(\mu-\sigma^2/2)T+\log(x/S_0))^2/(2\sigma^2 T)}, \qquad x > 0, \end{aligned}$$

where

$$\varphi(y) = \Phi'(y) := \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-y^2/2}, \qquad y \in \mathbb{R},$$

denotes the standard Gaussian probability density function. □

The next R code is generating sample paths of geometric Brownian motion together with the histogram of terminal values fitted to the lognormal probability density function (1.11), see Figure 1.14.

```
N=1000; t <- 0:N; dt <- 1.0/N; nsim <- 100 # using Bernoulli samples
sigma=0.2;r=0.5;a=(1+r*dt)*(1-sigma*sqrt(dt))-1;b=(1+r*dt)*(1+sigma*sqrt(dt))-1
X <- matrix(a+(b-a)*rbinom( nsim * N, 1, 0.5), nsim, N)
X<-cbind(rep(0,nsim),t(apply((1+X),1,cumprod))); X[,1]=1;H<-hist(X[,N],plot=FALSE);
      dev.new(width=16,height=7);
layout(matrix(c(1,2), nrow =1, byrow = TRUE)); par(mar=c(2,2,2,0), oma = c(2, 2, 2, 2))
plot(t*dt,X[1,],xlab="time",ylab="",type="l",ylim=c(0.8,3), col = 0,xaxs='i',las=1,
      cex.axis=1.6)
for (i in 1:nsim){lines(t*dt, X[i, ], xlab = "time", type = "l", col = i)}
lines((1+r*dt)^t, type="l", lty=1, col="black",lwd=3,xlab="",ylab="", main="")
for (i in 1:nsim){points(0.999, X[i,N], pch=1, lwd = 5, col = i)}; x <- seq(0.01,3, length=100);
px <- exp(-(-(r-sigma^2/2)+log(x))^2/2/sigma^2)/x/sigma/sqrt(2*pi); par(mar = c(2,2,2,2))
plot(NULL , xlab="", ylab="", xlim = c(0, max(px,H$density)),ylim=c(0.8,3),axes=F, las=1)
rect(0, H$breaks[1:(length(H$breaks) - 1)], col=rainbow(20,start=0.08,end=0.6), H$density,
      H$breaks[2:length(H$breaks)])
lines(px,x, type="l", lty=1, col="black",lwd=3,xlab="",ylab="", main="")
```

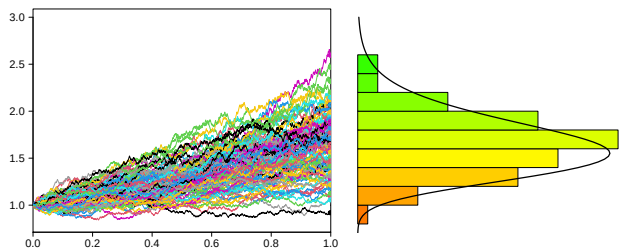


Fig. 1.14: Statistics of geometric Brownian paths *vs.* lognormal distribution.

Time-dependent coefficients

The above construction can be extended to time-dependent interest rate, drift and volatility processes $(r(t))_{t\in\mathbb{R}_+}$, $(\mu(t))_{t\in\mathbb{R}_+}$ and $(\sigma(t))_{t\in\mathbb{R}_+}$. In this case, we let $(A_t)_{t\in\mathbb{R}_+}$ be the risk-free asset with price given by

$$\frac{dA_t}{A_t} = r(t)dt, \qquad A_0 = 1, \qquad t \geqslant 0,$$

i.e.

$$A_t = A_0 \mathrm{e}^{\int_0^t r(s)ds}, \qquad t \geqslant 0,$$

and the price process $(S_t)_{t\in[0,T]}$ is defined by the stochastic differential equation

$$dS_t = \mu(t)S_t dt + \sigma(t)S_t dB_t, \qquad t \geqslant 0,$$

i.e., in integral form,

$$S_t = S_0 + \int_0^t \mu(u) S_u du + \int_0^t \sigma(u) S_u dB_u, \qquad t \geqslant 0,$$

with solution

$$S_t = S_0 \exp\left(\int_0^t \sigma(s) dB_s + \int_0^t \left(\mu(s) - \frac{1}{2}\sigma^2(s) \right) ds \right),$$

$t \in \mathbb{R}_+$.

