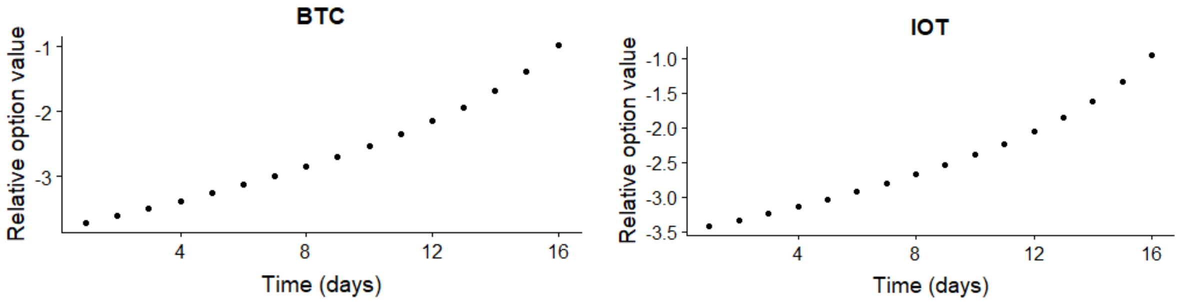
SIMULATING CRYPTOCURRENCY PRICE TRAJECTORIES USING JUMP-DIFFUSION MODELS

Aurél György Prósz, XGRP0J, Fizikus MSc

December 07, 2018
Computer simulations in physics
Second project

Tartalom


BTC
Relative option value
-1
-2
-3
4
8
12
16
Time (days)
IOT
Relative option value
-1.0
-1.5
-2.0
-2.5
-3.0
-3.5
4
8
12
16
Time (days)

Introduction

Blockchain technology opened a whole new class of decentralized digital currencies, which can be traded without a central bank or a single administrator. The cryptocurrency market has a market capitalization of **$184 000 000 000** as of November 2018. The market itself has unique properties, like extremely high trading volume and price volatility in relatively short amount of time. The causes include media hype, new assets released to the market every day, and activities considered illegal in the classical markets (so called “pump and dump” groups). In this project I utilize jump-diffusion models, which are originate from other fields, like condensed matter physics, to simulate the price trajectories of certain cryptocurrencies and indexes. I first calibrate the specific models using the historical time series of the cryptocurrency price and log returns. After this I simulate the trajectory of new price values and use this information to price derivative market assets, for example options. I also plan to use this method to explore the dynamics of Initial Coin Offerings. [1][2]

Motivation

I first heard about the blockchain technology (and particularly Bitcoin) a year ago, and I was very excited about the possibilities it could bring. I bought some bitcoin and other alternative cryptocurrencies and participated in Initial Coin Offerings. However, I quickly realized that the cryptocurrency market possesses enormous volatility and is very sensitive to perturbations created by the media, or the news and new regulations associated with the market. I wondered how the price can be simulated, and how one can assess these jumps in the currency price, which are rarely seen in the classical stock market. The results may help me to create better portfolios and to manage my risks better.

Project goals

My plan is the following:

- Select two of the most volatile cryptocurrencies and a less traded one from [4], and download their corresponding historical price data. I also included the EUR/USD pair in the analysis to compare the forex and the cryptocurrency market.
- Implement a basic diffusion model which serves as a baseline, and the more complex jump-diffusion model
- Calibrate the models by calculating the parameters based on the historical data
- Run simulations with the obtained parameters and plot the trajectories [3]
- Run simulations with different jump intensities to see if it fits the original data better
- Using the results of the simulations implement a simple option pricing method

My own contribution to the topic

Simulating price trajectories to make the option pricing more efficient is a widespread method in the stock market. However applying this technique to the cryptocurrency market is still a less explored field. My report reproduces the results of the application of the jump-diffusion models in Bitcoin, and I explore its usage on two other cryptocurrency, which were not investigated before and have much more different price histories than Bitcoin has.

Also, as far as I know, my approach is novel in the sense of there were not any application of jump-diffusion models regarding minor cryptocurrencies, where hype generated by Initial Coin Offering plays a central role in determining the jumps and the trajectories in the price.

