R Package Development

Yuze Zhai

January 13, 2023

Contents

1 Introduction **3**

1.1 Abstract . . . 3

1.2 Literature Review . . . 3

2 Concepts **5**

2.1 Financial Options . . . 5

2.1.1 Option Styles . . . 5

2.1.2 Pricing of Option . . . 5

2.2 R . . . 5

2.2.1 R Package Development . . . 5

2.2.2 Objective Oriented Programming in R . . . 5

3 Existing Option Pricing Packages within R Ecosystem **7**

3.1 Packages Review . . . 7

3.1.1 derivmkts . . . 7

3.1.2 fOptions . . . 7

3.1.3 RQuantLib . . . 7

3.2 General Comments . . . 7

4 Package Development **8**

4.1 Package Structure . . . 8

4.1.1 R Scripts . . . 8

4.2 Functions . . . 8

4.3 Deterministic Methods . . . 8

4.3.1 Black-Scholes Model . . . 8

4.3.2 Binomial Lattice Tree . . . 9

4.3.3 Trinomial Lattice . . . 9

4.3.4 Discussion on Multinomial Option Pricing Models & Their Relationship with Binomial Model . . . 9

4.4 Monte Carlo Methods 10
4.4.1 Vanilla Option Pricing 10
4.4.2 Asian Option Pricing 10
4.4.3 Barrier Option Pricing 10
4.4.4 Binary Option Pricing 10
4.4.5 Lookback Option Pricing 10

5 Discussion **11**
5.1 Package Limitation 11

Chapter 1

Introduction

1.1 Abstract

This report discuss the development of the R package `rmcop`, an objective oriented financial option pricing package. The name "rmcop" stands for R Monte Carlo Option Pricing.

In chapter 2, fundamental technical concepts are introduced. These includes explainations on financial options and programming techniques in R that are essential for understanding the development phase described in chapter 4.

In chapter 3, the report enumerates multiple existing R packages for financial options pricing. We will discuss their functionalities and identify the advantages and limitations `rmcop` has comparing to them.

In chapter 4, we give detailed explainations on quantitative models in option pricing and their R implementations. This part will be roughly divided into two sections: deterministic methods and Monte Carlo methods.

In the last chapter, we will explain the current functionalities and limitations of `rmcop` comprehensively, and briefly discuss the possible extension of the package in the future.

1.2 Literature Review

The very first attempt of applying quantitative method in option pricing (perhaps in all finance) is by the French mathematician Louis Bachelier in 1900 [1]. In his paper "The Theory of Speculation," he deducted deterministic formulas for pricing European (vanilla) call and put options as follows:

Being the earlist approach, Bachelier's formula had already outlined the relationship between option price and asset price S, strike price X, and volatility measure σ, which are essential fragments in modern formulas. However, based on limited data, Bachelier's solution was built under some unrealistic assumptions. The normality assumption violates the non-negativity of the stock price, and the formula's discrete measure in time omitted the effect of continuous movements in the stock price. Also, the formula did not discount the effect of interest rate. These errors cause Bachelier's model fails to price options accurately.

It wasn't until 1960s had further improvement been made to quantitative option pricing. In 1961, Case Sprenkle [2] introduced the Sprenkle formula. The formula addressed the above issues by describing the stock price by the more suitable log-normal distribution and discounting for the effect of interest rate, which successfully explained the time value of an option. In the following decade, improvements have been made by scholars such as Boness and Samuelson [3], who introduced emperical constants to increase the effectiveness of Sprenkle's model.

The model was finalised by Black and Scholes in 1973 [3], who explained the stock price movement by General Brownian Motion (GBM). The GBM was described in the form of a Stochastic Differential Equation (SDE), which effectly model the continuity of price movement. The solution (derived through Itô's lemma) of Black-Scholes formula under an risk-neutral approach [1] eliminates the emperical measure of Sprenkle et al's model, which resulted in an objective and deterministic estimation of option price, as is used by most contemporary pricing methods.

In 1979, based on the risk-neutral methods introduced by Black and Scholes, Cox, Ross and Rubinstein [4] introduced a Binomial lattice tree model for modelling stock prices. Unlike the continuous movement described by the Black-Scholes' SDE, the Binomial model discretised the time period of interest into n steps. Initiate from the current price, after passing each time step, the price moves either up or down with fixed ratios throughout the modelling process. By applying the risk-neutral formulation in the Black-Scholes model, one can obtain an risk-neutral probability p for an upward movement (and $1-p$ for the downward movement). By completing the tree "branches" until maturity T and recursing back from each node to current time, one may obtain a deterministic estimation of the option's fair value at current time.

