UNIVERSIDAD DE CANTABRIA

Facultad de Ciencias

Estimación del precio en opciones de compra asociado al Bitcoin

Estimation of the price of call options associated with Bitcoin

Trabajo de fin de grado
para acceder al

GRADO EN MATEMÁTICAS

Autor: Aram Diego López

Director: Domingo Gómez Pérez

junio de 2021

Agradecimientos

Quiero dedicar este trabajo a mis padres y familiares, los cuales siempre me han apoyado y animado durante esta etapa de la universidad.

También agradecer a Domingo Gómez que ha sido el que me ha guiado y también apoyado para que fuese posible entregar este trabajo.

Estimación del precio en opciones de compra asociado al Bitcoin

Resumen

palabras clave: Matemática Financiera, Bitcoin, Criptomoneda, Moneda Electrónica, Opciones De Compra

Las opciones de compra son instrumentos financieros que se utilizan en los mercados de valores para asegurar la adquisición de activos, normalmente acciones, en un futuro.

Una de las opciones de compra es la llamada opción de compra Europea, donde el poseedor de esta opción de compra tiene el derecho de adquirir acciones a un precio fijado en una fecha concreta.

El precio de este instrumento financiero, denominado prima, se modela en función de la ecuación de Black-Scholes, una ecuación estocástica diferencial que depende de la volatilidad de la acción, el precio inicial y el tiempo de expiración.

En este trabajo se va a estudiar esta ecuación en relación al Bitcoin, estudiando la evolución del precios además de una implementación en Python que permita resolver de forma numérica la ecuación de Black-Scholes, estimando la volatilidad.

Más específicamente, lo que realizará serán las siguientes tareas:

- Definición de los conceptos básicos relativos a la matemática financiera (bonos, acciones, volatilidad, opciones de compra, etc).
- Introducción al origen de bitcoin, relacionando las pautas de diseño con conceptos financieros (tiempo de diez minutos entre bloques, número máximo de bitcoins disponibles, tasa minera, etc).
- Aplicación de los métodos clásicos estadísticos para el cálculo de parámetros poblacionales, estos son el método de los momentos y el método de máxima verosimilitud.
- Estudio de las ecuaciones estocásticas diferenciales, así como su resolución numérica.
- Estudio de las diferentes opciones de compra y el problema numérico asociado.
- Implementación de los métodos mencionados en lenguaje python para el estudio de la evolución de precios de Bitcoin.

Estimation of the price of call options associated with Bitcoin

Abstract

keywords: Mathematical Finance, Bitcoin, Cryptocurrency, Electronic Currency, Call Options

Call options are financial instruments that are used in the stock markets to ensure the acquisition of assets, usually stocks, in the future.

One of the call options is the so-called European call options, where the holder of this call option has the right to acquire shares at a fixed price on a specific date.

The price of this financial instrument, called premium, is modeled on the basis of the Black-Scholes equation, a differential stochastic equation that depends on the volatility of the share, the initial price and the expiration time.

In this work, this equation will be studied in relation to Bitcoin, studying the evolution of prices and also an implementation in Python that allows solving the Black-Scholes equation numerically, estimating the volatility.

More specifically, what we will perform will be the following tasks:

- Definition of the basic concepts related to financial mathematics (bonds, stocks, volatility, call options, etc.).
- Introduction to the origin of bitcoin, relating the design guidelines with financial concepts (time of ten minutes between blocks, maximum number of available bitcoins, mining taxes, etc).
- Application of the classic statistical methods for the calculation of population parameters, these are the method of moments and the method of maximum likelihood.
- Study of differential stochastic equations, as well as their numerical resolution.
- Study of the different call options and the associated numerical problem.
- Implementation of the mentioned methods in python language for the study of the price evolution of Bitcoin.

Índice

1 Introducción

Comenzaremos introduciendo al lector en el contexto de las matemáticas financieras y de las técnicas de “Algorithmic Trading”, apoyándonos para estas últimas en el libro de Kissel [2020, Capítulo 1].

Las *matemáticas financieras* son una rama de las matemáticas que estudian las variaciones cuantificables que se producen en el mercado financiero durante el transcurso del tiempo.[1] El *mercado financiero* es un lugar diseñado para facilitar las transacciones de activos financieros.[2]

Las nuevas tecnologías han cambiado el mercado financiero, encontrando entre estos cambios el uso de las comunicaciones electrónicas y la ejecución de operaciones automáticas para la negociación de instrumentos financieros. Hay que diferenciar dos tecnologías causantes de este cambio, el “Algorithmic Trading”, que permite la automatización de los intercambios financieros mediante ciertos algoritmos, y el “Electronic Trading”, que permite el intercambio de instrumentos financieros mediante comunicación electrónica. Tal y como podemos ver el la Ilustración 1, en 2019, el 99,9 % de las negociaciones ya eran electrónicas y el 92 % era comprendido por el “Algorithmic Trading”.

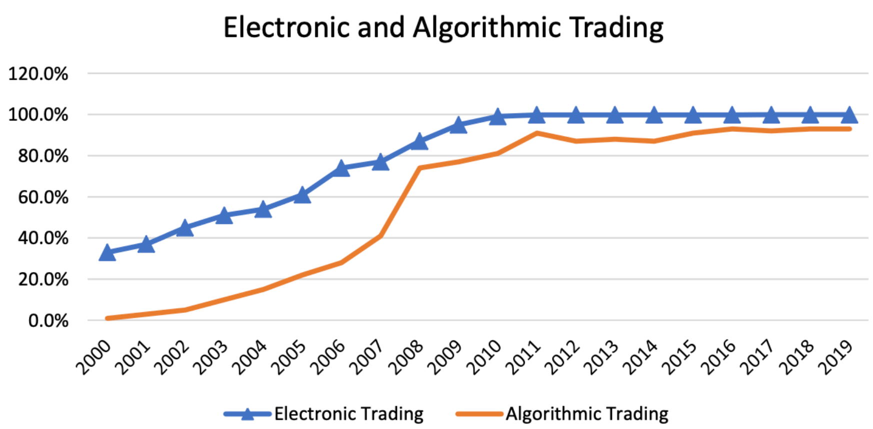

Ilustración 1: Evolución de las negociaciones.