Materials and methods

I used R to generate the price trajectories and the corresponding calcualtions, and its **ggplot2** package to visualize the results.

Possible source of errors and difficulties

I planned to obtain the parameters of the jump-diffusion model by applying Markov Chain Monte Carlo Method, but It turned out to be too difficult to apply in R, and I have found a quicker way to estimate the model parameters analiticaly.

Possible errors could be assessed from the simulation part itself, where more than a thousand independent price trajectories are generated with random processes. This of course can involve errors due to its stochastic nature of the method. By taking the mean or the median of the trajectories, one can reduce these errors.

There are multiple methods for option pricing, which can be very complex. In this project I apply a very simple option pricing formula, which is only applicable in short time-frames.

Jump-diffusion models in physics

Jump-diffusion models are applied in broad fields in physics. Here I present some examples according to [Neutron Scattering from a Liquid on a Jump Diffusion Model]:

- In crystals, atomic diffusion typically consists of jumps between vacant lattice sites.
- Magnetic reconnection
- Coronal mass ejections

Jump-diffusion models in finance

There are many models in finance that can describe the patterns of an asset attributes, like the price of a cryptocurrency. One of the most famous is the geometrical Brownian motion, where the price is assumed as continous as follows:

$$dS_t = \mu S_t dt + \sigma S_t dB_t$$

where S is the price, μ is the long term trend, σ is the volatility constant, dB_t is a Brownian process.

Solving this equation for S_t yields the following result (the derivation can be found in []):

$$S_t = S_0 exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)t + \sigma B_t\right)$$

Then the parameters μ and σ can be obtained from the log-returns (R), defined in the following way:

$$R = ln\left(\frac{S_{t_i+\Delta t}}{S_{t_i}}\right) = \left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\Delta B_{t_i+\Delta t}$$

For the followings let $\Delta t = 1$.

The drift parameter μ is estimated by the empirical mean of the log-returns from the historical data, and the the volatility constant σ is simply the standard deviation of the log-returns from the same dataset. The log-returns then have a normal distribution $N\left(\mu - \frac{1}{2}\sigma^2, \sigma\right)$. $\Delta B_{t_i+\Delta t}$ can be drawn from a standard normal distribution $N(0, 1)$. This way S_t can be simulated iteratively, after obtaining the parameters from historical data.

However, one of the shortcoming of this model is that it does not account for random jumps accounted for events like media hype and other rapid price-changing factors, which are more common in the cryptocurrency economy, but can be seen in the stock and the forex market aswell. An advanced model which should account for these jumps is a simple jump-diffusion model, first proposed by Merton []:

$$dS_t = \mu S_t dt + \sigma S_t dB_t + (y - 1)S_t dN_t$$

Where y is absolute jump size, dN_t is a counting process, which is a Poisson process.

Then the asset price can be simulated iteratively in the following way (similar to the geometrical Brownian motion):

$$S_{t_i+\Delta t} = exp\left(\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\Delta B_{t_i+\Delta t} + \sum_{N_{t_i+1}}^{N_{t_i+\Delta t}} ln(y_i)\right)$$

Where $N_{t_i+\Delta t}$ and N_{t_i+1} have a Poisson distribution with parameter λ, and y is the absolute jump size, which has a normal distribution $N(0, \delta^2)$. λ and δ can be determined by the method of moments described in [], μ in the same way as with the geometrical Brownian motion, and σ as:

$$\sigma^2 = \sigma_T^2 - \lambda\delta^2$$

Where σ_T^2 is the second moment of the log-return values.

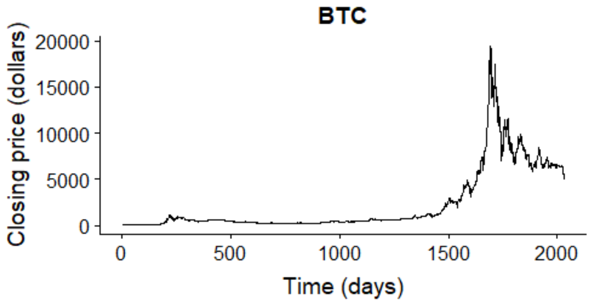
Importing data and visualization of cryptocurrency prices