As the financial market develops and more complicated options emerge, in many realistic cases, one cannot find a deterministic solution for pricing option. However, thanks to the advancement of computer power, one can simulate price trajectories for enormous times, and obtain estimate of the stock price and corresponding options' prices accroding to the results of these simulations. Such method is known as the Monte Carlo method. The very first attempt of applying computational method in option pricing is by Phelim Boyle in 1977. More serious (and effective) approach was introduced by Paul Glasserman [5] in the 1990s. Until now, the field of Monte Carlo option pricing is still under active development and is widely used by "quants".

[1] The risk-neutral approach, in simple terms, is constructing a portfolio at a moment in time such that the portfolio value will be identical at the next moment in time regardless of the price movement, so the portfolio will be riskless to the price movement

Chapter 2

Concepts

2.1 Financial Options

2.1.1 Option Styles

2.1.2 Pricing of Option

2.2 R

2.2.1 R Package Development

2.2.2 Objective Oriented Programming in R

Existing packages in R ecosystem provides comprehensive pricing algorithms for financial derivatives. However, their function are based on Procedural Oriented Programming (POP). POP functions can be called directly by passing in required arguments. For simple options pricing cases, such as pricing individual options, using POP functions is intuitive. However, many real scenarios require pricing options in a complex formulation, such as compounded options (i.e. options with underlying assets being another option) and combination of options (i.e. spread, straddle, stringle, and other option strategies). In these situations, managing numerous arguments for POP functions can be difficult and inefficient.

The `rmcop` package proposed an Objective Oriented Programming (OOP) approach for pricing financial options in R. It encapsulates multiple arguments, such as option style, type, strike price, and maturity time, into an `option` class object. It also allows encapsulation of other market environemnt arguments, such as interest rate, dividend yield rate, and volatility measure into an `option.env` class object. The OOP structure enables easier variables managements and facilitates the comparison of prices among different sets of options and market environments.

The codes below demonstrates the calculation of an theoretical European vanilla call option with strike price $K = 20$, maturity $t = 0.75$, under the market condition such that the current price is $S = 20$, fixed interest rate is $r = 1\%$, and market volatility measured by $\sigma = 0.1$. We use Monte Carlo (i.e. `mc`) method with $n = 100$ replications and number of time steps per replication $steps = 1$.

OOP approach requires extra steps declaring objects before using the function, but once the declaration is completed, calling the pricing funcion is much simpler than the POP approach. As above, it only required two (object) arguments, `obj` and `env`.

R provides two ways to perform OOP, the S3 and S4 methods. The S3 class objects are based on R `list` objects. An R list contains a `class` attribute, which can be customised into string (or vector of strings) that can be interpreted as a list's corresponding S3 class (i.e. so that the list itself became an object of that class). Other items within the list can be treated as object's properties under the OOP scope, and can be extracted using the $ operator. Here is an example on how to implement OOP using S3 method in R.

```
John <- list(
    "age" = 20,
    "gender" = "male",
    "nation" = "UK"
)
class(John) <- "student"
```

We first define a new `list` object named "John", which contains three items (age, gender, and nationality). Then, we redefined the class of this list using the `class()` function to "student". Now, we have created a new object named "John" of the class "student" under the S3 scheme.

The S4 methods requires more rigorous class and object definition, as one would typically see within an OOP language such as Python and Java. The development of our pricing functions does not require rigorous OOP structure or defining generic functions, so S3 method is sufficient for its development.

Chapter 3

Existing Option Pricing Packages within R Ecosystem

3.1 Packages Review

3.1.1 derivmkts

3.1.2 fOptions

3.1.3 RQuantLib

An R interface to the QuantLib library, which embeded C++ programming.