El concepto de “Algorithmic Trading” hace referencia a la ejecución computarizada de un instrumento financiero siguiendo un conjunto preestablecido de reglas e instrucciones comerciales. Es decir, los inversores usan recursos informáticos para hacer realizar operaciones de forma mucho más rápida que un humano puede conseguir. El objetivo principal es el de llevar un mejor control de lo que sucede en el mercado, gestionar los costes de transacción globales de la orden y conseguir precios favorables. El “Algorithmic Trading” se clasifica en tres categorías:

- **Execution Algorithms:** o algoritmo de ejecución, se encarga de realizar la transacción de la decisión de inversión tomada por el inversor. El agente determina las características generales de

[1] ver la página web de Rankia

[2] Obtenida esta definición en los apuntes de la asignatura de Análisis de los Mercados de Valores del Grado en Administración de Empresas

la inversión, y luego ingresa la orden en el algoritmo. Por último, el algoritmo ejecutará la orden e implementará la decisión siguiendo las reglas especificadas por el administrador de la cartera, es decir, la persona encargada de controlar que se realicen los pagos.

- **Profit Seeking Algorithms:** o algoritmo de búsqueda de ganancias, determinará qué comprar y/o vender en el mercado y ejecutará esas decisiones sin la interacción del administrador de la cartera. La ejecución de estos algoritmos requiere, entre otros datos, información de precios en tiempo real y datos de mercado, y luego se ejecutarán la operación cuando las condiciones sean favorables para el inversor. Este algoritmo no tiene en cuenta otros objetivos, sólo tiene como finalidad obtener ganancias.

- **High Frequency Trading:** o comercio de alta frecuencia (HFT), es un algoritmo de búsqueda de ganancias que busca obtener una ganancia comercial a corto plazo, llegando a durar unos pocos segundos o menos. Los algoritmos HFT buscan obtener ganancias explotando las condiciones de liquidez y precios incorrectos del mercado y también son conocidos por tratar de descubrir las intenciones de compra y/o venta de los inversores a largo plazo para luego utilizar esta información en su beneficio con el objetivo de lograr una ganancia. Todos los algoritmos HFT son algoritmos de búsqueda de ganancias, pero no todos los algoritmos de búsqueda de ganancias son HFT.

Dentro del "Algorithmic Trading" también nos podemos encontrar en estos estilos de algoritmos:

- **Agresivos:** Son algoritmos que corren un mayor riesgo con el objetivo de comprar activos con mayor solvencia y vender los que tienen menos. La solvencia mide la capacidad para devolver en el futuro las deudas que ha contraído o que planea contraer.[3]

- **"Working Order":** Son algoritmos que operan en el mercado según reglas preestablecidas basadas en las necesidades de los inversores. Estos algoritmos buscan equilibrar el costo de negociación y el riesgo de mercado, así como maximizar los objetivos de inversión.

- **Pasivos:** Son algoritmos que operan según unas reglas fijas, normalmente comprando y vendiendo valores a unos valores prefijados. Estos algoritmos no comprueban si los precios del mercado han cambiado y sólo cambiarán de reglas si tú mismo lo indicas.

Más conceptos que serán necesarios que conozcan los lectores son estos tres recogidos en el libro de Leobacher y Pillichshammer [2014, Capítulo 7]:

- Los *bonos* son instrumentos financieros emitidos por las empresas o por el gobierno, los cuales adquirimos a una determinada cantidad de dinero. Estos bonos tienen un determinado plazo, el cual al expirar el emisor se compromete a devolvernos lo que pagamos por ellos más unos ciertos intereses.

- Las *acciones* son instrumentos financieros emitidos por las empresas a un determinado precio. Estos, a diferencia de los bonos, te otorgan una pequeña parte de la empresa convirtiéndote en socio de esta, pero las ganancias no están aseguradas.

- Las *opciones de compra* son instrumentos financieros y un tipo de derivado cuyo valor es definido en el futuro en términos de otros instrumentos financieros. Veremos distintos tipos de opciones de compra, las cuales son las siguientes:

  - La *opción de compra europea* sobre una acción es un contrato que tiene una determinada duración T, denominada *madurez*, el cual otorga a su poseedor la posibilidad de que únicamente cuando haya vencido ese periodo decidir si adquirir la acción a un precio K prefijado, llamado *precio de ejecución*.

[3]Obtenida esta definición a partir de la definición de "Liquidation Cost Analysis" del glosario del libro de Kissel

- La *opción de compra americana* consta de los mismos elementos que la europea, ya sea la madurez y el precio de ejecución; y en lo único que se diferencia de esta es que se puede ejecutar en cualquier momento de la madurez.
- La *opción de compra asiática* es igual que la opción de compra europea, salvo por el detalle de que el precio de ejecución se calcula como la media de los precios del activo financiero durante un periodo de tiempo.

Como compensación, al principio de cada una de estas opciones hay que pagar una determinada cantidad de dinero, llamada *prima*. Además, durante este trabajo solamente nos dedicaremos a las opciones de compra europeas.

A lo largo de este capítulo denotaremos el precio de la acción en el momento T como S_T. Por lo tanto, en el momento T el valor de la opción es de $S_T - K$ si $S_T > K$ y de 0 si $S_T \leq K$.

