

Jump Diffusion Models for Option Pricing with Emphasis on Stochastic Calculus

Dhruv Vansraj Rathore

*A dissertation submitted for the partial fulfillment of
BS-MS dual degree in Science*



Indian Institute of Science Education and
Research Mohali

April 2022

Certificate of Examination

This is to certify that the dissertation titled **Jump Diffusion Models for Option Pricing with Emphasis on Stochastic Calculus** submitted by Dhruv Vansraj Rathore (Reg. No. MS17142) for the partial fulfillment of BS-MS dual degree programme of the institute, has been examined by the thesis committee duly appointed by the institute. The committee finds the work done by the candidate satisfactory and recommends that the report be accepted.

Dr. Abhik Ganguli

Dr. Amit Kulshrestha

Dr. Neeraja Sahasrabudhe
(Local Supervisor)

Dr. Arun Kumar
(External Supervisor)

Dated: April 28, 2022

Declaration

The work presented in this dissertation has been carried out by me under the guidance of Dr. Arun Kumar at Indian Institute of Technology, Ropar and Dr. Neeraja Sahasrabudhe at the Indian Institute of Science Education and Research, Mohali.

This work has not been submitted in part or in full for a degree, a diploma, or a fellowship to any other university or institute. Whenever contributions of others are involved, every effort is made to indicate this clearly, with due acknowledgement of collaborative research and discussions. This thesis is a bonafide record of work done by me and all sources listed within have been detailed in the bibliography.

Dhruv Vansraj Rathore
(Candidate)

Dated: April 28, 2022

In my capacity as the supervisor of the candidate's project work, I certify that the above statements by the candidate are true to the best of my knowledge.

Dr. Arun Kumar
(External Supervisor)

Dr. Neeraja Sahasrabudhe
(Local Supervisor)

Acknowledgement

I would like to thank my thesis Supervisors **Dr. Arun Kumar** and **Dr. Neeraja Sahasrabudhe** for giving me the opportunity to learn and work under their guidance. It is due to their patience and helpfulness that I have been able to complete the thesis project. I would also like to thank my family members and all my friends for constantly encouraging me and always keeping me motivated whenever I felt low during hard times to finish my degree. Further I would like to thank **IISER Mohali** and **IISER Mohali Library** for providing me opportunities and a good environment to carry out my study.

Abstract

The aim of thesis is to cover the jump diffusion models for option pricing. A detailed theory on stochastic processes and stochastic calculus has been covered rigorously. The Black Scholes option pricing formula has been derived using the Equivalent Martingale Measure approach which utilises the application of Girsanov's theorem to change the underlying probability measure. The jump diffusion models proposed by Merton(M.J.D. model) and Kou(K.J.D. model) has been covered and the parameter estimation of these models has been covered for Infosys stock using Maximum Likelihood Estimate(M.L.E.) approach. It has been shown that the K.J.D. model fits best to the Infosys stock. A brief discussion on implied volatility has been carried out and implied volatility plots for TATA Motors stock and NIFTY50 stock index has been obtained respectively. Finally for modeling time varying volatility the GARCH(1,1) model using gaussian and student's t distribution innovation respectively with parameter estimation for Reliance stock has been covered and it has been shown that the GARCH(1,1) model with student's t distribution innovation models the volatility better for Reliance stock as compared to gaussian innovation.

Contents

Certificate of Examination	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
1 Introduction	1
2 Preliminaries	3
2.1 A Primer on Probability Theory	3
2.2 A Primer on Stochastic Processes	5
2.3 A Primer on Option theory	8
3 Stochastic Calculus and the Black Scholes Model	11
3.1 Properties of Brownian Motion	11
3.2 Stochastic Calculus	14
3.2.1 Itô Integral	15
3.2.2 Itô formula	18
3.3 Black Scholes Model for Option pricing	20
3.3.1 Deriving the Black Scholes formula by Equivalent martingale measure technique	22
3.3.2 Parameter estimation and some statistics of BS modelled returns	27
3.3.3 Drawbacks of the Black Scholes Model	28
4 Jump Diffusion Models for Option Pricing	30
4.1 Jump diffusion S.D.E. and its solution	30
4.2 Merton Jump Diffusion Model	31
4.3 Kou Jump Diffusion Model	35
4.4 Option pricing under Jump Diffusion Models and parameter estimation .	37

4.5	Application of the Black Scholes and the Jump Diffusion models for Infosys stock	39
4.5.1	Parameter estimation and comparative study of the models	39
4.5.2	Pricing Infosys Call options under the BS and the Jump diffusion models	43
5	Volatility	45
5.1	The Implied Volatility	45
5.2	GARCH(1,1) Model	47
5.2.1	Parameter estimation	50
5.2.2	Application of GARCH(1,1) model on Reliance stock	50
A	Python Code used for the project	57
A.1	Code used for simulation of BM,Poisson,GBM,MJD and KJD paths . . .	57
A.2	Code used for estimating BS, MJD and KJD parameters for Infosys equity	59
A.3	Code used for pricing Infosys call options under BS, MJD and KJD models respectively	65
A.4	Code used for computing implied volatilities for TATA MOTORS stock and NIFTY50 stock index data respectively	67
A.5	Code used for estimating the GARCH(1,1) model parameters for Reliance equity	70
B	Computed Implied Volatilities of Call options for TATA MOTORS stock and NIFTY50 stock index data respectively	75

List of Figures

2.1	Simulation of Brownian motion paths.	7
2.2	Simulation of Poisson process paths with $\lambda = 5$	8
2.3	Options pay off diagram.	9
3.1	Simulation of GBM paths with $S_0 = 100, \mu = 0.2$ and $\sigma = 0.3$	20
4.1	Simulation of MJD paths with $S_0 = 100, \mu = 0.01, \sigma = 0.1, \lambda = 5, m = 0$ and $\delta = 0.1$	32
4.2	Simulation of KJD paths with $S_0 = 100, \mu = 0.01, \sigma = 0.1, \lambda = 5, p =$ $0.5, \eta_1 = 10$ and $\eta_2 = 10$	35
4.3	Daily closing price of Infosys stock.	39
4.4	Daily returns of Infosys stock.	40
4.5	Comparison of the empirical returns with the simulated returns of the BS, MJD and KJD models respectively.	41
4.6	Comparison of the kernel density estimate plots of empirical returns with the BS, MJD and KJD models simulated returns respectively.	42
5.1	TATA MOTORS stock call options implied volatility plot.	46
5.2	NIFTY50 stock index call options implied volatility and volatility surface plot.	46
5.3	Daily closing price of Reliance stock from 2012-01-01.	51
5.4	Daily log-returns of Reliance stock.	51
5.5	ACF plot of the squared daily log-returns.	52
5.6	Q-Q plots of the GARCH(1,1) and t-GARCH(1,1) innovations respectively.	54
5.7	Conditional variance computed through the estimated t-GARCH(1,1) model.	55

List of Tables

4.1	Some statistics of daily returns of Infosys stock.	40
4.2	BS estimated parameters.	40
4.3	MJD estimated parameters through M.L.E.	40
4.4	KJD estimated parameters through M.L.E.	41
4.5	Comparison of some statistics of the BS, MJD and KJD modelled returns respectively with the empirical returns.	43
4.6	Computed Option prices of Infosys call options under BS, MJD and KJD models respectively.	44
5.1	Some statistics of log-returns of Reliance stock.	52
5.2	Augmented Dickey-Fuller test for stationarity of the log-return series. . .	52
5.3	Ljung-Box test for first 5 lags of the squared log-return series.	53
5.4	GARCH(1,1) estimated parameters using M.L.E.	53
5.5	t-GARCH(1,1) estimated parameters using M.L.E.	53
5.6	Mean and variance of the GARCH(1,1) and t-GARCH(1,1) innovations respectively.	54
5.7	Normality test for the GARCH(1,1) innovations.	55
5.8	Kurtosis for empirical, GARCH(1,1) and t-GARCH(1,1) returns respectively.	55
B.1	Computed I.V. values using Newton-Raphson method and the I.V. values listed on NSE option chain site of TATA MOTORS stock call options. . .	75
B.2	Computed I.V. values of call options of NIFTY50 stock index with different maturities.	76

Chapter 1

Introduction

Option contracts are type of derivatives which play a crucial role in financial markets and in contributing towards the growth of economy. Option pricing is important to traders in Stock markets and since option contracts involve a stock as its underlying the modeling of stock price then becomes crucial in order to price the option. The pioneering works of Fischer Black and Myron Scholes(1973) in the paper titled “The Pricing of Options and Corporate Liabilities”[BS73] and the work of Robert C. Merton(1974) in the paper titled “On the pricing of corporate debt: the risk structure of interest rates”[Mer74] involved modeling the stock price process through Geometric Brownian Motion (G.B.M.) to price European options.

The Black Scholes Model using G.B.M. for stock price process cannot incorporate the jumps that are usually seen in the stock prices and can only model continuous sample paths. In 1976, Robert Merton in his paper titled “Option pricing when the underlying stock returns are discontinuous”[Mer76] incorporated jumps in G.B.M. through a compound poisson process. Merton proposed the jump sizes to be normally distributed and the model also covered the two major drawbacks of the Black Scholes model namely the implied volatility smile and the asymmetric leptokurtic feature of the returns.

In 2002, S.G. Kou in his paper titled “A Jump-Diffusion Model for Option Pricing” [Kou02] proposed the use of jump sizes in Merton jump diffusion(M.J.D.) model to follow double exponential distribution. Kou Jump diffusion(K.J.D.) model also explains the implied volatility smile and the asymmetric leptokurtic feature with the later being more pronounced in K.J.D. model as compared to M.J.D. model. In this thesis we have only covered the asymmetric leptokurtic feature of the jump diffusion models.

This thesis is divided into 5 chapters. In chapter 2 we have presented concepts of

probability theory, stochastic processes and some details of option pricing theory. In chapter 3 we have discussed important properties of Brownian motion, the Itô Integral, the Itô formula and the Black Scholes model is covered. The Black Scholes option pricing formula is derived using the Equivalent Martingale Measure approach which involves the use of Girsanov's theorem to change the probability measure to a new equivalent probability measure such that under which the discounted stock price process becomes a martingale. In chapter 4 we have discussed the Jump diffusion models proposed by Merton and S.G. Kou respectively. At the end of the chapter the parameter estimation of the Black Scholes and the jump diffusion models have been done by using Maximum Likelihood Estimate(M.L.E.) method for Infosys stock. It has been shown that the K.J.D. model fits best to the Infosys stock. The pricing of Infosys call options under the Black Scholes and the jump diffusion models is also done at the very end. In chapter 5 we have discussed about implied volatility and has obtained the implied volatility plots for TATA Motors stock and NIFTY50 stock index respectively. Then the GARCH(1,1) model is covered for modeling time-varying volatility and parameter estimation of the model is done using the M.L.E. approach for both gaussian and student's t distribution innovation respectively for Reliance Stock. It has been shown that the GARCH(1,1) model with student's t distribution innovation models the volatility of Reliance stock better as compared to gaussian innovation.

Chapter 2

Preliminaries

2.1 A Primer on Probability Theory

This section is meant to review probability theory and some key measure theory concepts required for the project. Theory in this section has been covered from [Kle20].

Definition 2.1.1 (σ -algebra). *Let Ω be a nonempty set. A collection of subsets of Ω denoted by \mathcal{F} is said to be σ -algebra if following holds:*

1. $\phi \in \mathcal{F}$.
2. If $A \in \mathcal{F}$ then $A^c \in \mathcal{F}$.
3. If the sequence of sets A_1, A_2, \dots belongs to \mathcal{F} , then $\cup_{i=1}^{\infty} A_i \in \mathcal{F}$.

Definition 2.1.2. *Consider (Ω, \mathcal{F}) , where Ω is a non-empty set with \mathcal{F} as σ -algebra on Ω . Suppose \mathbb{P} is a function $\mathbb{P} : \mathcal{F} \rightarrow [0, 1]$ such that,*

1. $\mathbb{P}(\Omega) = 1$.
2. For any countable pairwise disjoint collection $\{A_i\}_{i \geq 1}$:

$$\mathbb{P}(\cup_{i \geq 1} A_i) = \sum_{i \geq 1} \mathbb{P}(A_i).$$

Then \mathbb{P} is called a probability measure and the triplet $(\Omega, \mathcal{F}, \mathbb{P})$ is called a probability space.

Definition 2.1.3. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. A random variable X is a real-valued function $X : (\Omega, \mathcal{F}) \rightarrow (\mathbb{R}, \mathcal{B}(\mathbb{R}))$ such that we have

$$X^{-1}(A) \in \mathcal{F}, \quad \forall A \in \mathcal{B}(\mathbb{R}),$$

where $\mathcal{B}(\mathbb{R})$ is the Borel sigma algebra of \mathbb{R} .

The sigma algebra generated by a random variable X is the smallest sigma algebra with respect to which X is measurable and is given by

$$\sigma(X) = \{X^{-1}(A) : A \in \mathcal{B}(\mathbb{R})\}.$$

We now present some of the notions of convergence of random variable which are required in this project.

Definition 2.1.4. Consider a sequence of random variables $(X_n)_{n \geq 0}$. Then as $n \rightarrow \infty$ we say that

1. the sequence $(X_n)_{n \geq 1}$ converges with probability 1 or almost surely (a.s.) to a random variable X if and only if $P(\{\omega : X_n(\omega) \rightarrow X(\omega)\}) = 1$. We denote it by writing $X_n \xrightarrow{a.s.} X$.
2. the sequence $(X_n)_{n \geq 1}$ converges in probability to a random variable X if and only if $\mathbb{P}(\{\omega : |X_n(\omega) - X(\omega)| \geq \epsilon\}) \xrightarrow{n \rightarrow \infty} 0, \forall \epsilon > 0$. We denote it by writing $X_n \xrightarrow{\mathbb{P}} X$.
3. the sequence $(X_n)_{n \geq 1}$ converges in r -th mean ($1 \leq r < \infty$) to a random variable X if and only if $E[|X_n - X|^r] \xrightarrow{n \rightarrow \infty} 0$. We denote it by writing $X_n \xrightarrow{L^r} X$.

We note that there are some relationships between the above three mentioned notions of convergence. We only present those which are required and they are as follows:

- (1.) $X_n \xrightarrow{a.s.} X \implies X_n \xrightarrow{\mathbb{P}} X$.
- (2.) $X_n \xrightarrow{L^r} X \implies X_n \xrightarrow{\mathbb{P}} X \implies X_{n_k} \xrightarrow{a.s.} X$ along some subsequence $(n_k)_{k \geq 0}$.

We now present two important theorems namely the Monotone convergence theorem (M.C.T.) and the Dominated Convergence theorem (D.C.T.) which provide sufficient condition under which the expectation of a sequence of converging random variables converges.

Theorem 2.1.1 (M.C.T.). Let $(X_n)_{n \geq 0}$ be a sequence of integrable random variables such that $X_n \uparrow X$ a.s. Then we have that $E[X_n] \uparrow E[X]$.

Theorem 2.1.2 (D.C.T.). *Let $(X_n)_{n \geq 0}$ be a sequence of integrable random variables satisfying $|X_n| \leq Y$ where Y is some random variable with $E[Y] < \infty$ and $X_n \uparrow X$ a.s. Then we have that $E[X_n] \uparrow E[X]$.*

For a more detailed discussion of the above notions of convergence refer to [Kle20]. We now discuss the notion of Conditional Expectation.

Definition 2.1.5. *Let X be a integrable random variable defined on probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and let $\mathcal{G} \subset \mathcal{F}$ be a sub σ -field. Then the conditional expectation of X given \mathcal{G} is a random variable $E[X|\mathcal{G}]$ such that,*

1. $E[X|\mathcal{G}]$ is \mathcal{G} -measurable.
2. For any $A \in \mathcal{G}$ we have that $\int_A E[X|\mathcal{G}] dP = \int_A X dP$.

We now list some of the important properties of the conditional expectation. Let X, Y be two random variables.

1. $E[aX + bY|\mathcal{G}] = aE[X|\mathcal{G}] + bE[Y|\mathcal{G}]$ for some constants a and b .
2. $E[XY|\mathcal{G}] = XE[Y|\mathcal{G}]$ if X is \mathcal{G} - measurable.
3. If $\sigma(X)$ is independent of \mathcal{G} then $E[X|\mathcal{G}] = E[X]$.
4. $E[E[X|\mathcal{G}]|\mathcal{H}] = E[E[X|\mathcal{H}]|\mathcal{G}] = E[X|\mathcal{H}]$ if $\mathcal{H} \subset \mathcal{G}$.

The fourth property is known as the tower property using it one can also show that $E[E[X|\mathcal{G}]] = E[X]$.

Definition 2.1.6. *Let \mathbb{P} and \mathbb{Q} be two probability measures defined on the common space (Ω, \mathcal{F}) . Then \mathbb{Q} is said to be absolutely continuous with respect to \mathbb{P} if for every set $A \in \mathcal{F}$, $\mathbb{P}(A) = 0 \implies \mathbb{Q}(A) = 0$. We denote it by $\mathbb{Q} \ll \mathbb{P}$.*

Now two probability measures \mathbb{P} and \mathbb{Q} are said to be equivalent if $\mathbb{Q} \ll \mathbb{P}$ and $\mathbb{P} \ll \mathbb{Q}$.

2.2 A Primer on Stochastic Processes

Stochastic process are used to describe randomly occurring non-deterministic phenomenon. Stock price process is also a non-deterministic phenomena i.e., we cannot exactly predict the stock price in future. Theory of Stochastic process can help us to model the stock

price process. In this section we will cover the theory of stochastic process including the two important stochastic processes namely Brownian motion and Poisson process where the former process plays a crucial role in modelling stock price process in the Black Scholes Model and the later plays a crucial role in incorporating jumps in stock price process for Jump diffusion models. Theory in this section is covered from [Mik11],[Kle20].

Definition 2.2.1 (Stochastic Process). *A stochastic process is a family or collection of random variables denoted by $\mathbf{X} = \{X_t(\omega), t \in \mathcal{I}, \omega \in \Omega\}$ where each of the random variables X_t are defined on a common probability sample space Ω are real valued and indexed by $t \in \mathcal{I}$ where the indexing set $\mathcal{I} \subset \mathbb{R}$.*

A stochastic process therefore is a function of two variables. For fixed t , it is a random variable $X_t(\omega)$, $\omega \in \Omega$ and for a fixed ω it is a function of time $X_t = X_t(\omega)$, $t \in \mathcal{I}$ which is known as the sample path of the process.

Note: For notational convenience we will drop ω as defined in definition of stochastic process and denote the stochastic process by $\mathbf{X} = \{X_t, t \in \mathcal{I}\}$.

Definition 2.2.2 (Stationary process). *A stochastic process $\mathbf{X} = \{X_t, t \in \mathcal{I}\}$ is called strictly stationary if for all possible choices of time indices $t_1, t_2, \dots, t_n \in \mathcal{I}$ where $n \geq 1$ and for all $h \in \mathbb{R}$ such that $t_1 + h, t_2 + h, \dots, t_n + h \in \mathcal{I}$ we have that the distribution of random vectors $(X_{t_1}, X_{t_2}, \dots, X_{t_n})$ is invariant under time shift i.e.,*

$$(X_{t_1}, X_{t_2}, \dots, X_{t_n}) \stackrel{d}{=} (X_{t_1+h}, X_{t_2+h}, \dots, X_{t_n+h}). \quad (2.1)$$

A stochastic process $\mathbf{X} = \{X_t, t \in \mathcal{I}\}$ is called weakly stationary if for all $s, t \in \mathcal{I}$ and for all $h \in \mathbb{R}$ such that $s + h, t + h \in \mathcal{I}$ we have the following:

$$\begin{aligned} E[X_{t+h}] &= E[X_t]. \\ Cov[X_{s+h}, X_{t+h}] &= Cov[X_s, X_t]. \\ E[|X_t|^2] &< \infty. \end{aligned} \quad (2.2)$$

This implies that the expectation and variance of the process is constant and the covariance depends only on the difference of the time indices for weakly stationary process.

Definition 2.2.3 (Gaussian process). *A stochastic process $\mathbf{X} = \{X_t, t \in \mathcal{I}\}$ is called a gaussian process if for all $t_1, t_2, \dots, t_n \in \mathcal{I}$ where $n \geq 1$ we have that the random vector $(X_{t_1}, X_{t_2}, \dots, X_{t_n})$ is distributed as multivariate Gaussian random variable or equivalently every linear combinations of $X_{t_1}, X_{t_2}, \dots, X_{t_n}$ is normally distributed.*

Definition 2.2.4 (Brownian Motion). A stochastic process $\mathbf{B} = \{B_t, t \geq 0\}$ defined on $(\Omega, \mathcal{F}, \mathbb{P})$ is called a standard Brownian Motion if it satisfies the following:

- (1.) $B_0 = 0$.
- (2.) For $0 \leq s < t$, $B_t - B_s \stackrel{d}{=} \mathcal{N}(0, t - s)$.
- (3.) For $0 < t_1 < t_2 < \dots < t_k$ we have that $B_{t_1}, B_{t_2} - B_{t_1}, \dots, B_{t_k} - B_{t_{k-1}}$ are independent.
- (4.) The sample paths of \mathbf{B} are continuous a.s.

Fig 2.1 shows simulation of some brownian motion sample paths.

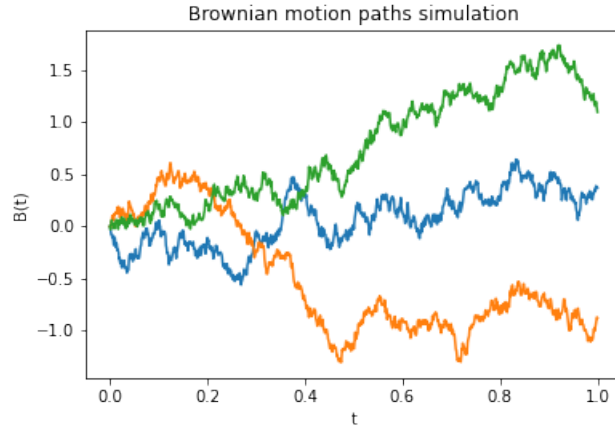


Figure 2.1: Simulation of Brownian motion paths.

Definition 2.2.5 (Poisson Process). A stochastic process $\mathbf{N} = \{N_t, t \geq 0\}$ defined on $(\Omega, \mathcal{F}, \mathbb{P})$ is called a Poisson process with rate λ if it satisfies the following:

- (1.) $N_0 = 0$.
- (2.) For $0 \leq s < t$, $N_t - N_s \stackrel{d}{=} \text{Pois}(\lambda(t - s))$ i.e., for $k \in \mathbb{N}$ we have that,

$$\mathbb{P}(N_t - N_s = k) = \frac{(\lambda(t - s))^k e^{-\lambda(t-s)}}{k!}. \quad (2.3)$$

- (3.) For $0 < t_1 < t_2 < \dots < t_k$ we have that $N_{t_1}, N_{t_2} - N_{t_1}, \dots, N_{t_k} - N_{t_{k-1}}$ are independent.

