

# Contents

<b>Introduction</b>	<b>2</b>
<b>1 Methodology</b>	<b>3</b>
1.1 Calibration . . . . .	3
1.1.1 Option prices based calibration . . . . .	4
1.1.2 Asset prices based calibration . . . . .	5
1.1.3 Merton's model calibration . . . . .	6
1.1.4 Heston's model calibration . . . . .	9
1.2 Delta hedging . . . . .	12
1.2.1 Determination of the option price . . . . .	13
1.2.2 Construction of the delta-neutral portfolio . . . . .	14
1.2.3 Computation of the delta . . . . .	14
1.2.4 Measuring the hedging performance . . . . .	16

# Introduction

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

Using R. R Core Team [2017]

# Chapter 1

## Methodology

The objective of this master thesis is to measure the performances of the Black-Scholes-Merton (BSM) pricing method when the assumption of normality for the log-returns distribution is not met. The analysis will be focused on the Apple stock data and its related calls options.

The methodology followed to measure such performances consists of three main parts. First, The computation of the options price by using, in turn, the BSM, Merton mixed jump-diffusion (MJD) and Heston stochastic volatility (HSV) models. After that, the implied volatility of those computed prices, split by models and maturities, are going to be compared. Finally, the construction of delta-neutral portfolios in order to measure the hedging performances of the models mentioned above.

Instead of exclusively using market data, whether to collect the options prices or the stock time-series, theoretically based algorithms have been purposely created. Depending on the framework to explore, the options prices will be either computed by using the BSM equation (??), if the underlying process relates to a geometric Brownian motion or by using the method developed by Heston [1993] (??), if the underlying process relates to the model MJD or HSV. In order to assess the delta-neutral portfolio, time-series will be needed to measure its evolution across time. Those series will be simulated based on the theories developed by Merton [1976] and Heston [1993] to respectively obtain paths with jumps and others with stochastic volatility as a correlated process. Consequently, functions have been created in the R language to perform those tasks. Table 1.1 is a summary of a few used in that chapter and in the analysis. More details on them are given in appendix ??.

Even though the hedging won't directly be constructed on market data, they are needed to calibrate the parameters to pass into the functions above listed in table 1.1. Therefore, the functions with the objective to calculate the European calls prices will be calibrated with the Apple European calls market data and, to be consistent, those that simulate time-series will be adjusted using the Apple stock data available on the market. These processes of calibration are fully explained in section 1.1.

### 1.1 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 that will be applied in the current section concerns two

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_delta	$\{S(0), T, k, r, \sigma\}$	Compute the BSM delta of an option
mjd_delta	$\{S(0), T, k, r, \lambda, \mu, \delta\sigma\}$	Compute the Merton delta of an option
hsv_delta	$\{S(0), T, k, r, V(0), \theta, \kappa, \sigma, \rho\}$	Compute the heston delta 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

distinctive groups of parameters. Those intended to the functions that compute the European call price and those used by the functions that simulate the possible stock market movements. Consequently, the methods to adjust 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 in a risk-averse world.

Therefore, the market available options prices data will be used to calibrate the functions that compute options prices, while those on the Apple stock's prices evolution will serve to estimate the right values to be passed as parameters 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 a provider. The datasets so downloaded are available in appendix ??.

Section 1.1.1 explains how to operate for the options' parameters, whereas section 1.1.2 shows the followed procedure to adjust the assets simulations' arguments.

### 1.1.1 Option prices based calibration

In accordance with Heston [1993] and Crisóstomo [2015], provided that the characteristic functions of the MJD and HSV models are known, the European calls prices of such underlying processes can be computed using ??. Although known, these characteristic functions (????) need that their parameters are fed with appropriate values to best fit with what is observed in reality. That is why both functions will be adjusted with referential values before being used.

To do so, the method followed by Crisóstomo [2015] will be applied. It consists of finding the arguments that minimize the square of the difference between the options prices provided by the market and those generated by a pricing function. Such a procedure can be resource consuming because it is iterative. Consequently to avoid endless computations and since the market comes with a large number of data, the maturities, and strikes listed in tables 1.2 and 1.3 will be those considered during the analysis.