Este trabajo está destinado al estudio de un algoritmo HFT que nos permita el cálculo de la prima de una opción de compra. En nuestro caso, no lo estudiaremos en el caso de opciones sobre acciones, si no que trabajaremos con Bitcoin. Lo que haremos será ofrecer bitcoins a través de opciones con el objetivo de hacer que el cliente pueda ver aumentados sus capitales, llamado este proceso *apalancamiento*. Para hacer este estudio necesitaremos plantear un proceso estocástico, en el que la variable precio será tratada como la variable estocástica.

Se considerarán conocidos por el lector los conceptos de espacio probabilístico, variable aleatoria, función de densidad, ..., esperanza y varianza; definidos estos dos últimos conceptos a través de la integral de la función de densidad $f(x)$:

$$\mathbb{E}[X] = \int_0^{+\infty} x f(x) \ \mathrm{dx} \tag{1}$$

$$Var[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \int_0^{+\infty} x^2 f(x) \ \mathrm{dx} - \mathbb{E}[X]^2 \tag{2}$$

Continuaremos definiendo lo que es un *proceso estocástico* y viendo un ejemplo de este. Para esto usaremos el libro de Ash y Doléans-Dade [2000, Capítulo 9].

Definición: Un proceso estocástico sobre un espacio probabilístico $(\Omega, \sigma, \mathbb{P})$ es una familia de variables aleatorias $(X_t)_{t\in T}$ donde T marca el tiempo. Se utilizará $T = \mathbb{N}$ en el caso de tiempos discretos y $T = \mathbb{R}$ en el caso de continuos.

Ejemplo: En el caso de las finanzas, un proceso estocástico puede ser entendido como en el cual un jugador apuesta en los tiempos 0,1,2,3,...; que X_0 es la cantidad de dinero con la que empezamos y que X_n es la cantidad de dinero que tenemos en el tiempo n.

Una vez hecha esta introducción del "Algorithmic Trading", de los instrumentos financieros con los que vamos a trabajar y del objetivo del trabajo, vamos a ilustrar distintos algoritmos HFT para la fijación de precios de derivados.

1.1. Principio de arbitraje

El *arbitraje financiero* es una estrategia financiera en la que se aprovecha de las diferencias de precios entre los mercados para obtener beneficios sin tomar ningún riesgo. En el siguiente ejemplo se muestra un caso en el que hay arbitraje:

- El valor de un bono al inicio lo denotaremos por $B_0 > 0$ y el valor de este en el momento 1 será de $B_1 = B_0(1 + r)$ con $r \geq 0$, es decir, el valor del bono inicial más unos ciertos intereses.

- El valor de una acción al inicio lo denotaremos como $S_0 > 0$ y el valor en el momento 1 vendrá definido de la siguiente forma:

$$S_1 = \begin{cases} S_0 u & con\ probabilidad\ p \\ S_0 d & con\ probabilidad\ 1 - p \end{cases} \tag{3}$$

  donde $0 < d < 1 + r < u$.

- El valor de una opción de compra sobre una acción en el momento 1 es el siguiente:

$$C_1 = max(S_1 - K, 0) = \begin{cases} max(S_0 u - K, 0) & con\ probabilidad\ p \\ max(S_0 d - K, 0) & con\ probabilidad\ 1 - p \end{cases} \tag{4}$$

Por lo tanto, C_0 sería la esperanza de la variable aleatoria expuesta en la ecuación 4.

$$C_0 = \frac{B_0}{B_1}\,\mathbb{E}[max(S_1 - K, 0)] = \frac{1}{1 + r}(p\ max(S_0 u - K, 0) + (1 - p)\ max(S_0 d - K, 0)) \tag{5}$$

Sin embargo, veamos como con el siguiente ejemplo la fórmula puede dar un valor sin sentido:

Ejemplo: Supongamos que $r = 0$, $u = 2$, $d = \frac{1}{2}$, $S_0 = K = 1$ y $p = \frac{1}{2}$, con lo que sustituyendo en la ecuación 5 nos queda $C_0 = \frac{1}{2}$. Con lo que el vendedor puede hacer lo siguiente:

- En el momento 0 puede escribir 4 opciones y venderlas por un precio de 2€ en total (por $\frac{1}{2}$€ cada una). Tomando un 1€ prestado tendría 3€ que podría invertir en comprar 3 acciones.

- En el momento 1 habría dos opciones:

  1. El precio de las acciones ha crecido y por lo tanto, estas han duplicado su valor, dejando al vendedor con 6€ al venderlas. Las opciones que vendió se ejecutarán, dejándolo a pagar 4€ (1€ por cada opción, puesto que el valor actual de las acciones es de 2€ y nosotros las prefijamos a 1€) y tendría que devolver el préstamo de 1€. Con lo que al final del proceso acabaría con un beneficio neto de 1€.
  2. El precio de las acciones baja y por lo tanto las opciones no se ejecutarían. Al vender las acciones que adquirió el vendedor recibirá $\frac{3}{2}$€, con lo que al devolver el préstamo de 1€ se quedaría con unas ganancias de $\frac{1}{2}$€

Visto todo esto, aquí tenemos un ejemplo de arbitraje financiero, puesto que el vendedor siempre sale ganando sin hacer ninguna inversión y no tomando riesgos. Esta estrategia no se da en el mercado financiero puesto que no se ajusta a la *hipótesis del mercado eficiente*.

Definición: La hipótesis del mercado eficiente dice que si al revelar cierta información sobre el mercado a todos los participantes, no afecte al precio de un activo y sea imposible obtener un beneficio económico del mismo. Esta definición está obtenida a partir de la definición de hipótesis del mercado eficiente en el sentido fuerte del trabajo de Uribe Gil y Ulloa Villegas [2011].