The historical cryptocurrency prices were obtained from []. The EUR/USD is from … The following table summarizes the metadata on the selected cryptocurrencies:

Currency name	Short description	Date
Bitcoin (BTC)	Bitcoin was the very first, and also curently the number one most traded cryptocurrency used as digital cash.	Apr28. 2013 - Nov19. 2018
Iota (IOT)	The first open-source distributed ledger that is being built to power the future of the Internet of Things with feeless microtransactions and data integrity for machines.	Jun13. 2017 - Nov23. 2018
BlockCAT (CAT)	An ICO project. BlockCAT lets anyone create, manage, and deploy smart contracts on the Ethereum blockchain with just a few clicks. No programming required. Select from our catalog of pre-built contracts, or design your own with a drag-and-drop interface.	Aug13. 2017 - Nov23. 2018
EUR/USD	Rate of the price of the two most used currency in the world.	Oct24. 2013-Nov23. 2018

First I read in the historical data for all currencies, then visualised the corresponding time series. After calculating the log-returns for each of them I also visualized them with Q-Q plot, which shows the discrepancy between their distribution and a normal distribution under Brownian motion.

1. Figure


BTC
Closing price (dollars)
20000
15000
10000
5000
0
0
500
1000
1500
2000
Time (days)

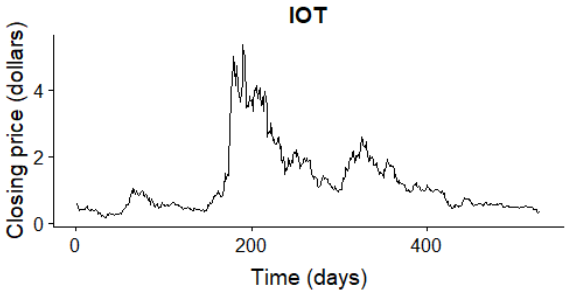

IOT
Closing price (dollars)
4
2
0
0
200
400
Time (days)