As for Crisóstomo [2015], the optimization method used is the least-square non-linear analysis. To work, such a method needs (i) a function that returns dummy data – i.e., the one to be adjusted –, (ii) a dataset that serves as a template, (iii) a cost function

Table 1.2: Maturities took into account during the calibration stage

63 91 126 154 182 245 399

Table 1.3: Strikes took into account during the calibration stage

130 140 150 160 170 180 190 200 210 220

to minimize, and (iv) a set of parameters to optimize. To perform that analysis, an algorithm developed by Borchers [2018] is directly available in the R language through the function *lsqnonlin* from the R package *pracma*.

The functions that output the artificial data are those exhibited in table 1.1, that is to say, and for what matters here, *mjd\_call* and *hsv\_call* for the computation of European calls prices using MJD and HSV, respectively. The template of options data is available 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).

$$cst = \sum_{k \in K} \sum_{t \in T} (C_{k,t}^{mkt} - C_{k,t}^{m \oplus h}(arguments))^2, \quad (1.1)$$

$$with K = \{130, 140, 150, 160, 170, 180, 190, 200, 210, 220\},$$

$$T = \{63, 91, 126, 154, 182, 245, 399\}$$

On the one hand, The subscripts  $K$  and  $T$  respectively stand for the sets of strike prices and maturity dates, while on the other hand, the superscript  $mkt$  denotes the "market price" whilst the one  $m \oplus h$  refers to either MJD (m) or HSV (h) process based prices.

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

One difficulty when dealing with such an algorithm is that the least-square non-linear analysis 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 select only one set among all the results provided by the optimization method, will be chosen that with which, at strikes and maturity dates equal, the pricing function gives the more of its outputs within the bid-ask spread of the corresponding options whose price is giving by the market, as shown by equation (1.2).

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

Even though the cost function to be optimized is the same for both HSV and MJD options pricing procedures, the parameters to adjust are different. sections 1.1.3 and 1.1.4 respectively illustrate the MJD and HSV related calibration processes along with the associated results.

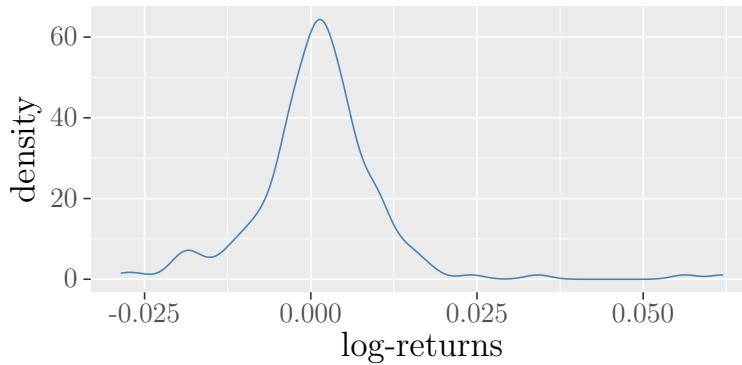
### 1.1.2 Asset prices based calibration

To calibrate the parameters to pass to the MJD and HSV models aimed to generate all the dummy times-series that will serve for the analysis of the delta hedging, historical market data on the Apple stock will be used as a template.

To perform such an upstream analysis, 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-returns will be applied. The list of arguments to adjust is smaller than the one needed to option calibration because the starting point is the set of the already calibrated parameters gotten from options data and will only be readjusted, those they are 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 on Apple from 18th May 2017 to 18th May 2018. figure 1.1 shows the density curve generated by the log-returns of historical data.

Figure 1.1: Historical Apple stock Log-returns distribution



**Notes.** The above density function is constructed over the historical data of the Apple share of stock price evolution from 18th May 2017 to 18th May 2018. It represents the empirical distribution of its daily log-returns observed during that period.

The functions that output the dummy time-series are those exhibited in table 1.1, namely, *mjd\_ts* and *hsv\_ts* for the generation of MJD and HSV processes, respectively. The procedures to calibrate the arguments for those functions are explained in sections 1.1.3 and 1.1.4.