Moments and cumulants of random variables

In the sequel we will use the sequence $(\kappa_n(X))_{n\geqslant 1}$ of cumulants of a random variable X, defined from its Moment Generating Function (MGF)

$$\mathcal{M}_X(t) := \mathbb{E}\big[\mathrm{e}^{tX}\big] = 1 + \sum_{n\geqslant 1} \frac{t^n}{n!} \mathbb{E}[X^n], \tag{1.13}$$

for t in a neighborhood of 0, see Thiele (1899).

Definition 1.9. *The* cumulants *of a random variable X are the coefficients $(\kappa_n(X))_{n\geqslant 1}$ appearing in the series expansion*

$$\log \mathbb{E}\big[\mathrm{e}^{tX}\big] = \sum_{n\geqslant 1} \kappa_n(X) \frac{t^n}{n!}, \tag{1.14}$$

of the logarithmic moment generating function (log-MGF) of X.

The first cumulants of X can be identified as follows.

a) First moment and cumulant. We have $\kappa_1(X) = \mathbb{E}[X]$.

b) Variance and second cumulant. We have

$$\kappa_2(X) = \mathbb{E}\big[X^2\big] - (\mathbb{E}[X])^2 = \mathbb{E}\big[(X - \mathbb{E}[X])^2\big],$$

and $\sqrt{\kappa_2(X)}$ is the *standard deviation* of X.

c) The third cumulant of X is given as the third central moment

$$\kappa_3(X) = \mathbb{E}[(X - \mathbb{E}[X])^3].$$

d) Similarly, the fourth cumulant of X satisfies

$$\begin{aligned} \kappa_4(X) &= \mathbb{E}[(X - \mathbb{E}[X])^4] - 3(\kappa_2(X))^2 \\ &= \mathbb{E}[(X - \mathbb{E}[X])^4] - 3\big(\mathbb{E}\big[(X - \mathbb{E}[X])^2\big]\big)^2. \end{aligned}$$

Example: Gaussian moments and cumulants

When X is centered we have $\kappa_1^X = 0$ and $\kappa_2^X = \mathbb{E}\big[X^2\big] = \mathrm{Var}\,[X]$, and X becomes Gaussian if and only if $\kappa_n^X = 0$, $n \geqslant 3$, *i.e.*

$$\kappa_n^X = \mathbb{1}_{\{n=2\}}\sigma^2, \qquad n \geqslant 1,$$

or

$$\big(\kappa_1^X, \kappa_2^X, \kappa_3^X, \kappa_4^X, \ldots\big) = (0, \sigma^2, 0, 0, \ldots).$$

Example: Poisson moments and cumulants

In the particular case of a Poisson random variable $Z \simeq \mathcal{P}(\lambda)$ with intensity $\lambda > 0$, we have

$$\begin{aligned}\mathbb{E}_\lambda\big[\mathrm{e}^{tZ}\big] &= \sum_{n\geqslant 0} \mathrm{e}^{nt}\mathbb{P}(Z = n)\\ &= \mathrm{e}^{-\lambda}\sum_{n\geqslant 0}\frac{(\lambda \mathrm{e}^t)^n}{n!}\\ &= \mathrm{e}^{\lambda(\mathrm{e}^t-1)}, \qquad t \in \mathbb{R}_+, \end{aligned} \tag{1.15}$$

hence $\kappa_n^Z = \lambda$, $n \geqslant 1$, or

$$\big(\kappa_1^Z, \kappa_2^Z, \kappa_3^Z, \kappa_4^Z, \ldots\big) = (\lambda, \lambda, \lambda, \lambda, \ldots),$$