Figure SEQ Figure * ARABIC 2

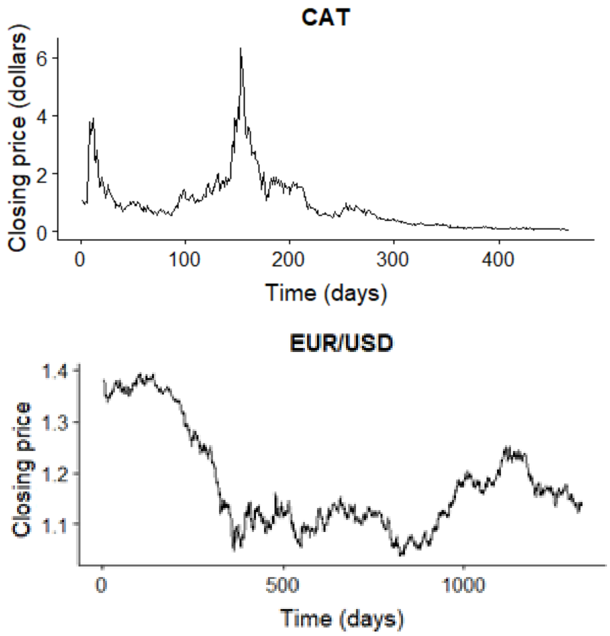
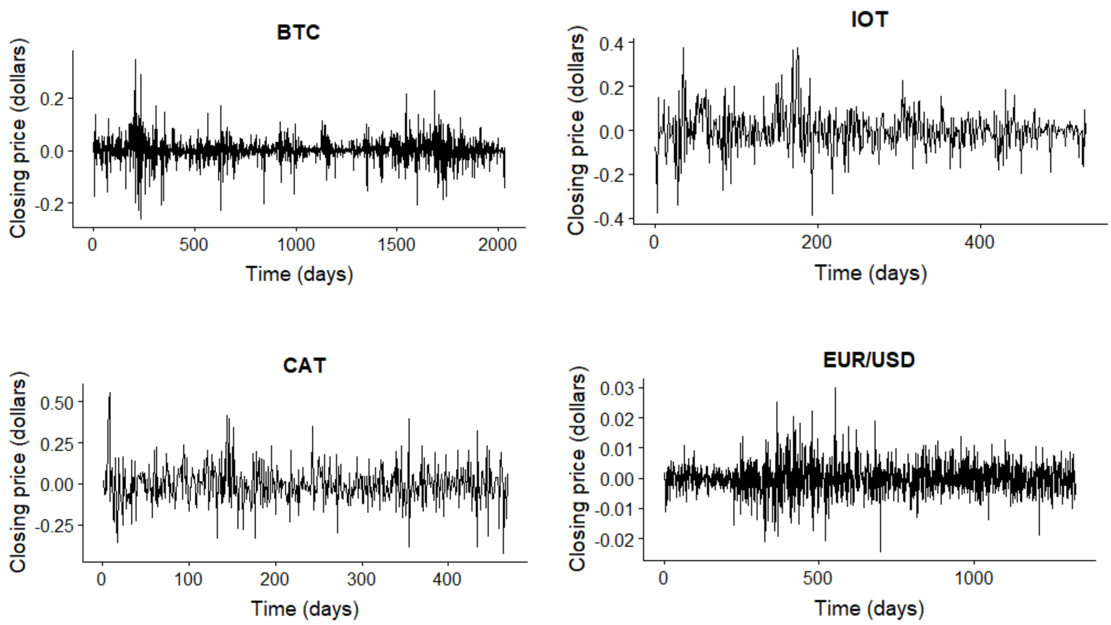
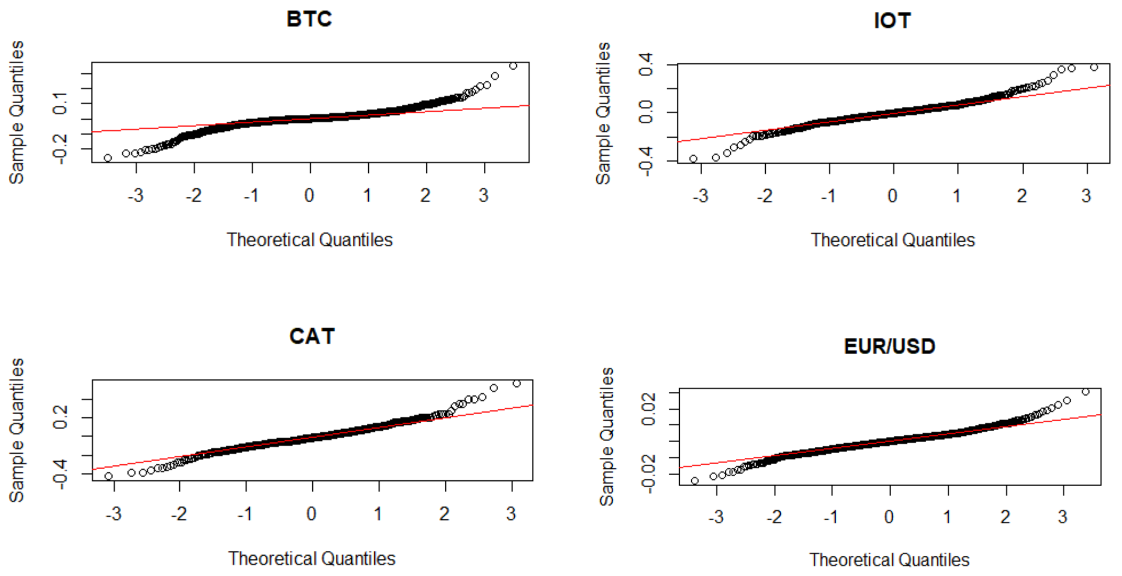