### 1.1.3 Merton's model calibration

#### Calibration of parameters for option pricing

In order to use the function *mjd\_call* based on ?? to compute the price of European call options on an underlying with increments driven by the MJD model, the parameters  $\{\lambda, \mu, \delta, \sigma\}$  must be calibrated with respect to the available market data.

To do so, the aforementioned least-square non-linear analysis will be used 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 *mjd\_call*.

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

Indeed,  $\sigma$  and  $\delta$ , as standard deviation parameters, must range between 0 and 1. Otherwise, no limit is fixed for both  $\mu$  and  $\lambda$ , letting the frequency of jumps and their average magnitudes fully free.

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

Table 1.4: Best estimates for MJD call option model

lambda	mu	delta	sigma
0.07915	-0.37583	0.08771	0.18651
0.11606	-0.30724	0.09993	0.18445
0.14411	-0.24174	0.12023	0.18676
0.06839	-0.37680	0.18072	0.18870
0.10312	-0.29738	0.19545	0.18583
0.05698	-0.38007	0.28030	0.19043
0.08216	-0.28573	0.27560	0.18822

However, the set of arguments 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\_merton*, for all strikes and maturities from tables 1.2 and 1.3, more than 68% of the so generated prices are within the bid-ask spread of the Apple option historical data.

$$\left\{ \begin{array}{l} \lambda = 0.1031218, \\ \mu = -0.2973769, \\ \delta = 0.1954519, \\ \sigma = 0.1858289 \end{array} \right\} \quad (1.3)$$

Figure 1.2 confronts the blue colored volatility smiles computed from market data, with those dotted in red, calculated from data provided by the function *mjd\_call*, which takes the items of the set 1.3 as parameters.

Consequently, for the purpose of that master thesis, the set 1.3 is the one that will be used together with the function *mjd\_call* whenever performing the computation of option prices for the MJD model.

### Calibration of parameters for time series

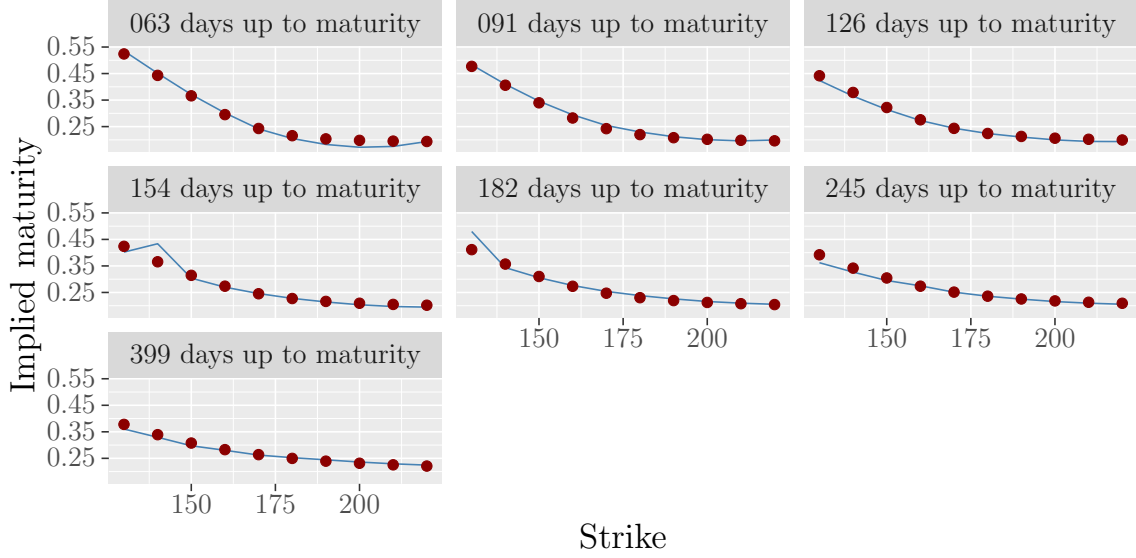
The set 1.3 is calibrated under the risk-neutral world and cannot, therefore, be used as is to simulate the dummy time-series that will serve the analysis aimed to measure the delta hedging performance. Furthermore, in addition to those parameters, the drift rate  $\alpha$  has to be estimated as well. It was not present in the set 1.3 because it was 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 *mjd\_ts()* fed with the risk-neutral parameters 1.3, for the red one.