Note: For a Poisson process for small interval dt we have that:

$$\mathbb{P}(N_{t+dt} - N_t = k) = \mathbb{P}(dN_t = k) = \begin{cases} \lambda dt + o(dt), & \text{if } k = 1 \\ o(dt), & \text{if } k \geq 2 \\ 1 - \lambda dt + o(dt), & \text{if } k = 0 \end{cases} \quad (2.4)$$

Fig 2.2 shows simulation of some poisson process sample paths.

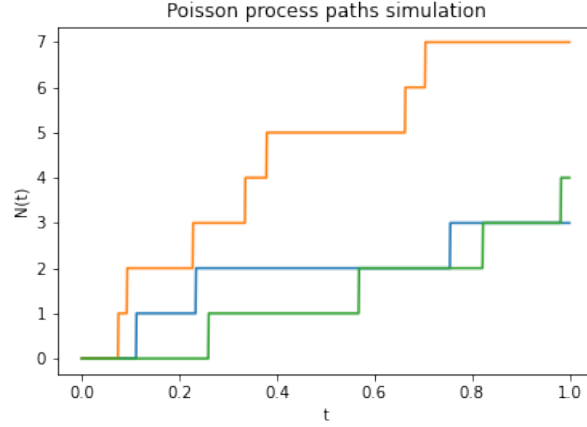


Figure 2.2: Simulation of Poisson process paths with $\lambda = 5$.

Definition 2.2.6 (Filtration). A filtration on a $(\Omega, \mathcal{F}, \mathbb{P})$ is a non-decreasing family of sub sigma algebras of \mathcal{F} denoted by $\{\mathcal{F}_t, t \in \mathcal{I}\}$ such that for $s \leq t$ we have that $\mathcal{F}_s \subseteq \mathcal{F}_t \subseteq \mathcal{F}$.

Definition 2.2.7 (Adapted Stochastic process). A stochastic process $\mathbf{X} = \{X_t, t \in \mathcal{I}\}$ is said to be adapted to a filtration $\{\mathcal{F}_t, t \in \mathcal{I}\}$ if X_t is \mathcal{F}_t -measurable for every $t \in \mathcal{I}$.

Note: A stochastic process $\mathbf{X} = \{X_t, t \in \mathcal{I}\}$ is always adapted to the filtration it generates denoted by $\{\mathcal{F}_t^X, t \in \mathcal{I}\}$ where $\mathcal{F}_t^X = \sigma(\{X_s : s \leq t\})$.

Definition 2.2.8 (Martingale process). A stochastic process $\mathbf{X} = \{X_t, t \in \mathcal{I}\}$ is said to be a martingale with respect to a filtration $\{\mathcal{F}_t, t \in \mathcal{I}\}$ if the following holds:

1. (**Adapted**) X_t is \mathcal{F}_t -measurable for all $t \in \mathcal{I}$.
2. (**Integrable**) $E[|X_t|] < \infty$.
3. (**Martingale property**) $E[X_t | \mathcal{F}_s] = X_s$ for $s \leq t$.

Note: From the martingale property we have that $E[X_t] = E[E[X_t | \mathcal{F}_s]] = E[X_s]$. This implies that Martingale processes have constant expectation.

2.3 A Primer on Option theory

In this section we give an introduction to Option theory and some financial terminologies that are required for the project. The Theory in this section is covered from [Cam14]

and [Pap19].

A security is a financial instrument that has monetary value and which can be traded in Financial markets. Financial markets often make use of securities with stocks and bonds as its common types. A derivative is a type of security whose value is derived from another underlying asset. An option contract is also a derivative such that the underlying asset in this case is stock. Options are of two types namely call and put options. We give the proper definition of an Option Contract.

Definition 2.3.1. *A call(put) option is a contract that gives its holder the right, but not the obligation, to buy(sell) a specific asset at a specified price(called the strike price) on or before a specific time(called the maturity time).*

Options are further classified into two types namely European and American options with the former can be exercised only at the maturity time and the latter can be exercised on or before the maturity time. In this project we will only deal with European options. Let S_t be the price of a stock at time t . Consider a European option with the stock as its underlying with strike price denoted by K and maturity time of the contract denoted by T . Then the call option payoff and put option payoff denoted by δ_{Call} and δ_{Put} respectively are given by,

$$\delta_{Call} = \max(S_T - K, 0) = (S_T - K)^+.$$

$$\delta_{Put} = \max(K - S_T, 0) = (K - S_T)^+.$$

Fig 2.3 shows the payoff diagram for both put and call option with $K = 50$.

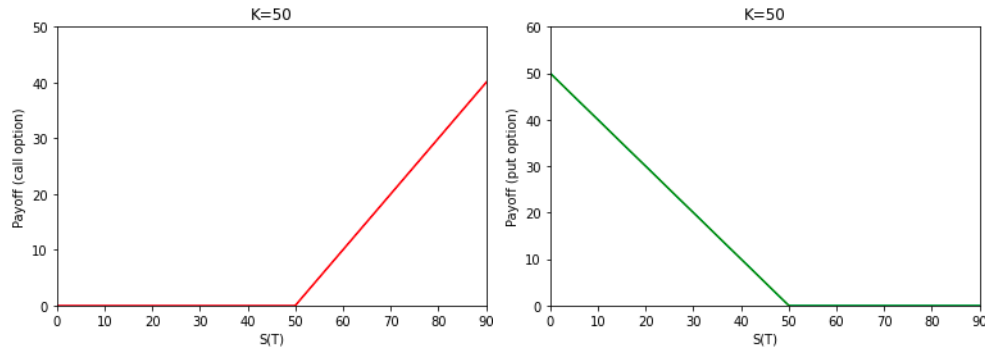


Figure 2.3: Options pay off diagram.

We now discuss the notion of moneyness for option. An option is said to be in the money when the option yields positive payoff (i.e., when $K < S_t$ for call option and

$S_t < K$ for put option.) It is said to be out of the money when the option yields no payoff (i.e., when $K > S_t$ for call option and $S_t > K$ for put option.) It is said to be at the money when $S_t = K$ for both call and put option.

We now give an important relationship known as put-call parity that holds between the price of a European call and put option defined on a same underlying asset. Let C and P be the price of the European call and put option respectively. The put call parity is given by,

$$C + Ke^{-rT} = P + S_0.$$

Put-call parity relation is quite important as it can be used to know the price of a put option given that we know the price of a call option on the same underlying and vice versa. For a detailed proof of put-call parity the reader is encouraged to refer to [Cam14].

Definition 2.3.2. (*Definition 1.1.1*, [Pap19]) *An arbitrage opportunity is a trading opportunity that allows an investor to have a non zero probability of getting a profit and zero probability of getting a loss respectively given that his initial wealth was zero.*

Theorem 2.3.1. (*Theorem 4.5*, [Cam14]) *Under the assumption of absence of arbitrage followings are true :*

$$\begin{aligned} (S_0 - Ke^{-rT})^+ &\leq C < S_0. \\ (Ke^{-rT} - S_0)^+ &\leq P < Ke^{-rT}. \end{aligned} \tag{2.5}$$

where C and P are the European Call and Put option price respectively.

For a proof of the above bounds please refer to [Cam14].

Chapter 3

Stochastic Calculus and the Black Scholes Model

In this chapter we will first cover some important properties of Brownian motion which then will be followed by rigorously covering Stochastic Calculus. At the end we will cover the Black Scholes model for option pricing. Theory in this chapter is mainly covered from [Mik11],[Shr11], [Res92] and [Øk13].

3.1 Properties of Brownian Motion

We now list out some of the important properties of Brownian Motion with a brief outline of proof of some of them.

1. $\mathbf{B} = \{B_t, t \geq 0\}$ is a **Gaussian process**

Proof: We have to check that for $0 \leq t_1 < t_2 < \dots < t_k$ the random vector $(B_{t_1}, B_{t_2}, \dots, B_{t_k})$ is multivariate gaussian. This can be checked by observing that for $c_i \in \mathbb{R}, 1 \leq i \leq k$ we have $c_1 B_{t_1} + c_2 B_{t_2} + \dots + c_k B_{t_k} = c_k (B_{t_k} - B_{t_{k-1}}) + (c_k + c_{k-1})(B_{t_{k-1}} - B_{t_{k-2}}) + \dots + (c_k + c_{k-1} + \dots + c_1) B_{t_1}$. The right hand side by the independent increments of \mathbf{B} is normally distributed.

Now by Definition 3.1.1 it can be checked that \mathbf{B} is a Gaussian process with mean zero and covariance given by,

$$Cov(B_s, B_t) = s \wedge t.$$

2. **Differential property:** For any $s > 0$, $\{B_{t+s} - B_s, t \geq 0\}$ is a standard Brownian motion independent of $\{B(u), u \leq s\}$.

3. **Scaling property:** For $c \geq 0$, $\{\sqrt{c}B(\frac{t}{c}), t \geq 0\}$ is a standard Brownian motion.
4. **Symmetry property:** $\{-B_t, t \geq 0\}$ is a standard Brownian motion.
5. **Time inversion property:** Let $\{B^1(t), t \geq 0\}$ be a stochastic process defined as,

$$B^1(t) = \begin{cases} tB(\frac{1}{t}), & \text{if } t > 0 \\ 0 & \text{if } t = 0 \end{cases}$$

then $\{B^1(t), t \geq 0\}$ is a standard Brownian motion.

Proof: We check each of the conditions of Definition 3.1.1.

(i) Clearly, $B^1(0) = 0$.

(ii) For $0 < s < t$, we check that $B^1(t) - B^1(s) \stackrel{d}{=} \mathcal{N}(0, t - s)$.

$$B^1(t) - B^1(s) = (t - s)B\left(\frac{1}{t}\right) - s\left(B\left(\frac{1}{s}\right) - B\left(\frac{1}{t}\right)\right).$$

The right hand side is a linear combination of independent normal random variables and so $B^1(t) - B^1(s)$ is normally distributed. Now we have that,

$$\begin{aligned} E[B^1(t) - B^1(s)] &= E\left[tB\left(\frac{1}{t}\right) - sB\left(\frac{1}{s}\right)\right] = 0. \\ Var[B^1(t) - B^1(s)] &= Var\left[(t - s)B\left(\frac{1}{t}\right) - s\left(B\left(\frac{1}{s}\right) - B\left(\frac{1}{t}\right)\right)\right] \\ &= \frac{(t - s)^2}{t} + s^2\left(\frac{1}{s} - \frac{1}{t}\right). \\ &= t - s. \end{aligned}$$

(iii) We check the independent increment condition. Let $t_1 < t_2 < t_3$ then,

$$\begin{aligned} Var[B^1(t_3) - B^1(t_1)] &= t_3 - t_1. \\ &= (t_3 - t_2) + (t_2 - t_1). \\ &= Var[B^1(t_2) - B^1(t_1)] + Var[B^1(t_3) - B^1(t_2)]. \end{aligned}$$

This implies that the increments $B^1(t_2) - B^1(t_1)$ and $B^1(t_3) - B^1(t_2)$ are uncorrelated and since they are jointly gaussian as well, therefore the increments are

independent.

(iv) Clearly, $B^1(t) = tB(\frac{1}{t})$ is continuous for $\forall t > 0$. For the proof of continuity at $t=0$ refer to [Res92].

6. The sample paths of Brownian motion are **nowhere differentiable**.

Proof: For the detailed proof refer to [Res92].

Nowhere differentiability can also be shown by using distributional arguments by observing that $\frac{B_{t+h}-B_t}{h} \stackrel{d}{=} \frac{\mathcal{N}(0,h)}{h} \stackrel{d}{=} \mathcal{N}(0, \frac{1}{h})$ which implies that the slope of the sample paths has infinite variance almost everywhere, therefore the sample paths are nowhere differentiable.

7. **Finite Quadratic variation:** Let $\pi := \{0 = t_0 < t_1 < \dots < t_{n-1} < t_n = T\}$ be a partition of $[0, T]$ such that $|\pi| = \max_{1 \leq i \leq n} (t_i - t_{i-1}) \xrightarrow{n \rightarrow \infty} 0$.

Then we have the following:

$$\sum_{i=1}^n (B_{t_i} - B_{t_{i-1}})^2 \xrightarrow[|\pi| \rightarrow 0]{L^2} T. \quad (3.1)$$

Proof: Denote the quadratic variation over $[0, T]$ by $[\mathbf{B}, \mathbf{B}](T)$. Then we have that,

$$\begin{aligned} \mathbb{E}[\sum_{i=1}^n (B_{t_i} - B_{t_{i-1}})^2] &= \sum_{i=1}^n \mathbb{E}[(B_{t_i} - B_{t_{i-1}})^2] = \sum_{i=1}^n (t_i - t_{i-1}) \\ &= T. \\ \text{Var}[\sum_{i=1}^n (B_{t_i} - B_{t_{i-1}})^2] &= \sum_{i=1}^n \text{Var}[(B_{t_i} - B_{t_{i-1}})^2] = 2 \sum_{i=1}^n (t_i - t_{i-1})^2 \\ &\leq 2|\pi|T \xrightarrow{|\pi| \rightarrow 0} 0. \end{aligned}$$

Thus $\sum_{i=1}^n (B_{t_i} - B_{t_{i-1}})^2 \xrightarrow[|\pi| \rightarrow 0]{L^2} \mathbb{E}[\sum_{i=1}^n (B_{t_i} - B_{t_{i-1}})^2] = T$ and $[\mathbf{B}, \mathbf{B}](T) = T$.

8. **Unbounded first variation:** Let π be the same partition of $[0, T]$ as defined in property (7). Then the first variation of Brownian motion is infinite a.s. i.e.,

$$\sup_{\pi} \sum_{i=1}^n |B_{t_i} - B_{t_{i-1}}| = \infty \quad a.s.$$

Proof: Let $T = 1$ and $\Delta_i B = B_{t_i} - B_{t_{i-1}}$. Suppose $\sup_{\pi} \sum_{i=1}^n |B_{t_i} - B_{t_{i-1}}| < \infty$. Then,

$$\begin{aligned} \sum_{i=1}^n (\Delta_i B)^2 &\leq \max_{i=1, \dots, n} |(\Delta_i B)| \sum_{i=1}^n |(\Delta_i B)| \leq \max_{i=1, \dots, n} |(\Delta_i B)| \sup_{\pi} \sum_{i=1}^n |(\Delta_i B)| \\ &\xrightarrow{|\pi| \rightarrow 0} 0 \quad a.s. \end{aligned} \quad (3.2)$$

3.2 holds because $\max_{i=1, \dots, n} |(\Delta_i B)| \xrightarrow{|\pi| \rightarrow 0} 0$ as B_t is uniformly continuous a.s. From property (7) we have that,

$$\begin{aligned} \sum_{i=1}^n (\Delta_i B)^2 &\xrightarrow[|\pi| \rightarrow 0]{L^2} 1 \implies \sum_{i=1}^n (\Delta_i B)^2 \xrightarrow[|\pi| \rightarrow 0]{\mathbb{P}} 1 \\ &\implies \sum_{i=1}^{n_k} (\Delta_i B)^2 \xrightarrow[|\pi| \rightarrow 0]{a.s.} 1. \end{aligned}$$

(along some subsequence n_k)

Then we must have that 3.2 happens only on a measure zero set.

$\implies \mathbb{P}(\{\sup_{\pi} \sum_{i=1}^n |B_{t_i} - B_{t_{i-1}}| = \infty\}) = 1$ and we are done.

9. Martingale property: $\{B_t, t \geq 0\}$ is a martingale process w.r.t. $\{\mathcal{F}_t, t \geq 0\}$ where $\mathcal{F}_t = \sigma(B_s, s \leq t)$.

Proof: Adaptibility of Brownian motion process is clear. Further from Cauchy-Schwarz inequality we have that $E[|B_t|] \leq \sqrt{E[B_t^2]} = \sqrt{t} < \infty$.

We now prove the martingale property let $s < t$,

$$\begin{aligned} E[B_t | \mathcal{F}_s] &= E[(B_t - B_s) | \mathcal{F}_s] + E[B_s | \mathcal{F}_s]. \\ &= E[B_t - B_s] + B_s \quad (\text{As } B_t - B_s \text{ is independent of } \mathcal{F}_s \text{ and } B_s \in \mathcal{F}_s). \\ &= B_s. \end{aligned}$$

Further it can also be shown that $\{B_t^2 - t, t \geq 0\}$ and $\{\exp(cB_t - \frac{1}{2}c^2t), t \geq 0\}$ are both also martingales w.r.t. $\{\mathcal{F}_t, t \geq 0\}$.

3.2 Stochastic Calculus

We first give a proper motivation for introducing Stochastic Calculus. We are particularly interested in integrals of the form $\int_0^T f(s) dB_s(\omega)$ and $\int_0^T B_s(\omega) dB_s(\omega)$ where f is a

deterministic function or a function of stochastic process which is adapted to the brownian motion filtration. In ordinary calculus we deal with these integrals by the Riemann-Stieljes approach. We first discuss the major problem that arises if we proceed with this approach. For simplicity we suppose $T = 1$.

Definition 3.2.1. For $p > 0$, the p -variation of a function $f : \mathbb{R} \rightarrow \mathbb{R}$ on $[0, 1]$ is defined as:

$$V^p(f) = \sup_{\pi} \sum_{i=1}^n |f(t_i) - f(t_{i-1})|^p. \quad (3.3)$$

where the supremum varies over the partitions π of $[0, 1]$. f is said to be a function of bounded p -variation on $[0, 1]$ if $V^p(f) < \infty$.

The following theorem provides sufficient condition under which the Riemann-Stieljes integral $\int_0^1 f(x)dg(x)$ exists for some functions f and g .

Theorem 3.2.1. The Riemann-Stieljes integral $\int_0^1 f(x)dg(x)$ exists provided the following are satisfied:

1. f and g do not have discontinuities at the same points on $[0, 1]$.
2. $V^p(f) < \infty$ and $V^q(g) < \infty$ for some $p, q > 0$ such that $\frac{1}{p} + \frac{1}{q} > 1$.

Further for Brownian motion sample paths it is known that $V^p(B_t) < \infty$ for $p > 2$ and $V^p(B_t) = \infty$ for $p \leq 2$.

Since $V^1(B_t) = \infty$ we will have that $\int_0^1 f(s)dB_s(\omega)$ exists in the Riemann-Stieljes sense provided $V^1(f_t) < \infty$ i.e., we are forced to consider only functions of bounded first variation and further $\int_0^1 B_s(\omega)dB_s(\omega)$ does not exist in the Riemann-Stieljes sense. To deal with these problems the theory of Stochastic calculus is introduced.

3.2.1 Itô Integral

Let \mathcal{L}_2 be the space of all stochastic processes $\mathbf{X} = \{X_t, t \geq 0\}$ adapted to the Brownian motion filtration s.t. $E[\int_0^T X_s^2 ds] < \infty$ and the Riemann integral $\int_0^T X_s ds$ exists a.s., $\forall T > 0$. We first define the Itô Integral for simple processes in this space of stochastic processes and then extend the definition of the Itô Integral for any general process in this space.

Definition 3.2.2. A stochastic process $\mathbf{X} \in \mathcal{L}_2$ is called simple if there exists a countable partition $\{t_n\}_{n \in \mathbb{N} \cup \{0\}}$ s.t. $X_t(\omega) = X_{t_j}(\omega)$, $\forall t \in [t_j, t_{j+1})$ and $\forall \omega \in \Omega$.

The Itô Integral denoted $I_t(\mathbf{X})$ for such simple processes for $t_n \leq t < t_{n+1}$ is given by,

$$I_t(\mathbf{X}) = \int_0^t X_s dB_s = \sum_{j=0}^{n-1} X_{t_j} (B_{t_{j+1}} - B_{t_j}) + X_{t_n} (B_t - B_{t_n}). \quad (3.4)$$

The following important properties holds for the Itô Integral of simple processes.

Theorem 3.2.2. *The Itô Integral for simple processes satisfies the following:*

1. (**Linearity**) $I_t(a\mathbf{X} + b\mathbf{Y}) = aI_t(\mathbf{X}) + bI_t(\mathbf{Y})$.
2. (**Itô Isometry**) $E[(I_t(\mathbf{X}))^2] = E[\int_0^t X_s^2 ds]$.
3. $\{I_t(\mathbf{X}), t \geq 0\}$ is a martingale w.r.t. Brownian motion filtration.
4. $I_t(\mathbf{X})$ has continuous sample paths.
5. (**Quadratic variation**) $[I, I](t) = \int_0^t X_s^2 ds$.

Now given any general stochastic process in \mathcal{L}_2 it can be shown that it can be approximated by a sequence of simple processes in the sense defined in the below theorem.

Theorem 3.2.3. *Given any general stochastic process $\mathbf{X} \in \mathcal{L}_2$, \exists a sequence of simple processes $\mathbf{X}^n \in \mathcal{L}_2$ such that,*

$$\lim_{n \rightarrow \infty} E \left[\int_0^T (X_s^n - X_s)^2 ds \right] = 0. \quad (3.5)$$

For the detailed proof of above theorem refer to [Res92].

Theorem 3.2.4. *Let $\mathbf{X}^n \in \mathcal{L}_2$ be a simple process that satisfies the above theorem. Then there exists a stochastic process Z_t satisfying,*

$$\lim_{n \rightarrow \infty} E [(Z_t - I_t(X^n))^2] = 0, \forall t \in [0, T]. \quad (3.6)$$

The stochastic process Z_t is unique in the sense that if \hat{X}^n is another simple process satisfying the above theorem and let \hat{Z}_t be its limit in the above sense then,

$$\mathbb{P}(\hat{Z}_t = Z_t, \forall t \in [0, T]) = 1. \quad (3.7)$$

Theorem 3.2.5. *Let $\mathbf{X} \in \mathcal{L}_2$ be a general stochastic process then its Itô Integral $I_t(X)$ is defined as the unique stochastic process Z_t defined in the above theorem.*

Thus to compute $I_t(X)$ for general process $\mathbf{X} \in \mathcal{L}_2$ we first need to obtain a sequence of simple processes in \mathcal{L}_2 satisfying theorem 3.2.3 and then $I_t(X)$ is given by the limit of the Itô Integral of those simple processes in the mean square sense.

Theorem 3.2.6. *The Itô Integral for any general process in \mathcal{L}_2 satisfies the following:*

1. (**Linearity**) $I_t(a\mathbf{X} + b\mathbf{Y}) = aI_t(\mathbf{X}) + bI_t(\mathbf{Y})$.
2. (**Itô Isometry**) $E[(I_t(\mathbf{X}))^2] = E[\int_0^t X_s^2 ds]$.
3. $\{I_t(\mathbf{X}), t \geq 0\}$ is a martingale w.r.t. Brownian motion filtration.
4. $I_t(\mathbf{X})$ has continuous sample paths.
5. (**Quadratic variation**) $[I, I](t) = \int_0^t X_s^2 ds$.

