Contents

Introduction

Talk about what is done to price a vanilla option throuhout the BSM method. How does the BSM model is fair under its assumption. What about if we are going beyond ? How perfomant is it ? What about other model such as . . . ?

Using R. R Core Team [2017]

Chapter 1

Methodology

1.1 Overview

The objective of this master thesis is to measure the performances of the Black-Scholes-Merton pricing method when the assumption of normality for the log-returns distribution is not met.

In order to measure such performances, I will (i) compute the options price by using in turn the models BSM, Merton mixed jump-diffusion and Heston stochastic volatility models. Thereafter, I will (ii) compare the implied volatility of those computed prices split by maturities and by models. Finally, I will (iii) construct delta-neutral portfolios to measure the hedging performance of the aforementioned models.

Instead of exclusively using market data, whether to gather option prices or to build the delta-neutral portfolio, I will rather construct some theoretical based algorithm. Depending on the framework to explore, the option prices will be computed either by using the BSM equation, if the underlying process relates to a geometric Brownian motion or by using the method developed by Heston [1993], it the underlying process relates to the model MJD or HSV. To assess the delta-neutral portfolio, I will need time series to measure its evolution across time. those series will be simulated based on the theories developed by Merton [1976] and Heston [1993] in order to respectively obtain paths with jumps and others with stochastic volatility. Consequently, I have built some functions in the R language to perform those tasks. table 1.1 is a summary of a few of them used in that chapter and in the analysis. More details on them are given in appendix **??**.

Table 1.1: R functions dealing with options and time series

Function name	Arguments	Purpose
bsm_call	$\{S(0), T, k, r, \sigma\}$	Compute the BSM price of an option
mjd_call	$\{S(0), T, k, r, \lambda, \mu, \delta\sigma\}$	Compute the Merton price of an option
hsv_call	$\{S(0), T, k, r, V(0), \theta, \kappa, \sigma, \rho\}$	Compute the heston price of an option
bsm_ts	$\{S(0), T, \sigma, \alpha, dt\}$	Simulate BSM time series
mjd_ts	$\{S(0), T, \sigma, \alpha, \lambda, \mu, \delta dt\}$	Simulate MJD time series
hsv_ts	$\{S(0), T, V(0), \alpha, \rho, \kappa, \theta, \sigma, dt\}$	Simulate HSV time series

Even though I won't construct my hedge directly on market data, I need those data to calibrate the parameters to pass into the functions listed in table 1.1. Therefore, the

functions with the objective to provide european call price will be calibrated with the Apple european call market data while, to stay consistent, the functions that simulate time series will be benchmarked using the Apple stock data available on the market. This process of calibration is fully explained at section 1.2.

1.2 Calibration

Whenever one deals with functions aimed to reproduce some real-life experiments, the calibration process is crucial because it gives to the functions the capacity to act within appropriate boundaries.

The process of calibration which will be applied in the current section concerns two distinctive groups of parameters. Those intended to the functions that compute the European call price and those used by the functions that simulate the hypothetical stock market movements. Consequently, the methods to calibrate both kinds of arguments differ, mainly because the options prices calculation must be performed under risk-neutral environment and the delta hedging is measured on time-series evolving under a risk-averse world.

To do so, I am going to use the available option price data to calibrate the arguments aimed to the functions that compute the price of the latter. Whilst I am going to use the stock's historical data to estimate the right values for the time-series output-related functions. The option market prices and stock data were downloaded using the package *quantmod*, developed by Ryan and Ulrich [2018] which uses Yahoo! finance as provider. The datasets so downloaded are available in appendix **??**.

Section 1.2.1 explains how I am going to operate for the options' parameters, whereas section 1.2.2 shows the procedure I am going to follow for the assets simulations' arguments.

1.2.1 Option prices based calibration

In accordance to Heston [1993] and Crisóstomo [2015], provided that the characteristic functions of the MJD and HSV models are known, the European call option price of such underlying processes can be computed using **??**. Although known, these characteristics functions (**????**) need some parameters to work and these parameters need appropriates values to best fit with what is observed in reality. That is why both functions need to be benchmarked with referential values before being used.

To do so, I am going to apply the same method followed by Crisóstomo [2015], namely, iteratively minimizing the difference between the options market prices and those generated by the functions that compute the option prices based on the characteristic functions. As the market data comes with a large number of maturities and stikes, tables 1.2 and 1.3 list those considered during the analysis.