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 *mjd\_ts* fit together.

Practically, to do so, 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* will be purposely used. The goal of that algorithm is the find the parameters of a density function that make it reproduce the distribution of a sample of

Figure 1.2: Implied volatility of Apple option prices computed with MJD



**Notes.** The implied volatilities that represent the above volatility smiles have been computed by using an iterative method on the BSM equation to solve  $\sigma$ . The parameters given to the function *bsm\_call* were the stock price  $S$ , the strike price  $K$ , the riskless rate  $r$  and the time to maturity  $T$ , letting  $\sigma$  unknown. The method used was to find the root of  $C(S(0), K, T, r; \sigma) - C_{K,T}$ , where  $C_{K,T}$  is either the market price or the price giving by the function *mjd\_call*, and  $C(S(0), K, T, r; \sigma)$  is the function *bsm\_call* with  $\sigma$  as a cursor.

data. To do the job, that algorithm needs (i) a density function along with (ii) a sample of random data as a template. As defined in Matsuda [2004], the density of the log-returns of time-series generated by the MJD model is given by equation (1.4).

$$P_t(x) = \sum_{i=0}^{\infty} \frac{e^{-\lambda t} (\lambda t)^i}{i!} N \left[ x; \left( \alpha - \frac{\sigma^2}{2} - \lambda \kappa \right) t + i\mu, \sigma^2 t + i\delta^2 \right] \quad (1.4)$$

where

$$\kappa = e^{\mu + \frac{1}{2}\delta^2} - 1$$

The term  $e^{-\lambda t} (\lambda t)^i / i!$  gives the probability that  $\lambda$  jumps occur during the time period  $t$ , while  $N$  denotes the normal probability density function.

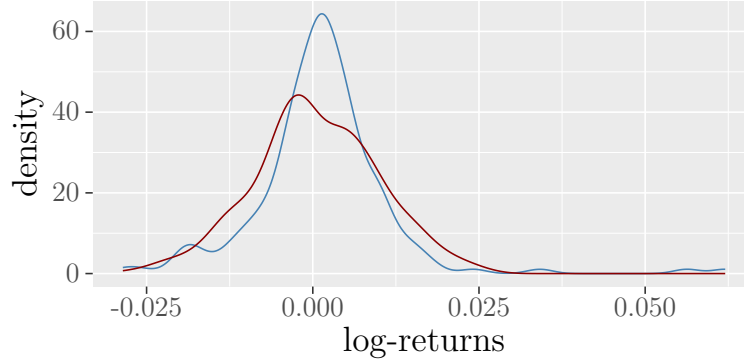
The parameters to be assessed are the drift and variance rates of the stock prices process, namely,  $\alpha$  and  $\sigma$ , the frequency of the jumps occurrence  $\lambda$  and their intensity parameters, that is,  $\mu$  and  $\delta$ . All of those parameters are fully described in ??.

By giving to the optimization function the equation (1.4) as a density function, the Apple stock data ?? as a template and the set of variables  $\{\alpha, \sigma, \lambda, \mu, \delta\}$   $\lambda$  as a cursor, that algorithm outputs the set 1.5 as the best fit arguments for the real world parameters.

$$\left\{ \begin{array}{l} \alpha = 0.4816753642, \\ \sigma = 0.1020368856, \\ \lambda = 99.5434345733, \\ \mu = -0.0006791266, \\ \delta = 0.0161092106, \end{array} \right\} \quad (1.5)$$



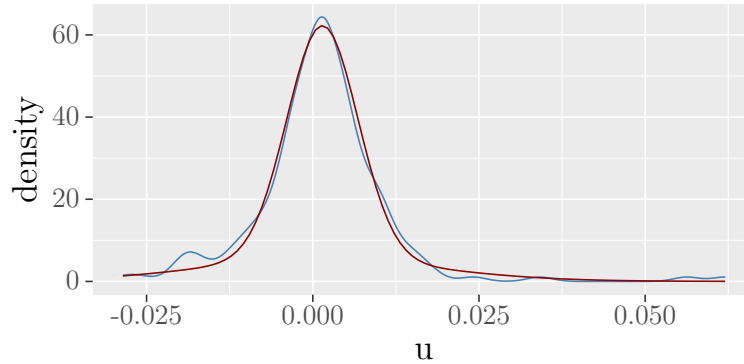
Figure 1.3: Historical and MJD related Apple stock Log-returns distribution



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 generated by the function *hsv\_ts* taking the risk-neutral parameters 1.3 as arguments.