Example 3.2.1. We will now compute $I_T(\mathbf{B}) = \int_0^T B_t dB_t$.

Consider the sequence of partitions $\pi_n := \{0 = t_0 < t_1 < \dots < t_n = T\}$ such that $|\pi_n| \xrightarrow{n \rightarrow \infty} 0$. Now consider the following process \mathbf{B}^n where $B_t^n = B_{t_j}, \forall t \in [t_j, t_{j+1})$.

Clearly \mathbf{B}^n is a sequence of simple processes such that,

$$\begin{aligned} E\left[\int_0^T (B_t - B_t^n)^2 dt\right] &= E\left[\sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} (B_t - B_{t_j})^2 dt\right] = \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} E[(B_t - B_{t_j})^2] dt \\ &= \sum_{j=0}^{n-1} \frac{(t_{j+1} - t_j)^2}{2} \\ &\leq \frac{1}{2} |\pi_n| T \xrightarrow{n \rightarrow \infty} 0. \end{aligned}$$

Thus the sequence of simple processes \mathbf{B}^n satisfies theorem 3.2.3 and so we need to compute the limit of $I_T(\mathbf{B}^n)$ in mean square sense.

Note: $I_T(\mathbf{B}^n) = \sum_{j=0}^{n-1} B_{t_j} (B_{t_{j+1}} - B_{t_j})$. Then we have that,

$$\begin{aligned} B^2(T) - B^2(0) &= \sum_{j=0}^{n-1} (B_{t_{j+1}}^2 - B_{t_j}^2) \\ \implies B^2(T) &= \sum_{j=0}^{n-1} (B_{t_{j+1}} - B_{t_j})^2 + 2 \sum_{j=0}^{n-1} B_{t_j} (B_{t_{j+1}} - B_{t_j}) \\ \implies I_T(\mathbf{B}^n) &= \frac{1}{2} B^2(T) - \frac{1}{2} \sum_{j=0}^{n-1} (B_{t_{j+1}} - B_{t_j})^2 \\ &\xrightarrow[n \rightarrow \infty]{L^2} \frac{1}{2} B^2(T) - \frac{T}{2}. \end{aligned}$$

Thus $I_T(\mathbf{B}) = \int_0^T B_t dB_t = \frac{1}{2}B^2(T) - \frac{T}{2}$.

We observe that for the Itô Integral the convention is that the simple processes are evaluated at the left end point of the partitions. One important property about the Stochastic Integral is that the integral actually depends on where the simple processes are evaluated as opposed to the Riemann-Stieljes approach where the choice of evaluation point of the integrand doesn't matter provided that the Riemann-Stieljes integral exists.

To demonstrate this consider again example 3.2.1 with the following simple processes \mathbf{B}^n where $B_t^n = B_{t_j^*}, \forall t \in [t_j, t_{j+1})$ where $t_j^* = \frac{t_{j+1} + t_j}{2}$. Then it can be shown that,

$$\sum_{j=0}^{n-1} B_{t_j^*} (B_{t_{j+1}} - B_{t_j}) \xrightarrow[n \rightarrow \infty]{L^2} \frac{1}{2} B^2(T).$$

The stochastic integral where the simple processes are evaluated at mid point is known as the **Stratonovich Integral** denoted by $\int_0^T B_t \circ dB_t = \frac{1}{2} B^2(T)$.

3.2.2 Itô formula

Itô formula is a analogous chain rule of differentiation for stochastic calculus. If f is a differentiable function then from the ordinary chain rule of calculus the differential $df(B_t) = f'(B_t)dB_t$ is no longer valid due to non-differentiability of Brownian motion sample paths and it has to be corrected by an additional term due to non-zero quadratic variation of Brownian motion sample paths which is given by the Itô formula. We first give the definition of Itô processes.

Definition 3.2.3. *A stochastic process \mathbf{X} is called Itô process if it can be written as:*

$$X_t = X_0 + \int_0^t a(s, X_s) ds + \int_0^t b(s, X_s) dB_s. \quad (3.8)$$

where both $a(s, X_s)$ and $b(s, X_s)$ are adapted to Brownian motion filtration. It is assumed here that the first integral exists in the Riemannian sense and second integral in the Itô sense.

We observe that Brownian motion is also a type of Itô process for the case when $a(s, X_s) = 0$ and $b(s, X_s) = 1$ in 3.8. It can be checked that the quadratic variation of Itô process is given by $[\mathbf{X}, \mathbf{X}](T) = \int_0^T (b(s, X_s))^2 ds$. For computational convenience we often write 3.8 in the following differential form:

$$dX_t = a(t, X_t)dt + b(t, X_t)dB_t. \quad (3.9)$$

Now using $(dB_t)^2 = dt$ and $(dt)^2 = dB_t dt = 0$ (in mean square sense) we will have,

$$(dX_t)^2 = \left(a(t, X_t)dt + b(t, X_t)dB_t \right) \left(a(t, X_t)dt + b(t, X_t)dB_t \right) = (b(t, X_t))^2 dt.$$

Theorem 3.2.7. Itô Formula for Ito Process Let X_t be an Ito process as defined in 3.8. Let $f(t, x)$ be a function whose partial derivatives $f_1(t, x)$, $f_2(t, x)$ and $f_{22}(t, x)$ exists and are continuous. Then we have:

$$f(t, X_t) - f(X_0, 0) = \int_0^t f_1(s, X_s)ds + \int_0^t f_2(s, X_s)dX_s + \frac{1}{2} \int_0^t f_{22}(s, X_s)(dX_s)^2. \quad (3.10)$$

We write 3.10 in the following differential form for computational purpose:

$$df(t, X_t) = f_1(t, X_t)dt + f_2(t, X_t)dX_t + \frac{1}{2}f_{22}(t, X_t)(dX_t)^2. \quad (3.11)$$

$$df(t, X_t) = \left[f_1(t, X_t) + a(t, X_t)f_2(t, X_t) + \frac{1}{2}b(t, X_t)^2 f_{22}(t, X_t) \right] dt + b(t, X_t)f_2(t, X_t)dB_t. \quad (3.12)$$

Itô Formula is used to solve Itô stochastic differential equations of the form 3.9. We illustrate it through an example of stochastic differential equation.

Example 3.2.2 (Geometric Brownian Motion). *The S.D.E. of Geometric Brownian Motion is obtained by considering $a(t, X_t) = \mu X_t$ and $b(t, X_t) = \sigma X_t$ in 3.9 where μ and $\sigma > 0$ are constants.*

$$dX_t = \mu X_t dt + \sigma X_t dB_t. \quad (3.13)$$

We consider the following function $f(t, x) = \log(x)$ for applying Itô's Formula. Now $f_1(t, x) = 0$, $f_2(t, x) = \frac{1}{x}$ and $f_{22}(t, x) = -\frac{1}{x^2}$. We have that all the required partial derivatives exist and are continuous then by 3.12 we have,

$$\begin{aligned} d\log(X_t) &= \left[\mu X_t \frac{1}{X_t} - \frac{1}{2}(\sigma X_t)^2 \frac{1}{(X_t)^2} \right] dt + \sigma X_t \frac{1}{X_t} dB_t \\ d\log(X_t) &= \left[\mu - \frac{1}{2}\sigma^2 \right] dt + \sigma dB_t \\ \log\left(\frac{X_t}{X_0}\right) &= \int_0^t \left(\mu - \frac{1}{2}\sigma^2 \right) ds + \int_0^t \sigma dB_s \\ \implies X_t &= X_0 e^{\sigma B_t + (\mu - \frac{1}{2}\sigma^2)t}. \end{aligned} \quad (3.14)$$

The stochastic process X_t in 3.14 is the solution of the S.D.E. 3.13 known as Geometric Brownian Motion and it is an important stochastic process for modelling stock prices.

Fig 3.1 shows simulation of some paths of geometric brownian motion.

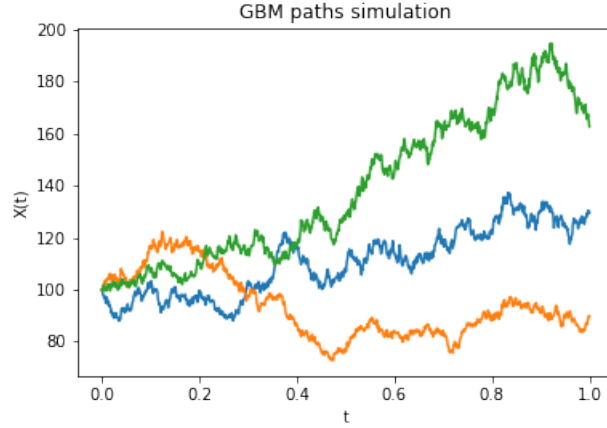


Figure 3.1: Simulation of GBM paths with $S_0 = 100$, $\mu = 0.2$ and $\sigma = 0.3$.

3.3 Black Scholes Model for Option pricing

We have covered the theory required to understand the mathematical framework of continuous time option pricing model. In this section we cover the benchmark option pricing model i.e., the Black Scholes Model. The following assumptions are to be made for the model:

- (1.) We only consider European options.
- (2.) We assume that there exists no arbitrage opportunities in the market.
- (3.) We also assume that the stock pays no dividend and there are no transaction costs.
- (4.) The Stock price follows lognormal distribution¹ i.e., the log-returns are normally distributed.
- (5.) The risk free interest rate r and volatility of asset σ are known and they are constant throughout the option's life.

We also have in our model that the risk free bank account β_t evolves as :

$$d\beta_t = r\beta_t dt. \quad (3.15)$$

We also assume that stock price process S_t evolves according to the geometric brownian motion:

¹This assumption is due to the efficient market hypothesis and an application of the Central limit theorem. For more detail refer to [Ros09]

$$dS_t = \mu S_t dt + \sigma S_t dB_t. \quad (3.16)$$

Let $C(t, S_t)$ be price of call option at time t with strike price K and maturity time T . We assume that all the partial derivatives of $C(t, S_t)$ required for the Itô Formula exist and are continuous. We will first derive the famous Black Scholes P.D.E. by using a replicating portfolio argument. We consider a portfolio of the risk free bank account and the stock with a_t (amount of shares in stock) and b_t (amount of shares in bank) such that both the stochastic process a_t and b_t are adapted to $\mathcal{F}_t, \forall t \in [0, T]$ where $\mathcal{F}_t = \sigma(B_s, s \leq t)$. Let $V(t, S_t)$ denote the value of the portfolio at time t then,

$$V_t = a_t S_t + b_t \beta_t. \quad (3.17)$$

Further we assume that the portfolio is self-financing i.e., the portfolio value evolves only due to the changes in the value of the stock and bank respectively and we do not inject any external money or take out any money from the portfolio during the process. Mathematically the self-financing condition is given by:

$$dV_t = d(a_t S_t + b_t \beta_t) = a_t dS_t + b_t d\beta_t. \quad (3.18)$$

Now for the replication argument we want the portfolio to match the price of the call option at maturity i.e., $V(T, S_T) = C(T, S_T) = (S_T - K)^+$. Note that under this condition we must have $V_t = C(t, S_t), \forall t \in [0, T]$ due to no-arbitrage assumption. Indeed if we have $V(t, S_t) < C(t, S_t)$ for any $t \in [0, T]$ then one can sell the call option and buy the portfolio at time t . This strategy is an arbitrage opportunity as it will yield a profit of $(C(t, S_t) - V(t, S_t))$ because at maturity we have that $C(T, S_T) = V(T, S_T)$ i.e., the call option payoff that has to be paid will be covered by the bought replicating portfolio at maturity. Similarly the other case $V(t, S_t) > C(t, S_t)$ will also result in arbitrage opportunity.

Now by the Itô formula we have that,

$$dC_t = \left[C_1(t, S_t) + \mu C_2(t, S_t) S_t + \frac{1}{2} \sigma^2 S_t^2 C_{22}(t, S_t) \right] dt + \sigma C_2(t, S_t) S_t dB_t. \quad (3.19)$$

We also have from 3.15 and 3.16,

$$dV_t = a_t dS_t + r b_t \beta_t dt = a_t \left[\mu S_t dt + \sigma S_t dB_t \right] + r \beta_t \left(\frac{V_t - a_t S_t}{\beta_t} \right) dt$$

$$= \left[\mu a_t S_t + r(V_t - a_t S_t) \right] dt + \sigma a_t S_t dB_t.$$

We have that $dV_t = dC_t, \forall t \in [0, T]$. Equating the dB_t coefficient gives,

$$a_t = \frac{\partial C(t, S_t)}{\partial S_t}. \quad (3.20)$$

Equating the drift terms and from 3.20 we get,

$$\frac{\partial C(t, S_t)}{\partial S_t} + \frac{(\sigma S_t)^2}{2} \frac{\partial^2 C(t, S_t)}{\partial S_t^2} + r S_t \frac{\partial C(t, S_t)}{\partial S_t} - r C(t, S_t) = 0. \quad (3.21)$$

This is the famous Black scholes P.D.E. for European call options with the boundary conditions $C(t, 0) = 0$, $C(T, S_T) = (S_T - K)^+$ and $C(t, S_t) \xrightarrow{S_t \rightarrow \infty} S_t$. The Black Scholes option pricing formula can be obtained from the above P.D.E. by transforming it into a heat equation and using Fourier transform technique to solve it. In this project the Black Scholes formula will be derived using the Equivalent Martingale Measure which is covered in next section.

3.3.1 Deriving the Black Scholes formula by Equivalent martingale measure technique

In this section we will derive the Black Scholes formula by the Equivalent martingale measure technique. The idea of the proof is to change the probability measure \mathbb{P} to another equivalent measure \mathbb{Q} such that under \mathbb{Q} we have the following:

(1.) $\{\tilde{B}_t = B_t + ct, 0 \leq t \leq T\}$ (Brownian motion with drift) becomes a standard Brownian motion under \mathbb{Q} .

(2.) $\{e^{-rt} S_t, 0 \leq t \leq T\}$ and $\{e^{-rt} V_t, 0 \leq t \leq T\}$ are martingales under \mathbb{Q} .

An Equivalent Martingale Measure (E.M.M.) is an equivalent measure to \mathbb{P} such that under it some stochastic process becomes a martingale. Girsanov's theorem is an important result which will give us the required equivalent measure \mathbb{Q} and once we have shown point (2.) then this new measure \mathbb{Q} will become the required E.M.M.

Theorem 3.3.1 (Girsanov's Theorem). *Suppose $\Theta(t)$, $0 \leq t \leq T$ be an adapted process to Brownian motion filtration. Define:*

$$Z(t) = e^{-\int_0^t \Theta(u) dB_u - \frac{1}{2} \int_0^t \Theta^2(u) du}.$$

$$\tilde{B}_t = B_t + \int_0^t \Theta(u) du.$$

Let $Z = Z(T)$, we define,

$$\mathbb{Q}(A) = \int_A Z d\mathbb{P}(\omega), \quad A \in \mathcal{F}.$$

Then,

1. $\mathbb{E}_{\mathbb{P}}[Z] = 1$ and \mathbb{Q} is an equivalent probability measure to \mathbb{P} .

2. \tilde{B}_t is a standard Brownian motion under \mathbb{Q} .

In our case we want $\tilde{B}_t = B_t + ct$ to be a standard Brownian motion where $c = \frac{\mu-r}{\sigma}$. Let $Z = e^{-cB_T - \frac{1}{2}c^2T}$ and define,

$$\mathbb{Q}(A) = \int_A Z(\omega) d\mathbb{P}(\omega), \quad A \in \mathcal{F}. \quad (3.22)$$

Then by Girsanov's theorem it follows that \mathbb{Q} is the required equivalent probability measure such that under it \tilde{B}_t becomes a standard Brownian motion.

Note: Under \mathbb{P} the stock price process is given by,

$$S_t = S_0 e^{\sigma B_t + (\mu - \frac{\sigma^2}{2})t}. \quad (3.23)$$

Under \mathbb{Q} the stock price process becomes,

$$\begin{aligned} S_t &= S_0 e^{\sigma \tilde{B}_t + (r - \frac{\sigma^2}{2})t} \\ \implies E_{\mathbb{Q}}[S_t] &= S_0 E_{\mathbb{Q}}[e^{\sigma \tilde{B}_t + (r - \frac{\sigma^2}{2})t}] = S_0 e^{rt}. \end{aligned} \quad (3.24)$$

Therefore under this measure \mathbb{Q} , the stock on an average behaves as the risk free bank account and that is why this Equivalent Martingale Measure is also known as the Risk Neutral Measure.

Now we will prove that $\{e^{-rt}S_t, 0 \leq t \leq T\}$ is martingale under \mathbb{Q} . Let $\tilde{S}_t = e^{-rt}S_t$ and consider $f(t, x) = e^{-rt}x$ we have that,

$$f_1(t, x) = -re^{-rt}x, f_2(t, x) = e^{-rt}, f_{22}(t, x) = 0.$$

Now apply the Itô Formula 3.11 for \tilde{S}_t ,

$$\begin{aligned}
d\tilde{S}_t &= -re^{-rt}S_tdt + e^{-rt}dS_t \\
&= \left[-re^{-rt}S_t + \mu S_te^{-rt} \right]dt + \sigma S_te^{-rt}dB_t \\
&= \tilde{S}_t \left[(\mu - r)dt + \sigma dB_t \right] \\
&= \sigma \tilde{S}_t d\tilde{B}_t. \\
\implies \tilde{S}_t - \tilde{S}_0 &= \int_0^t \sigma \tilde{S}_u d\tilde{B}_u.
\end{aligned}$$

The right hand side is an Itô integral under the measure \mathbb{Q} and which by theorem 3.2.6(3.) we know that it is a martingale. Therefore $\tilde{S}_t = e^{-rt}S_t$ is a martingale process under \mathbb{Q} .

Now we will prove that $\{e^{-rt}V_t, 0 \leq t \leq T\}$ is martingale under \mathbb{Q} . Let $\tilde{V}_t = e^{-rt}V_t$ now again apply the Itô's Formula for \tilde{V}_t ,

$$\begin{aligned}
d\tilde{V}_t &= -re^{-rt}V_tdt + e^{-rt}dV_t \\
&= e^{-rt}[-rV_tdt + a_t dS_t + b_t d\beta_t] \\
&= e^{-rt}[-ra_t S_t dt + \mu a_t S_t dt + \sigma a_t S_t dB_t] \\
&= a_t \tilde{S}_t [(\mu - r)dt + \sigma dB_t] \\
&= \sigma a_t \tilde{S}_t d\tilde{B}_t. \\
\implies \tilde{V}_t - \tilde{V}_0 &= \int_0^t \sigma a_u \tilde{S}_u d\tilde{B}_u.
\end{aligned}$$

The right hand side is a martingale under \mathbb{Q} and therefore $\tilde{V}_t = e^{-rt}V_t$ is a martingale under \mathbb{Q} . Now by martingale property for $t \in [0, T]$ we will have that,

$$e^{-rt}V_t = E_{\mathbb{Q}}[e^{-rT}V_T | \mathcal{F}_t] \quad (3.25)$$

$$V_t = E_{\mathbb{Q}}[e^{-r(T-t)}V_T | \mathcal{F}_t]. \quad (3.26)$$

We have that $V_t = C_t, \forall t \in [0, T]$ and $V_T = (S_T - K)^+$ this gives,

$$C(t, S_t) = E_{\mathbb{Q}}[e^{-r(T-t)}(S_T - K)^+ | \mathcal{F}_t]. \quad (3.27)$$

For $t = 0$ the above becomes,

$$\begin{aligned}
C(0, S_0) &= E_{\mathbb{Q}}[e^{-rT}(S_T - K)^+] \\
&= E_{\mathbb{Q}}[e^{-rT}(S_T - K)\mathbb{I}_{\{S_T > K\}}] \\
&= E_{\mathbb{Q}}[e^{-rT}S_T\mathbb{I}_{\{S_T > K\}}] - Ke^{-rt}E_{\mathbb{Q}}[\mathbb{I}_{\{S_T > K\}}] \\
&= E_{\mathbb{Q}}[e^{-rT}S_T\mathbb{I}_{\{S_T > K\}}] - Ke^{-rt}\mathbb{Q}(\{S_T > K\}).
\end{aligned}$$

Note: $S_T = S_0 e^{\sigma \widetilde{B}_T + (r - \frac{\sigma^2}{2})T} \stackrel{d}{=} S_0 e^{\sigma \sqrt{T}Z + (r - \frac{\sigma^2}{2})T}$ where Z is the standard normal random variable. Now,

$$\begin{aligned}
S_T > K &\iff S_0 e^{(r - \frac{1}{2}\sigma^2)T + \sigma\sqrt{T}Z} > K \\
&\iff Z > \frac{\log(K/S_0) - (r - \sigma^2/2)T}{\sigma\sqrt{T}} \\
&= \sigma\sqrt{T} - d_1,
\end{aligned}$$

where

$$d_1 = \frac{\log(\frac{S_0}{K}) + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}.$$

$$\begin{aligned}
E_{\mathbb{Q}}[e^{-rT}S_T\mathbb{I}_{\{S_T > K\}}] &= S_0 e^{-rT} E_{\mathbb{Q}}[e^{(r - \frac{1}{2}\sigma^2)T + \sigma\sqrt{T}Z} \mathbb{I}_{\{Z > \sigma\sqrt{T} - d_1\}}] \\
&= S_0 e^{-rT} \int_{\sigma\sqrt{T} - d_1}^{\infty} \left(e^{(r - \frac{1}{2}\sigma^2)T + \sigma\sqrt{T}z} \right) \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz \\
&= S_0 \frac{1}{\sqrt{2\pi}} \int_{\sigma\sqrt{T} - d_1}^{\infty} \left(e^{-\frac{1}{2}\sigma^2 T + \sigma\sqrt{T}z - \frac{z^2}{2}} \right) dz \\
&= S_0 \frac{1}{\sqrt{2\pi}} \int_{\sigma\sqrt{T} - d_1}^{\infty} e^{-\frac{(z - \sigma\sqrt{T})^2}{2}} dz \\
&= S_0 \frac{1}{\sqrt{2\pi}} \int_{-d_1}^{\infty} e^{-\frac{y^2}{2}} dy \quad (\text{Substitute } y = z - \sigma\sqrt{T}) \\
&= S_0(1 - \Phi(-d_1)) \\
&= S_0\Phi(d_1).
\end{aligned}$$

Now,

$$\begin{aligned}
\mathbb{Q}(\{S_T > K\}) &= \mathbb{Q}(\{Z > \sigma\sqrt{T} - d_1\}) \\
&= \mathbb{Q}(\{Z > -d_2\}) \\
&= 1 - \mathbb{Q}(\{Z \leq -d_2\}) \\
&= 1 - \Phi(-d_2) \\
&= \Phi(d_2),
\end{aligned}$$

where

$$d_2 = d_1 - \sigma\sqrt{T}.$$

Thus the Black Scholes call option price at $t = 0$ is given by,

$$C(0, S_0) = S_0\Phi(d_1) - Ke^{-rT}\Phi(d_2), \quad (3.28)$$

where

$$d_1 = \frac{\log(\frac{S_0}{K}) + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}, d_2 = d_1 - \sigma\sqrt{T}. \quad (3.29)$$

In general the Black Scholes call option price at any time $t \in [0, T]$ using 3.26 is given by,

$$C(t, S_t) = S_t\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2), \quad (3.30)$$

where

$$d_1 = \frac{\log(\frac{S_t}{K}) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}, d_2 = d_1 - \sigma\sqrt{T-t}. \quad (3.31)$$

Similarly we can price the put options using the Put-Call parity. First we prove the Put-Call parity using 3.26 and the result that the discounted stock price process is martingale under \mathbb{Q} . Let $P(t, S_t)$ denote the price of put option with strike price K and maturity time T . We observe that the following holds at $t = T$:

$$\begin{aligned}
&(S_T - K)^+ - (K - S_T)^+ = S_T - K \\
\implies E_{\mathbb{Q}}[e^{-r(T-t)} ((S_T - K)^+ - (K - S_T)^+) | \mathcal{F}_t] &= E_{\mathbb{Q}}[e^{-r(T-t)} (S_T - K) | \mathcal{F}_t] \\
\implies C(t, S_t) - P(t, S_t) &= S_t - Ke^{-r(T-t)} \textbf{(Put-Call parity)}.
\end{aligned}$$

Now the Black Scholes put option price at any time $t \in [0, T]$ using 3.30 is given by,

$$\begin{aligned} P(t, S_t) &= C(t, S_t) - S_t + Ke^{-r(T-t)} \\ &= S_t(\Phi(d_1) - 1) + Ke^{-r(T-t)}(1 - \Phi(d_2)) \\ &= Ke^{-r(T-t)}\Phi(-d_2) - S_t\Phi(-d_1), \end{aligned}$$

where d_1 and d_2 are given by 3.31.