Para corregir este problema definiremos una probabilidad $\tilde{p}$ que siga una estrategia del tipo *martingala*.

Definición: La martingala es un proceso para las ganancias de un jugador en un juego limpio, en el cual si partimos de X_t, pasado un tiempo s obtendremos $X_{t+s} = X_t$, es decir, que esperamos ganar lo mismo que apostamos. De forma más técnica y según podemos ver en el libro de Glasserman [2003, Apéndice B], una martingala es un proceso estocástico $\{X_t, t \geq 0\}$ que cumple las siguientes condiciones:

1. $E[X_t] < \infty \; \forall t \geq 0$
   En el caso estudiado en este TFG, esta condición siempre se va a cumplir, puesto que el precio nunca será infinito.

2. $E[X_t | X_1, ..., X_{t-1}] = X_{t-1}$
   Esta condición es modelada con los datos de precios anteriores, con el objetivo de obtener que las ganancias finales obtenidas sean 0.

Ejemplo: Sea el espacio probabilístico $(\Omega, \sigma, \mathbb{P})$ donde $\Omega = \{cara = 0, cruz = 1\}$, σ son las tuplas de posibles resultados de tirar dos veces una moneda y $\mathbb{P}$ es la función de probabilidad que asigna la misma probabilidad a cada suceso. También consideramos las variables aleatorias $X_1(t_1, t_2) = t_1$ y $X_2(t_1, t_2) = t_2$, donde vemos que X_1 nos devuelve el resultado de la primera tirada y X_2 el de la segunda. Además, consideramos que la apuesta cuesta 1€, que vamos a cara y que si sale esta ganamos 2€, pero si sale cruz no ganamos nada. Ahora veamos si cumple que esto es una martingala:

1. $\mathbb{E}[X_1] = 2 \cdot \mathbb{P}[X_1 = 0] + 0 \cdot \mathbb{P}[X_1 = 1] = 2 \cdot \frac{1}{2} + 0 \cdot \frac{1}{2} = 1 < \infty$. De forma análoga obtenemos que $E[X_2] = 1$.

2. $\mathbb{E}[X_2 | X_1] = 2 \cdot \mathbb{P}[X_2 = 0 | X_1] + 0 \cdot \mathbb{P}[X_1 = 1 | X_1] = 2 \cdot \frac{1}{2} + 0 \cdot \frac{1}{2} = 1 = \mathbb{E}[X_1]$.

En efecto se cumplen las condiciones, por lo tanto estamos ante un ejemplo de martingala en el que lo que se espera ganar es lo mismo que se apostó.

Ahora calculemos esa probabilidad $\tilde{p}$ para que la ecuación 5 siga la estrategia del tipo martingala:

$$\frac{1}{1+r}(\tilde{p} S_0 u + (1-\tilde{p}) S_0 d) = S_0 \Leftrightarrow \tilde{p} S_0 u + S_0 d - \tilde{p} S_0 d = S_0(1+r) \Leftrightarrow S_0(\tilde{p} u + d - \tilde{p} d) = S_0(1+r) \Leftrightarrow$$

$$\Leftrightarrow \; \tilde{p} u + d - \tilde{p} d = 1 + r \; \Leftrightarrow \; \tilde{p}(u-d) = 1 + r - d \; \Leftrightarrow \; \tilde{p} = \frac{1+r-d}{u-d}$$

Con lo que al final obtendríamos que

$$C_0 = \frac{1}{1+r}(\tilde{p} \; max(S_0 u - K, 0) + (1-\tilde{p}) \; max(S_0 d - K, 0)) \tag{6}$$

donde $\tilde{p} = \frac{1+r-d}{u-d}$.

1.2. Modelos EDE

Para la exposición de este modelo necesitaremos entender el concepto de movimiento browniano, el cual podemos encontrar en el libro de Leobacher y Pillichshammer [2014, Capítulo 8].

Definición: Un movimiento browniano un proceso estocástico de tiempo continuo definido en un espacio probabilístico $(\Omega, \sigma, \mathbb{P})$ y que tiene las siguientes propiedades:

1. $W_0 = 0$ c.s.

2. Sus trayectorias son continuas

3. Sus incrementos son independientes, es decir, $W_{t_1} - W_{t_0}$,...,$W_{t_d} - W_{t_{d-1}}$ son independientes para $t_0 := 0 < t_1 < t_2 < ... < t_d$

4. Para $0 \leq t_1 < t_2$, tenemos que $W_{t_2} - W_{t_1} \sim N(0, \sigma(t_2 - t_1))$

La utilidad del movimiento browniano en el ámbito de las finanzas es el de describir la volatilidad de los precios a medida que pasa el tiempo.

En la explicación de este método no entraremos en cuentas y nos basaremos en los resultados obtenidos en el libro de Leobacher y Pillichshammer [2014, Capítulo 7]. En muchas ocasiones el precio de las acciones en las diferentes etapas no es dado explícitamente, sino que está definido por una *ecuación diferencial estocástica*. Según el libro de Glasserman [2003, Apéndice B], una EDE puede ser usada en la mayor parte de los modelos financieros con la siguiente forma:

$$\begin{cases} dX_t = a(X_t, t)dt + b(X_t, t)dW_t \\ X_0 = x_0 \end{cases} \tag{7}$$

donde W es un movimiento browniano de dimensión k, a es una aplicación de $\mathbb{R}^n \times [0, \infty]$ en $\mathbb{R}^n$, b es una aplicación de $\mathbb{R}^n \times [0, \infty]$ en $\mathbb{R}^{n \times k}$ y X_0 es un vector de dimensión d linealmente independiente de W.