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 *mjd\_ts()* fed with the risk-averse parameters 1.5, for the red one.

Figure 1.4: Historical and MJD 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 generated by the function *mjd\_ts* taking the risk-averse parameters 1.5 as arguments.

Consequently, for the purpose of that master thesis, the set 1.5 is the one that will be used together with the function *mjd\_ts()* whenever performing the computation of time series using the MJD model within the real world constraints.

### 1.1.4 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 market data. Therefore, the so estimated parameters will be in line with the risk-neutral measure, or more specifically  $\kappa$  and  $\theta$ , which are the only ones needed to be adapted to make ?? risk-neutral.

To do so and similarly as it was done for the MJD option model, the least-square non-linear analysis will be used 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 ( $\kappa$ ), the long-run variance ( $\theta$ ) and the volatility of the volatility ( $\sigma$ ) need to take values that together respect the Feller's condition (see ??). Additionally,  $\sigma$  must range between 0 and 1 and, according to ?  $\kappa$  should be positive to avoid mean aversion. Although the correlation coefficient  $\rho$  may take any value from  $[-1, 1]$ , typically the correlation between the stock price increments and their intrinsic volatility is negative. Consequently, the boundaries of  $\rho$  are set as  $]0, -1[$ .

According to those constraints and due to the presence of multiple local minima, the *lsqnonlin* function returns the table 1.5 as the whole sets of parameters that make the function *hsv\_call* better fit with reality.

Table 1.5: Best estimates for HSV call option model

v0	theta	sigma	rho	kappa
0.03851	0.05142	0.29539	-0.49338	1.99931
0.03774	0.04832	0.30687	-0.40213	3.00036
0.03910	0.04948	0.40736	-0.50193	3.00005
0.03798	0.04872	0.50379	-0.39878	4.00106
0.03740	0.04754	0.30583	-0.60126	4.00096
0.03661	0.04604	0.30659	-0.40295	5.00152
0.03921	0.04709	0.50753	-0.50170	5.00152

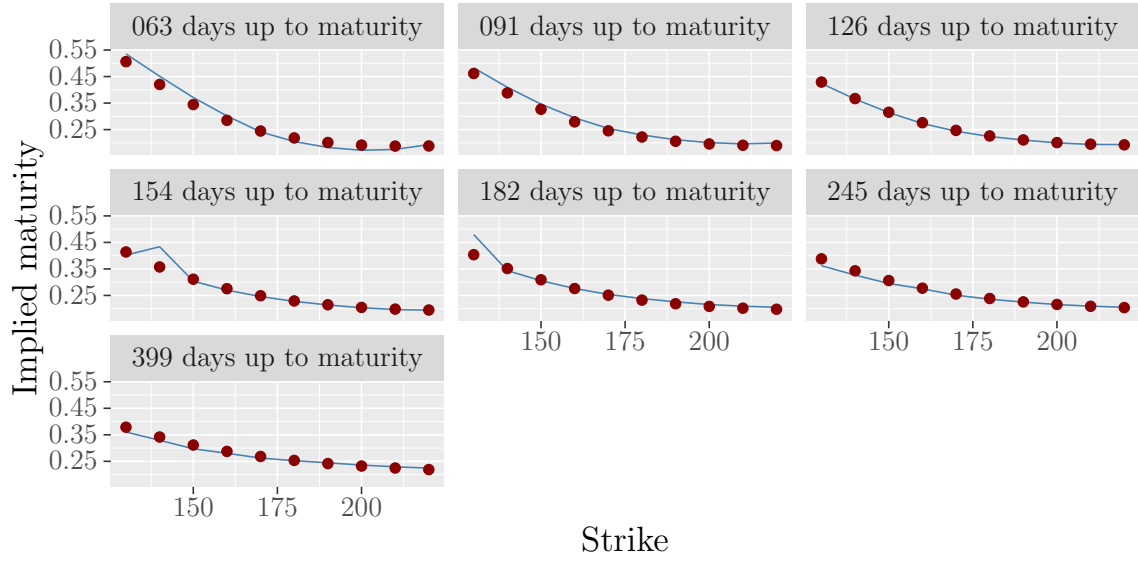
However, the set of arguments 1.6 is the one making respond the model the best with what is observed in reality. Indeed when passing that set to the R function *call\_heston*, for all strikes and maturities from tables 1.2 and 1.3 more than 70% of the so generated prices are within the bid-ask spread of the historical data.