Figure 3





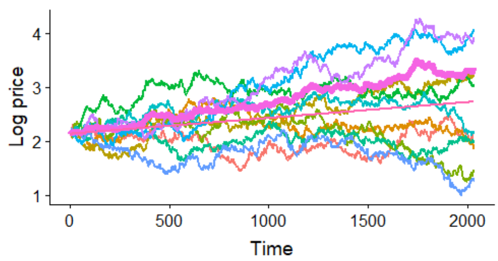
Discussion

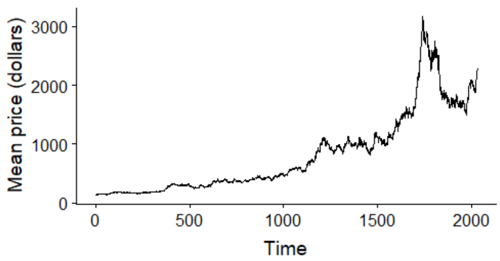
Results

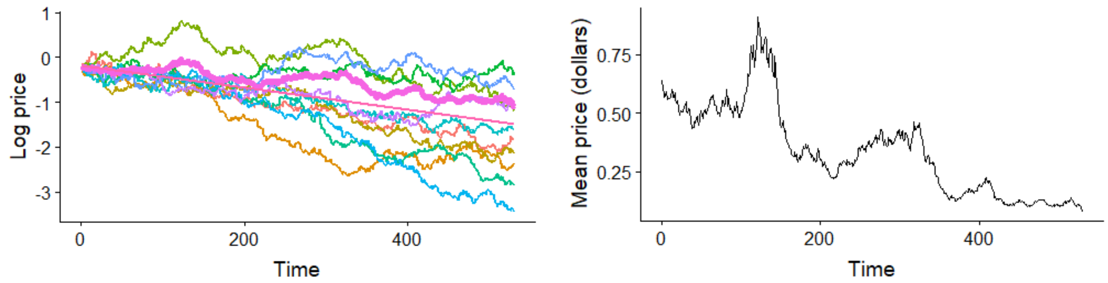
Diffusion model

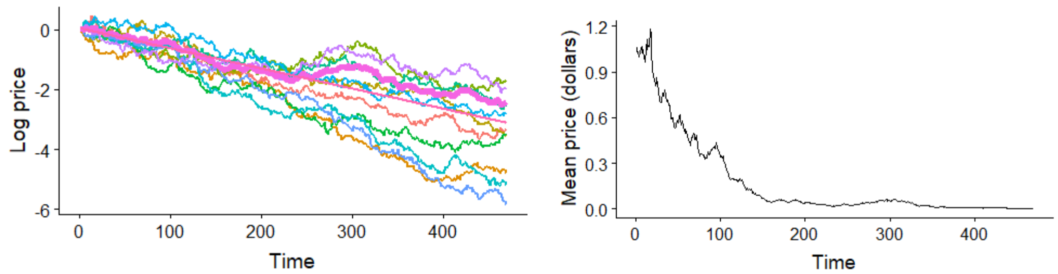
The following table summarises the obtained parameters from the historical data for the diffusion model:

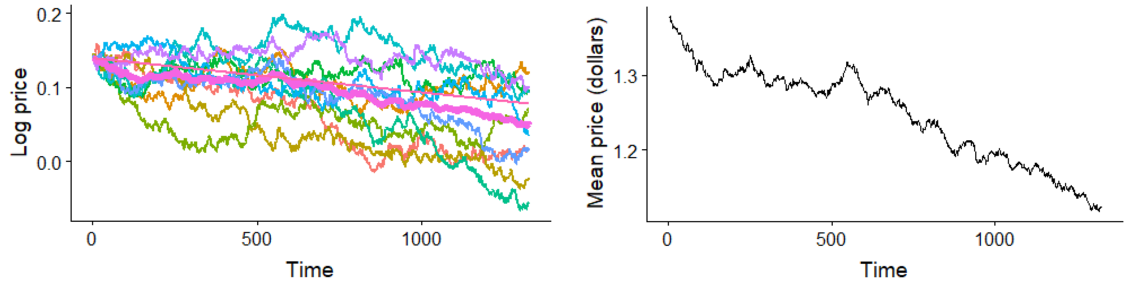
Currency name	μ	σ
BTC	0.001627597	0.04363496
IOT	-0.001556947	0.09050432
CAT	-0.007532748	0.1256792
EUR/USD	-9.273341e-05	0.005265627