3.2 General Comments

Chapter 4

Package Development

4.1 Package Structure

4.1.1 R Scripts

The package consists of 7 R scripts, with their contents defined as below:

File	Contents
`Option.R`	Methods creating and updating option and option.env objects
`Price.R`	Pricing functions takes objects input and calls specific pricing engines
`MonteCarlo.R`	Monte Carlo option pricing method engine functions
`BlackScholes.R`	Black-Scholes option pricing method engine functions
`Binomial.R`	Binomial option pricing method engine functions
`Trinomial.R`	Trinomial option pricing method engine functions
`Tools.R`	Other supplementary functions used in package

For the simplicity of user access, only four functions are exported, they are:

Function	Description
`option()`	Create new `"option"` class object, which represents the option of interest
`option.env()`	Create new `"option.env"` class object, which represents the market environment of interest
`update.option.env()`	Update the variables within a defined `"option.env"` class object
`price.option()`	Pricing the option based on specified option, market environment, and method input

4.2 Functions

4.3 Deterministic Methods

4.3.1 Black-Scholes Model

The movement of asset prices is intuitively described by an Geometric Brownian Motion (GBM). Through 1960s to 1970s, deterministic formula have been driven by numerous scholars.

Introduced by Fisher Black and Myron Scholes in 1973, the Black-Scholes model provides a deterministic method in valuing options under the assumption that the underlying asset's price is described by a Geometric Brownian Motion (GBM).

the Black-Scholes valuation formula provides a deterministic estimation for European option price.

Introduced by Fischer Black and Myron Scholes, the Black-Scholes model perceive the movement of the stock price as an Geometric Brownian Motion (GBM). The general form of the Brownian motion is specified by the following Stochastic Differential Equation (SDE) [5]:

$$\frac{dS(t)}{S(t)} = \mu(S(t), t)dt + \sigma dW(t) \tag{4.1}$$

Where μ is the drift term measuring the "direction" of the price movement at time t, σ measuring the volatility of the motion, and $W(t)$ the Standard Brownian Motion.

4.3.2 Binomial Lattice Tree

Introduced by Cox, Ross, and Rubinstein in 1979, the Binomial Model

4.3.3 Trinomial Lattice

Extending the Binomial model, the Trinomial Lattice model was introduced by Phelim Boyle in 1988 [6].

4.3.4 Discussion on Multinomial Option Pricing Models & Their Relationship with Binomial Model

4.4 Monte Carlo Methods

Due to the complexity of American option pricing using Monte Carlo method, the package (until now) has only includes European option pricing.

4.4.1 Vanilla Option Pricing

4.4.2 Asian Option Pricing

Different from a vanilla option whose payoff only depends on stock price at maturity (i.e. $t = T$), an Asian option's payoff is determined by the average stock price throughout the option's life $\bar{S}$ and the strike price K. To calculate $\bar{S}$, we need to consider the price movement throughout $t \in [0, T]$, making the pricing of an Asian option path-dependent , i.e. we should store the price at any time $t \in [0, T]$ for each simulated price trajectory [7].

A generalised case of the stock pricing model is given by:

$$dS(t) = rS(t)dt + \sigma(S(t))S(t)dW(t) \tag{4.2}$$

4.4.3 Barrier Option Pricing

4.4.4 Binary Option Pricing

4.4.5 Lookback Option Pricing

Chapter 5

Discussion

5.1 Package Limitation

Bibliography

[1] L. Bachelier, "Théorie de la spéculation," *Annales scientifiques de l'École normale supérieure*, vol. 17, pp. 21–86, 1900. Earliest attempt of quantitative method in option pricing.

[2] C. M. Sprenkle, "Warrant prices as indicators of expectations and preferences," *Yale economic essays*, vol. 1, pp. 179–232, 1961.

[3] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *The Journal of political economy*, vol. 81, pp. 637–654, 1973. Black-Scholes Model.

[4] J. C. Cox, S. A. Ross, and M. Rubinstein, "Option pricing: A simplified approach," *Journal of financial economics*, vol. 7, pp. 229–263, 1979. Binomial Lattice Model.

[5] P. Glasserman, *Monte Carlo methods in financial engineering.* Springer, 2003. Explaination on Monte Carlo method, MC European vanilla option pricing, MC Asian option pricing.

[6] P. P. Boyle, "A lattice framework for option pricing with two state variables," *Journal of Financial and Quantitative Analysis*, vol. 23, 3 1988. Trinomial Lattice Model.

[7] D. Higham, *An Introduction to Financial Option Valuation.* Cambridge University Press, 2004. Exotic Options Pricing.