Development Report of R Financial Option Pricing Package `rmcop`

Yuze Zhai

January 20, 2023

Contents

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 this chapter 1, we will briefly introduces the function of `rmcop` in Section 1.2. The literature review below in Section 1.3 will discuss the option pricing models that are included in the package, accompanying by some formula deductions. The implementation of them using R programming are explained in details in Chapter 3.

In Chapter 2, we will introduce three existing option pricing packages in R (`derivmkts`, `fOptions`, `RQuantLib`), and explain their functionalities. We will also discuss their advantages and limitations.

In Chapter 3, we will discuss the package structure of `rmcop`. Section 3.1 explains R's Objective Oriented Programming techniques that are used in developing the package. Section 3.2 outlines the R scripts and functions the package contains and their respective purposes. Section 3.3 will detailedly explain the implementation of the option pricing models in R.

In the last chapter, Chapter 4, we will illustrate `rmcop`'s functionalities by providing examples of R code as well as results' intrepretations under Section 4.1. Finally in Section 4.2, we will conclude the report by a discussion on `rmcop`'s limitations and plans for further development.

1.2 Package Description

`rmcop` is an R package used for pricing financial options. The name "rmcop" stands for "R Monte Carlo Option Pricing". The package supports Monte Carlo based pricing for European vanilla options and several European exotic options, as well as some deterministic methods for pricing European and American vanilla options, which will be discussed in more details in Chapter 3.

1.3 Literature Review

1.3.1 Pricing of Financial Options

A financial option is a common derivative[1] in the market. A option can be generally be classified into two types, either "call" or "put". An (European) call option, by definition [1], "gives its holder the right (but not the obligation) to purchase from the writer a prescribed asset for a prescribed price (strike price) at a prescribed time (maturity / expiration) in the future." In constst, an (European) put option gives the right to sale an asset.

The two most common styles of financial options are European and American. An European option can only be exercised at maturity, whereas an American option allows exercising at anytime prior to maturity (which is a more complex setting for option pricing).

The most typical options are so called "vanilla options", which includes no special or unusual features. For vanilla cases, an European's call option has a payoff of $(S_T - K)^+ := \max(S_T - K, 0)$, and an European put option has a payoff of $(K - S_T)^+ := \max(K - S_T, 0)$ [2] at the point of exercise T (i.e. maturity). Here, S_t is the underlying asset price at time $t \in [0, T]$, and K is the option's strike price.

Options with "special or unusual features" are called "exotic options". The exotic options whose pricing are supported by `rmcop` is introduced as follows based on the definitions given in *An Introduction to Financial Option Valuation* [1].

- **Asian Options** Asian options' payoff are determined by the average price of the underlying asset.
  - An average price Asian call option has payoff at the expiry T given by $\max(\bar{S} - K)$.
  - An average price Asian put option has payoff at the expiry T given by $\max(K - \bar{S})$.
  - An average strike Asian call option has payoff at the expiry T given by $\max(S_T - \bar{S})$.
  - An average strike Asian put option has payoff at the expiry T given by $\max(\bar{S} - S_T)$.
- **Barrier Options** Barrier options have a payoff that switches on or off depending on whether the asset crosses a pre-defined level (barrier) B.
  - A down-and-out call option has a payoff that is zero if the asset crosses some predefined barrier $B < S_0$ at some time in $[0, T]$. If the barrier is not crossed then the payoff becomes that of a European call, $\max(S_T - K, 0)$.
  - A down-and-in call option has a payoff that is zero unless the asset crosses some predefined barrier $B < S_0$ at some time in $[0, T]$. If the barrier is crossed then the payoff becomes that of a European call, $\max(S(T) - K, 0)$.
- **Binary Options** A binary (a.k.a. cash-or-nothing) option have payoff at expiry being either some specified value A or zero.
  - A binary call option has payoff A if $S_T > K$ and zero otherwise.
  - A binary put option has payoff A if $S_T < K$ and zero otherwise.
- **Lookback Options** The payoff for a lookback option depends upon either the maximum $S^{\max}$ or the minimum value $S^{\min}$ attained by the asset throughout the price trajectory.
  - A fixed strike lookback call option has payoff at expiry T given by $\max(S^{\max} - K, 0)$.

[1] A financial derivative is a security whose value depends on an the value of an underlying (i.e.corresponding) asset/stock.