Table 1.2: Taking into account maturities during calibration stage

63	91	126	154	182	245	399

The optimization method used is the least-square non-linear analysis. To work, such a method need (i) a function that returns dummy data, (ii) a dataset of data to serve as a template, (iii) a cost function to minimize and (iv) the parameters to optimize.

4

Table 1.3: Taking into account strikes during calibration stage

130	140	150	160	170	180	190	200	210	220

The functions that return the artificial data are those exhibited in table 1.1, that is to say, *mjd_call* and *hsv_call* for the computation of call price using MJD and HSV, respectively. The dataset template is the one in **??** (see appendix **??**). While the cost function is given by equation (1.1), the parameters to assess depend on the underlying model (either MJD or HSV).

$$\left(C_{K,T}^{mkt} - C_{K,T}^{h|m}(arguments) \right)^2 \tag{1.1}$$
$$\forall K \in \{130, 140, 150, 160, 170, 180, 190, 200, 210, 220\},$$
$$T \in \{63, 91, 126, 154, 182, 245, 399\}$$

Where the subscripts K and T respectively stand for strike price and maturity date, while the superscripts mkt denotes the "market price" and $h|m$ refers to either Heston or Merton process based prices.

Consequently, the equation (1.1) is that to be minimized using the least-square non-linear analysis, for all strikes and maturities. The output of this process will eventually be the calibrated arguments that maximize the performance of the model with respect to what is observed in the reality.

One difficulty when dealing with such an algorithm is that the least-square non-linear approach may return several best-fit sets of arguments due to the existence of multiple local minima in the cost function. Therefore, in order to choose among all the outputs provided by the optimization method, I will select those that ensure that the pricing function gives the most of its returns within the bid-ask spread of the option data, as shown by equation (1.2).

$$C_{bid}^{mkt} \leq C_{K,T}^{h|m}(arguments) \leq C_{ask}^{mkt} \tag{1.2}$$

Even though the cost function to be optimized is the same for both HSV and MJD pricing option procedures, the arguments to calibrate are different. sections 1.2.3 and 1.2.4 respectively illustrate the MJD and HSV calibration process and results.

1.2.2 Asset prices based calibration

In order to calibrate the parameters to pass to the MJD and HSV model, to generate all the dummy times series that will serve for the analysis of the delta hedging, I am going to use the historical market data on the Apple stock as a template.

To perform such an upstream analysis, I will use an approximation method to estimate the arguments with which the distribution of the log-returns generated by both MJD and HSV models better fit the density curve of the historical log-return. The list of arguments to estimate will be smaller than the one needed to option calibration because I will start with the set of the already assessed parameters from options data and only (re)calibrate those risk-neutralized.

To get the historical data, I will once more use the package *quantmod*, developed by Ryan and Ulrich [2018] which uses Yahoo! finance as a provider. The dataset so

downloaded is available in appendix **??** and concerns the daily stock price from 18th May 2017 to 18th May 2018. figure 1.1 shows the density curve generated by the log-return of the historical data.

Figure 1.1: Historical Apple stock Log-returns distribution

The above densities function is constructed over the historical data of the Apple share of stock price evolution from 18th May 2017 to 18th May 2018.

The procedures to calibrate the arguments for MJD and HSV processes are respectively explained at sections 1.2.3 and 1.2.4.

1.2.3   Merton's model calibration

1.2.4   Heston's model calibration

Calibration of parameters for option pricing

In order to use the function *hsv_call* based on **??** to compute the price of European call options on an underlying with increments driven by the HSV model, the parameters $(V(0), \kappa, \theta, \sigma, \rho)$ must be calibrated with respect to the available option market data. Consequently, the so estimated parameters will be in line with the risk-neutral measure, or more specifically κ and θ, which are the only ones adapted to make **??** risk-neutral.

To do so, I am going to use the aforementioned least-square non-linear analysis together with data on Apple call option as a template. In that respect, the theoretical models to calibrate are given by **??**–**??** along with **??**, implemented by the R function *hsv_call*.

Moreover, to stay in a range of acceptable values, the parameters of the HSV model should lie between some defined boundaries.