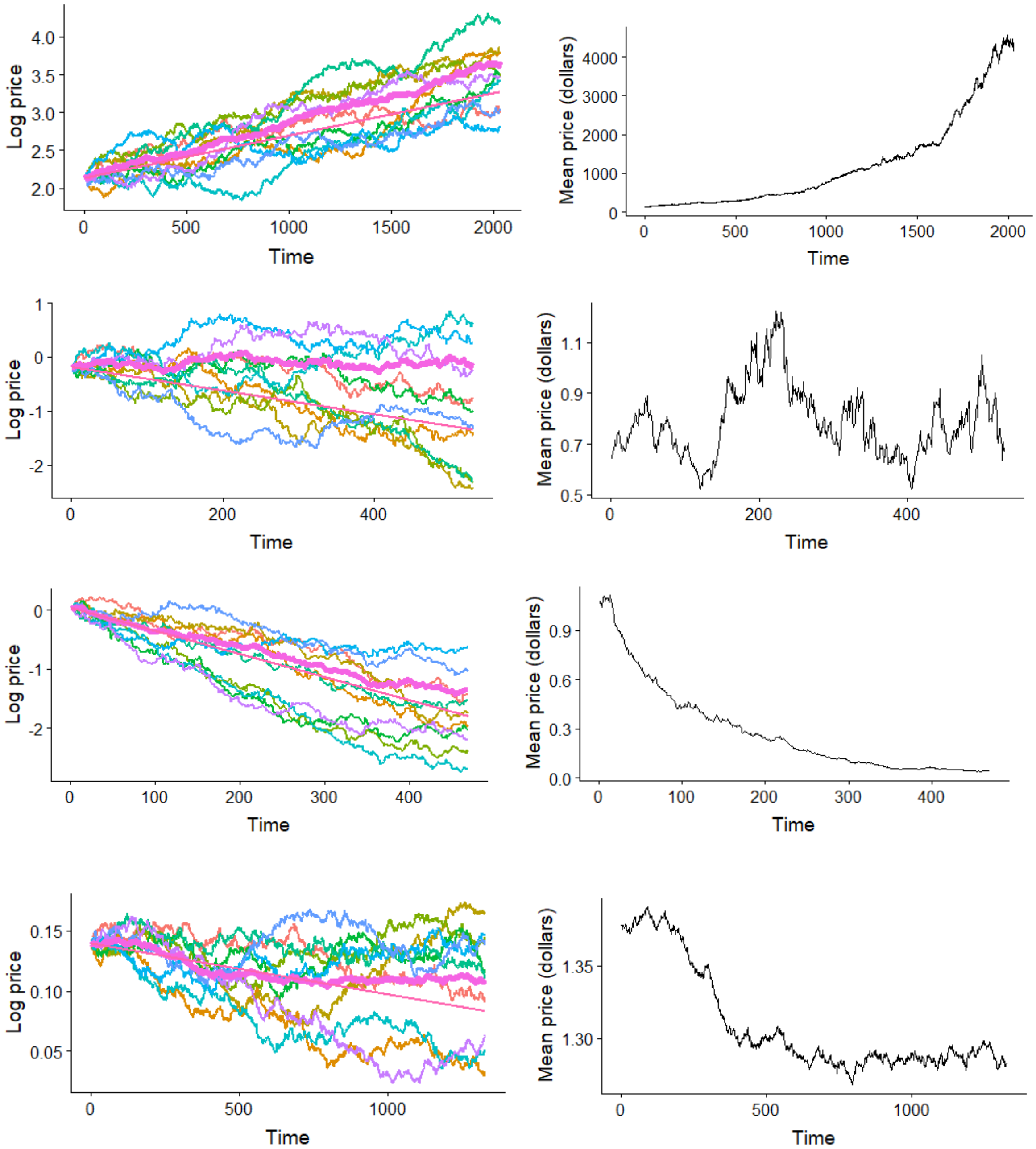




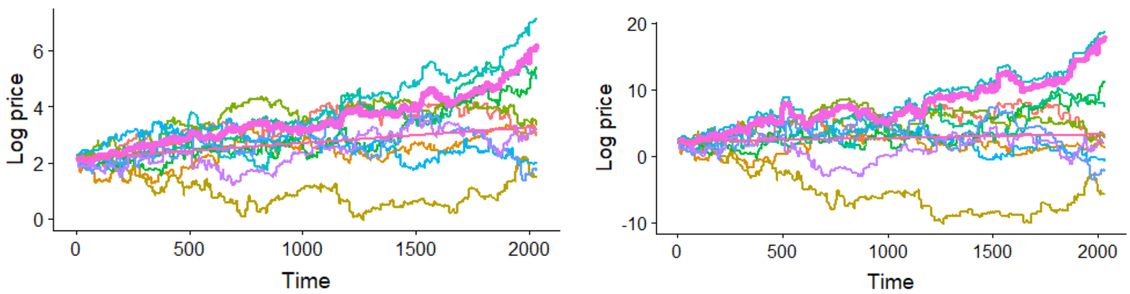


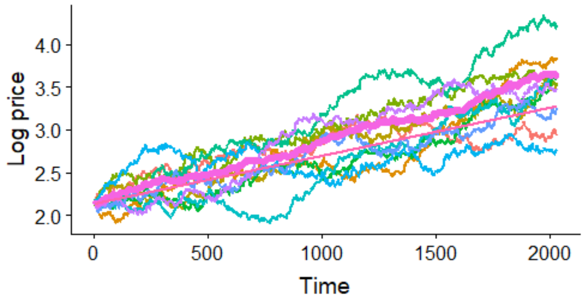
Jump-diffusion model

Currency name	μ	σ	λ	δ
BTC	0.001627597	0.02597066	0.1153685	0.1033068
IOT	-0.001556947	0.03643603	0.3526627	0.1393727
CAT	-0.007532748	0.05393077	0.4024358	0.179106
EUR/USD	-9.273341e-05	0.003009303	0.2429572	0.008763448


Log price
Time
Mean price (dollars)
Time
Log price
Time
Mean price (dollars)
Time
Log price
Time
Mean price (dollars)
Time
Log price
Time
Mean price (dollars)
Time

Game


Log price
Time
Log price
Time



Discussion

Evaluation

ARPE