3.3.2 Parameter estimation and some statistics of BS modelled returns

We need to estimate the parameters μ and σ of the Black Scholes model in order to fit an empirical data. Let R_{dt} denote the log-return of a stock during the period $[t, t + dt]$ then,

$$\begin{aligned} R_{dt} &= \log\left(\frac{S_{t+dt}}{S_t}\right) \\ &= \log\left(e^{\sigma(B_{t+dt}-B_t)+(\mu-\frac{1}{2}\sigma^2)dt}\right) \\ &= \sigma(B_{t+dt} - B_t) + (\mu - \frac{1}{2}\sigma^2)dt \\ &\stackrel{d}{=} \mathcal{N}\left((\mu - \frac{1}{2}\sigma^2)dt, \sigma^2dt\right). \end{aligned}$$

This shows that the PDF of the log-returns R_{dt} is given by,

$$f_{R_{dt}}(x) = \frac{1}{\sqrt{2\pi\sigma^2dt}} e^{\frac{(x-(\mu-\frac{1}{2}\sigma^2)dt)^2}{2\sigma^2dt}}. \quad (3.32)$$

Let $\{x_1, x_2, \dots, x_n\}$ be empirical log-returns. The Black Scholes model parameters can be estimated by using the Maximum Likelihood Estimate method which involves maximizing the following likelihood function:

$$L = \prod_{i=1}^n f_{R_{dt}}(x_i), \quad (3.33)$$

where the likelihood $f_{R_{dt}}(x_i)$ is given by 3.32. For optimisation purpose we maximise the log of the likelihood function which is equivalent to minimising the negative of the log of the likelihood function given by,

$$-\log(L) = -\sum_{i=1}^n \log(f_{R_{dt}}(x_i)). \quad (3.34)$$

The parameters μ and σ which minimises 3.34 are known as the M.L.E. estimates. Now for Normal distribution it can be shown that the sample mean and sample variance are the M.L.E. estimates for the mean and variance of the normal distribution. In our case we will have the following estimates,

$$\begin{aligned}(\mu - \frac{1}{2}\sigma^2)dt &= \tilde{E}[R_{dt}] = \frac{1}{n} \sum_{i=1}^n x_i. \\ \sigma^2 dt &= \tilde{Var}[R_{dt}] = \frac{1}{n} \sum_{i=1}^n (x_i - \tilde{E}[R_{dt}])^2.\end{aligned}$$

Therefore μ and σ are estimated by,

$$\begin{aligned}\mu &= \frac{\tilde{E}[R_{dt}] + \frac{1}{2}\tilde{Var}[R_{dt}]}{dt}. \\ \sigma &= \sqrt{\frac{\tilde{Var}[R_{dt}]}{dt}}.\end{aligned}\tag{3.35}$$

Now using $R_{dt} \stackrel{d}{=} \mathcal{N}((\mu - \frac{1}{2}\sigma^2)dt, \sigma^2 dt)$ the mean, variance, skewness and excess kurtosis of R_{dt} can be easily computed and are given by,

$$\begin{aligned}E[R_{dt}] &= (\mu - \frac{1}{2}\sigma^2)dt. \\ Var[R_{dt}] &= \sigma^2 dt. \\ Skew[R_{dt}] &= \frac{E[(R_{dt} - E[R_{dt}])^3]}{Var[R_{dt}]^{3/2}} = 0. \\ Excess\ Kurt[R_{dt}] &= \frac{E[(R_{dt} - E[R_{dt}])^4]}{Var[R_{dt}]^2} - 3 = 0.\end{aligned}\tag{3.36}$$

3.3.3 Drawbacks of the Black Scholes Model

Following are the major drawbacks of the Black Scholes Model:

- (1.) The assumption that the risk free rate r , the mean rate of return μ and the volatility σ is constant throughout the option's life is not valid in real life scenario. Infact all these parameters changes with respect to time and the phenomena of volatility smile or skew² suggests that σ cannot be assumed to be constant.
- (2.) The model also does not take in account the jumps in stock prices that are often seen in market.

²The volatility smile or skew will be discussed in Chapter 5

(3.) The model also fails to capture the skewness of the return distribution as the skewness of the BS modeled returns is zero in 3.36. Usually stocks return distribution is either skewed to the left or right rather than being symmetric.

(4.) In general the market stock returns tends to have non zero excess kurtosis (leptokurtic feature) which suggests that the market log-returns tends to have fatter tails than that implied by the log-normal distribution assumption of stock price in BS model. Therefore the BS model also fails to account for the heavy tails of the market stock returns distribution.

Chapter 4

Jump Diffusion Models for Option Pricing

In this chapter we will cover the jump diffusion models which incorporates the modeling of jumps that are seen in the stock prices. We will see in this chapter that the Jump diffusion models tries to cover some of the drawbacks of the Black Scholes model (namely asymmetric leptokurtic feature of the market stock returns). Theory in this chapter is covered mainly from [Mat17],[Tsa16], [CT15] and [Cam14].

4.1 Jump diffusion S.D.E. and its solution

In the jump diffusion models the stock price process follows the dynamics of a geometric brownian motion with some jumps imposed on it by using a compound poisson process and between the jump times the stock price follows geometric brownian motion process. We first illustrate the derivation of the S.D.E. in this case. Suppose that during the time increment dt stock price process S_t jumps from S_t to $V_t S_t$ where the random variable V_t measure the amount by which stock price gets multiplied by during a jump. Then the relative price change is given by,

$$\frac{dS_t}{S_t} = \frac{V_t S_t - S_t}{S_t} = V_t - 1.$$

Then by incorporating this relative price change in geometric brownian S.D.E. gives

$$\frac{dS_t}{S_t} = \mu dt + \sigma dB_t + (V_t - 1)dN_t \quad (4.1)$$

where N_t is the poisson process with intensity λ . The S.D.E. in 4.1 is used for the stock price process under Jump diffusion models. It is assumed that all sources of randomness

B_t, N_t and V_t are independent of each other. We will now derive the solution of S.D.E. 4.1 using the Itô formula for Jump diffusion processes ([CT15]) X_t of the form:

$$X_t = X_0 + \int_0^t b_s ds + \int_0^t \sigma_s dB_s + \sum_{i=1}^{N_t} \Delta X_i.$$

The Itô formula for sufficiently smooth function $f(t, X_t)$ is given by

$$\begin{aligned} df(t, X_t) = & f_1(t, X_t)dt + b_t f_2(t, X_t)dt + \frac{1}{2}(\sigma_t)^2 f_{22}(t, X_t)dt + \sigma_t f_2(t, X_t)dB_t \\ & + \left[f(X_{t-} + \Delta X_t) - f(X_{t-}) \right]. \end{aligned} \quad (4.2)$$

In our case from 4.1 we have $b_t = \mu S_t$, $\sigma_t = \sigma S_t$. Now for $f(t, S_t) = \log(S_t)$ we have $f_1(t, S_t) = 0$, $f_2(t, S_t) = \frac{1}{S_t}$ and $f_{22}(t, S_t) = -\frac{1}{S_t^2}$. Applying the Itô formula gives,

$$\begin{aligned} d\log(S_t) = & \mu S_t \frac{1}{S_t} dt - \frac{1}{2}(\sigma S_t)^2 \frac{1}{(S_t)^2} dt + \sigma S_t \frac{1}{S_t} dB_t + \left[\log(S_t + (V_t - 1)S_t) - \log(S_t) \right] \\ d\log(S_t) = & \left[\mu - \frac{1}{2}\sigma^2 \right] dt + \sigma dB_t + \log(V_t) \\ \log\left(\frac{S_t}{S_0}\right) = & \int_0^t \left(\mu - \frac{1}{2}\sigma^2 \right) dt + \int_0^t \sigma dB_t + \sum_{i=1}^{N_t} \log(V_i). \end{aligned}$$

Therefore the stock price process under Jump diffusion models is given by:

$$S_t = S_0 \exp \left\{ \left(\mu - \frac{1}{2}\sigma^2 \right) t + \sigma B_t \right\} \prod_{i=1}^{N_t} V_i. \quad (4.3)$$

Let $Y_i = \log(V_i)$ then the stock price process in 4.3 becomes,

$$S_t = S_0 \exp \left\{ \left(\mu - \frac{1}{2}\sigma^2 \right) t + \sigma B_t + \sum_{i=1}^{N_t} Y_i \right\}. \quad (4.4)$$

4.2 Merton Jump Diffusion Model

Robert C Merton ([Mer76]) proposes the jump sizes of the compound poisson process in 4.4 to be normally distributed. More specifically $Y_i \stackrel{d}{=} \mathcal{N}(m, \delta^2)$. Then the mean and variance of V_i in 4.3 are given by,

$$\mathbb{E}[V_i] = \mathbb{E}[e^{Y_i}] = e^{m + \frac{\delta^2}{2}}.$$

$$\text{Var}[V_i] = \mathbb{E}[(V_i - \mathbb{E}[V_i])^2] = e^{2m + 2\delta^2}.$$

Fig 4.1 shows paths simulations of stock price process under Merton Jump diffusion model.

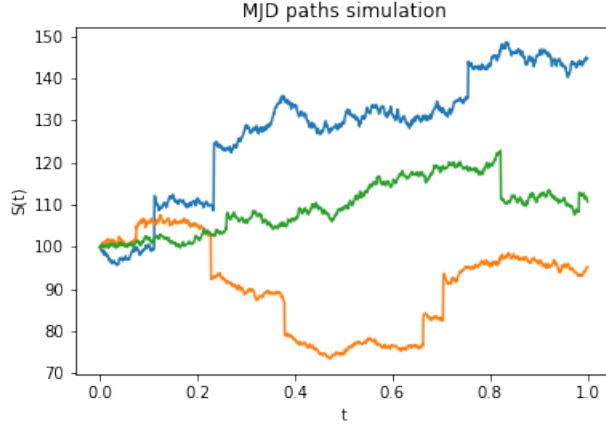


Figure 4.1: Simulation of MJD paths with $S_0 = 100$, $\mu = 0.01$, $\sigma = 0.1$, $\lambda = 5$, $m = 0$ and $\delta = 0.1$.

Now when the time increment dt is small enough the distribution of (arithmetic) returns can be approximately computed using $e^x \approx 1 + x + \frac{x^2}{2}$, $(dB_t)^2 = dt$ and ignoring terms of order higher than dt (in mean square sense). We illustrate it as follows:

$$\begin{aligned}
R_{dt} &= \frac{dS_t}{S_t} = \frac{S_{t+dt}}{S_t} - 1 \\
&= \frac{e^{((\mu - \frac{1}{2}\sigma^2)(t+dt) + \sigma B_{t+dt} + \sum_{i=1}^{N_{t+dt}} Y_i)}}{e^{((\mu - \frac{1}{2}\sigma^2)t + \sigma B_t + \sum_{i=1}^{N_t} Y_i)}} - 1 \\
&= e^{((\mu - \frac{1}{2}\sigma^2)dt + \sigma dB_t + \sum_{i=N_t+1}^{N_{t+dt}} Y_i)} - 1 \\
&\approx (\mu - \frac{1}{2}\sigma^2)dt + \sigma dB_t + \sum_{i=N_t+1}^{N_{t+dt}} Y_i + \frac{1}{2} \left(\left[(\mu - \frac{1}{2}\sigma^2)dt \right]^2 \right. \\
&\quad \left. + 2\left(\mu - \frac{1}{2}\sigma^2\right) \left[\sigma dB_t + \sum_{i=N_t+1}^{N_{t+dt}} Y_i \right] dt + \left[\sigma dB_t + \sum_{i=N_t+1}^{N_{t+dt}} Y_i \right]^2 \right) \\
&\approx (\mu - \frac{1}{2}\sigma^2)dt + \sigma dB_t + \sum_{i=N_t+1}^{N_{t+dt}} Y_i + \frac{1}{2} [\sigma (dB_t)^2] \\
&= \mu dt + \sigma dB_t + \sum_{i=N_t+1}^{N_{t+dt}} Y_i.
\end{aligned}$$

In the above expansion the terms with order higher than dt (in the mean square sense) has been dropped off. Therefore the returns R_{dt} approximately follows the distribution

given by:

$$\begin{aligned}
R_{dt} &= \mu dt + \sigma dB_t + \sum_{i=N_t+1}^{N_{t+dt}} Y_i \\
&\stackrel{d}{=} \mu dt + \sigma dB_t + \sum_{i=1}^{dN_t} Y_i.
\end{aligned} \tag{4.5}$$

In above we have that $dN_t = N_{t+dt} - N_t$. Now since dt is small enough using definition of poisson process (2.4) the term dN_t either produces jump of size 1 with probability λdt or no jump with probability $1 - \lambda dt$. Therefore we have that

$$R_{dt} \stackrel{d}{=} \mu dt + \sigma \sqrt{dt} Z + BY. \tag{4.6}$$

where Z is standard normal random variable, $Y \stackrel{d}{=} \mathcal{N}(m, \delta^2)$ and B is Bernoulli like random variable with $\mathbb{P}(B = 1) = \lambda dt$ and $\mathbb{P}(B = 0) = 1 - \lambda dt$. Now on conditional on the event that $B = 1$ and using the independence of Z and Y we have:

$$R_{dt} \stackrel{d}{=} \mu dt + \sigma \sqrt{dt} Z + Y \stackrel{d}{=} \mathcal{N}(\mu dt + m, \sigma^2 dt + \delta^2). \tag{4.7}$$

Similarly conditional on the event that $B = 0$ we have:

$$R_{dt} \stackrel{d}{=} \mu dt + \sigma \sqrt{dt} Z \stackrel{d}{=} \mathcal{N}(\mu dt, \sigma^2 dt). \tag{4.8}$$

Then the C.D.F. of R_{dt} is given by:

$$\begin{aligned}
F_{R_{dt}}(x) &= \mathbb{P}(R_{dt} \leq x) = \mathbb{P}(R_{dt} \leq x | B = 1) \mathbb{P}(B = 1) + \mathbb{P}(R_{dt} \leq x | B = 0) \mathbb{P}(B = 0) \\
&= \lambda dt \Phi\left(\frac{x - (\mu dt + m)}{\sqrt{\sigma^2 dt + \delta^2}}\right) + (1 - \lambda dt) \Phi\left(\frac{x - \mu dt}{\sigma \sqrt{dt}}\right).
\end{aligned}$$

This shows that the P.D.F. of returns under Merton jump diffusion model is given by,

$$f_{R_{dt}}(x) = \frac{\lambda dt}{\sqrt{\sigma^2 dt + \delta^2}} \phi\left(\frac{x - (\mu dt + m)}{\sqrt{\sigma^2 dt + \delta^2}}\right) + \frac{(1 - \lambda dt)}{\sigma \sqrt{dt}} \phi\left(\frac{x - \mu dt}{\sigma \sqrt{dt}}\right). \tag{4.9}$$

where Φ and ϕ are the C.D.F. and P.D.F. respectively of standard normal random variable. The below theorem gives the first four central moments of the returns.

Theorem 4.2.1 (Theorem 5.17, [Han07]). *Let R_{dt} be the returns that satisfy 4.5 and for convenience let $Y \stackrel{d}{=} Y_i$. Let $\mu_j = \mathbb{E}[Y]$ and $\sigma_j^2 = \text{Var}[Y]$ then the first four central moments are given by,*

$$\begin{aligned}
E[R_{dt}] &= (\mu + \mu_j \lambda) dt. \\
E[(R_{dt} - E[R_{dt}])^2] &= (\sigma^2 + (\sigma_j^2 + \mu_j^2) \lambda) dt. \\
E[(R_{dt} - E[R_{dt}])^3] &= \left(E[(Y - \mu_j)^3] + (3\sigma_j^2 + \mu_j^2) \mu_j \right) \lambda dt. \\
E[(R_{dt} - E[R_{dt}])^4] &= \left(E[(Y - \mu_j)^4] + 4\mu_j E[(Y - \mu_j)^3] + 6\mu_j^2 \sigma_j^2 + \mu_j^4 \right) \lambda dt \\
&\quad + 3(\sigma^2 + (\sigma_j^2 + \mu_j^2) \lambda)^2 (dt)^2.
\end{aligned} \tag{4.10}$$

The above third and fourth central moments can be further simplified and are given by,

$$\begin{aligned}
E[(R_{dt} - E[R_{dt}])^3] &= (E[Y^3]) \lambda dt. \\
E[(R_{dt} - E[R_{dt}])^4] &= (E[Y^4]) \lambda dt + 3(\sigma^2 + (\sigma_j^2 + \mu_j^2) \lambda)^2 (dt)^2.
\end{aligned} \tag{4.11}$$

The mean, variance, skewness and excess kurtosis of the Merton jump diffusion modeled returns using 4.10 are given by,

$$\begin{aligned}
E[R_{dt}] &= \mu dt + m \lambda dt. \\
\text{Var}[R_{dt}] &= \sigma^2 dt + (\delta^2 + m^2) \lambda dt. \\
\text{Skew}[R_{dt}] &= \frac{E[(R_{dt} - E[R_{dt}])^3]}{(\text{Var}[R_{dt}])^{\frac{3}{2}}} \\
&= \frac{(3\delta^2 + m^2) m \lambda dt}{(\sigma^2 dt + (\delta^2 + m^2) \lambda dt)^{3/2}}. \\
\text{Excess Kurt}[R_{dt}] &= \frac{E[(R_{dt} - E[R_{dt}])^4]}{(\text{Var}[R_{dt}])^2} - 3 \\
&= \frac{(3\delta^4 + 6m^2 \delta^2 + m^4) \lambda dt}{(\sigma^2 dt + (\delta^2 + m^2) \lambda dt)^2}.
\end{aligned} \tag{4.12}$$

It is to be noted from the skewness in 4.12 that Merton jump diffusion model can capture the positive or negative skewness of the returns depending on the sign of m . We also note that the excess kurtosis of the model returns is non-zero provided $\lambda > 0$. Thus the model also takes into account the heaviness of the tail of return distributions and a high value of λ implies that the excess kurtosis will also be higher. Therefore the model tries to overcome some of the drawbacks that the Black Scholes model faces.

4.3 Kou Jump Diffusion Model

S.G. Kou ([Kou02]) proposes the jump sizes of the compound poisson process in 4.4 to be given by a double exponential distribution. The jump size follows the distribution:

$$Y \stackrel{d}{=} \begin{cases} J^+ & \text{with probability } p \\ -J^- & \text{with probability } q, \end{cases}$$

where $J^+ \stackrel{d}{=} \exp(\eta_1)$ and $J^- \stackrel{d}{=} \exp(\eta_2)$ where the former accounts for positive jumps with p probability and the latter accounts for negative jumps with q probability. The density of Y is given by,

$$f_Y(y) = p \eta_1 e^{-\eta_1 y} \mathbb{I}_{\{y \geq 0\}} + q \eta_2 e^{\eta_2 y} \mathbb{I}_{\{y < 0\}}, \quad \eta_1 > 1, \eta_2 > 0. \quad (4.13)$$

Now using 4.13 the following moments can be obtained. The condition $\eta_1 > 1$ is required for $E[V] < \infty$.

$$\begin{aligned} E[Y] &= \frac{p}{\eta_1} - \frac{q}{\eta_2}. \\ \text{Var}[Y] &= pq \left(\frac{1}{\eta_1} + \frac{1}{\eta_2} \right)^2 + \left(\frac{p}{\eta_1^2} + \frac{q}{\eta_2^2} \right). \\ E[V] &= E[e^Y] \\ &= p \frac{\eta_1}{\eta_1 - 1} + q \frac{\eta_2}{\eta_2 + 1}. \end{aligned} \quad (4.14)$$

Fig 4.1 shows simulation of paths of stock price process under Kou Jump Diffusion model.

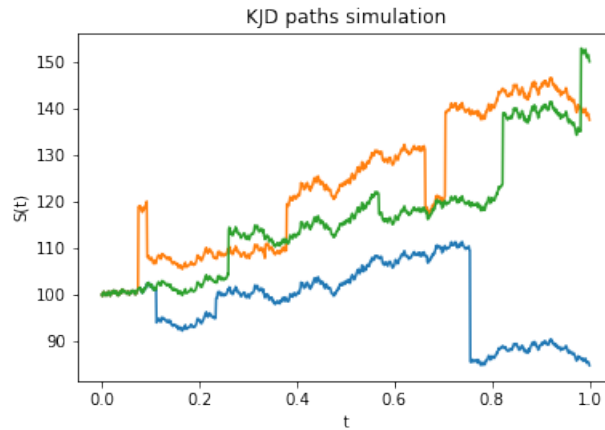


Figure 4.2: Simulation of KJD paths with $S_0 = 100, \mu = 0.01, \sigma = 0.1, \lambda = 5, p = 0.5, \eta_1 = 10$ and $\eta_2 = 10$.