De facto, the mean-reversion speed (κ), the long-run variance (θ) and the volatility of the volatility (σ) need to take values that together respect the Feller's condition (see **??**). Additionally, σ should range between 0 and 1 and, according to **?** κ should be positive to avoid mean aversion. Although the correlation coefficient ρ may take any value from

$[-1, 1]$, typically the correlation between the stock price increments and their intrinsic volatility is negative. Consequently, I set the borders of ρ as being $]0, -1[$.

According to those constraints and due to the presence of multiple local minima, the *lsqnonlin* function returns the table 1.4 as the whole sets of parameters that make the function *hsv_call* better fit with reality.

Table 1.4: Best estimates for HSV call option model

v0	theta	sigma	rho	kappa
0.03835	0.05144	0.30882	-0.40228	1.99821
0.03851	0.05142	0.29539	-0.49338	1.99931
0.03850	0.05241	0.30601	-0.59979	1.99843
0.03774	0.04832	0.30687	-0.40213	3.00036
0.03827	0.04954	0.40690	-0.40189	3.00011
0.03883	0.04798	0.30747	-0.50211	3.00015
0.03910	0.04948	0.40736	-0.50193	3.00005
0.03695	0.04703	0.30697	-0.40288	4.00092
0.04449	0.04443	0.40682	-0.40289	4.00079
0.03798	0.04872	0.50379	-0.39878	4.00106
0.04091	0.04651	0.40705	-0.50227	4.00097
0.03740	0.04754	0.30583	-0.60126	4.00096
0.03890	0.04792	0.40565	-0.60086	4.00091
0.03661	0.04604	0.30659	-0.40295	5.00152
0.04034	0.04547	0.40591	-0.40298	5.00153
0.04111	0.04558	0.50652	-0.40298	5.00144
0.04034	0.04720	0.60647	-0.40250	5.00138
0.03921	0.04709	0.50753	-0.50170	5.00152
0.03789	0.04656	0.30488	-0.80072	5.00164

Otherwise, The set 1.3 is the one making respond the model the best with what is observed in reality. Indeed when passing that set to the R function *call_heston*, for all strikes of table 1.3 and maturities of table 1.2, more than 70% of the so generated prices are within the bid-ask spread of the Apple option historical data.

$$\left\{ \begin{array}{rcl} V(0) & = & 0.03798218, \\ \theta & = & 0.04871543, \\ \sigma & = & 0.50378803, \\ \rho & = & -0.39877827, \\ \kappa & = & 4.00105546 \end{array} \right\} \tag{1.3}$$

figure 1.2 confronts the blue colored volatility smiles computed from market data, with those dotted in red, calculated from the function *hsv_call*, which takes the items of the set 1.3 as parameters.