– A fixed strike lookback put option has payoff at expiry T given by $\max(K - S^{\min}, 0)$.
– A floating strike lookback call option has payoff at expiry T given by $S_T - S^{\min}$.
– A floating strike lookback put option has payoff at expiry T given by $S^{\max} - S_T$.

1.3.2 Black-Scholes Model

The very first attempt of applying quantitative method in option pricing (perhaps in all finance) is by the French mathematician Louis Bachelier in 1900 [3]. In his paper "The Theory of Speculation," he deducted deterministic formulas for pricing European (vanilla) call and put options as follows:

$$C(S,T) = SN(\frac{S-X}{\sigma\sqrt{t}}) - XN(\frac{S-X}{\sigma\sqrt{t}}) + \sigma\sqrt{t}N(\frac{S-X}{\sigma\sqrt{t}}) \tag{1.1}$$

$$P(S,T) = XN(\frac{S-X}{\sigma\sqrt{t}}) - SN(\frac{S-X}{\sigma\sqrt{t}}) + \sigma\sqrt{t}N(\frac{S-X}{\sigma\sqrt{t}}) \tag{1.2}$$

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 [4] 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 [5], who introduced emperical constants to increase the effectiveness of Sprenkle's model.

The model was finalised by Black and Scholes in 1973 [5], who explained the stock price movement by Geometric 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[2] 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.

The Black-Scholes model describes the price movement of an underlying asset via a Geometric Brownian Motion (GBM), defined by the Stochastic Differential Equation (SDE) [5]:

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dW(t) \tag{1.3}$$

Where $W \in \mathbb{R}$ is the Standard Brownian Motion (SBM). The solution of this SDE is such that [2]:

[2]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.

$$S(t) = S(0)e^{(r-\frac{1}{2}\sigma^2)t+\sigma W(t)} \tag{1.4}$$

By the properties of Brownian motions, a SBM W has $W(t) \sim N(0,t)$. Thus, we can substitute $W(t)$ by $Z \sim N(0,1)$ so that $W(t) = \sqrt{t}Z$, which evaluates the above formula to:

$$S(t) = S(0)e^{(r-\frac{1}{2}\sigma^2)t+\sigma\sqrt{t}Z} \tag{1.5}$$

Under the risk-neutral assumption, the payoff of a (European vanilla) call option with strike price K and expiration T is given by the expectation $E[e^{-rT}(S(T)-K)^+]$, and the corresponding payoff of a (European vanilla) put option at the same strike price and expiration is given by $E[e^{-rT}(K-S(T))^+]$. Using the solution of the Black-Scholes SDE we can deterministically eavluate the payoffs as follows [1]:

$$C(S(0),T) = S(0)\Phi(d_1) - e^{-rT}K\Phi(d_2) \tag{1.6}$$
$$P(S(0),T) = e^{-rT}K\Phi(-d_2) - S(0)\Phi(-d_1) \tag{1.7}$$

Where Φ is the cumulative normal distribution, $d_1 \coloneqq \frac{\log(S(0)/K)+(r+\frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}$, and $d_2 = d_1 - \sigma\sqrt{T}$. The Equations 1.6 and 1.7 are the so-called Black-Scholes formula.

A genearlised solution which addressed the impact of the presence of dividends (assuming the option of interest has an annual dividend yield rate of δ)specified in Glasserman's *Monte Carlo Methods in Financial Engineering*, as presented below:

$$C(S(0),T) = e^{-\delta t}S(0)\Phi(d_1) - e^{-rT}K\Phi(d_2) \tag{1.8}$$
$$P(S(0),T) = e^{-rT}K\Phi(-d_2) - e^{-\delta t}S(0)S(0)\Phi(-d_1) \tag{1.9}$$

Where now $d_1 \coloneqq \frac{\log(S(0)/K)+(r-\delta+\frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}$.

A further result to the American option cases is such that an American call option is never optimal (i.e. the estimated value would be the same as the European call option under same conditions), and that an American put option's value does not have an analytical form and required numerical methods to calculate[3].

1.3.3 Binomial Lattice Model

In 1979, based on the risk-neutral approach used by the Black-Scholes model, Cox, Ross and Rubinstein (CRR) [6] introduced a Binomial lattice tree model for modelling stock prices.

To construct a Binomial tree of stock prices, one breaks down the option's life $[0,T]$ into n time steps with fixed interval $\Delta t = T/n$. For each time steps, the stock price can move either up by a factor u with probablity $\hat{p}$ or down by a factor d with probability $\hat{q} = 1 - \hat{p}$.