Now using the distributional approximation $R_{dt} \stackrel{d}{=} \mu dt + \sigma \sqrt{dt} Z + BY$ we will derive the density of the returns. Similar to the previous case when conditional on the event $B = 0$, the returns are distributed as $R_{dt} \stackrel{d}{=} \mu dt + \sigma \sqrt{dt} Z$ which implies that the density of returns in this case is given by $\frac{1-\lambda dt}{\sigma \sqrt{dt}} \phi\left(\frac{x-\mu dt}{\sigma \sqrt{dt}}\right)$. Now conditional on the event $B = 1$, we have that the returns are distributed as $R_{dt} \stackrel{d}{=} \mu dt + \sigma \sqrt{dt} Z + Y \stackrel{d}{=} N + Y$ which is the sum of independent normal random variable (N) and a double exponentially distributed random variable(Y) respectively where $N \stackrel{d}{=} \mathcal{N}(\mu dt, \sigma^2 dt)$. Then the density of $N + Y$ is given by the convolution of the densities of N and Y respectively we illustrate it as follows:

$$\begin{aligned}
f_{N+Y}(x) &= (f_N * f_Y)(x) \\
&= \int_{-\infty}^{\infty} f_N(x-y) f_Y(y) dy \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2 dt}} e^{-\frac{(x-y-\mu dt)^2}{2\sigma^2 dt}} \left[p \eta_1 e^{-\eta_1 y} \mathbb{I}_{\{y \geq 0\}} + q \eta_2 e^{\eta_2 y} \mathbb{I}_{\{y < 0\}} \right] dy \\
&= p \eta_1 \int_0^{\infty} \frac{1}{\sqrt{2\pi\sigma^2 dt}} e^{-\frac{-y^2 + 2y(x-\mu dt) - (x-\mu dt)^2 - 2\eta_1 y \sigma^2 dt}{2\sigma^2 dt}} dy \\
&\quad + q \eta_2 \int_{-\infty}^0 \frac{1}{\sqrt{2\pi\sigma^2 dt}} e^{-\frac{-y^2 + 2y(x-\mu dt) - (x-\mu dt)^2 + 2\eta_2 y \sigma^2 dt}{2\sigma^2 dt}} dy \\
&= p \eta_1 e^{-\frac{(x-\mu dt)^2}{2\sigma^2 dt}} \int_0^{\infty} \frac{1}{\sqrt{2\pi\sigma^2 dt}} e^{-\frac{-y^2 + 2y(x-\mu dt - \eta_1 \sigma^2 dt)}{2\sigma^2 dt}} dy \\
&\quad + q \eta_2 e^{-\frac{(x-\mu dt)^2}{2\sigma^2 dt}} \int_{-\infty}^0 \frac{1}{\sqrt{2\pi\sigma^2 dt}} e^{-\frac{-y^2 + 2y(x-\mu dt + \eta_2 \sigma^2 dt)}{2\sigma^2 dt}} dy \\
&= p \eta_1 e^{-\frac{(x-\mu dt)^2}{2\sigma^2 dt}} e^{\frac{(x-\mu dt - \eta_1 \sigma^2 dt)^2}{2\sigma^2 dt}} \int_0^{\infty} \frac{1}{\sqrt{2\pi\sigma^2 dt}} e^{-\frac{(y - (x-\mu dt - \eta_1 \sigma^2 dt))^2}{2\sigma^2 dt}} dy \\
&\quad + q \eta_2 e^{-\frac{(x-\mu dt)^2}{2\sigma^2 dt}} e^{\frac{(x-\mu dt + \eta_2 \sigma^2 dt)^2}{2\sigma^2 dt}} \int_{-\infty}^0 \frac{1}{\sqrt{2\pi\sigma^2 dt}} e^{-\frac{(y - (x-\mu dt + \eta_2 \sigma^2 dt))^2}{2\sigma^2 dt}} dy \\
&= p \eta_1 e^{\frac{\sigma^2 \eta_1^2 dt}{2}} e^{-(x-\mu dt)\eta_1} \Phi\left(\frac{x - \mu dt - \sigma^2 \eta_1 dt}{\sigma \sqrt{dt}}\right) \\
&\quad + q \eta_2 e^{\frac{\sigma^2 \eta_2^2 dt}{2}} e^{(x-\mu dt)\eta_2} \Phi\left(\frac{-(x - \mu dt + \sigma^2 \eta_2 dt)}{\sigma \sqrt{dt}}\right).
\end{aligned}$$

Therefore the density of returns for Kou Jump Diffusion model is given by:

$$f_{R_{dt}}(x) = \frac{1 - \lambda dt}{\sigma\sqrt{dt}} \phi\left(\frac{x - \mu dt}{\sigma\sqrt{dt}}\right) + \lambda dt \left\{ p\eta_1 e^{\frac{\sigma^2\eta_1^2 dt}{2}} e^{-(x - \mu dt)\eta_1} \Phi\left(\frac{x - \mu dt - \sigma^2\eta_1 dt}{\sigma\sqrt{dt}}\right) + q\eta_2 e^{\frac{\sigma^2\eta_2^2 dt}{2}} e^{(x - \mu dt)\eta_2} \Phi\left(\frac{-(x - \mu dt + \sigma^2\eta_2 dt)}{\sigma\sqrt{dt}}\right) \right\}. \quad (4.15)$$

where ϕ and Φ are the P.D.F. and C.D.F. of standard normal random variable respectively.

The mean, variance, skewness and excess kurtosis of the Kou jump diffusion modeled returns using 4.10 and 4.11 are given by,

$$\begin{aligned} E[R_{dt}] &= \mu dt + \left(\frac{p}{\eta_1} - \frac{q}{\eta_2}\right) \lambda dt. \\ \text{Var}[R_{dt}] &= \sigma^2 dt + \left\{ pq \left(\frac{1}{\eta_1} + \frac{1}{\eta_2}\right)^2 + \left(\frac{p}{\eta_1^2} + \frac{q}{\eta_2^2}\right) \right\} \lambda dt \\ &\quad + \left(\frac{p}{\eta_1} - \frac{q}{\eta_2}\right)^2 \lambda dt. \\ \text{Skew}[R_{dt}] &= \frac{E[(R_{dt} - E[R_{dt}])^3]}{(\text{Var}[R_{dt}])^{\frac{3}{2}}} \\ &= \frac{6 \left(\frac{p}{\eta_1^3} - \frac{q}{\eta_2^3}\right) \lambda dt}{(\text{Var}[R_{dt}])^{\frac{3}{2}}}. \\ \text{Excess Kurt}[R_{dt}] &= \frac{E[(R_{dt} - E[R_{dt}])^4]}{(\text{Var}[R_{dt}])^2} - 3 \\ &= \frac{24 \left(\frac{p}{\eta_1^4} + \frac{q}{\eta_2^4}\right) \lambda dt}{(\text{Var}[R_{dt}])^2}. \end{aligned} \quad (4.16)$$

From 4.16 it can be noted that the Kou Jump diffusion model also tries to capture the non-zero skewness of the returns distribution and the non-zero excess kurtosis (provided $\lambda > 0$) suggests that the model also tries to capture the heaviness of the tails of the returns distribution.

4.4 Option pricing under Jump Diffusion Models and parameter estimation

We know from section 4.1 that the stock price process under jump diffusion model is given by,

$$S_t = S_0 \exp \left\{ \left(\mu - \frac{1}{2} \sigma^2 \right) t + \sigma B_t \right\} \prod_{i=1}^{N_t} V_i. \quad (4.17)$$

We first compute $E[\prod_{i=1}^{N(t)} V_i]$ as follows:

$$\begin{aligned} E\left[\prod_{i=1}^{N(t)} V_i\right] &= E\left[E\left[\prod_{i=1}^{N(t)} V_i \mid N(t)\right]\right] = \sum_{n=0}^{\infty} E\left[\prod_{i=1}^{N(t)} V_i \mid N(t) = n\right] \cdot \mathbb{P}(N(t) = n) \\ &= \sum_{n=0}^{\infty} E\left[\prod_{i=1}^n V_i\right] \cdot \frac{e^{-\lambda t} (\lambda t)^n}{n!} \\ &= e^{-\lambda t} \sum_{n=0}^{\infty} (E[V_i])^n \cdot \frac{(\lambda t)^n}{n!} \\ &= e^{-\lambda t (1 - E[V_i])}. \end{aligned}$$

Now for pricing the options we want $\{e^{-rt} S_t, t > 0\}$ to be a martingale. We will find the condition under which the discounted stock price process in 4.17 becomes a martingale. Suppose $\{e^{-rt} S_t, t > 0\}$ is a martingale then it must have constant expectation which implies that

$$\begin{aligned} E[e^{-rt} S_t] = S_0 &\iff e^{-rt} S_0 e^{\mu t} e^{-\lambda t (1 - E[V_i])} = S_0 \\ &\iff e^{(\mu - r - \lambda(1 - E[V_i]))t} = 1 \\ &\iff \mu = r - \lambda(E[V_i] - 1). \end{aligned}$$

We denote $E[V_i - 1]$ by k . Therefore the risk neutral stock price process is given by,

$$S_t = S_0 \exp \left\{ \left(r - \lambda k - \frac{1}{2} \sigma^2 \right) t + \sigma B_t \right\} \prod_{i=1}^{N_t} V_i. \quad (4.18)$$

Let K be the strike price and T be the maturity time of call option. Then the call option price at $t=0$ under the jump diffusion models is given by,

$$C(0, S_0) = E[e^{-rT} (S_T - K)^+], \quad (4.19)$$

where S_T is given by the risk neutral stock price process 4.18.

Now for parameter estimation we have used the M.L.E. method. Let $\{x_1, x_2, \dots, x_n\}$ be the empirical (arithmetic) returns then the parameters of the models will be estimated by maximising the likelihood function,

$$L = \prod_{i=1}^n f_{R_{dt}}(x_i). \quad (4.20)$$

For Merton jump diffusion model the density in 4.9 will be used for the likelihood and similarly for Kou Jump diffusion model the density in 4.15 will be used for the likelihood.

Further in this case the initial estimates of the parameters are required for optimisation purpose. To deal with this we have used the methodology and approach discussed in [Tan18].

4.5 Application of the Black Scholes and the Jump Diffusion models for Infosys stock

In this section we will cover the application of the Black scholes, Merton jump diffusion and Kou jump diffusion models for Infosys stock. We will be using around 8 years of historical data (01/01/2014 - 21/04/2022) of closing prices of Infosys stock ¹. Fig 4.3 shows daily closing price of the Infosys stock from 2014-01-01.

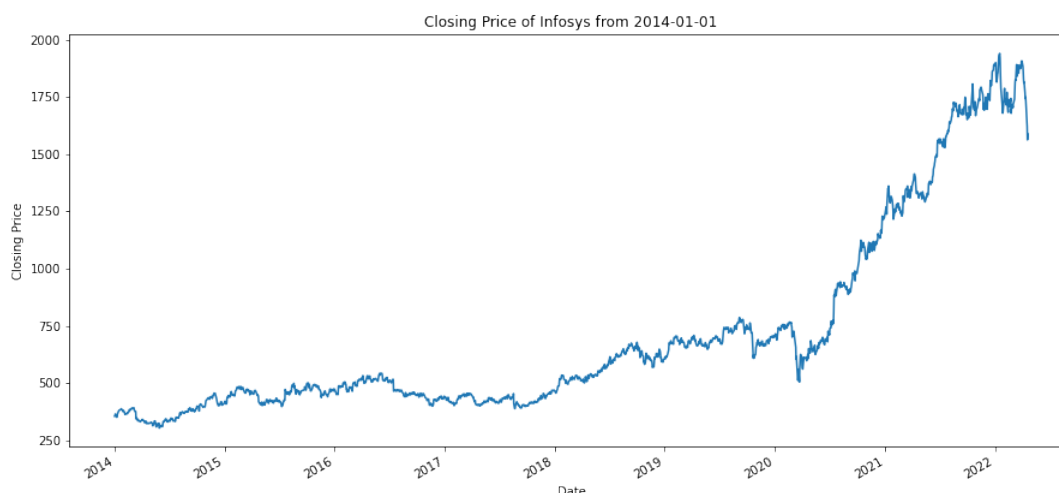


Figure 4.3: Daily closing price of Infosys stock.

4.5.1 Parameter estimation and comparative study of the models

In this section we will estimate the parameters of BS, MJD and KJD models respectively. After estimation process we will then do a comparative study to conclude which model fits better to the Infosys stock. Fig 5.4 shows the daily (arithmetic) returns of Infosys stock from 2014-01-01.

¹Historical data of closing price of Infosys has been obtained from <https://finance.yahoo.com>

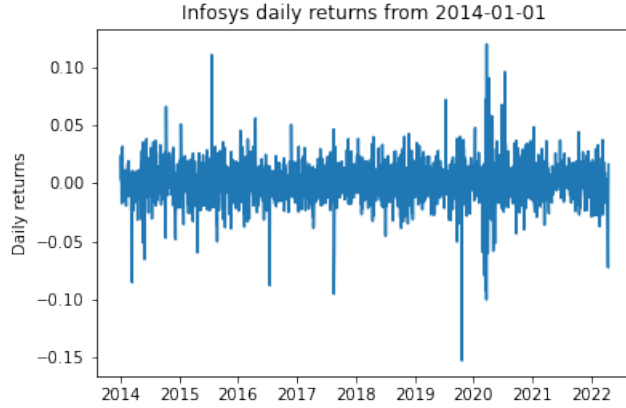


Figure 4.4: Daily returns of Infosys stock.

Table 4.1 shows some statistics of the daily returns of Infosys stock. Negative value of skewness suggests that the distribution of returns is skewed to left side rather than being symmetric. High kurtosis of returns suggests that the distribution of returns have heavier tails which means that the tail of the returns distribution dies out slowly.

Mean	Variance	Skewness	Kurtosis
0.000881	0.000292	-0.299524	11.922975

Table 4.1: Some statistics of daily returns of Infosys stock.

We now use the M.L.E. method discussed in section 3.3.2 to estimate the BS model parameters. Table 4.2 shows the estimated μ and σ .

μ	σ
0.258676	0.271156

Table 4.2: BS estimated parameters.

Table 4.3 and 4.4 shows the estimated parameters of the MJD and the KJD models respectively using the M.L.E. method discussed in section 4.4

μ	σ	λ	m	δ
0.253644	0.205033	18.628368	-0.001703	0.041080

Table 4.3: MJD estimated parameters through M.L.E.

μ	σ	λ	p	η_1	η_2
0.106614	0.190477	49.741324	0.630307	59.434328	44.687403

Table 4.4: KJD estimated parameters through M.L.E.

Then using the estimated parameters we then generated simulations of returns from each of the models. Fig 4.5 shows the plots of the simulated returns from each model and its comparison with the empirical returns.

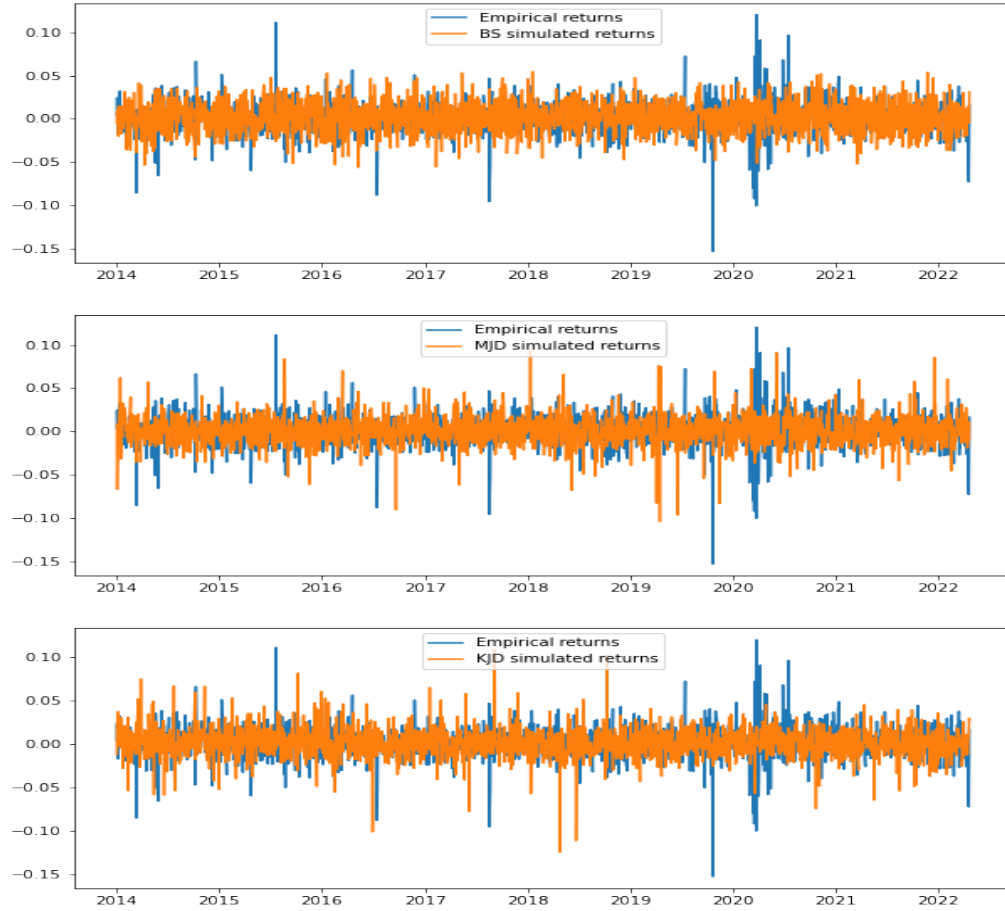


Figure 4.5: Comparison of the empirical returns with the simulated returns of the BS, MJD and KJD models respectively.

From fig 4.5 we can observe that the BS model fails to capture the occasional spikes(jump points) that are present in the empirical returns. The MJD and KJD models on the other hand show some occasional spikes in the plots indicating that they can model the occasional jumps of Infosys stock returns. We then covered the distributional comparison of the empirical returns with that of the returns modeled from each of the three models. Fig 4.6 shows the Kernel Density Estimate plot of the empirical returns and the simulated returns of fig 4.5 from each models.

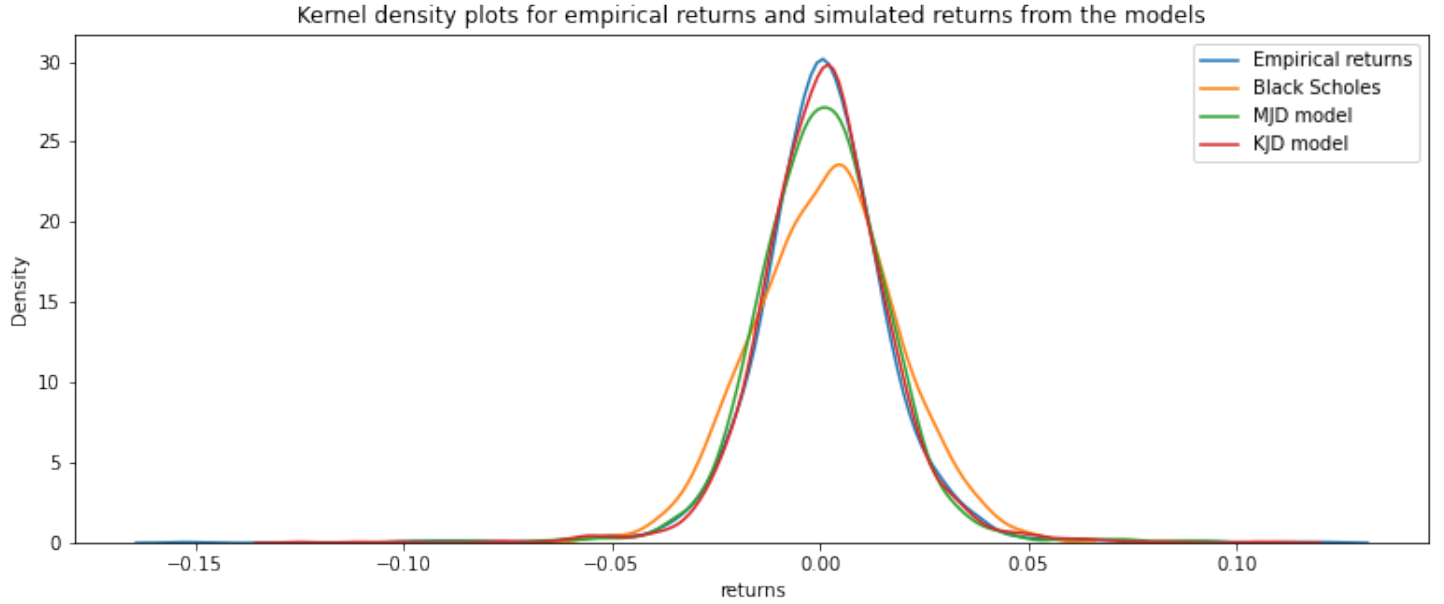


Figure 4.6: Comparison of the kernel density estimate plots of empirical returns with the BS, MJD and KJD models simulated returns respectively.

The K.D.E. plot suggests that the density of MJD and the KJD simulated returns fits better to that of the empirical returns as compared to the BS simulated returns density. It can be observed that the KJD model better captures the peak of the empirical returns distribution and also the tails of the KJD distribution is heavier than the other models. We then computed some statistics implied by each of the three models using 3.36, 4.12 and 4.16. Table 4.5 shows the computed statistics and its comparison with that of empirical returns.

	Mean	Variance	Skewness	Kurtosis
Empirical Returns	0.000881	0.000292	-0.299524	11.922975
BS Model	0.000881	0.000292	0	3
MJD Model	0.000881	0.000292	-0.127939	10.443763
KJD Model	0.000883	0.000287	-0.277089	11.208304

Table 4.5: Comparison of some statistics of the BS, MJD and KJD modelled returns respectively with the empirical returns.

It can be noted through the kurtosis values that the MJD and the KJD captures the heaviness of the tail of the distribution of returns and the heaviness of tails is more pronounced in the case of KJD model which the BS model fails to capture. Similarly both the MJD and KJD model respectively captures the negative skewness of the Infosys returns and the value of skewness of KJD model is more close to that of the Infosys returns as compared to the MJD model. Therefore comparatively we can conclude that the KJD model is a better fit for the Infosys stock data.

4.5.2 Pricing Infosys Call options under the BS and the Jump diffusion models²

In this section we will be pricing some Infosys call options dated 21/04/2022 using the estimated BS, MJD and KJD model of the previous section. For BS model the closed form pricing formula 3.28 will be used. The call price under MJD and KJD model using 4.19 and the central limit theorem can be approximated by,

$$Price = e^{-rT} \left(\frac{\sum_{i=1}^N (S_T - K)^+}{N} \right). \quad (4.21)$$

where S_T follows the risk neutral equity process as mentioned in 4.18. We simulated N=100000 values of S_T and computed the call option prices for MJD and KJD model using 4.21. The initial price S_0 was 1615.70 rupees on 21-04-2022 with expiry on 26-05-2022. We have taken $r = 0.045$ which is the current 1-Year Government Bond Yield. Table 4.6 shows the computed option prices of the three models for varying strike prices of Infosys call options. We can note that all the three models performs fine in terms of pricing and the prices are close to the MID PRICES.