Figure 1.2: Implied volatility of Apple option prices computed with HSV
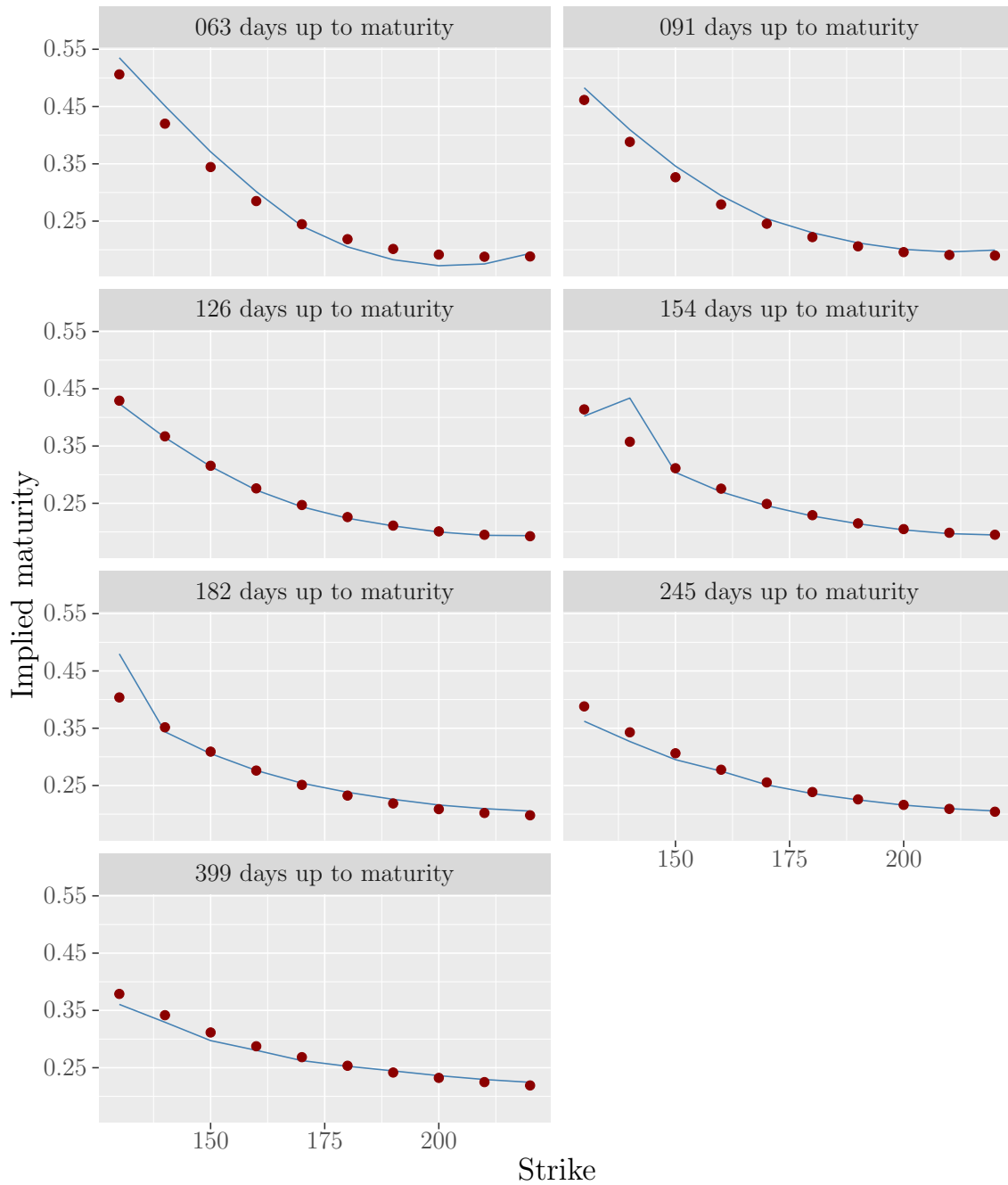
The blue curves represent the implied volatility computed from the option market data on Apple while the ....

Consequently, the set 1.3 is the one I will use together with the *hsv_call* function whenever I have to compute option price using the HSV model on Apple data for the purpose of that master thesis.

Calibration of parameters for time series

The set 1.3 is calibrated under the risk-neutral world and cannot, therefore, be used as is to simulate the dummy time-series that will be used in the analysis aimed to measure

the delta hedging performance. Furthermore, in addition to those parameters, the drift rate α has to be estimated as well. It was not present in the set 1.3 because it is not taken into account in the computation of the options prices.

Figure 1.3 shows the empirical density curves illustrating the distributions of the log-returns computed either from historical Apple stock data for the blue curve or from dummy time series generated by the function $hsv\_ts()$ fed with the risk-neutral parameters1.3, for the red one.

Figure 1.3: Historical and HSV related Apple stock Log-returns distribution
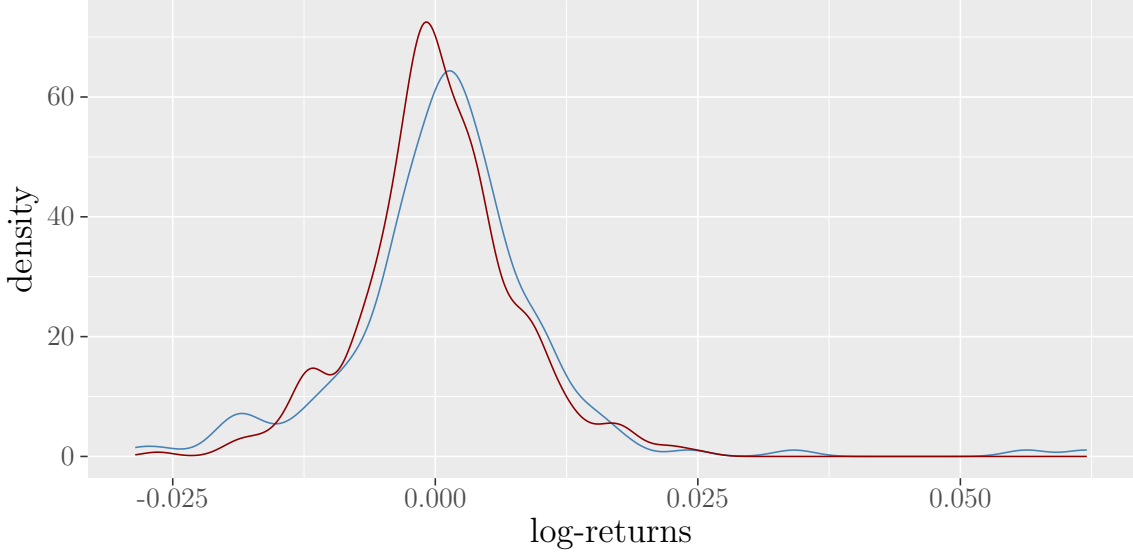
The above blue density curve is constructed over the historical data of the Apple share of stock price evolution from 18th May 2017 to 18th May 2018. while the red curve is constructed from time-series genereated by the function hsv_ts taking the risk-neutral parameters 1.3 as arguments.