As we can see, we will obtain a total of $n+1$ nodes at maturity T. For each node at the final step, we can obtain the corresponding payoff of an option related to that stock price.

[3]This is related to the studies on Monte Carlo pricing for American Options, which is beyond the scope of this report.

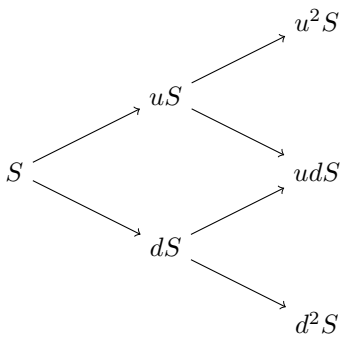

Figure 1.1: Binomial Lattice Tree for Stock Price

For example, recall from the above section, the $n+1$ possible payoffs $P(S(T))$ of an vanilla European call option would be:

$$P(S(T)) = (S(T) - K)^+ \tag{1.10}$$

The probability $\hat{p}$ is set to be "risk neutral" in a way such that:

$$S(t_i) = e^{-r\Delta t}\mathbb{E}[S(t_{i+1})] \tag{1.11}$$

$$= e^{-r\Delta t}[\hat{p}uS(t_i) + \hat{q}dS(t_i)], \quad \text{for } i = 0, ..., n-1 \tag{1.12}$$

Meaning that the current node's stock price is equal to the expected value of the two future nodes it can go in the next time step, discounting the interest. In such way, a portfolio which consist only of this stock would have a fixed return under price movement.

Under such setting, the option payoff at each node is computed in similar way. Suppose payoff P_{ij} denotes the payoff of the jth node at time step i, it can be computed via recursive form:

$$P_{ij} = e^{-rT}[\hat{p}P_{(i+1),j+1} + \hat{q}P_{(i+1),j}] \tag{1.13}$$

As mentioned above, the payoff of an option at maturity T is obvious to obtain. By applying this recursive method forward from $t = T$ back to $t = 0$, we will obtain an deterministic estimate of the current payoff (fair value) of the option of interest.

1.3.4 Trinomial Lattice Model

As a continuation of the Binomial model, Phelim Boyel in 1988 introduces the Trinomial Lattice Tree in modelling stock price movements [7]. Different from the Binomial tree, a Trinomial tree has three possible directions of movements: either up by a factor u with probability $\hat{p}_1$, down by a factor d with probability $\hat{p}_3$, or horizontally and remains unchanged with probability $\hat{p}_2$. We have $\hat{p}_1 + \hat{p}_2 + \hat{p}_3 = 1$.

The probabilities are also set to be risk neutral in such way that the current node's price is equal to the expected future nodes' price, discounting the interest:

$$S(t_i) = e^{-r\Delta t}\mathbb{E}[S(t_{i+1})] \tag{1.14}$$

$$= e^{-r\Delta t}[\hat{p}_1 uS(t_i) + \hat{p}_2 S(t_i) + \hat{p}_3 uS(t_i)], \quad \text{for } i = 0, ..., n-1 \tag{1.15}$$

However, in order to find unique solutions for $\hat{p}_i$'s, one need to imply more constraints on other parameters. One of the approach, which is implemented in `rmcop`, is that we have $u \times d = 1$ and $u \neq 1$. Under such setting, the solutions for the risk neutral probabilities are:

$$\hat{p}_1 = \frac{(V + M^2 - M)u - (M-1)}{(u-1)(u^2-1)} \qquad \hat{p}_3 = \frac{u^2(V + M^2 - M) - u^3(M-1)}{(u-1)(u^2-1)} \tag{1.16}$$

$$\hat{p}_2 = 1 - \hat{p}_1 - \hat{p}_3 \tag{1.17}$$

Where $V \coloneqq M^2(e^{\sigma^2\Delta t} - 1)$, $M \coloneqq e^{r\Delta t}$, and $\Delta t = T/n$.

The payoff of an option is calculated similarly as mentioned in the Binomial model.

The Trinomial model, as well as multinomial models, allows exponentially more stock price possibilities to be generated within the same number of time steps given. However, they are not necessarily computationally advantagous than the Binomial model, and solving for risk-neutral probabilities will be infeasible as more directions of movements are considered.

Relevant researches[8] have shown that, for some option styles, multinomial models are simply "transformation from Binomial lattice by removing the intraday periods." These ideas shows that pursue of further multinomial methods are unnecessary.