A partir de lo visto en la ecuación 7 obtenemos la EDE de dimensión 1:

$$\begin{cases} dS_t = \hat{\mu} S_t dt + \sigma S_t dW_t \\ S_0 = s_0 \end{cases} \tag{8}$$

donde dW_t es el incremento infinitesimal del movimiento browniano.

Obtenemos que la solución de la ecuación diferencial 8 es la siguiente:

$$S_t = s_0 \; exp\left(\tilde{\mu} + \sigma W_t - \frac{\sigma^2}{2} t\right) \tag{9}$$

Ahora vamos a considerar el modelo de la ecuación 7 con dimensión $m+1$:

$$\begin{cases} dS_t = \mu(t, S_t)dt + \sigma(t, S_t)dW_t \\ S_0 = s_0 \end{cases} \tag{10}$$

donde $S = (S^0, ..., S^m)$ es un proceso estocástico de dimensión $m+1$ y $s_0 = (s^0, ..., s^m) \in \mathbb{R}^{m+1}$.

Si S_T es la solución de la ecuación 10 en el tiempo T y $\hat{S}$ es una solución aproximada en los nodos de tiempo $0, h, ..., nh = T$ de la forma

$$\begin{cases} \hat{S}_0 = S_0 \\ \hat{S}_{k+1} = \hat{S}_k + \mu(kh, \hat{S}_k)h + \sigma(kh, \hat{S}_k)\Delta W_{k+1} \end{cases} \tag{11}$$

donde $\Delta W_{k+1} := W_{(k+1)h} - W_{kh}$.

Operando con la ecuación 11 se llegaría a la siguiente, que es la versión discreta:

$$\hat{S}_{k+1} = \hat{S}_k + \mu(\hat{S}_k)h + \sigma(\hat{S}_k)\Delta W_{k+1} + \frac{1}{2}(\sigma(Y_k) - \sigma(\hat{S}_k))(\Delta W_{k+1}^2 - h)\frac{1}{\sqrt{h}} \tag{12}$$

donde $Y_k = \hat{S}_k + \sigma(\hat{S}_k)\sqrt{h}$.

1.3. Modelo de Black-Scholes

Este modelo es la concreción del modelo anterior pero en una dimensión, en el que no profundizaremos en las cuentas y nos basaremos en lo ya calculado en el libro de Leobacher y Pillichshammer [2014, Capítulo 7]. Nuevamente tomaremos una probabilidad que nos asegure un precio libre de arbitraje. Ahora haremos tender n al infinito de forma que obtendremos el siguiente modelo:

$$\begin{aligned} B_t &= B_0 \ exp(rt) \\ S_t &= S_0 \ exp(\mu t + \sigma W_t) \end{aligned} \tag{13}$$

donde $t \in [0, T]$ y W es un *movimiento browniano* unidimensional.

Como en el modelo anterior, existe una probabilidad $\tilde{\mathbb{P}}$ para la cual esto se convierte en una martingala:

$$S_t = S_0 \ exp\left(\left(r - \frac{\sigma^2}{2}\right) t + \sigma \tilde{W}_t\right) \tag{14}$$

donde $\tilde{W}_t = W_t - \frac{1}{\sigma}(r - \mu - \frac{\sigma^2}{2})t$ es el movimiento browniano bajo $\tilde{\mathbb{P}}$. $\tilde{\mathbb{P}}$ también se puede dar en términos del teorema de Radon–Nikodym expuesto en el libro de Ash y Doléans-Dade [2000, Capítulo 2] y que dice lo siguiente:

Definición: Dadas $\mathbb{P}$ y $\tilde{\mathbb{P}}$ dos funciones de probabilidad en el mismo espacio medible (Ω, σ), se dice que $\tilde{\mathbb{P}}$ es absolutamente continua respecto a $\mathbb{P}$ si para todo $A \in \sigma$, tal que $\mathbb{P}(A) = 0$, se tiene que $\tilde{\mathbb{P}}(A) = 0$.[4]

Teorema: Dado un espacio probabilístico $(\Omega, \sigma, \mathbb{P})$ y sea $\tilde{\mathbb{P}}$ una medida con signo absolutamente continua, se cumple que existe una función medible f tal que:

$$\tilde{\mathbb{P}}(A) = \int_A f \ d\mathbb{P} \quad \forall A \in \sigma \tag{15}$$

Aplicando la ecuación 15 obtenemos que $\frac{d\tilde{\mathbb{P}}}{d\mathbb{P}} = exp(\nu W_t - \frac{\nu^2}{2}T)$, con $\nu = \frac{1}{\sigma}(r - \mu - \frac{\sigma^2}{2})$. Esto significa que la ganancia esperada $\tilde{\mathbb{E}}[X]$ de cualquier variable aleatoria medible viene dada por la fórmula

$$\tilde{\mathbb{E}}[X] = E\left[exp\left(\nu W_t - \frac{\nu^2}{2}T\right) X\right]. \tag{16}$$

Ahora esta nueva medida de probabilidad es usada para fijar el precio de la opción de compra en su etapa inicial de forma que este libre de arbitraje.