The goal of that section is to find the risk-averse parameters that make both the distributions of the log-returns generated from market data or from hsv_ts fit together. To do so, and according to the differences between **????** from **????**, the parameters to modify are the drift rate $r \rightarrow \alpha$, the mean reversion speed $\kappa^* \rightarrow \kappa$ and the long-run volatility $\theta^* \rightarrow \theta$. Accordingly, the correlation parameter ρ and the volatility of the volatility σ stay unchanged from the risk-neutral to risk-averse world.

I followed the method exhibited at **??** to get a rough estimation of the drift rate α. From the log-returns' first and second moments, I find the value 0.4822917 for the estimate of α.

According to **????**, in order to calculate κ and θ, the risk premium λ has to be guessed. Practically, to do so, I am using an approximation algorithm developed by Venables and Ripley [2002] which is directly available in the R language through the function *fitdistr* from the R package *MASS*. The goal of that algorithm is the find the parameters of a density function that make it reproduce the distribution of a sample of data. To do the job, that algorithm needs (i) a density function to be fitted along with (ii) a sample of random data as a template. As defined in Dragulescu and Yakovenko [2002], the density of the log-returns generated by the HSV model is given by equation (1.4).

9

$$P_t(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\phi x + F_t(\phi)} d\phi \tag{1.4}$$

where

$$F_t(\phi) = \frac{\kappa\theta}{\sigma^2}\gamma t - \frac{2\kappa\theta}{\sigma^2} \ln\left(\cosh\frac{\Omega t}{2} + \frac{\Omega^2 - \gamma^2 + 2\kappa\gamma}{2\kappa\Omega} \sinh\frac{\Omega t}{2}\right)$$

and

$$\Omega = \sqrt{\gamma^2 + \sigma^2(\phi^2 - i\phi)}, \gamma = \kappa + i\rho\phi\sigma$$

By applying the optimization function together with equation (1.4) as a density function, the Apple stock data [REF] as a template and the variable λ as a cursor, the optimization algorithm outputs $\lambda = 5.4883278229$ as the best fit argument for the risk premium. Therefore, the set of the risk-averse parameters is fully described here below in 1.5.

$$\left\{ \begin{array}{lcl} V(0) & = & 0.03798218, \\ \theta & = & 0.02054013, \\ \sigma & = & 0.50378803, \\ \rho & = & -0.39877827, \\ \kappa & = & 9.489383, \\ \alpha & = & 0.4822917 \end{array} \right\} \tag{1.5}$$

Figure 1.4 shows the empirical density curves illustrating the distributions of the log-returns computed either from historical Apple stock data for the blue curve or from dummy time series generated by the function *hsv_ts()* fed with the risk-averse parameters1.5, for the red one.

Consequently, the set 1.5 is the one I will use together with the *hsv_ts()* function whenever I have to compute time series using the HSV model within the real world constraints for the purpose of that master thesis.

1.3 Delta hedging

In order to measure the hedging performance of the MJD and HSV models, I will (i) compute the prices of some options and then virtually sell them at that price, (ii) construct the delta-neutral portfolio at time zero and thereafter continuously rebalance it, in accordance with the underlying asset price evolutions and (iii) measure the overall cost of the hedge as a performance indicator.