Definition 1.10. *i) The* skewness *of X is defined as*

$$\mathrm{Sk}_X := \frac{\kappa_3(X)}{(\kappa_2(X))^{3/2}} = \frac{\mathbb{E}\big[(X - \mathbb{E}[X])^3\big]}{(\mathbb{E}[(X - \mathbb{E}[X])^2])^{3/2}}.$$

ii) The excess kurtosis *of X is defined as*

$$\mathrm{EK}_X := \frac{\kappa_4(X)}{(\kappa_2(X))^2} = \frac{\mathbb{E}[(X - \mathbb{E}[X])^4]}{(\mathbb{E}[(X - \mathbb{E}[X])^2])^2} - 3.$$

The cumulants of X were originally called "semi-invariants" due to the property $\kappa_n(X+Y) = \kappa_n(X) + \kappa_n(Y)$, $n \geqslant 1$, when X and Y are independent random variables. Indeed, in this case we have

$$\begin{aligned}\sum_{n\geqslant 1}\kappa_n(X+Y)\frac{t^n}{n!} &= \log\big(\mathbb{E}\big[\mathrm{e}^{t(X+Y)}\big]\big)\\ &= \log\big(\mathbb{E}\big[\mathrm{e}^{tX}\big]\mathbb{E}\big[\mathrm{e}^{tY}\big]\big)\end{aligned}$$

This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html

$$
\begin{aligned}
&= \log \mathbb{E}\left[\mathrm{e}^{tX}\right] + \log \mathbb{E}\left[\mathrm{e}^{tY}\right] \\
&= \sum_{n\geqslant 1} \kappa_n(X)\frac{t^n}{n!} + \sum_{n\geqslant 1} \kappa_n(Y)\frac{t^n}{n!} \\
&= \sum_{n\geqslant 1} \big(\kappa_n(X)+\kappa_n(Y)\big)\frac{t^n}{n!},
\end{aligned}
$$

showing that $\kappa_n(X+Y) = \kappa_n(X) + \kappa_n(Y)$, $n \geqslant 1$.

1.3 Distribution of Market Returns

Market returns *vs.* Gaussian and power tails

Consider for example the market returns data obtained from fetching DJI and STI index data using the R package quantmod and the following scripts.

```
library(quantmod)
getSymbols("^STI",from="1990-01-03",to="2015-01-03",src="yahoo");stock=Ad(`STI`);
getSymbols("^DJI",from="1990-01-03",to=Sys.Date(),src="yahoo");stock=Ad(`DJI`);
stock.rtn=diff(log(stock));returns <- as.vector(stock.rtn)
m=mean(returns,na.rm=TRUE);s=sd(returns,na.rm=TRUE);times=index(stock.rtn)
n = sum(is.na(returns))+sum(!is.na(returns));x=seq(1,n);y=rnorm(n,mean=m,sd=s)
dev.new(width=16,height=8)
plot(times,returns,pch=19,xaxs="i",yaxs="i",cex=0.03,col="blue", ylab="", xlab="", main = '',
      las=1, cex.lab=1.8, cex.axis=1.8, lwd=3)
segments(x0 = times, x1 = times, y0 = 0, y1 = returns,col="blue")
points(times,y,pch=19,cex=0.3,col="red")
abline(h = m+3*s, col="black", lwd =1);abline(h = m, col="black", lwd =1);abline(h =
      m+-3*s, col="black", lwd =1)
length(returns[abs(returns-m)>3*s])/length(stock.rtn)
length(y[abs(y-m)>3*s])/length(y);2*(1-pnorm(3*s,0,s))
```

Figure 1.15 shows the mismatch between the distributional properties of market log-returns *vs.* standardized Gaussian returns, which tend to underestimate the probabilities of extreme events. Note that when $X \simeq \mathcal{N}(0,\sigma^2)$, 99.73% of samples of X are falling within the interval $[-3\sigma,+3\sigma]$, *i.e.* $\mathbb{P}(|X| \leqslant 3\sigma) = 0.9973002$.