$$\left\{ \begin{array}{lcl} V(0) & = & 0.03798218, \\ \theta & = & 0.04871543, \\ \sigma & = & 0.50378803, \\ \rho & = & -0.39877827, \\ \kappa & = & 4.00105546 \end{array} \right\} \quad (1.6)$$

Figure 1.5 confronts the blue colored volatility smiles computed from market data, with those dotted in red, calculated from data provided by the function *hsv\_call* which takes the items of the set 1.6 as parameters.

Consequently, for the purpose of that master thesis, the set 1.6 is the one that will be used together with the function *hsv\_call* function whenever performing the computation of option prices for the HSV model.

Figure 1.5: Implied volatility of Apple option prices computed with HSV



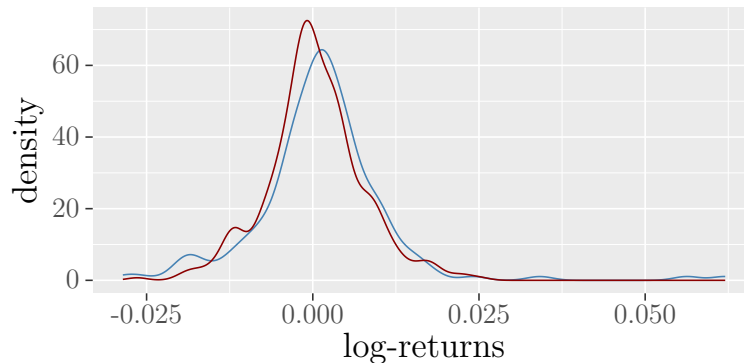
The blue curves represent the implied volatility computed from the option market data on Apple while the .....

### Calibration of parameters for time series

The set 1.6 is calibrated under the risk-neutral world and cannot, therefore, be used as is to simulate the dummy time-series that will serve the analysis aimed to measure the delta hedging performance. Furthermore, in addition to those parameters, the drift rate  $\alpha$  has to be estimated as well. It was not present in the set 1.6 because it was not taken into account in the computation of the options prices.

Figure 1.6 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 parameters 1.6, for the red one.

Figure 1.6: 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 generated by the function *hsv\_ts* taking the risk-neutral parameters 1.6 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 *????* and *????*, 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  $\rho$  and the volatility of the volatility  $\sigma$  stay unchanged from the risk-neutral to risk-averse world.

The drift rate  $\alpha$  has been estimated by following the method exhibited at *??*. From the market empirical log-returns' first and second moments, the value 0.4822917 has been found for  $\alpha$ .

According to *????*, in order to calculate  $\kappa$  and  $\theta$ , the risk premium  $\lambda$  has to be assessed. Practically, such as it was the case for the MJD model, the function *fitdistr* from the R package *MASS* will be purposely used. As defined in Dragulescu and Yakovenko [2002], the density of the log-returns generated by the HSV model is given by equation (1.7).

$$P_t(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\phi x + F_t(\phi)} d\phi \quad (1.7)$$

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 giving to the optimization function the equation (1.7) as a density function, the Apple stock data *??* as a template and the variable  $\lambda$  as a cursor, that algorithm outputs  $\lambda = 5.4883278229$ , as the best fit argument, for the risk premium. The set of the risk-averse parameters is fully described here below in 1.8.

$$\left\{ \begin{array}{lcl} V(0) & = & 0.03798218, \\ \theta & = & 0.02054013, \\ \sigma & = & 0.50378803, \\ \rho & = & -0.39877827, \\ \kappa & = & 9.489383, \\ \alpha & = & 0.4822917 \end{array} \right\} \quad (1.8)$$