1.3.1 Determination of the option price

The computation of the option price at time zero will depend on the underlying model to be hedged. Indeed, the function *mjd_call* will be used to price options for which the underlying asset price evolution is driven by the MDJ model, whereas *mjd_call* will be used for HSV processes.

Figure 1.4: Historical and HSV related Apple stock Log-returns distribution in the risk-averse world
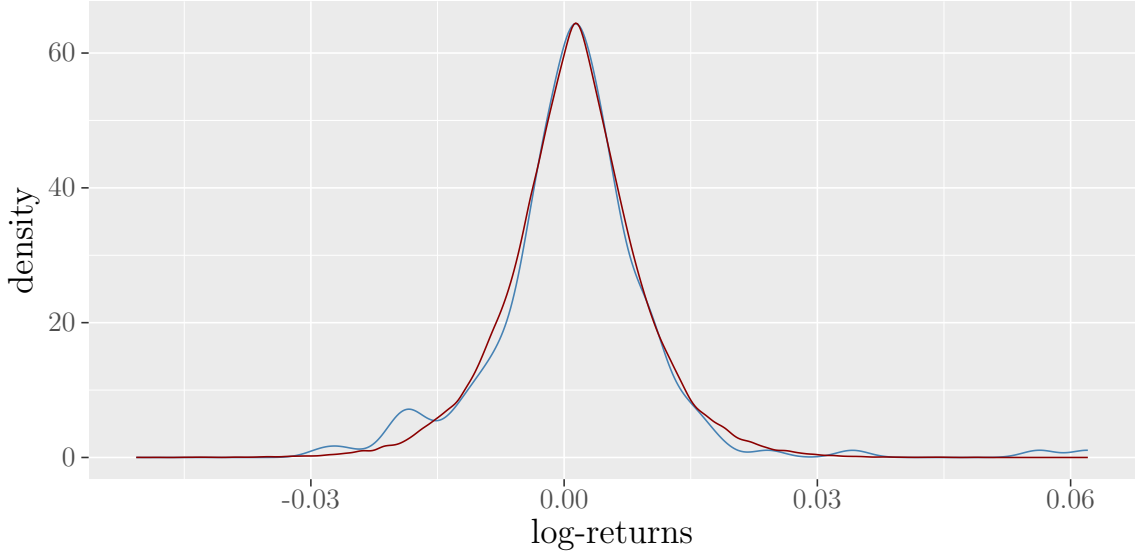
The above blue density curve is constructed over the historical data of the Apple share of stock price evolution from 18th May 2017 to 18th May 2018. while the red curve is constructed from time-series genereated by the function *hsv_ts* taking the risk-averse parameters 1.5 as arguments.

As explain as an introduction, I will virtually sell the options I price. Therefore, as the writer of the options, I will raise \$ $X(0) = C_K^{m/h}(0)$. That cash will directly be put into a money market account to make it grow at a constant riskless rate all along the duration of the hedging process, namely, during the whole life of the option.

The T-bills will serve as riskless rate. According to the available data on `https://www.treasury.gov/`, quotes of these are given by table 1.5.

Table 1.5: Treasury bill quotes on 18 May 2018

T-bills			
4 weeks	13 weeks	26 weeks	52 weeks
1.66%	1.90%	2.09%	2.30%

The rates of table 1.5 are annual-base, consequently to get their countinuously compounded counterpart, EQUATION X can be used to give those given by table 1.6.

Table 1.6: Treasury bill quotes on 18 May 2018

T-bills			
4 weeks	13 weeks	26 weeks	52 weeks
1.659%	1.896%	2.068%	2.274%

Thereafter by applying the bootstrap method along with linear interpolation, one can find the maturity-wise related riskless rates of the Apple call option used for the current analysis. Those rates are given by table 1.7.

Table 1.7: Maturities explored during the hedging performance measurement

Maturity(in days)	63	91	126	154	182	245	399
Riskless rate	1.817%	1.896%	1.962%	2.015%	2.068%	2.139%	2.311%

1.3.2 Construction of the delta-neutral portfolio

At time zero, the built of the delta-neutral portfolio is achieved by the purchase of $\Delta(0)$ share of stock multiplying by the number of assets involving in the derivative product. Simply put, if one wants to construct such a portfolio to cover against the possible losses incurred by the selling of a call option on one hundred shares, the number of underlying to buy at that time would be of $\Delta(t) \times 100$. For the sake of clarity, in the analysis I will perform, it is implicit that any European call option involves the purchase of only one underlying at maturity, though. Consequently, at time zero, the cost of such a portfolio is given by equation (1.6).

$$p(t_0) = \Delta^{m/h}(t_0)S(t_0) \tag{1.6}$$

Where the superscripts m (Merton) and h (Heston) respectively stand for the computation of MJD and HSV delta.