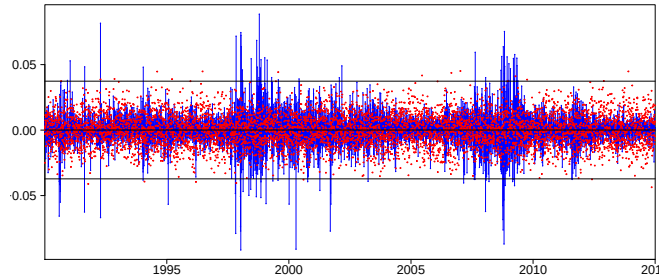


Fig. 1.15: Market returns (blue) *vs.* normalized Gaussian returns (red).

```
stock.ecdf=ecdf(as.vector(stock.rtn));x <- seq(-0.15, 0.15, length=200);px <- pnorm((x-m)/s)
dev.new(width=16,height=8)
plot(stock.ecdf, xlab = '', col="red",ylab = '', ylim=c(-0.002,1.002), main = '', xaxs="i",
     yaxs="i", las=1, cex.lab=1.8, cex.axis=1.8, lwd=4)
lines(x, px, type="l", lty=2, col="blue",xlab="",ylab="", main="", lwd=4)
legend("topleft", legend=c("Empirical CDF", "Gaussian CDF"),col=c("red", "blue"), lty=1:2,
     cex=2, lwd = 4);grid(lwd = 3)
```

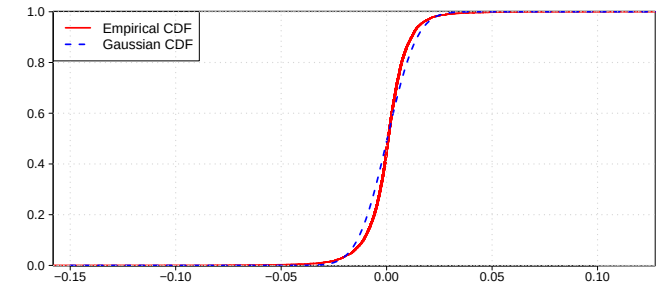


Fig. 1.16: Empirical *vs.* Gaussian CDF.

The following Quantile-Quantile plot is plotting the normalized empirical quantiles against the standard Gaussian quantiles, and is obtained with the `qqnorm(returns)` command.

This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html

```
dev.new(width=16,height=8)
qqnorm(returns, col = "blue", xaxs="i", yaxs="i", las=1, cex.lab=1.4, cex.axis=1, lwd=3)
grid(lwd = 2)
```

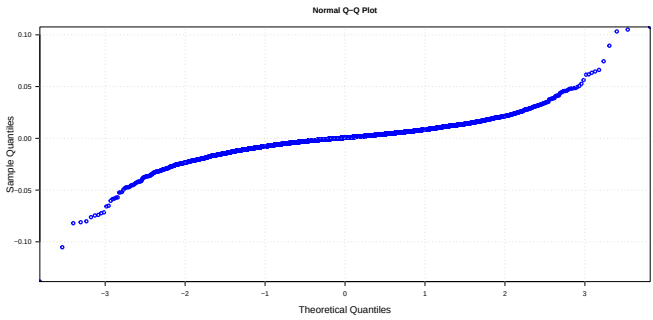


Fig. 1.17: Quantile-Quantile plot.

The following Kolmogorov-Smirnov test clearly rejects the null (normality) hypothesis of market returns.

```
n = sum(is.na(returns))+sum(!is.na(returns));x=seq(1,n);y=rnorm(n,mean=m,sd=s)
ks.test(y,"pnorm",mean=m,sd=s)
ks.test(returns,"pnorm",mean=m,sd=s)
```

```
One-sample Kolmogorov-Smirnov test

data:  returns
D = 0.075577, p-value < 2.2e-16
alternative hypothesis: two-sided
```