$$C_0 = \tilde{\mathbb{E}}[B_T^{-1} C_T] \tag{17}$$

Tomando una opción de compra europea con el precio de las acciones $(S_t)_{t \geq 0}$, una prima K y una madurez T con unos pagos de $C_T = max(S_T - K, 0)$ y aplicando la ecuación 19 obtenemos que el precio inicial de la opción viene dado por:

$$C_0 = exp(-rT)\tilde{\mathbb{E}}[max(S_T - k, 0)] \tag{18}$$

Cuando $S_t = S_0 \ exp((r - \frac{\sigma^2}{2})t + \sigma \tilde{W}_t)$ y cuando $(r - \frac{\sigma^2}{2})T + \sigma \tilde{W}_T$ es una variable aleatoria normal con media $(r - \frac{\sigma^2}{2})T$ y varianza $\sigma^2 T$ obtenemos

$$C_0 = \frac{exp(-rT)}{\sqrt{2\pi\sigma^2 T}} \int_{-\infty}^{\infty} max(S_0 \ exp(x) - K, 0) \ exp\left(-\frac{(x - (r - \frac{\sigma^2}{2})T)^2}{2\sigma^2 T}\right) \mathrm{dx}$$

[4] Obtenida esta definición del TFM de Molina [2020]

$$= \frac{exp(-rT)}{\sqrt{2\pi\sigma^2 T}} \int_{log(\frac{K}{S_0})}^{\infty} (S_0\ exp(x) - K)\ exp\left(-\frac{(x-(r-\frac{\sigma^2}{2})T)^2}{2\sigma^2 T}\right) \mathrm{dx}$$

Ejemplo: Vamos a calcular la prima de una opción de compra sobre una acción cuyo precio actual es de 1€, tiene una rentabilidad del 20 % y una volatilidad del 10 %. Esta opción de compra tendrá una madurez de 1 año y un precio de ejercicio de 1,15€, con lo que obtenemos lo siguiente:

$$C_0 = \frac{exp(-0,2)}{\sqrt{0,02\pi}} \int_{log(1,15)}^{\infty} (exp(x) - 1,15)\ exp\left(-\frac{(x-0,195)^2}{0,02}\right) \mathrm{dx} \approx 0,075€$$

Como resultado a lo expuesto anteriormente se obtiene la fórmula de Black-Scholes para el precio de una opción

$$C_0 = S_0\Phi(d_1) - exp(-rT)K\Phi(d_2), \tag{19}$$

donde $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} exp(-t^2/2)dt$,

$$d_1 = \frac{log\frac{S_0}{K} + (r+\frac{\sigma^2}{2})T}{\sigma\sqrt{T}} \qquad y \qquad d_2 = \frac{log\frac{S_0}{K} + (r-\frac{\sigma^2}{2})T}{\sigma\sqrt{T}} \tag{20}$$

2 Historia del Bitcoin

Para la realización de este capítulo tomaremos datos de los libros de Varios [2018] y de Champagne [2008].

En agosto de 2008, el matemático y criptógrafo Nakamoto [2008] (el cuál sólo era un apodo y a día de hoy todavía se desconoce su verdadera identidad) publicó un artículo que trataba sobre el *sistema P2P* de dinero digital. Este sistema consistía en una transacción de dinero entre pares, de forma que esta no involucrara exponer todos nuestros datos personales a la hora de un pago online. También describió por primera vez la tecnología *blockchain*, la cual era una base de datos distribuida donde cada usuario en la red ejecuta y registra transacciones agrupándolas en forma de bloques, capaz de registrar todo tipo de transacciones persona a persona de manera eficiente, segura, verificable e inmutable. El blockchain tiene su origen en las criptomonedas, en concreto en bitcoin, descrita esta moneda electrónica como una cadena de bloques de firmas digitales. El propietario de una moneda puede transferirla a otra persona añadiendo, al final de la cadena, la firma digital del código de transacción anterior y la llave pública del nuevo comprador:

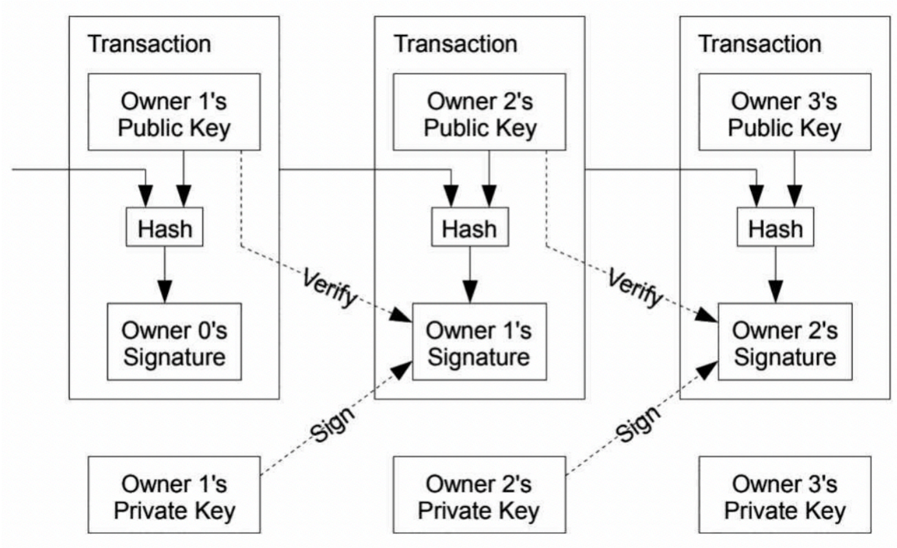

Ilustración 2: Funcionamiento del blockchain tomada del artículo de Nakamoto.