The objective of that portfolio is to accurately replicate the reverse position taken in a derivative, i.e., to reproduce a position in a long European call in the case that matters for that analysis. To be efficient in that role, it must be frequently rebalanced.

Accordingly, in order to do so, I will observe the underlying's time-series at every period of time $\delta t = 1 days$ and compute the updated value of $\Delta(t)$ at each of those observational timesteps. Therefore, the requisite expenses aimed to keep the portfolio delta-neutral will evolve over time, as shown here below by equation (1.7).

$$p(t_i) = \Delta^{m/h}(t_i - t_{i-1})S(t_i) \tag{1.7}$$

Where

$$i \in \mathbb{Z} : i \in \{1, T\}$$

The aforementioned time-series that will be daily scrutinized are obviously hypothetical. They will be generated by the functions mjd_ts and hsv_ts for which the risk-averse parameters will be passed in.

1.3.3 Computation of the delta

By definition, the delta is the first derivative of the option's pricing function with respect to the stock price. It consequently, represents the instantaneous rate of change in the option value as the price of its underlying evolves, such so illustrated by **??**.

Although Black and Scholes [1973] gives to delta the value $N(d_+(\Delta t, x))$, with d_+ explained by **??**, latter can only be used within the framework developed by Black and Scholes [1973], that is, if the underlying is driven by a geometric Brownian motion, the volatility is deterministic, or if no jump occurs. Accordingly, the solution to compute the

delta provided by Black and Scholes [1973] cannot be used for such the MJD and HSV models.

By following the definition of delta and because the pricing function of a European call option, for which the underlying asset price evolution is driven by either the MDJ or HSV models, is given by **??**, then, the equation (1.8) gives the solution to compute the delta for such models.

$$\Delta^{m/h}\left(t, S(t)\right) = P_1^{m/h} + S(t)\frac{\partial P_1^{m/h}}{\partial S(t)} - e^{-r(T-t)}K\frac{\partial P_2^{m/h}}{\partial S(t)} \tag{1.8}$$

where

$$\frac{\partial P_0^{m/h}}{\partial S(t)} = \frac{1}{\pi}\int_0^\infty Re\left[\frac{i\phi\exp\left(-i\phi\ln K\right)}{-\phi^2\psi^{m/h}(-i)^2}\times\right.$$
$$\left.\left(\frac{\partial\psi^{m/h}(\phi-i)}{\partial S(t)}\psi^{m/h}(-i) - \psi^{m/h}(\phi-i)\frac{\partial\psi^{m/h}(-i)}{\partial S(t)}\right)\right]d\phi \tag{1.9}$$

and

$$\frac{\partial P_2^{m/h}}{\partial S(t)} = \frac{1}{\pi}\int_0^\infty Re\left(\frac{\exp\left(-i\phi\ln K\right)}{i\phi}\frac{\partial\psi^{m/h}(\phi)}{\partial S(t)}\right)d\phi \tag{1.10}$$

Bibliography

Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.

Ricardo Crisóstomo. An analysis of the heston stochastic volatility model: Implementation and calibration using matlab. *arXiv preprint arXiv:1502.02963*, 2015.

Adrian A Dragulescu and Victor M Yakovenko. Probability distribution of returns in the heston model with stochastic volatility. *Quantitative Finance*, 2(6):443–453, 2002.

Steven L Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2):327–343, 1993.

Robert C Merton. Option pricing when underlying stock returns are discontinuous. *Journal of financial economics*, 3(1-2):125–144, 1976.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL `https://www.R-project.org/`.

Jeffrey A. Ryan and Joshua M. Ulrich. *quantmod: Quantitative Financial Modelling Framework*, 2018. URL `https://CRAN.R-project.org/package=quantmod`. R package version 0.4-13.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL `http://www.stats.ox.ac.uk/pub/MASS4`. ISBN 0-387-95457-0.