The mismatch in distributions observed in Figures 1.15-1.17 can be further illustrated by the empirical probability density plot in Figure 1.18, which is obtained from the following R code.

```
dev.new(width=16,height=8)
x <- seq(-0.25, 0.25, length=100);qx <- dnorm(x,mean=m,sd=s)
stock.dens=density(stock.rtn,na.rm=TRUE)
plot(stock.dens, xlab = 'x', lwd=4, col="red",ylab = '', main = '', xlim =c(-0.1,0.1),
     ylim=c(0,65), xaxs="i", yaxs="i", las=1, cex.lab=1.8, cex.axis=1.8)
lines(x, qx, type="l", lty=2, lwd=4, col="blue",xlab="x value",ylab="Density", main="")
legend("topleft", legend=c("Empirical density", "Gaussian density"),col=c("red", "blue"),
     lty=1:2, cex=1.5);grid(lwd = 2)
```

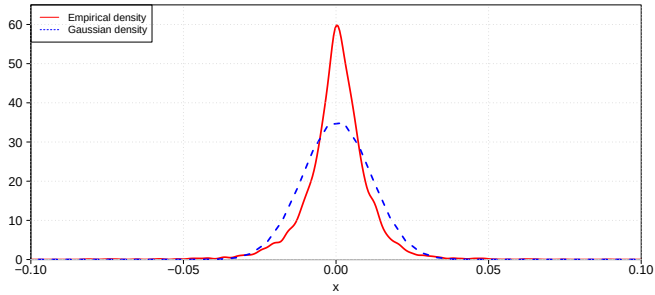


Fig. 1.18: Empirical density *vs.* normalized Gaussian density.

Form the above figure, we note that market returns have kurtosis higher than that of the normal distribution, i.e. their distribution is *leptokurtic*, in addition to showing some negative skewness.

The next R code and graph present a comparison of market prices to a calibrated lognormal distribution.

```
x <- seq(0, max(stock), length=100);qx <- dlnorm(x,mean=mean(log(stock)), sd=sd(log(stock)))
stock.dens=density(stock,na.rm=TRUE);dev.new(width=10, height=5)
plot(stock.dens, xlab = 'x', lwd=3, col="red",ylab = '', main = '', panel.first = abline(h = 0,
     col='grey', lwd =0.2), las=1, cex.axis=1, cex.lab=1, xaxs='i', yaxs='i')
lines(x, qx, type="l", lty=2, lwd=3, col="blue",xlab="x value",ylab="Density", main="")
legend("topright", legend=c("Empirical density", "Lognormal density"),col=c("red", "blue"),
     lty=1:2, cex=1.2)
```

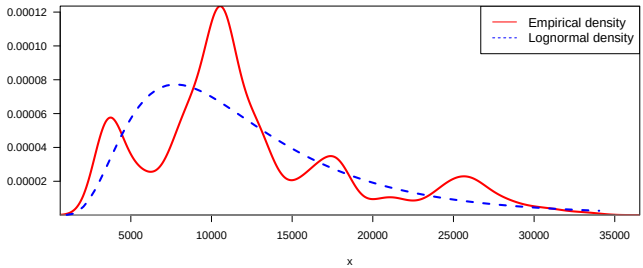


Fig. 1.19: Empirical density *vs.* normalized lognormal density.

Power tail distributions

We note that the empirical density has significantly higher kurtosis and non zero skewness in comparison with the Gaussian probability density. On the other hand, power tail probability densities of the form $\varphi(x) \simeq C_\alpha/|x|^\alpha$, $|x| \to \infty$, can provide a better fit of empirical probability density functions, as shown in Figure 1.20.

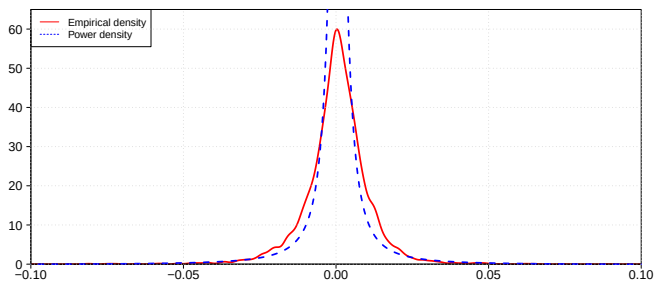


Fig. 1.20: Empirical density *vs.* power density.