In order to compare the results by the two models I considered the error between the empirical and the simulated data. As in [], I used the Average Relative Percentage Error (ARPE) to compare the models, defined as:

$$ARPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - s_i}{x_i}\right| \times 100$$

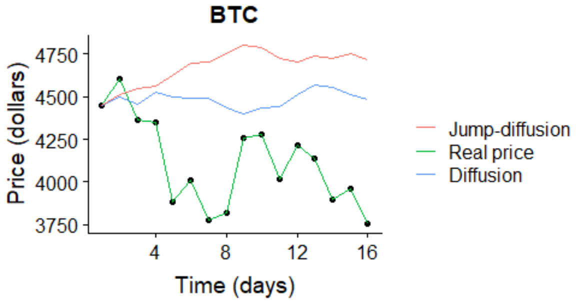
Where x_i and s_i are the i-th empirical and simulated data, n is the data szize. The lower value indicates the better model.

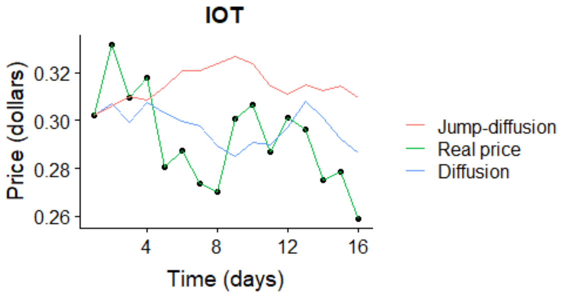
In table [] I compare the models with the ARPE values.

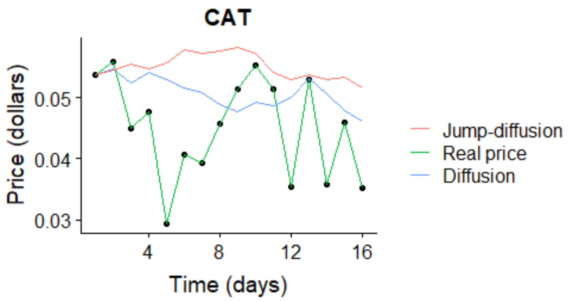
	Diffusion model	Jump-diffusion model
BTC	50	60
IOT	71	52
CAT	81	60
EUR/USD	8	12