El objetivo de este sistema es la verificación de la propiedad y la no duplicidad de las transacciones. Cualquier usuario en el sistema puede solicitar que se agregue una transacción a la cadena, pero esta sólo será aceptada si todos los usuarios validan su legitimidad. A este proceso de verificación se le llama *minería*. Para que tu transacción sea incluida deberás pagar una *tasa minera*, la cual puede ser de cualquier cantidad en bitcoins. Los mineros no podrán escoger todas las transacciones, puesto que en cada bloque sólo entran un número limitado de estas, con lo que escogerán las que más paguen. Una vez añadido un nuevo bloque, sólo podrá ser eliminado si existiera otra cadena más larga que no contuviese ese bloque. Esto es así, puesto que el blockchain trabaja con la cadena más larga de ese momento, llamada así no por su longitud, sino porque es la que necesita más energía para ser recorrida. Actualmente, encontrar una cadena más larga que la utilizada es casi imposible, puesto que requeriría mucho poder de cómputo.

Este proceso se enfrentó a un gran problema, denominado como el ataque del 51 %, que consistía en que si la mayoría de usuarios en la verificación de una nueva transacción decidían operar de forma fraudulenta podrían evitar esa transacción o en el peor de los casos provocar un doble gasto, es decir, gastar sus bitcoins en dos destinatarios a la vez. Esto a día de hoy es inviable en las cadenas de transacciones del Bitcoin, ya que debido a su larga extensión, poseer más del 51 % de los nodos o conseguir poner de acuerdo a tantas personas es casi imposible.

No fue hasta enero de 2009 cuando Satoshi Nakamoto lanzó un software libre denominado Bitcoin que contenía todos los conceptos anteriormente explicados. Este se componía de un protocolo (Bitcoin), de una unidad de cuenta o token (bitcoin, o en su forma abreviada BTC) y de una Blockchain (base de datos transaccional). Este era un blockchain público, es decir, una red a la que cualquier persona podía acceder, podía crear bloques y podía participar en el proceso de validación. El Bitcoin representa un concepto revolucionario, es decir, permite a un sujeto que realiza una compra en bitcoins sólo proporcionar al comerciante su firma electrónica y su dirección de envío, mientras que de la forma tradicional tendríamos que dar nuestra tarjeta de crédito, corriendo el riesgo que esa empresa sea hackeada y nuestra información se vea comprometida. Sin embargo, el significado de Bitcoin no se limita a la simplicidad del sistema de pago como ya veremos posteriormente.

Sólo 21 millones de bitcoins llegarán a existir, llevando ya creados unos 18,5 millones hasta el momento y a la espera de que el último sea creado alrededor del año 2140. Esta moneda es altamente divisible, donde la partición más pequeña que permite el software actual es de 0,00000001 BTC (10^{-8} BTC), definido como 1 satoshi, en honor a su creador. Otro concepto del Bitcoin es la tasa de hash o "hash rate", que es la unidad de medida de la potencia de procesamiento de la red Bitcoin, la cual debe hacer intensivas operaciones matemáticas por razones de seguridad. Cuando esta alcanza un hash rate de 10 TH/s significa que puede hacer 10 billones de cálculos por segundo.[1]

Bitcoin tiene una dificultad de minería dinámica, es decir, la dificultad de minería de un bloque en Bitcoin, no es siempre igual. El término "dificultad" se emplea como una unidad de medida en el proceso de minería que hace referencia a cómo de difícil es encontrar el hash del bloque. En Bitcoin se ha estipulado en su código que debe minarse un bloque cada 10 minutos de media. Una persona que tenga mucha potencia de cálculo podrá romper la media de los 10 minutos que se busca, y es ahí donde Bitcoin añadió uno de sus componentes más importantes: el reajuste de la dificultad de minería, ya que sin este reajuste todos los bitcoins estarían minados en poco tiempo. Para evitar esto, el reajuste de minado lo que hace es subir la complejidad del problema matemático buscado, estabilizando de nuevo el tiempo de cada bloque a 10 minutos.[2]

El 3 de enero de 2009 se crearon los primeros 50 bitcoins, y no fue hasta unos días después, el 12 de enero de 2009, cuando se realizó la primera transacción entre Satoshi Nakamoto y el criptógrafo Hal Finney, situándose el precio de esta moneda en 0,00076 dólares. Tal y como podemos ver en el siguiente gráfico del libro de Champagne [2008], en febrero de 2011, 1 bitcoin equivalía a 1 dólar.

[1]ver la página web de Bitcoin
[2]ver la página web de Bit2Me Academy

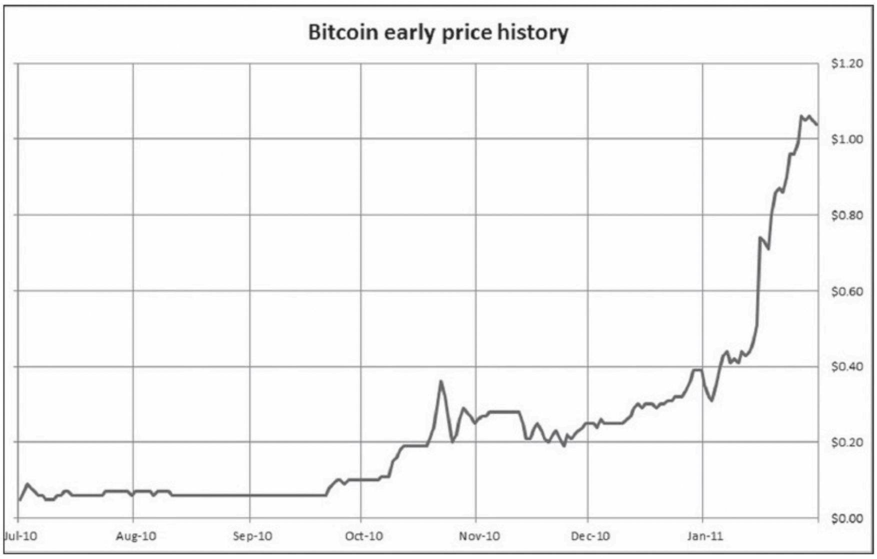

Ilustración 3: Gráfica de precios del bitcoin en sus inicios.