The above fitting of empirical probability density function is using a power probability density function defined by a rational fraction obtained by the following R script.

```
install.packages("pracma")
library(pracma); x <- seq(-0.25, 0.25, length=1000)
stock.dens=density(returns,na.rm=TRUE, from = -0.1, to = 0.1, n = 1000)
a<-rationalfit(stock.dens$x, stock.dens$y, d1=2, d2=2)
dev.new(width=16,height=8)
plot(stock.dens$x,stock.dens$y, lwd=4, type = "l",xlab = '', col="red",ylab = '', main = '', xlim
      =c(-0.1,0.1), ylim=c(0,65), xaxs="i", yaxs="i", las=1, cex.lab=1.8, cex.axis=1.8)
lines(x,(a$p1[3]+a$p1[2]*x+a$p1[1]*x^2)/(a$p2[3]+a$p2[2]*x+a$p2[1]*x^2),
      type="l",lty=2,col="blue",xlab="x value",lwd=4, ylab="Density",main="")
legend("topleft", legend=c("Empirical density", "Power density"),col=c("red", "blue"), lty=1:2,
      cex=1.5);grid(lwd = 2)
```

The output of the `rationalfit` command is

```
$p1
[1] -0.184717249 -0.001591433  0.001385017

$p2
[1]  1.000000e+00 -6.460948e-04  1.314672e-05
```

which yields a rational fraction of the form

$$x \longmapsto \frac{0.001385017 - 0.001591433 \times x - 0.184717249 \times x^2}{1.314672\ 10^{-5} - 6.460948\ 10^{-4} \times x + x^2}$$
$$\simeq -0.184717249 - \frac{0.001591433}{x} + \frac{0.001385017}{x^2},$$

which approximates the empirical probability density function of DJI returns in the least squares sense.

A solution to this tail problem is to use stochastic processes with jumps, that will account for sudden variations of the asset prices. On the other hand, such jump models are generally based on the Poisson distribution which has a slower tail decay than the Gaussian distribution. This allows one to assign higher probabilities to extreme events, resulting in a more realistic modeling of asset prices. *Stable distributions* with parameter $\alpha \in (0, 2)$ provide typical examples of probability laws with power tails, as their probability density functions behave asymptotically as $x \mapsto C_\alpha / |x|^{1+\alpha}$ when $x \to \pm\infty$.

1.4 Gram-Charlier Expansions

In this section, we search for a better fit of market return distributions using the additional information provided by higher order moments. Let now

$$\varphi(x) := \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2}, \qquad x \in \mathbb{R},$$

denote the standard normal density function, and let

$$\Phi(x) := \int_{-\infty}^{x} \varphi(y)dy, \qquad x \in \mathbb{R},$$

denote the standard normal cumulative distribution function. Let also

$$H_n(x) := \frac{(-1)^n}{\varphi(x)} \frac{\partial^n \varphi}{\partial x^n}(x), \qquad x \in \mathbb{R},$$

denote the Hermite polynomial of degree n, with $H_0(x) = 1$. The next proposition summarizes the Gram-Charlier expansion method to obtain series expansion of a probability density function, see Gram (1883), Charlier (1914) and § 17.6 of Cramér (1946).

Proposition 1.11. *(Proposition 2.1 in Tanaka et al. (2010)) The Gram-Charlier expansion of the continuous probability density function $\phi_X(x)$ of a random variable X is given by*

$$\phi_X(x) = \frac{1}{\sqrt{\kappa_2(X)}}\varphi\left(\frac{x-\kappa_1(X)}{\sqrt{\kappa_2(X)}}\right) + \frac{1}{\sqrt{\kappa_2(X)}}\varphi\left(\frac{x-\kappa_1(X)}{\sqrt{\kappa_2(X)}}\right)\sum_{n=3}^{\infty} c_n H_n\left(\frac{x-\kappa_1(X)}{\sqrt{\kappa_2(X)}}\right),$$

where $c_0 = 1$, $c_1 = c_2 = 0$, and the sequence $(c_n)_{n\geqslant 3}$ is given from the cumulants $(\kappa_n(X))_{n\geqslant 1}$ of X as

$$c_n = \frac{1}{(\kappa_2(X))^{n/2}} \sum_{m=1}^{[n/3]} \sum_{\substack{l_1+\cdots+l_m=n \\ l_1,\ldots,l_m\geqslant 3}} \frac{\kappa_{l_1}(X)\cdots\kappa_{l_m}(X)}{m!l_1!\cdots l_m!}, \qquad n \geqslant 3.$$