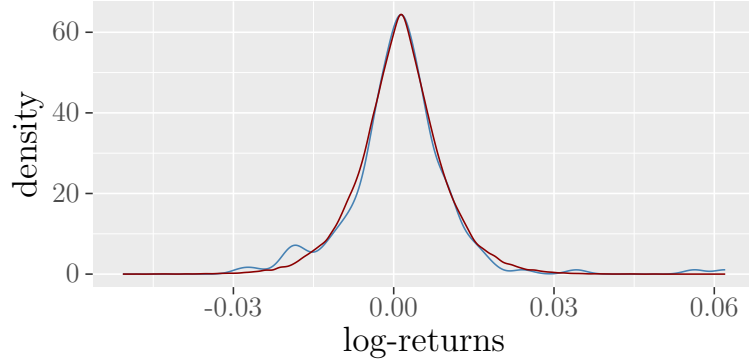
Figure 1.7 shows the empirical density curves illustrating the distributions of the log-returns computed either from historical data for the blue curve or from dummy time-series generated by the function *hsv\_ts()* fed with the risk-averse parameters 1.8, for the red one.

Consequently, for the purpose of that master thesis, the set 1.8 is the one that will be used together with the function *hsv\_ts()* whenever performing the computation of time series using the HSV model within the real world constraints.

## 1.2 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.

Figure 1.7: 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 generated by the function *hsv\_ts* taking the risk-averse parameters 1.8 as arguments.

### 1.2.1 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.

As explain as an introduction, I will virtually sell the options I price. Therefore, as the writer of the options, I will raise \$  $X(0) = C_K^{m\oplus h}(0)$ . 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 1.6.

Table 1.6: 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 1.6 are annual-base, consequently to get their countinuously compounded counterpart, EQUATION X can be used to give those given by table 1.7.

Table 1.7: 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 1.8.

Table 1.8: 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%

### 1.2.2 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 (1.9).

$$p(t_0) = \Delta^{m \oplus h}(t_0)S(t_0) \quad (1.9)$$

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  $\delta t = 1days$  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 (1.10).

$$p(t_i) = \Delta^{m \oplus h}(t_i - t_{i-1})S(t_i), \forall i \in \mathbb{Z} : i \in \{1, T\} \quad (1.10)$$

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.

### 1.2.3 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 (1.11) gives the solution to compute the delta for such models.

$$\Delta^{m\oplus h}(t, S(t)) = P_1^{m\oplus h} + S(t) \frac{\partial P_1^{m\oplus h}}{\partial S(t)} - e^{-r(T-t)} K \frac{\partial P_2^{m\oplus h}}{\partial S(t)} \quad (1.11)$$

where

$$\begin{aligned} \frac{\partial P_1^{m\oplus h}}{\partial S(t)} = \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left[ \frac{i\phi \exp(-i\phi \ln K)}{-\phi^2 \psi^{m\oplus h}(-i)^2} \right. \\ \left. \times \left( \frac{\partial \psi^{m\oplus h}(\phi - i)}{\partial S(t)} \psi^{m\oplus h}(-i) - \psi^{m\oplus h}(\phi - i) \frac{\partial \psi^{m\oplus h}(-i)}{\partial S(t)} \right) \right] d\phi \end{aligned} \quad (1.12)$$

and

$$\frac{\partial P_2^{m\oplus h}}{\partial S(t)} = \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left( \frac{\exp(-i\phi \ln K)}{i\phi} \frac{\partial \psi^{m\oplus h}(\phi)}{\partial S(t)} \right) d\phi \quad (1.13)$$

Hence, the equation (1.11) is the one to solve for both MJD and HSV models to find them the delta at time  $t$ . Equations (1.12) and (1.13) can be solved by finding the first derivatives of the characteristic functions of MJD and HSV processes with respect to the stock price. By the way, it turns out that both of these have the same form. Equation (1.14) gives the solution of these derivatives, the term  $\psi^{m\oplus h}(x)$  have to be replaced by the appropriate characteristic function.

$$\frac{\partial \psi^{m\oplus h}(x)}{\partial S(t)} = \psi^{m\oplus h}(x) \frac{ix}{S(t)} \quad (1.14)$$