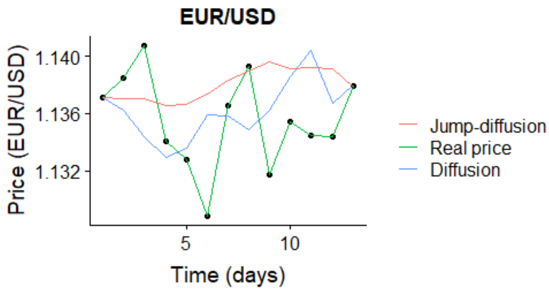
One can see that aside of BTC, the jump-diffusion model is always better than the simple diffusion one. This can be the result of some yet unexplained behaviour of the BTC market.

Forecasting









Pricing option in the cryptocurrency market

https://www.coinbureau.com/education/cryptocurrency-options/

Options are derivative instruments that give the holder the right to buy or sell a cryptocurrency at a predetermined price (Strike price) sometime in the future (expiry time).

Options have been a part of the general financial markets for decades and were originally used by farmers in order to secure the price of their crops when they were brought to the market. Since then, the option markets have grown to almost eclipse the traditional financial markets.

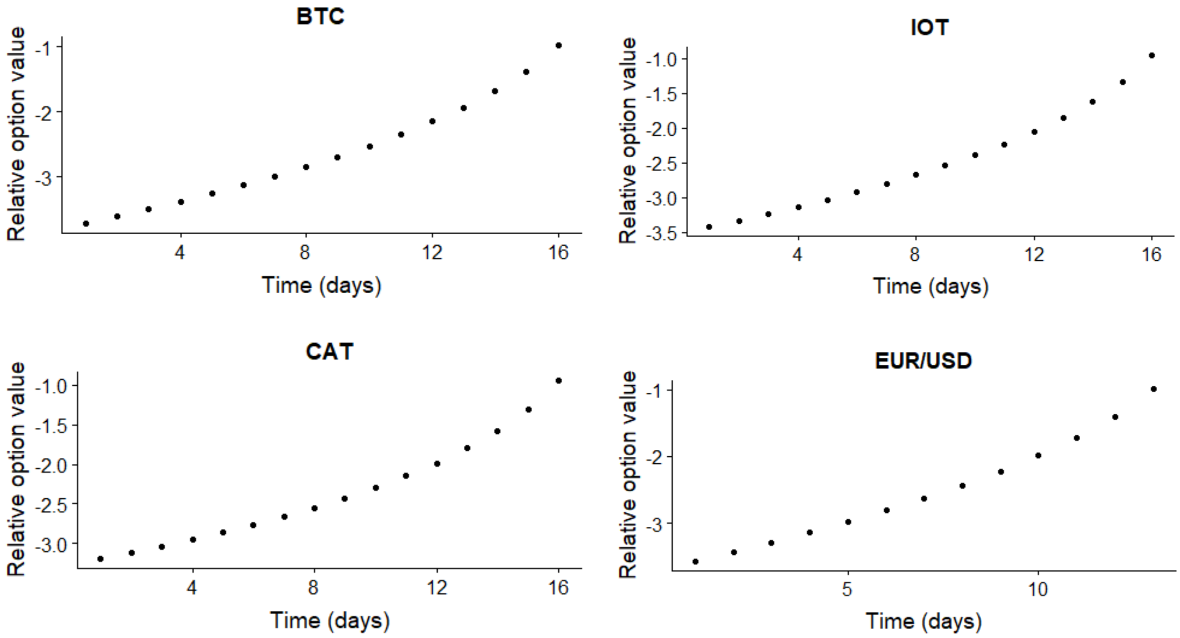
In this part of the project I use the mean price trajectories from the previous parts to determine the value of options on the corresponding market of the specific cryptocurrency.

Option pricing

There are multiple ways to price options, but most of them are very complex, like the Black–Scholes model. Here I apply a relative simple method, which is applicable only for short durations. I set this time limit to 13-16 days, which seemed an appropriate choice.

$$P = 0.4S\sigma T - \sqrt{T}$$

where P is the value of the option, S is the current price of the underlying asset, σ is the volatility of the market determined from the simulations and T is the remaining time to the end of the option.



Discussion

Summary