The coefficients c_3 and c_4 can be expressed from the skewness $\kappa_3(X)/(\kappa_2(X))^{3/2}$ and the excess kurtosis $\kappa_4(X)/(\kappa_2(X))^2$ as

$$c_3 = \frac{\kappa_3(X)}{3!(\kappa_2(X))^{3/2}} \quad \text{and} \quad c_4 = \frac{\kappa_4(X)}{4!(\kappa_2(X))^2}.$$

a) The second-order expansion

$$\phi_X^{(1)}(x) = \frac{1}{\sqrt{\kappa_2(X)}}\varphi\left(\frac{x-\kappa_1(X)}{\sqrt{\kappa_2(X)}}\right)$$

corresponds to normal moment matching approximation.

b) The third-order expansion is given by

$$\phi_X^{(3)}(x) = \frac{1}{\sqrt{\kappa_2(X)}}\left(1 + c_3 H_3\left(\frac{x-\kappa_1(X)}{\sqrt{\kappa_2(X)}}\right)\right)\varphi\left(\frac{x-\kappa_1(X)}{\sqrt{\kappa_2(X)}}\right).$$

c) The fourth-order expansion is given by

$$\phi_X^{(4)}(x) = \frac{1}{\sqrt{\kappa_2(X)}}\left(1 + c_3 H_3\left(\frac{x-\kappa_1(X)}{\sqrt{\kappa_2(X)}}\right) + c_4 H_4\left(\frac{x-\kappa_1(X)}{\sqrt{\kappa_2(X)}}\right)\right)\varphi\left(\frac{x-\kappa_1(X)}{\sqrt{\kappa_2(X)}}\right).$$

```
install.packages("SimMultiCorrData");install.packages("PDQutils")
library(SimMultiCorrData);library(PDQutils)
dev.new(width=16,height=8);m<-calc_moments(returns[!is.na(returns)]);
x <- stock.dens$x; qx <- dnorm(x,mean=m[1],sd=m[2])
plot(x,stock.dens$y, xlab = 'x', type = 'l', lwd=4, col="red",ylab = '', main = '', xlim
     =c(-0.1,0.1), ylim=c(0,65), xaxs="i", yaxs="i", las=1, cex.lab=1.8, cex.axis=1.8)
grid(lwd = 2); lines(x, qx, type="l", lty=2, lwd=4, col="blue")
cumulants<-c(m[1],m[2]**2);d2 <- dapx_edgeworth(x, cumulants)
lines(x, d2, type="l", lty=2, lwd=4, col="blue")
cumulants<-c(m[1],m[2]**2,m[3]*m[2]**3);d3 <- dapx_edgeworth(x, cumulants)
lines(x, d3, type="l", lty=2, lwd=4, col="green")
cumulants<-c(m[1],m[2]**2,0.5*m[3]*m[2]**3,0.2*m[4]*m[2]**4)
d4 <- dapx_edgeworth(x, cumulants);lines(x, d4, type="l", lty=2, lwd=4, col="purple")
legend("topleft", legend=c("Empirical density", "Gaussian density", "Third order
       Gram-Charlier", "Fourth order Gram-Charlier"),col=c("red", "blue", "green", "purple"),
       lty=1:2,cex=1.5)
```