1.3.5 Monte Carlo Option Pricing

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 estimate the payoff of the option of interest by simply taking the average of the option payoff under each simulated trajectory. 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 [2] in the 1990s based on the Black-Scholes model. Until now, the field of Monte Carlo option pricing is still under active development and is widely used by "quants".

`rmcop`'s Monte Carlo pricing functionalities are limited to pricing European options. The supported scenarios are vanilla options and exotic options displayed in List 1.3.1 above. The methods implementing Monte Carlo option follow the framework introduced in Glasserman's *Monte Carlo Methods in Financial Engineering* [2], and will be elaborated below.

A generalised form of the Black-Scholes SDE shown in Equation 1.3 can be written as:

$$dS(t) = rS(t)dt + \sigma(S(t))S(t)dW(t) \tag{1.18}$$

Here, we allow the volatility σ to be a function of the stock price[4]. To simulate the movement of

[4]Normally, the volatility of a stock price is positively related to the stock price itself, i.e. higher price indicates more investments are involved in the stock, and thus leading to larger movements.

the stock price for each moment in time, we discretise Equation 1.18 by using Δt to approximate the infinitesimal dt:

$$dS(t + \Delta t) = S(t) + rS(t)\Delta t + \sigma(S(t))S(t)\sqrt{\Delta t}Z \tag{1.19}$$

If we break down the interval of the options' life $t \in [0, T]$ into m time steps, combining the solution described in 1.5, we will obtain the following result:

$$dS(t_{i+1}) = S(t_i) \exp\left([r - \frac{1}{2}\sigma^2](t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i}Z_{i+1}\right) \tag{1.20}$$

Where $Z_1, ..., Z_n$ are independent Standard Normal random variables. Now, by simulating Z_i, we can randomly generate the stock price step by step from $S(0)$ to $S(T)$. The following pseudocode will generate a simulated stock price trajectory in $[0, T]$ with m steps:

```
# for i in 1,...,m:
#     generate Z_i
#     S(t_i) = S(t_{i-1}) exp([r - 1/2 σ^2](t_i - t_{i-1}) + σ sqrt(t_i - t_{i-1}) Z_i))
```

Repeating the above procedure many times, we will obtain multiple simulated stock prices. By using information within each trajetory to compute the payoff of the option for that simulation, and take the mean of the payoffs computed for all simulations, we will have an estimated of the expected option payoff. Discounting this payoff will gave us an estimation of the option's current price under the scope of Black-Scholes model.

Detailed description on implementing Monte Caro method upon each style of option that are relevant to `rmcop` will be discussed in Section 3.3.

Chapter 2

Existing Option Pricing Packages within R Ecosystem

2.1 Packages Review

2.1.1 derivmkts

2.1.2 fOptions

2.1.3 RQuantLib

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

2.2 General Comments

Chapter 3

Package Development

3.1 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 in 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.

3.2 Package Structure

3.2.1 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

Table 3.1: R Scripts for rmcop

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

In Table 3.2.1, we have ...

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

Table 3.2: Functions for rmcop

3.3 Implementation

3.4 Deterministic Methods

3.4.1 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) [2]:

$$\frac{dS(t)}{S(t)} = \mu(S(t), t)dt + \sigma dW(t) \tag{3.1}$$

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.

3.4.2 vanilla.binomial

3.4.3 vanilla.trinomial

3.5 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.

3.5.1 mc.engine

The `mc.engine` function is the core of the package's Monte Carlo pricing funcitions. It is an internal function that is used to simulate price trajectories.

Recall Equation 1.20, the Black-Scholes model allows us to simulate the stock price trajectory using normal random variables.

3.5.2 vanilla.mc

3.5.3 asian.mc

1.3.1

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 [1].

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

$$dS(t) = rS(t)dt + \sigma(S(t))S(t)dW(t) \tag{3.2}$$

3.5.4 barrier.mc

3.5.5 binary.mc

3.5.6 lookback.mc

Chapter 4

Results

4.1 Examples

4.2 Further Discussions

Bibliography

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

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

[3] 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.

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

[5] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *The Journal of political economy*, vol. 81, pp. 637–654, 1973. Definition of Financial Options¡br/¿Black-Scholes Model.

[6] 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.

[7] 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.

[8] F. N. F. Sudding and M. Y. T. Irsan, "Multinomial method for pricing lookback option," *Journal of Physics: Conference Series*, vol. 1341, p. 62036, 2019. Multinomial model.