²Call options data of Infosys has been obtained from <https://www.nseindia.com/option-chain>

STRIKE PRICE	BID PRICE	ASK PRICE	MID PRICE	BS PRICE	MJD PRICE	KJD PRICE
1500.0	132.7	135.85	134.275	134.40275	134.8983	134.27881
1520.0	111.55	121.3	116.425	118.62497	119.20429	118.17841
1540.0	101.05	103.5	102.275	103.79511	102.97129	102.97846
1560.0	87.7	88.75	88.225	90.0016	88.9934	90.07431
1580.0	73.9	75.35	74.625	77.31336	76.58405	76.77823
1600.0	61.8	62.85	62.325	65.77603	65.53064	64.64055
1620.0	52.1	52.5	52.3	55.40992	54.75655	54.19889
1640.0	42.55	43.0	42.775	46.20955	45.66211	45.45656
1660.0	34.65	35.4	35.025	38.14495	37.53703	37.3841
1680.0	28.0	28.8	28.4	31.16438	30.70015	30.18425
1700.0	22.75	22.9	22.825	25.19812	24.85551	24.33266
1720.0	18.25	18.75	18.5	20.16294	19.81938	19.51125
1740.0	14.55	14.9	14.725	15.96678	15.77687	15.12316
1760.0	12.0	12.35	12.175	12.51337	12.44429	12.09165

Table 4.6: Computed Option prices of Infosys call options under BS, MJD and KJD models respectively.

Note: Since we are relying on simulations for pricing in case of MJD and KJD model using 4.21. The prices under the next scenario of simulations might change but not by an insignificant amount.

Chapter 5

Volatility

5.1 The Implied Volatility

Let $C(S_0, K, T, r, \sigma)$ be the Black Scholes price of a call option with S_0, K and T as the initial price of stock, strike price and maturity time of the option respectively. Let M denote the market price of the call option. Then the implied volatility denoted is the value of σ that makes the two prices equal i.e.,

$$C(S_0, K, T, r, \sigma) = M. \quad (5.1)$$

Now from the Black Scholes formula it is not possible to express σ in terms of the other parameters and the Black Scholes price in closed form. We can compute $\frac{\partial C}{\partial \sigma}$ and is given by,

$$\frac{\partial C}{\partial \sigma} = S_0 \sqrt{T} \phi(d_1), \quad (5.2)$$

where ϕ is standard normal density and d_1 is given by 3.29. This implies that $\frac{\partial C}{\partial \sigma} > 0$, which means that the Black Scholes price is a monotonic increasing function of σ . Therefore to compute the implied volatility of a call option we can use Newton-Raphson method of root finding for the function $f(\sigma) = C(S_0, K, T, r, \sigma) - M$. The Newton-Raphson method for a function $f(x)$ to estimate its root first requires an interval (a, b) in which its root lies. Then with an initial guess of root $x_0 = \alpha$ where $\alpha \in (a, b)$. The root is estimated by the following iteration:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad f'(x_n) \neq 0. \quad (5.3)$$

Fig 5.1 shows the computed implied volatility(using Newton Raphson method) plot of TATA MOTORS call options¹ dated 28-01-22 with $S_0 = 497.30$ and maturity $T = 27$ days. For comparison NSE Option Chain site implied volatility values is also plotted. We can observe from the plot that Newton-Raphson method works well for computing implied volatility.

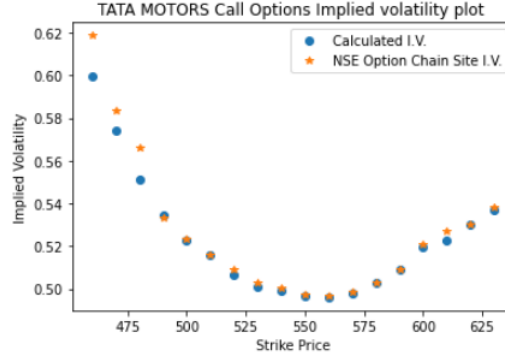


Figure 5.1: TATA MOTORS stock call options implied volatility plot.

Fig 5.2 shows the implied volatility plot of NIFTY50 stock index call options dated 29-11-21 with $S_0 = 17053.9$ and $T = 3, 10, 17, 31$ days. The volatility surface plot which is a plot between the implied volatility, strike price and maturities is also plotted.²

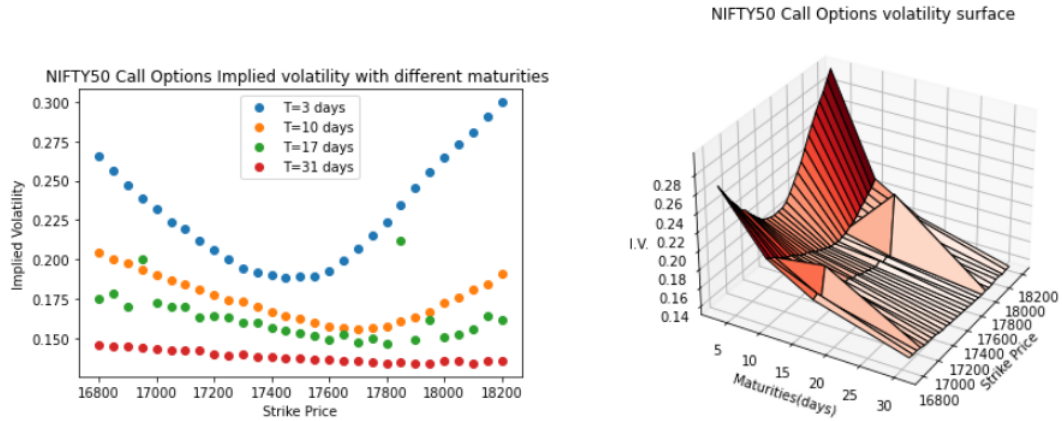


Figure 5.2: NIFTY50 stock index call options implied volatility and volatility surface plot.

¹Call options data used in this section has been obtained from <https://www.nseindia.com/option-chain> and as per the convention of NSE option chain site $r = 10\%$ is used for computing the implied volatility.

²The computed implied volatility used for plotting for both TATA MOTORS and NIFTY50 stock is shown in Appendix B

Now if the Black Scholes assumption that σ is constant was true then we would have obtained a flat line in the implied volatility plot but in general it is usually observed that the implied volatility plot is in the shape of a smile or it is skewed to one of the side as observed in Fig 5.1 and 5.2. This suggests that σ atleast as a function of K and T is not constant. One of reason for obtaining smile in the implied volatility plot is that the demands of in the money or out of the money options are more as compared to at the money options which led to more pricing of these options as compared to at the money options and as a result of that the implied volatility is more for these options. For a more detailed discussion on implied volatility the reader is encouraged to refer to [Hul18] and [Pap19].

In the next section we discuss the GARCH models which are used to model time varying volatility. Theory for the next section is covered from [Tsa16].

5.2 GARCH(1,1) Model

GARCH stands for Generalised Autoregressive Conditional Heteroskedasticity. In lay-man terms heteroskedasticity means volatility. Under GARCH the present volatility of the series depends on its past. GARCH model requires the assumption of stationarity (in the weak sense) of the series. In a GARCH(1,1) model³ the log-return series r_t follows:

$$r_t = \sigma_t \epsilon_t, \quad \sigma_t^2 = \omega + a r_{t-1}^2 + b \sigma_{t-1}^2. \quad (5.4)$$

where $\{\epsilon_t\}_{t>0}$ is a sequence of i.i.d. random variables (also known as innovations or noise) with mean 0 and variance 1 and $\omega = \gamma V_L$ where V_L is the long run variance. Usually we have that $\epsilon_t \stackrel{d}{=} \mathcal{N}(0, 1)$. We have the following conditions on the parameters $\omega > 0, a \geq 0, b \geq 0, (a + b) < 1$ and $\gamma + a + b = 1$. The terms σ_t^2 measures the conditional variance of the series r_t we will show this. Let \mathcal{F}_t denote the information about σ_t till time t then the conditional mean⁴ and conditional variance of r_t is given by,

$$E[r_t | \mathcal{F}_{t-1}] = \sigma_t E[\epsilon_t | \mathcal{F}_{t-1}] = \sigma_t E[\epsilon_t] = 0.$$

³In general a GARCH(p, q) model involves p lag terms and q lag terms of r_t and σ_t respectively in 5.4. We have restricted our study to GARCH(1,1) model only for this project.

⁴The GARCH model also involves modeling the conditional mean of r_t . In this project we have only considered the case when the conditional mean of r_t is zero.

$$\begin{aligned}
Var[r_t|\mathcal{F}_{t-1}] &= E[(r_t - E[r_t|\mathcal{F}_{t-1}])^2|\mathcal{F}_{t-1}] \\
&= E[r_t^2|\mathcal{F}_{t-1}] \\
&= \sigma_t^2 E[\epsilon_t^2|\mathcal{F}_{t-1}] \\
&= \sigma_t^2 E[\epsilon_t^2] \\
&= \sigma_t^2.
\end{aligned}$$

Now using law of total expectation and variance we can show the following,

$$\begin{aligned}
E[r_t] &= E[E[r_t|\mathcal{F}_{t-1}]] \\
&= E[\sigma_t E[\epsilon_t|\mathcal{F}_{t-1}]] \\
&= 0. \\
Var[r_t] &= E[Var[r_t|\mathcal{F}_{t-1}]] + Var[E[r_t|\mathcal{F}_{t-1}]] \\
&= E[\sigma_t^2] + 0 \\
&= E[\sigma_t^2].
\end{aligned}$$

From above it follows that $Var[r_t] = E[r_t^2] = E[\sigma_t^2]$. Now using this and the assumption of stationarity of the series r_t we can compute the unconditional variance as follows:

$$\begin{aligned}
Var[r_t] &= E[\sigma_t^2] \\
&= E[\omega + ar_{t-1}^2 + b\sigma_{t-1}^2] \\
&= \omega + aE[r_{t-1}^2] + bE[r_{t-1}^2] \\
&= \omega + aE[r_t^2] + bE[r_t^2] \\
&= \omega + aVar[r_t] + bVar[r_t].
\end{aligned}$$

This gives,

$$Var[r_t] = \frac{\omega}{1 - a - b}. \quad (5.5)$$

The restriction $a + b < 1$ is required for $Var[r_t]$ in 5.5 to be well defined. It can be observed that the unconditional variance $Var[r_t]$ is same as the long run variance V_L . The GARCH(1,1) model is a mean reverting model. The variance tends to get pulled towards the long run variance V_L with time and the speed of the reversion is governed by the parameter γ . For more detail to the mean reversion property of GARCH(1,1) model

the reader is encouraged to refer to [Hul18].

GARCH(1,1) model also captures the phenomena of volatility clustering in which large changes of volatility tends to be followed by large changes of volatility and similarly for the small changes of volatility. This can be observed from 5.4 that large value of σ_{t-1}^2 or r_{t-1}^2 will result in large value of σ_t in next period and similar reason for small values as well.

Now for the noise terms ϵ_t we have that,

$$E[\epsilon_t^4] = K_\epsilon + 3,$$

where K_ϵ is the excess kurtosis of the noise terms. Then,

$$E[r_t^4] = E[\sigma_t^4]E[\epsilon_t^4] = (K_\epsilon + 3)E[\sigma_t^4].$$

Using 5.4 it can be shown that,

$$E[\sigma_t^4] = \frac{\omega^2(1+a+b)}{(1-a-b)(1-(a+b)^2-a^2(K_\epsilon+2))}.$$

Note: We require $a+b < 1$ and $(1-(a+b)^2-a^2(K_\epsilon+2)) > 0$ for the existence of $E[\sigma_t^4]$. The Kurtosis of the GARCH(1,1) modelled returns denoted by K_r is found to be,

$$K_r = \frac{E[r_t^4]}{(Var[r_t])^2} = \frac{(K_\epsilon + 3)(1-(a+b)^2)}{(1-2a^2-(a+b)^2-K_\epsilon a^2)}.$$

Since $K_\epsilon \geq 0$ then we have that,

$$K_r = \frac{(K_\epsilon + 3)(1-(a+b)^2)}{(1-(a+b)^2-2a^2-K_\epsilon a^2)} > \frac{(K_\epsilon + 3)(1-(a+b)^2)}{(1-(a+b)^2)} > (K_\epsilon + 3) \geq 3.$$

This shows that the distribution of r_t under GARCH(1,1) model have heavier tails than that implied by normal distribution. Now if $\epsilon_t \stackrel{d}{=} \mathcal{N}(0, 1)$ then $K_\epsilon = 0$ and the kurtosis of r_t denoted by K_r^g is given by,

$$K_r^g = \frac{3(1-(a+b)^2)}{(1-2a^2-(a+b)^2)}. \quad (5.6)$$

In order to account for more heavier tails of distribution of r_t , Student's t distribution with ν degrees of freedom can be used for the distribution of noise terms. In this case $K_\epsilon = \frac{6}{\nu-4}$ and the kurtosis of r_t denoted by K_r^t is given by,

$$K_r^t = \frac{(\frac{6}{\nu-4} + 3)(1-(a+b)^2)}{(1-2a^2-(a+b)^2-\frac{6}{\nu-4}a^2)}. \quad (5.7)$$

5.2.1 Parameter estimation

The parameters of the GARCH(1,1) model can be estimated by M.L.E. method. Let $\{x_1, x_2, \dots, x_n\}$ be the empirical log-returns. The methodology is to maximise the following likelihood function,

$$L = \prod_{i=1}^n f_{r_t}(x_i). \quad (5.8)$$

Now if we are fitting the model to the empirical log-returns with $\epsilon_t \stackrel{d}{=} \mathcal{N}(0, 1)$ then the conditional P.D.F. of r_t using $r_t = \sigma_t \epsilon_t$ is given by,

$$f(r_t | \mathcal{F}_{t-1}) = \frac{1}{\sqrt{2\pi}\sigma_t} e^{-\frac{r_t^2}{2\sigma_t^2}}. \quad (5.9)$$

Then the P.D.F. mentioned in 5.9 can be used in 5.8 for estimating the parameters.

Now we discuss the case when the noise terms are assumed to be drawn from Student's t distribution with ν degrees of freedom and denote the random variable by T then its P.D.F. is given by,

$$f_T(x) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}},$$

where $\Gamma(y) = \int_0^\infty x^{y-1} e^{-x} dx$, $E[T] = 0$ and $Var[T] = \frac{\nu}{\nu-2}$.

Now since we require $Var[\epsilon_t] = 1$ we need to consider the standardised Student's t distribution given by $Y = \sqrt{\frac{\nu-2}{\nu}}T$ then $Var[Y] = 1$ and the P.D.F. of Y can be shown to be given by,

$$f_Y(y) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{(\nu-2)\pi}\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{y^2}{(\nu-2)}\right)^{-\frac{\nu+1}{2}}.$$

Now again using $r_t = \sigma_t \epsilon_t$ the conditional P.D.F. of r_t is given by,

$$f(r_t | \mathcal{F}_{t-1}) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{(\nu-2)\pi}\Gamma\left(\frac{\nu}{2}\right)} \frac{1}{\sigma_t} \left(1 + \frac{r_t^2}{(\nu-2)\sigma_t^2}\right)^{-\frac{\nu+1}{2}}. \quad (5.10)$$

In this case the obtained P.D.F. in 5.10 can be used in 5.8 for estimating the parameters.

5.2.2 Application of GARCH(1,1) model on Reliance stock

In this section we will estimate the GARCH(1,1) model parameters for Reliance stock followed by computing the conditional variances using 5.4. We will be using around 10

years of historical data (01/01/2012 - 21/04/2022) of closing prices of Reliance stock.⁵

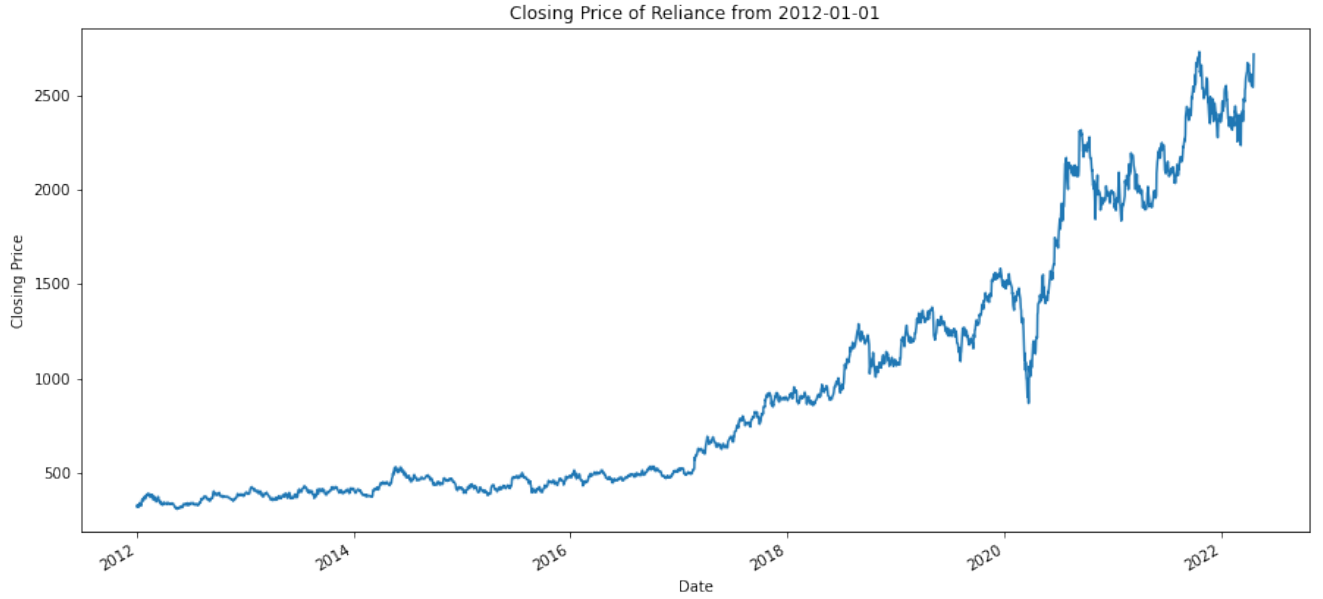


Figure 5.3: Daily closing price of Reliance stock from 2012-01-01.

Fig 5.4 shows the daily log-returns of Reliance stock and table 5.1 shows some statistics of the log-returns. Skewness value is positive which suggests that the log-returns are somewhat skewed to the right side rather than being symmetric . High value of kurtosis suggests that the distribution of log-returns have heavy tails which means that the tail of the log-returns distribution dies out slowly.

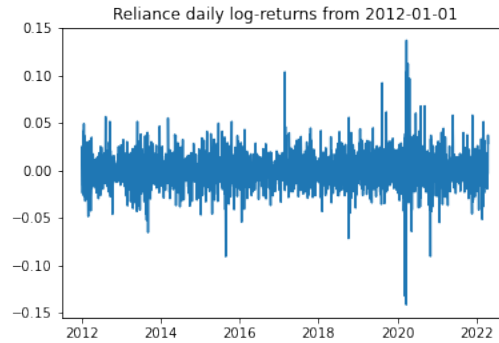


Figure 5.4: Daily log-returns of Reliance stock.

⁵Historical data of closing price of Reliance has been obtained from <https://finance.yahoo.com>.

Mean	Std. Dev.	Skewness	Kurtosis
0.000839	0.017917	0.076077	10.036623

Table 5.1: Some statistics of log-returns of Reliance stock.

We first check that the log-returns series is stationary by using the Augmented Dickey-Fuller test. The null hypothesis of this statistical test is that the series is not stationary and the alternate hypothesis is that the series is stationary. Table 5.2 shows that the test reports a p-value of 0 which is less than 0.05 therefore we can reject the null hypothesis and confirm the stationarity of the log-return series.

	p-value
ADF Test	0.0

Table 5.2: Augmented Dickey-Fuller test for stationarity of the log-return series.

We now check that the log-return series has GARCH effects(Conditional heteroskedasticity). This is observed by the ACF(Auto Correlation function) plot of the squared log-return series which computes the correlation coefficient of the squared series with its lag-terms to check for serial correlation. Fig 5.5 shows that ACF plot has some statistically significant spikes at various lags. Blue region in the plot is the confidence interval if a spike falls in this region then it is statistically zero(no autocorrelation).

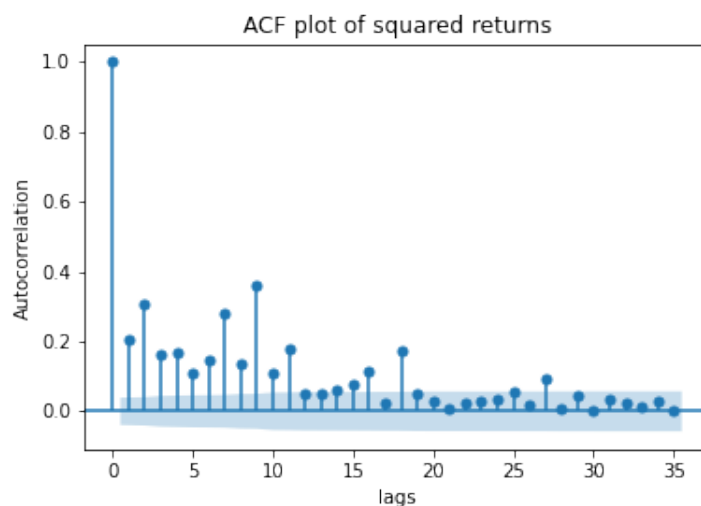


Figure 5.5: ACF plot of the squared daily log-returns.

We then use the Ljung Box test to confirm for serial correlation. The null hypothesis of the test is that the series does not exhibit serial correlation and the alternate hypothesis is that it exhibits serial correlation. Table 5.3 shows the test results for first 5 lags. Since all the p-values are less than 0.05 we thus reject the null hypothesis to confirm for the presence of serial correlation.

L.B. test \ Lags	1	2	3	4	5
p-value	1.748866e-25	1.525120e-75	2.069387e-89	1.075583e-103	1.853743e-109

Table 5.3: Ljung-Box test for first 5 lags of the squared log-return series.

We now use M.L.E. method as described in section 5.2.1 to estimate the parameters of the GARCH(1,1) model using both the Gaussian and Student's t distribution innovations (noise). For convenience we will denote our model by GARCH(1,1) when Gaussian innovation is used and by t-GARCH(1,1) when Student's t distribution innovation is used. Tables 5.4 and 5.5 shows the estimated parameters for both the models respectively.

ω	a	b
2.566257e-05	0.087744	0.827022

Table 5.4: GARCH(1,1) estimated parameters using M.L.E.

ω	a	b	ν
1.613329e-05	0.065359	0.879345	6.083066

Table 5.5: t-GARCH(1,1) estimated parameters using M.L.E.

Now the conditional volatilities σ_t can be computed using 5.4 for both models. Now to check the goodness of fit our models we can analyse the distributional properties of both the innovations. The series of both the innovations are computed using $\epsilon_t = \frac{r_t}{\sigma_t}$ from 5.4. Fig 5.6 shows the mean and variance of both the GARCH(1,1) and t-GARCH(1,1) innovations respectively. We observe that the mean and variance in both cases are close to 0 and 1 respectively as expected.