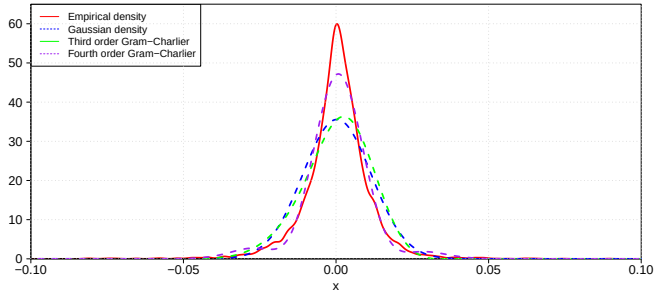


Fig. 1.21: Gram-Charlier expansions

Exercises

Exercise 1.1 Let $c > 0$. Using the definition of Brownian motion $(B_t)_{t\in\mathbb{R}_+}$, show that:

This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html

a) $(B_{c+t} - B_c)_{t\in\mathbb{R}_+}$ is a Brownian motion.
b) $(cB_{t/c^2})_{t\in\mathbb{R}_+}$ is a Brownian motion.

Exercise 1.2 Solve the stochastic differential equation

$$dS_t = \mu S_t dt + \sigma S_t dB_t, \tag{1.16}$$

defining geometric Brownian motion $(S_t)_{t\in\mathbb{R}_+}$, where $\mu, \sigma \in \mathbb{R}$.

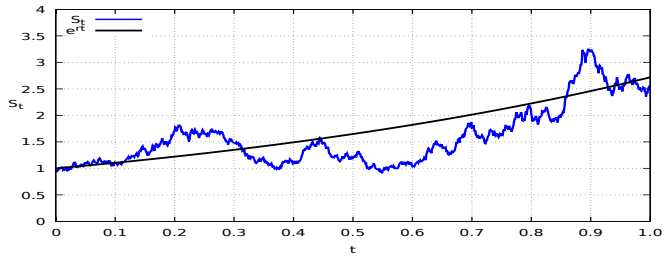


Fig. 1.22: Sample path of (1.16) with $r = 1$ and $\sigma^2 = 0.5$.

Exercise 1.3

a) Using the Gaussian moment generating function (MGF) formula (A.41), compute the *n-th* order moment $\mathbb{E}[S_t^n]$ for all $n \geqslant 1$.
b) Compute the lognormal mean and variance

$$\mathbb{E}[S_t] = S_0 e^{rt} \quad \text{and} \quad \mathrm{Var}[S_t] = S_0^2 e^{2rt}\big(e^{\sigma^2 t} - 1\big), \quad t \geqslant 0.$$

Exercise 1.4 Consider two assets whose prices $S_t^{(1)}$, $S_t^{(2)}$ at time $t \in [0, T]$ follow the geometric Brownian dynamics

$$dS_t^{(1)} = \mu S_t^{(1)} dt + \sigma_1 S_t^{(1)} dW_t^{(1)} \quad \text{and} \quad dS_t^{(2)} = \mu S_t^{(2)} dt + \sigma_2 S_t^{(2)} dW_t^{(2)},$$

$t \in [0, T]$, where $\big(W_t^{(1)}\big)_{t\in[0,T]}$, $\big(W_t^{(2)}\big)_{t\in[0,T]}$ are two Brownian motions with correlation $\rho \in [-1, 1]$, *i.e.* we have $\mathbb{E}\big[W_t^{(1)} W_t^{(2)}\big] = \rho t$. Compute $\mathrm{Var}\big[S_t^{(2)} - S_t^{(1)}\big]$, $t \in [0, T]$.

Exercise 1.5 We consider an economy in which individual income is modeled over time by a geometric Brownian motion

$$S_t = S_0 \mathrm{e}^{\sigma B_t + \mu t - \sigma^2 t/2}, \qquad t \geqslant 0.$$

Compute the value

$$\mathrm{Theil}_t := \log \mathbb{E}[S_t] - \mathbb{E}[\log S_t]$$

of the Theil (1967) inequality index at time $t \geqslant 0$ in this economy.

Hint: You may use the moment generating function (13.59) of the normal distribution.

This version: May 3, 2024
https://personal.ntu.edu.sg/nprivault/indext.html