By replacing all the derivative operations inside equations (1.12) and (1.13) by equation (1.14), one finds the solved version given by equations (1.15) and (1.16).

$$\frac{\partial P_1^{m\oplus h}}{\partial S(t)} = \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left[ \frac{\exp(-i\phi \ln K) \psi^{m\oplus h}(\phi - i)}{S(t) \psi^{m\oplus h}(-i)} \right] d\phi \quad (1.15)$$

$$\frac{\partial P_2^{m\oplus h}}{\partial S(t)} = \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left[ \frac{\exp(-i\phi \ln K) \psi^{m\oplus h}(\phi)}{S(t)} \right] d\phi \quad (1.16)$$

Consequently, in the extent that the term  $\psi^{m\oplus h}(\phi)$  is replaced by the appropriate characteristic function, substituting equations (1.15) and (1.16) into 1.11 gives 1.17 that allows the computation of the delta of an option for which the underlying asset is driven by either a MJD or HSV process.

$$\begin{aligned} \Delta^{m\oplus h}(t, S(t)) = P_1^{m\oplus h} + \frac{S(t)}{\pi} \int_0^\infty \operatorname{Re} \left[ \frac{\exp(-i\phi \ln K) \psi^{m\oplus h}(\phi - i)}{S(t) \psi^{m\oplus h}(-i)} \right] \\ - \frac{e^{-r(T-t)} K}{\pi} \int_0^\infty \operatorname{Re} \left[ \frac{\exp(-i\phi \ln K) \psi^{m\oplus h}(\phi)}{S(t)} \right] d\phi \end{aligned} \quad (1.17)$$

In that respect, the R functions *mjd\_delta* and *hsv\_delta*, respectively compute the MJD and HSV related deltas at time  $t$  by taking the risk-neutral parameters.

### 1.2.4 Measuring the hedging performance

Finally, to analyze how much the hedging works, one method is to measure its overall cost. To do so, the really first thing is to value at maturity all the buying and selling operations carried out in order to keep delta-neutral the previously constructed portfolio and add them together.

$$p_{tot} = \sum_{i=0}^T p(t_i) e^{T-t_i}, \forall i \in \mathbb{Z} : i \in \{1, T\} \quad (1.18)$$

On the other hand, at maturity, the amount of money  $X(0)$  received when entering into the contract at time zero will have continuously grown up at the constant riskless rate  $r$ , as described by equation (1.19).

$$X(T) = X(0) * e^{rT} \quad (1.19)$$

Incidentally, if the option is in-the-market at maturity date, it first implies that it will be exercised, so increasing the wealth of the short position in the derivative by  $K$ , and then that  $\Delta(T) = 1$ , meaning that the writer of the option has exactly one share of stock in his / her portfolio to honour the transaction. Otherwise, the option is out-of-the-market, inducing that no transaction occurs and that  $\Delta(T) = 0$ . Consequently, at maturity, someone with a short position in the derivative, i.e., the one tempting to be replicated in the current analysis, will receive a positive cash inflow of  $X(T) + \Delta(T)K$ .

Hence, if the hedge works well, the relation exhibited by 1.20 should be respected.

$$\sum_{i=0}^{n=T} P(i) \cong X(T) + \Delta(T) * k \quad (1.20)$$

Because the position could vary, we have to adapt the relation above (REF) in order to remove all unit from the comparison. According to Hull, a measure of performance of the hedge could be to take the ratio of the standard deviation of the cost to hedge to the option price. In that case, the measure could be compared among themselves even if the position taken into the underlying differs.

$$\frac{sd(hedge)}{C_0 e^{rT}} \quad (1.21)$$

where hedge is a vector containing  $n$  sample generated, and  $C_0$  is the price of the call option at time zero value up to maturity.



# Bibliography

- Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.
- Hans W. Borchers. *pracma: Practical Numerical Math Functions*, 2018. URL <https://CRAN.R-project.org/package=pracma>. R package version 2.1.4.
- 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.
- Kazuhisa Matsuda. Introduction to merton jump diffusion model. *Department of Economics. The Graduate Center, The City University of New York*, 2004.
- 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.