	Mean	Variance
GARCH(1,1) innovation	0.043317	0.999375
t-GARCH(1,1) innovation	0.043671	1.012925

Table 5.6: Mean and variance of the GARCH(1,1) and t-GARCH(1,1) innovations respectively.

Now to check the distributional assumptions of our innovations the Quantile-Quantile plot (probability plot) may be used. If the distribution of the computed innovations are normal and Student's t respectively then we expect the Q-Q plot to fit straight line in both cases. From fig 5.6 we observe that the gaussian innovations does not fit the line properly suggesting that the distribution of the GARCH(1,1) innovation is not gaussian. Similar thing is observed for t-Garch(1,1) innovations but comparatively it fits the line better in this case.

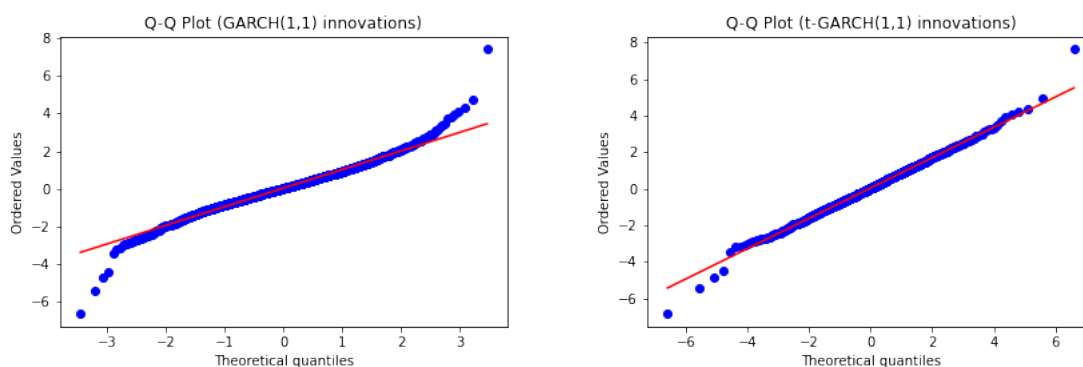


Figure 5.6: Q-Q plots of the GARCH(1,1) and t-GARCH(1,1) innovations respectively.

We then check the normality of GARCH(1,1) innovations using Jarque–Bera and Shapiro–Wilk tests. The null hypothesis is that the series data is normally distributed and the alternate hypothesis is that the series data is not normally distributed. Table 5.7 shows that both the tests report p-values of less than 0.05. Therefore we can reject the null hypothesis and conclude that the GARCH(1,1) innovations are not normally distributed.

Test	p-value
Jarque–Bera test	0.0
Shapiro–Wilk test	4.233365e-20

Table 5.7: Normality test for the GARCH(1,1) innovations.

Table 5.8 shows the kurtosis of empirical returns and the kurtosis implied(using 5.6 and 5.7) by both the GARCH(1,1) and t-GARCH(1,1) modeled returns. We observe that t-GARCH(1,1) modeled returns captures heavier tails than the GARCH(1,1) modeled returns.

	Empirical returns	GARCH(1,1) returns	t-GARCH(1,1) returns
Kurtosis	10.036623	3.312534	7.294554

Table 5.8: Kurtosis for empirical, GARCH(1,1) and t-GARCH(1,1) returns respectively.

Therefore we can conclude by analysing both the innovations that the t-GARCH(1,1) model fits better to Reliance stock data as compared to the GARCH(1,1) model where Gaussian innovation is used.

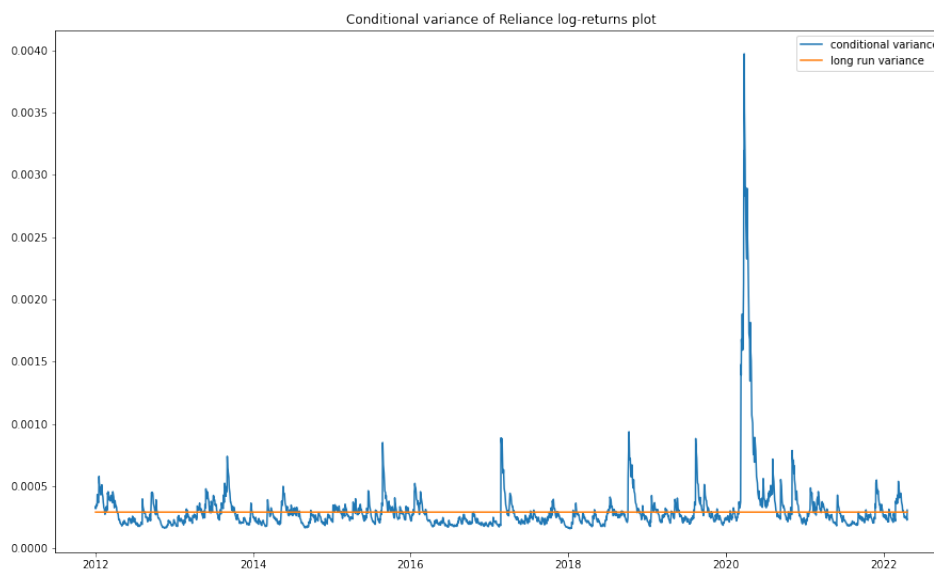


Figure 5.7: Conditional variance computed through the estimated t-GARCH(1,1) model.

Fig 5.7 shows the computed conditional variances (σ_t^2) through the estimated t-GARCH(1,1) model. It can be observed that the model captures the phenomena of volatility clustering and the mean reversion property of the conditional variances (σ_t^2) towards the long run variance V_L . The high volatile period during 2020-21 is due to the COVID-19 pandemic.

Appendix A

Python Code used for the project

A.1 Code used for simulation of BM,Poisson,GBM,MJD and KJD paths

```
1 import numpy as np
2 import random
3 import matplotlib.pyplot as plt

1 ## BM simulation
2 N = 1000
3 BM = np.empty((N,3))
4 t = np.linspace(0,1,N)
5 for i in range(3):
6     inc= np.sqrt(1/N)*np.random.normal(0,1,N-1)
7     cumsum=np.cumsum(inc).tolist()
8     b=[0] + cumsum
9     BM[:,i] = np.array(b)
10    plt.plot(t,b)
11 plt.title("Brownian motion paths simulation")
12 plt.xlabel("t")
13 plt.ylabel("B(t)")
14 plt.savefig("BM_paths")

1 ##GBM simulation
2 N = 1000
3 mu=0.2
4 sigma=0.3
5 t = np.linspace(0,1,N)
```

```

6 for i in range(3):
7     s = [100]
8     for j in range(N-1):
9         s.append(s[j] + s[j]*mu*(1/N) + s[j]*sigma*(BM[j+1][i]-BM[j][i]))
10    plt.plot(t,s)
11 plt.title("GBM paths simulation")
12 plt.xlabel("t")
13 plt.ylabel("X(t)")
14 plt.savefig("GBM_paths.png")

```

```

1 ## Poisson process generation for simulation purpose
2 lamda = 5
3 N=1000
4 t = np.linspace(0,1,N)
5 Nt = np.empty((N,3))
6 nt_inc = np.empty((N-1,3))
7 for i in range(3):
8     nt_inc[:,i]=np.random.poisson(lam=lamda*(1/N),size=N-1)
9     nt = np.cumsum(nt_inc[:,i]).tolist()
10    nt = [0] + nt
11    Nt[:,i] = nt
12 plt.plot(t,Nt)
13 plt.title("Poisson process paths simulation")
14 plt.xlabel("t")
15 plt.ylabel("N(t)")
16 plt.savefig("Poisson_simulation.png")

```

```

1 ## MJD paths simulations
2 for i in range(3):
3     sigma=0.1#diffusion volatility
4     mu=0.01#diffusion mean
5     m=0 #jump mean
6     delta=0.1#jump std dev
7     s0=100
8     diffusion=np.array([(mu - 0.5*sigma**2)*(1/N) + sigma*(BM[j+1,i]-BM[j,i]
9     for j in range(N-1))]
10    jump =[]
11    for k in nt_inc[:,i]:
12        if k == 1:
13            jump.append(np.random.normal(m, delta))
14        else: jump.append(0)
15    jump = np.array(jump)

```

```

15     r = diffusion + jump
16     s = np.exp(r).tolist()
17     mjd_st=[s0]
18     for i in range(N-1):
19         mjd_st.append(mjd_st[i]*s[i])
20     plt.plot(t, mjd_st)
21     plt.xlabel('t')
22     plt.ylabel('S(t)')
23     plt.title('MJD paths simulation')
24 plt.savefig('MJD_simulation.png')

1 ## KJD paths simulation
2 for i in range(3):
3     p = 0.5
4     n_1 = 10
5     n_2 = 10
6     kou_jumps = []
7     for k in nt_inc[:, i]:
8         if k == 1:
9             if np.random.binomial(1, p) == 1:
10                 kou_jumps.append(np.random.exponential(scale=1/n_1))
11             else: kou_jumps.append(-np.random.exponential(scale=1/n_2))
12         else: kou_jumps.append(0)
13     kou_jumps = np.array(kou_jumps)
14     kou_r = diffusion + kou_jumps
15     kou_s = np.exp(kou_r).tolist()
16     kou_st=[s0]
17     for i in range(N-1):
18         kou_st.append(kou_st[i]*kou_s[i])
19     plt.plot(t, kou_st)
20     plt.xlabel('t')
21     plt.ylabel('S(t)')
22     plt.title('KJD paths simulation')
23 plt.savefig('KJD_simulation.png')

```

A.2 Code used for estimating BS, MJD and KJD parameters for Infosys equity

```

1 import numpy as np

```

```

2 import random
3 import matplotlib.pyplot as plt
4 from scipy.stats import norm
5 from scipy.ndimage.interpolation import shift
6 from numpy import prod
7 import pandas as pd
8 from matplotlib.pyplot import figure
9 import warnings
10 warnings.filterwarnings("ignore")

1 def returns(data): ## compute daily simple returns
2     N=len(data)
3     r=np.array((N,))
4     r = (data/data.shift(1))-1
5     #r = r.dropna()
6     r = r[1:N]
7     return r

1 import yfinance as yf
2 data = yf.download('INFY.NS',start="2014-01-01",end="2022-04-21") ##Infosys
    8 years historical data
3 data['Adj Close'].plot(figsize = (15,7))
4 price = data['Adj Close']
5 plt.title("Closing Price of Infosys from 2014-01-01")
6 plt.ylabel('Closing Price')
7 plt.savefig('INFY_closing_prices.png')

1 returns_data = returns(price)
2 plt.plot(data.index[1:], returns_data)
3 plt.title("Infosys daily returns from 2014-01-01")
4 plt.ylabel("Daily returns")
5 plt.savefig('INFY_daily_returns')

1 ## Infosys statistics
2 from scipy.stats import kurtosis,skew
3 print(np.mean(returns_data))
4 print(np.var(returns_data))
5 print(skew(returns_data))
6 print(kurtosis(returns_data, fisher=False))

1 ## BS parameters estimation
2 dt = 1/252
3 bs_sigma = np.std(returns_data)/(dt)**0.5

```

```

4 bs_mu = (np.mean(returns_data) + 0.5*((bs_sigma)**2)*dt)/dt
5 print("BS_parameters - mu = {}, sigma = {}".format(bs_mu, bs_sigma))

1 N = len(returns_data)
2 jumps = 0
3 dt = 1/252
4 diffusion_component = []
5 for i in range(N): ## computing jumps
6     if np.abs(returns_data[i]) >= 3*np.std(returns_data):
7         jumps = jumps + 1
8     else : diffusion_component.append(returns_data[i])
9 ## initialisng parameters for MLE purpose
10 initial_sigma = np.std(diffusion_component)*(1/dt)**0.5
11 initial_mu = (np.mean(diffusion_component) + (0.5*(initial_sigma)**2)*dt)/
    dt
12 initial_lamda = jumps*252/len(returns_data)
13 print(jumps)
14 print(3*np.std(returns_data))
15 print(initial_mu)
16 print(initial_sigma)
17 print(initial_lamda)

1 def mjd_neg_log_likelihood(para, returns): ## computes mjd neg of
    log_likelihood
2     x=returns
3     N = len(x)
4     dt = 1/252
5     mu, sigma, m, delta, lam = para
6     likelihood = []
7     for i in range(N):
8         likelihood.append(lam*dt*norm.pdf(x[i], mu*dt + m, ((sigma**2)*dt +
            delta**2)**0.5) + (1- lam*dt)*norm.pdf(x[i], mu*dt, ((sigma**2)*dt)**0.5)
9     )
10    l = sum(np.log(likelihood))
11    return(-l) ## returns negative of log-likelihood

12 from scipy.optimize import minimize ## mjd log-likelihood optimisation for
    estimating parameters
13 bounds = ((-np.inf, np.inf), (0, np.inf), (-np.inf, np.inf), (0, np.inf), (0, np.inf)
    ) ## bounds for estimating the parameters
14 mjd_res = minimize(mjd_neg_log_likelihood, (initial_mu, initial_sigma
    , 0.5, 0.5, initial_lamda), args = returns_data, method='SLSQP', bounds=

```

```

    bounds, options={'disp': True} )
15 print(mjd_res)
16
17 ## MJD estimated parameters
18 print("MJD_parameters - mu = {},sigma = {},jump_mean = {},jump_std = {},
    lamda = {}".format(mjd_res.x[0], mjd_res.x[1], mjd_res.x[2], mjd_res.x[3],
    mjd_res.x[4]))

1 def kou_neg_log_likelihood(para, returns): ## Kou neg of log_likelihood for
    MLE purpose
2     x = returns
3     N = len(x)
4     dt = 1/252
5     mu, sigma, lam, p, n_1, n_2 = para
6     q = 1-p
7     likelihood = []
8     for i in range(N):
9         likelihood.append((((1-lam*dt)*norm.pdf((x[i]-mu*dt)/(sigma*(dt)
            **0.5)),0,1))/(sigma*(dt)**0.5) + lam*dt*(p*n_1*np.e**((((sigma*n_1)
            **2)*dt*0.5) - (((x[i]-mu*dt)*n_1)))*norm.cdf(((x[i]-mu*dt-n_1*dt*(
            sigma)**2)/(sigma*(dt)**0.5))) + q*n_2*np.e**((((sigma*n_2)**2)*dt*0.5)
            + ((x[i]-mu*dt)*n_2))*norm.cdf((-x[i]-mu*dt+n_2*dt*(sigma)**2)/(sigma
            *(dt)**0.5))))))
10    l = sum(np.log(likelihood))
11    return(-l) ## returns negative of log-likelihood
12
13 from scipy.optimize import minimize ## kou-log-likelihood optimisation for
    estimating parameters
14 bounds = ((-np.inf, np.inf), (0, np.inf), (0, np.inf), (0, 1), (1, np.inf), (0, np.inf)
    )) ## bounds for estimating the parameters
15 kou_res = minimize(kou_neg_log_likelihood, (initial_mu, initial_sigma,
    initial_lamda, 0.5, 2, 2), args = returns_data, bounds=bounds, method='SLSQP',
    options={'disp': True} )
16 print(kou_res)
17
18 ## Kou estimated parameters
19 print("KJD_parameters - mu = {},sigma = {},lamda = {},p = {},n_1 = {},n_2 =
    {}".format(kou_res.x[0], kou_res.x[1], kou_res.x[2], kou_res.x[3], kou_res
    .x[4], kou_res.x[5]))

1 ## Simulating BS,MJD and KJD returns through estimated parameters for
    comparison with empirical density of returns

```

```

2 N = len(returns_data)
3 bs_sim_returns = np.random.normal(np.mean(returns_data), np.std(returns_data
    ), N).tolist() ## BS_model simulated returns
4 mjd_sim_returns = []
5 kou_sim_returns = []
6 dt = 1/252
7 mjd_mu, mjd_sigma, mjd_m, mjd_delta, mjd_lamda = mjd_res.x
8 kou_mu, kou_sigma, kou_lamda, p, n_1, n_2 = kou_res.x
9 for i in range(N):
10     if np.random.binomial(1, mjd_lamda*dt) == 1: ## if there is jump for
        MJD_model
11         mjd_sim_returns.append(mjd_mu*dt + mjd_sigma*((dt)**0.5)*np.random.
            normal(0,1) + np.random.normal(mjd_m, mjd_delta))
12     else: mjd_sim_returns.append(mjd_mu*dt + mjd_sigma*((dt)**0.5)*np.
        random.normal(0,1)) ## if there is no jump for MJD_model
13
14     if np.random.binomial(1, kou_lamda*dt) == 1: ## if there is jump for
        Kou_model
15         if np.random.binomial(1, p) == 1: ## positive jump for Kou_model
16             kou_sim_returns.append(kou_mu*dt + kou_sigma*((dt)**0.5)*np.
                random.normal(0,1) + np.random.exponential(scale=1/n_1))
17             ## negative jump for Kou_model
18         else: kou_sim_returns.append(kou_mu*dt + kou_sigma*((dt)**0.5)*np.
                random.normal(0,1) - np.random.exponential(scale=1/n_2))
19
20     else : kou_sim_returns.append(kou_mu*dt + kou_sigma*((dt)**0.5)*np.
        random.normal(0,1)) ## if there is no jump for Kou_model
21 figure(figsize=(10, 15))
22 plt.subplot(3,1,1)
23 plt.plot(data.index[1:], returns_data)
24 plt.plot(data.index[1:], bs_sim_returns)
25 plt.legend(['Empirical returns', 'BS simulated returns'], loc='upper center')
26 plt.subplot(3,1,2)
27 plt.plot(data.index[1:], returns_data)
28 plt.plot(data.index[1:], mjd_sim_returns)
29 plt.legend(['Empirical returns', 'MJD simulated returns'], loc='upper center'
    )
30 plt.subplot(3,1,3)
31 plt.plot(data.index[1:], returns_data)
32 plt.plot(data.index[1:], kou_sim_returns)
33 plt.legend(['Empirical returns', 'KJD simulated returns'], loc='upper center'

```



```

    )
34 plt.savefig('simulated_returns.png')

1 ## Density plots
2 import seaborn as sns
3 figure(figsize=(13, 5))
4 sns.kdeplot(returns_data)
5 sns.kdeplot(bs_sim_returns)
6 sns.kdeplot(mjd_sim_returns)
7 sns.kdeplot(kou_sim_returns)
8 plt.xlabel('returns')
9 plt.legend(['Empirical returns', 'Black Scholes', 'MJD model', 'KJD model'])
10 plt.title('Kernel density plots for empirical returns and simulated returns
           from the models')
11 plt.savefig('KDE_plot-INFY.png')

1 ## BS statistics
2 dt = 1/252
3 bs_mean = (bs_mu - 0.5*(bs_sigma)**2)*dt
4 bs_std = bs_sigma*(dt)**0.5
5
6 ## MJD Parameters and statistics
7 mu,sigma,m,delta,lamda = mjd_res.x
8 dt = 1/252
9 mjd_mean = (mu)*dt + m*lamda*dt
10 mjd_variance = (sigma**2)*dt + ((lamda*dt))*(delta**2 + m**2)
11 mjd_skewness = ((3*delta**2 + m**2)*m*lamda*dt)/mjd_variance**(3/2)
12 mjd_kurtosis = ((3*(delta)**4 + 6*(m*delta)**2 + (m)**4)*lamda*dt)/(
    mjd_variance)**2 + 3
13
14 ## Kou Parameters and statistics
15 mu,sigma,lamda,p,n_1,n_2 = kou_res.x
16 q = 1-p
17 kou_mean = mu*dt + lamda*((p/n_1)-(q/n_2))*dt
18 kou_variance = (sigma**2)*dt + (p*q*((1/n_1 + 1/n_2))**2 + (p/(n_1)**2 + q
    /(n_2)**2 ))*lamda*dt + lamda*dt*(p/n_1 - q/n_2)**2
19 kou_skewness = 6*(p/n_1**3 - q/n_2**3)*lamda*dt/(kou_variance)**(3/2)
20 kou_kurtosis = (24*(p/(n_1)**4 + q/(n_2)**4)*lamda*dt)/(kou_variance)**2 +
    3

```

A.3 Code used for pricing Infosys call options under BS, MJD and KJD models respectively

```

1 import numpy as np
2 import random
3 from scipy.stats import norm
4 import pandas as pd

1 def BS_call(S0,K,T,r,sigma): ## calculate BS call price
2     T=T/365
3     d1 = (np.log(S0/K) + (r+0.5*(sigma)**2)*T)/(sigma*np.sqrt(T))
4     d2 = (np.log(S0/K) + (r-0.5*(sigma)**2)*T)/(sigma*np.sqrt(T))
5     call_price = S0*norm.cdf(d1) - (K*np.exp(-r*T)*norm.cdf(d2))
6     return round(call_price,5)
7
8 ## MJD option price calculation (Monte Carlo method)
9 def mjd_price_call(para,r,T,S0,Strike):
10     T = T/365
11     sigma,m,delta,lamda = para
12     k = (np.e**(m + ((delta)**2)*0.5)) - 1
13     sim = 100000
14     b_T = np.random.normal(0,1,sim)*(T)**0.5
15     n_T = np.random.poisson(lam=lamda*(T),size=sim)
16     jumps_component = np.array([np.sum(np.random.normal(m,delta,n_T[i]))
17     for i in range(sim)])
18     s_T = S0*np.e**((r-0.5*(sigma)**2-lamda*k)*T + sigma*b_T +
19     jumps_component)
20     discounted_payoff = []
21     for i in range(sim):
22         discounted_payoff.append(np.e**((-r)*T)*(max(s_T[i]-Strike,0)))
23     return round(np.mean(discounted_payoff),5)
24
25 ## Kou model option price calculation (Monte Carlo method)
26 def kou_price_call(para,r,T,S0,Strike):
27     T = T/365
28     sigma,lamda,p,n_1,n_2 = para
29     k= (p*n_1)/(n_1 -1) + (q*n_2)/(n_2 + 1) - 1
30     sim = 100000
31     b_T = np.random.normal(0,1,sim)*(T)**0.5
32     n_T = np.random.poisson(lam=lamda*(T),size=sim)

```

```

31 jumps = []
32 for i in range(sim):
33     temp = [] ##list for storing jump points
34     for j in range(n_T[i]):
35         if np.random.binomial(1,p) == 1:
36             temp.append(np.random.exponential(scale=1/n_1))
37         else : temp.append(-np.random.exponential(scale=1/n_2))
38     jumps.append(sum(temp))
39 jumps_component = np.array(jumps)
40 s_T = S0*np.e**((r-0.5*(sigma)**2-lamda*k)*T + sigma*b_T +
jumps_component)
41 discounted_payoff = []
42 for i in range(sim):
43     discounted_payoff.append(np.e**((-r)*T)*(max(s_T[i]-Strike,0)))
44 return round(np.mean(discounted_payoff),5)

```

```

1 Pricing Infosys Call options dated 21/04/22, S0 =1615.70 ,expiry -26/05/22
2
3 call_data = pd.read_csv('infy21april.csv')
4 call_data["ASK PRICE"] = [float(str(i).replace(",","")) for i in call_data
["ASK PRICE"]]
5 call_data["STRIKE PRICE"] = [float(str(i).replace(",","")) for i in
call_data["STRIKE PRICE"]]
6 call_data
7
8 df = call_data
9 K = [df['STRIKE PRICE'][i] for i in range(len(df))]
10 BS_prices = []
11 MJD_prices = []
12 Kou_prices = []
13 mid_prices = (call_data["ASK PRICE"] + call_data["BID PRICE"])/2
14 for i in range(len(df)):
15     BS_prices.append(BS_call(S0=1615.70,K=K[i],T=35,r=0.045,sigma=np.std(
returns_data)*(252)**0.5))
16     MJD_prices.append(mjd_price_call(mjd_res.x[1:5],r=0.045,T=35,S0
=1615.70,Strike=K[i]))
17     Kou_prices.append(kou_price_call(kou_res.x[1:6],r=0.045,T=35,S0
=1615.70,Strike=K[i]))
18
19 Prices = pd.DataFrame({'STRIKE PRICE':K,'BID PRICE':df['BID PRICE'],'ASK
PRICE':df['ASK PRICE'],'MID PRICE':mid_prices,
20                        'BS PRICE':BS_prices,'MJD PRICE':MJD_prices,'KJD

```

```

    PRICE': Kou_prices})
21 Prices ## Computed prices

```

A.4 Code used for computing implied volatilities for TATA MOTORS stock and NIFTY50 stock index data respectively

```

1 import numpy as np
2 import random
3 import pandas as pd
4 from scipy.stats import norm
5 import matplotlib.pyplot as plt
6 import warnings

1 def BS_call(S0,K,T,r,sigma): ## calculate BS call price
2     T=T/365
3     d1 = (np.log(S0/K) + (r+0.5*(sigma)**2)*T)/(sigma*np.sqrt(T))
4     d2 = (np.log(S0/K) + (r-0.5*(sigma)**2)*T)/(sigma*np.sqrt(T))
5     call_price = S0*norm.cdf(d1) - (K*np.exp(-r*T)*norm.cdf(d2))
6     return call_price
7 def vega_call(S0,K,T,r,sigma):# vega will be used in Newton-Raphson method
8     T=T/365
9     d1 = (np.log(S0/K) + (r+0.5*(sigma)**2)*T)/(sigma*np.sqrt(T))
10    vega = S0*norm.pdf(d1,0,1)*(T)**(0.5)
11    return vega

1 def imp_vol(price_data,S0,T,r): ## computes implied volatility
2     data = price_data
3     ## price_data preprocessing
4     if 'ASK PRICE' in data.columns:
5         data["ASK PRICE"] = [float(str(i).replace(",","")) for i in data["
ASK PRICE"]]# removing comas from data entries if found to convert them
        in float
6         data["BID PRICE"] = [float(str(i).replace(",","")) for i in data["
BID PRICE"]]
7         data["STRIKE PRICE"] = [float(str(i).replace(",","")) for i in
data["STRIKE PRICE"]]
8         price = data[['ASK PRICE', 'BID PRICE']].mean(axis=1) ##Market
        price that will be used

```

```

9     elif 'Settle Price' in data.columns:
10         data = data.rename(columns={'Strike Price': 'STRIKE PRICE'})
11         price = data['Settle Price']
12     else: warnings.warn("No price data found")
13     temp = np.linspace(0,1,1000)
14     vol = [] # to store imp vol
15     u=1 #right most point of interval
16     l=0 #left most point of interval
17 # Finding the interval of convergence for Newton-Raphson method
18     for i in range(len(price)):
19         K=data['STRIKE PRICE'][i]
20         for j in range(1,len(temp)):
21             if BS_call(S0,K,T,r,temp[j]) > price[i]:
22                 l = temp[j-1]
23                 break
24         for k in range(1,len(temp)):
25             if BS_call(S0,K,T,r,temp[-(k+1)]) < price[i]:
26                 u = temp[-k]
27                 break
28         g = random.uniform(l,u)## initial point for Newton-Raphson method
29         h = (BS_call(S0,K,T,r,g) - price[i])/(vega_call(S0,K,T,r,g))
30         while abs(h) > 0.001:
31             g = g - h
32             h = (BS_call(S0,K,T,r,g) - price[i])/(vega_call(S0,K,T,r,g))
33         vol.append(round(g,4))
34     return(np.array(vol))

```

```

1 Computing TATA Motors call options implied volatility
2 ## TATA_Motors,Dated-28/01/2022,S0=497.30,T=27
3 tata_data = pd.read_csv('tata.csv')
4
5 tata_iv = imp_vol(tata_data,S0=497.30,T=27,r=0.1)
6 MID_PRICE = (tata_data["BID PRICE"] + tata_data["ASK PRICE"])/2
7 plt.plot(tata_data['STRIKE PRICE'],tata_iv,'o')
8 iv_list = tata_data['IV'].tolist()
9 IV = [i*0.01 for i in iv_list]
10 rmse = np.sqrt(np.sum((np.array(IV) - np.array(tata_iv))**2)/len(IV))
11 plt.plot(tata_data['STRIKE PRICE'],IV,'*')
12 plt.legend(loc="upper right")
13 plt.legend(["Calculated I.V.", "NSE Option Chain Site I.V."])
14 plt.xlabel('Strike Price')
15 plt.ylabel('Implied Volatility')

```

```

16 plt.title("TATA MOTORS Call Options Implied volatility plot ")
17 plt.savefig("TATA_IV_plot.png")
18 print(rmse)

1 Computing NIFTY50 call options implied volatility ,S0=17053.95,dated
   29/11/2021
2 data1 =pd.read_csv('nifty_29nov_2dec.csv')
3 data1=data1[(data1['Strike Price'].between(16800,18200))&(data1['Date'] ==
   '29-Nov-2021')].reset_index()
4 data2 =pd.read_csv('nifty_29nov_9dec.csv')
5 data2=data2[(data2['Strike Price'].between(16800,18200))&(data2['Date'] ==
   '29-Nov-2021')].reset_index()
6 data3 =pd.read_csv('nifty_29nov_16dec.csv')
7 data3=data3[(data3['Strike Price'].between(16800,18200))&(data3['Date'] ==
   '29-Nov-2021')].reset_index()
8 data4 =pd.read_csv('nifty_29nov_30dec.csv')
9 data4=data4[(data4['Strike Price'].between(16800,18200))&(data4['Date'] ==
   '29-Nov-2021')]
10 data4 = data4.sort_values(by=['Strike Price'], ascending=True)
11 data4 = data4.reset_index()
12 d1=imp_vol(data1,S0=17053.95,T=3,r=0.1)
13 d2=imp_vol(data2,S0=17053.95,T=10,r=0.1)
14 d3=imp_vol(data3,S0=17053.95,T=17,r=0.1)
15 d4=imp_vol(data4,S0=17053.95,T=31,r=0.1)
16 plt.plot(data1['Strike Price'],d1,'o',label = 'T=3 days')
17 plt.plot(data2['Strike Price'],d2,'o',label = 'T=10 days')
18 plt.plot(data3['Strike Price'],d3,'o',label = 'T=17 days')
19 plt.plot(data4['Strike Price'],d4,'o',label = 'T=31 days')
20 plt.legend(loc="upper center")
21 plt.xlabel('Strike Price')
22 plt.ylabel('Implied Volatility')
23 plt.title("NIFTY50 Call Options Implied volatility with different
   maturities ")
24 plt.savefig("NIFTY50_IV_plot.png")
25
26 ## plotting NIFTY50 volatility surface
27 x1 =np.zeros((len(data1),4))
28 for i in range(len(data1)):
29     x1[i][0]=d1[i]
30 for i in range(len(data1)):
31     x1[i][1]=d2[i]
32 for i in range(len(data1)):

```

```

33     x1[i][2]=d3[i]
34 for i in range(len(data1)):
35     x1[i][3]=d4[i]
36 y1 = np.zeros((len(data1),4))
37 for j in range(4):
38     for i in range(len(data1)):
39         y1[i][j]=data1['Strike Price'][i]
40 z1 = np.zeros((len(data1),4))
41 l=[3,10,17,31]
42 for j in range(4):
43     for i in range(len(data1)):
44         z1[i][j]=l[j]
45 from mpl_toolkits.mplot3d import Axes3D
46 from matplotlib import pyplot as plt
47 from matplotlib.pyplot import figure
48 figure(figsize=(13, 10))
49 fig = plt.figure()
50 ax = Axes3D(fig, auto_add_to_figure=False)
51 ax.plot_surface(y1,z1,x1,cmap='Reds',edgecolor='black')
52 ax.invert_xaxis()
53 ax.set_ylabel('Maturities(days)')
54 ax.set_xlabel('Strike Price')
55 ax.set_zlabel('I.V. ')
56 fig.add_axes(ax)
57 #ax.set_zlim(0.20,0.45)
58 ax.view_init(30,30)
59 plt.title('NIFTY50 Call Options volatility surface')
60 plt.show()

```

A.5 Code used for estimating the GARCH(1,1) model parameters for Reliance equity

```

1 import numpy as np
2 import random
3 import matplotlib.pyplot as plt
4 import pandas as pd
5 from matplotlib.pyplot import figure
6 from scipy.ndimage.interpolation import shift
7 import warnings

```

```

8 warnings.filterwarnings("ignore")

1 def returns(data): ## compute daily log returns
2     N=len(data)
3     #r=np.array((N,))
4     r = np.log(data) - np.log(data.shift(1))
5     #r = r.dropna()
6     r = r[1:N]
7     return r
8
9 import yfinance as yf
10 data = yf.download('RELIANCE.NS',start="2012-01-01",end="2022-03-31") ##
    Reliance 10 years historical data
11 data['Adj Close'].plot(figsize = (15,7))
12 plt.title("Closing Price of Reliance from 2012-01-01")
13 plt.ylabel('Closing Price')
14 plt.savefig('Rel_closing_prices.png')
15 price = data['Adj Close']
16
17 returns_data = returns(price)
18 plt.plot(data.index[1:], returns_data)
19 plt.title('Reliance daily log-returns from 31/03/2012')
20 plt.savefig('Rel_daily_log-returns')

1 ## Reliance statistics
2 from scipy.stats import kurtosis,skew
3 print(np.mean(returns_data))
4 print(np.std(returns_data))
5 print(skew(returns_data))
6 print(kurtosis(returns_data, fisher=False))

1 from statsmodels.tsa.stattools import adfuller
2 res = adfuller(returns_data)
3 print('Augmented Dickey-Fuller Statistic: %f' % res[0])
4 print('p-value: %f' % res[1])

1 from statsmodels.graphics.tsaplots import plot_pacf, plot_acf
2 plot_acf((returns_data**2))## checking for correlation
3 plt.xlabel('lags')
4 plt.ylabel('Autocorrelation')
5 plt.title('ACF plot of squared returns')
6 plt.show()
7

```



```

8 from statsmodels.stats.diagnostic import acorr_ljungbox
9 acorr_ljungbox(returns_data**2, lags=5, return_df=True)

1 def Garch(para, returns): ## compute the volatilities , input parameters = (
   omega, a, b)
2     N = len(returns)
3     omega, a, b = para[0:3] ## cond_mean = 0 (assumption)
4     cond_variance = np.zeros((N,)) ## array to store the volatilities
5     #long_run_variance = omega/(1-a-b)
6     cond_variance[0] = np.std(returns)**2 ## initialising cond_variance
7     #cond_variance[0] = long_run_variance ## initialising cond_variance
8     for i in range(1, N):
9         cond_variance[i] = omega + a*(returns[i-1])**2 + b*(cond_variance[i-1])
10    return cond_variance ## sigma-squared values
11
12 from scipy.special import gamma
13 def neg_log_likelihood(para, returns, noise): ## computes negative of log-
   likelihood which will be minimized using scipy.optimize.minimize
14     N = len(returns) ## for estimating the parameters.
15     cond_variance = Garch(para, returns)
16     if noise == 'normal':
17         ll = -0.5*sum([np.log(2*(np.pi)) + np.log(cond_variance[j]) +
   returns[j]**2/cond_variance[j] for j in range(N)])
18     elif noise == 't':
19         nu = para[3]
20         c = np.log([gamma((nu+1)/2)/(gamma(nu/2)*(np.pi*(nu-2))**0.5)])
21         ll = c*N - sum([np.log((cond_variance[j])**0.5) for j in range(N)])
   - ((nu+1)/2)*sum([np.log(1+(returns[j]**2/((nu-2)*(cond_variance[j]))))
   for j in range(N)])
22     return (-ll)

```

```

1 Taking the noise as Gaussian
2
3 from scipy.optimize import minimize
4 bounds = ((0.0001, None), (0.0001, 0.9999), (0.0001, 0.9999)) ## bounds for
   estimating the parameters
5 res_gauss = minimize(neg_log_likelihood, (0.1, 0.05, 0.9), args=(np.array(
   returns_data)*100, 'normal'), method='SLSQP', options={'disp': True})
6 print(res_gauss)
7 omega, a, b = res_gauss.x
8

```

```

9 omega = omega/10000 ## For optimisation purpose the returns were converted
   into percentage and thus the estimate of omega was scaled by 10000.
10 print("Estimated parameters : omega={}, a={}, b={}".format(omega,a,b))
11
12 Gaussian Noise Analysis
13
14 cond_variance = Garch((omega,a,b),returns_data)
15 noise_gauss = np.array(returns_data)/np.sqrt(cond_variance)
16 print(np.mean(noise_gauss))
17 print(np.var(noise_gauss))
18
19 from scipy import stats
20 stats.probplot(noise_gauss,dist = stats.norm,plot=plt)
21 plt.title("Q-Q Plot (GARCH(1,1) innovations)")
22
23 from scipy.stats import jarque_bera,shapiro
24 print(jarque_bera(noise_gauss))
25 print(shapiro(noise_gauss))

```

```

1 Taking noise from "Student's t distribution"
2
3 from scipy.optimize import minimize
4 bounds = ((0.0001,None),(0.0001,0.9999),(0.0001,0.9999)) ## bounds for
   estimating the parameters
5 res_t = minimize(neg_log_likelihood, (0.1,0.05,0.9,6),args =(np.array(
   returns_data)*100,'t'), method='SLSQP',options={'disp': True})
6 print(res_t)
7 omega_t,a_t,b_t,nu_t = res_t.x
8
9 omega_t = omega_t/10000 ## For optimisation purpose the returns were
   converted into percentage and thus the estimate of omega was scaled by
   10000.
10 print("Estimated parameters : omega={}, a={}, b={}, nu={}".format(omega_t,
   a_t,b_t,nu_t))
11
12 Noise analysis t-GARCH(1,1) estimated model
13
14 cond_variance_t = Garch((omega_t,a_t,b_t,nu_t),np.array(returns_data))
15 noise_t = np.array(returns_data)/np.sqrt(cond_variance_t)
16 print(np.mean(noise_t))
17 print(np.var(noise_t))
18

```

```

19 from scipy import stats
20 stats.probplot(noise_t, dist = stats.t(df=nu_t), plot=plt)
21 plt.title("Q-Q Plot (t-GARCH(1,1) innovations)")
22 plt.show()
23
24 ## Implied Kurtosis from both innovations of GARCH(1,1) model
25 gauss_kurtosis = 3*(1-(a+b)**2)/(1-2*(a**2) - (a+b)**2)
26 t_kurtosis = (6/(nu_t-4) + 3)*(1-(a_t+b_t)**2)/(1-2*(a_t**2) - (a_t+b_t)**2
    - 6*((a_t)**2)/(nu_t-4))
27 print(gauss_kurtosis)
28 print(t_kurtosis)
29
30 ## From the Q-Q plot and kurtosis values we see that Student's t
    distribution performs well as compared to normal distribution for noise
    .
31
32 omega,a,b = res_t.x[0:3] ## estimated parameters of t-GARCH(1,1) model
33 long_term_variance = omega_t/((1-a_t-b_t))
34 figure(figsize=(15, 9))
35 plt.plot(data.index[1:], cond_variance_t) ## cond_volatility plot with the
    std. residuals as Student's t distribution
36 plt.plot(data.index[1:], np.array([long_term_variance for i in range(len(
    data.index[1:]))]))
37 plt.legend(['conditional variance', 'long run variance'])
38 plt.title('Conditional variance of Reliance log-returns plot')
39 plt.savefig("Cond_variance.png")

```

Appendix B

Computed Implied Volatilities of Call options for TATA MOTORS stock and NIFTY50 stock index data respectively

BID PRICE	ASK PRICE	MID PRICE	STRIKE PRICE	I.V.(NSE Site)	I.V.(computed)
54.95	56.1	55.525	460	0.6186	0.5998
47.6	48.1	47.85	470	0.5834	0.5739
40.5	40.9	40.7	480	0.5659	0.5509
34.2	34.4	34.3	490	0.5334	0.5347
28.6	28.7	28.65	500	0.5232	0.5227
23.75	23.9	23.825	510	0.516	0.5158
19.4	19.5	19.45	520	0.5088	0.5068
15.65	15.8	15.725	530	0.5027	0.5012
12.65	12.8	12.725	540	0.5005	0.4989
10.1	10.2	10.15	550	0.497	0.4964
8.1	8.2	8.15	560	0.4965	0.4961
6.5	6.55	6.525	570	0.4987	0.4977
5.25	5.35	5.3	580	0.5028	0.5027
.3	4.4	4.35	590	0.5093	0.5092
3.6	3.7	3.65	600	0.5207	0.5193
2.9	3	2.95	610	0.5272	0.5227
2.4	2.5	2.45	620	0.5302	0.5304
2	2.05	2.025	630	0.5382	0.5371

Table B.1: Computed I.V. values using Newton-Raphson method and the I.V. values listed on NSE option chain site of TATA MOTORS stock call options.

STRIKE PRICE	Settle Price(T=3)	I.V.(T=3)	Settle Price(T=10)	I.V.(T=10)	Settle Price(T=17)	I.V.(T=17)	Settle Price(T=31)	I.V.(T=31)
16800	330.45	0.2657	408.7	0.2045	455.45	0.1754	525.7	0.1459
16850	289.9	0.2567	371.1	0.2002	425.05	0.1783	490.0	0.1451
16900	249.9	0.2472	335.7	0.1981	382.35	0.1702	458.25	0.1453
16950	212.95	0.2389	300.35	0.1934	393.9	0.2005	424.85	0.1442
17000	179.3	0.2317	268.25	0.1904	324.75	0.1727	390.7	0.1429
17050	147.3	0.2233	236.6	0.1866	291.6	0.17	362.75	0.1425
17100	119.7	0.2191	208.35	0.1847	265.6	0.1701	333.2	0.1422
17150	94.25	0.2121	180.6	0.181	231.4	0.163	304.7	0.1421
17200	72.2	0.2061	155.25	0.1776	208.85	0.164	278.7	0.1402
17250	53.85	0.2002	132.45	0.1746	187.15	0.1636	252.85	0.1393
17300	38.65	0.1946	111.8	0.1731	162.95	0.1602	230.15	0.1396
17350	27.6	0.1919	93.15	0.1702	143.0	0.16	208.4	0.1383
17400	19.4	0.1899	76.25	0.1667	122.95	0.1568	187.45	0.1382
17450	13.5	0.1886	62.05	0.1646	107.05	0.1552	168.0	0.1377
17500	9.5	0.1892	49.65	0.1621	90.0	0.1536	151.3	0.1371
17550	6.65	0.1893	39.4	0.1599	75.85	0.1519	134.45	0.1368
17600	4.8	0.1927	30.7	0.1572	63.85	0.1494	119.5	0.1364
17650	3.85	0.1992	24.05	0.1568	57.15	0.1526	105.5	0.1357
17700	3.2	0.2066	18.6	0.1556	44.6	0.1473	92.9	0.1355
17750	2.75	0.2151	15.05	0.1568	39.2	0.1498	81.85	0.1348
17800	2.5	0.2233	12.35	0.1575	30.65	0.1463	71.35	0.1344
17850	2.4	0.2343	10.5	0.1608	80.35	0.2117	62.5	0.1349
17900	2.25	0.2457	8.8	0.1633	21.9	0.1487	54.6	0.1341
17950	2.1	0.2551	7.65	0.1667	25.7	0.1614	48.2	0.1344
18000	1.95	0.2649	7.2	0.1722	16.5	0.1512	42.6	0.1361
18050	1.75	0.2729	6.35	0.1757	13.9	0.1523	36.75	0.1358
18100	1.6	0.2805	5.8	0.1809	12.7	0.1554	31.35	0.1343
18150	1.55	0.2906	5.3	0.1844	13.7	0.1645	27.45	0.1353
18200	1.45	0.2997	5.05	0.1908	10.35	0.1617	23.75	0.1353

Table B.2: Computed I.V. values of call options of NIFTY50 stock index with different maturities.

Bibliography

- [BS73] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.
- [Cam14] Giuseppe Campolieti. *Financial mathematics: A comprehensive treatment*. 2014.
- [CT15] Rama Cont and Peter Tankov. *Financial modelling with Jump Processes*. CRC Press, 2015.
- [Han07] Floyd B. Hanson. *Applied Stochastic Processes and control for jump-diffusions: Modeling, analysis, and computation*. Siam, 2007.
- [Hul18] John Hull. *Options, futures, and other derivatives*. Pearson Education Limited, 2018.
- [Kle20] Achim Klenke. *Probability theory: A comprehensive course*. Springer, 2020.
- [Kou02] S. G. Kou. A jump-diffusion model for option pricing. *Management Science*, 48(8):1086–1101, 2002.
- [Mat17] Kazuhisa Matsuda. Introduction to merton jump diffusion model, Jun 2017.
- [Mer74] Robert C. Merton. On the pricing of corporate debt: The risk structure of interest rates. *The Journal of Finance*, 29(2):449, 1974.
- [Mer76] Robert C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1-2):125–144, 1976.
- [Mik11] Thomas Mikosch. *Elementary stochastic calculus with finance in View*. World Scientific, 2011.

- [Pap19] Andrew Papanicolaou. Introduction to stochastic differential equations (sdes) for finance, Jan 2019.
- [Res92] Sidney I. Resnick. *Adventures in stochastic processes*. Birkhauser, 1992.
- [Ros09] Walter A. Rosenkrantz. *Introduction to probability and statistics for Science, Engineering, and Finance*. Chapman amp; Hall, 2009.
- [Shr11] Steven E. Shreve. *Stochastic calculus for finance II*. Springer, 2011.
- [Tan18] Furui Tang. Merton jump-diffusion modeling of stock price data, Oct 2018.
- [Tsa16] Ruey S. Tsay. *Analysis of Financial Time Series*. Wiley India, 2016.
- [Øk13] B. K. Øksendal. *Stochastic differential equations: An introduction with applications*. Springer, 2013.