En este otro gráfico hecho por mi mismo a través de un histórico de precios del bitcoin obtenido de Internet[3] podemos ver como avanzó el precio del bitcoin a lo largo de la historia.

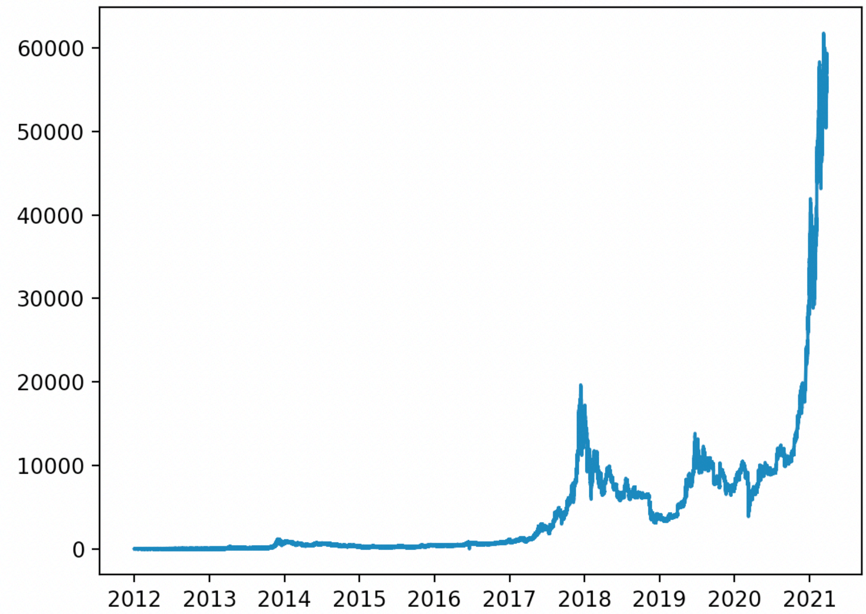

Ilustración 4: Gráfica de precios en dólar del bitcoin disponibles en el histórico descargado.

Otro momento histórico del bitcoin sería el 27 de noviembre de 2013, cuando esta moneda alcanza el precio de una onza de oro, situándose por encima de los mil dólares.

En lo que respecta al 2021, el precio del bitcoin a superado máximos históricos situándose el 7 de enero del 2021 con que 1 bitcoin equivalía a 40.000 dólares y tan sólo poco más de un mes después superaría la barrera de los 50.000 dólares. El 13 de marzo llegaría a su máximo actual de algo más de 63.000 dólares.[4]

[3]ver la página web de Kaggle
[4]ver la página web de la Oficina de Seguridad del Internauta

A continuación se muestra un gráfico extraído de Google con la evolución del precio del bitcoin a lo largo de toda su existencia:

Ilustración 5: Gráfica de precios del bitcoin a lo largo de toda su existencia.

Tal y como podemos ver en el gráfico anterior, actualmente el precio del bitcoin no pasa por su mejor etapa, situándose entre los 35.000 y los 40.000 dólares y habiendo llegado a caer por debajo de los 33.000 dólares el 23 de mayo.

El impacto que provocó el Bitcoin fue enorme, puesto que era un modelo descentralizado que permitía solucionar el problema del doble gasto anteriormente mencionado. Además, permitía transferir dinero por todo el mundo con tan sólo enviar un correo electrónico, cosa que ya era posible gracias a Mastercard y a Paypal. Esto facilitaba el envío de dinero por parte de extranjeros a sus propios países, puesto que si realizaban esta operación a través de una compañía les cobraban altas comisiones, mientras que si lo realizaban de esta otra forma sólo tenían una pequeña comisión por parte de la conversión de la moneda a bitcoin.

Otro beneficio que ofreció el Bitcoin, es el mencionado anteriormente de que facilitó la privacidad de los usuarios a la hora de hacer compras y donaciones por Internet, puesto que no tenían que dar la información de su tarjeta y que esta pudiese ser almacenada y usada por parte de la empresa.

Otro impacto muy importante es el de que el Bitcoin aspira a ser capaz de crear un nuevo sistema de ser dinero y no ser tan sólo una moneda. Esto quiere decir que el bitcoin, además de tener las propiedades de una moneda

- Medio de intercambio en el comercio
- Unidad de cuenta (cuantificable)
- Perdurable (tiene que ser resistente al paso del tiempo)
- Divisible (tiene que poder dividirse en unidades más pequeñas)
- Portátil (tiene que se fácilmente transportable)
- Fungible (lo que quiere decir que una unidad de una moneda específica puede ser cambiada por otra unidad de esa misma moneda)

aspira a tener una más, que es la capacidad de conservar su valor a largo plazo.

Las monedas están sujetas a la inflación, que es el hecho de que en una economía decida hacer más unidades de esta. El problema de esto lo vemos por ejemplo en la Alemania de los años 20, en la que tras la Primera Guerra Mundial y para hacer frente a la fuerte crisis económica que sufrían decidieron imprimir más dinero, lo que les llevó a que su moneda pasará de equivaler 1 dolar por cada 60 marcos a 1 dolar por cada 1 millón de marcos.[5] Es por esta situación por lo que los países no pueden decidir imprimir más dinero para solucionar sus problemas, puesto que el valor de su moneda caerá y esto provocará una subida de los precios. Actualmente Venezuela tiene una inflación del 2.959,8 % causada por la política de creación de dinero que se llevó a cabo en 2009.[6]

Para mantener el poder adquisitivo de una economía, se tiene que limitar la oferta monetaria. El oro y la plata fueron usados como dinero durante miles de años, por lo que si se quería introducir más dinero en la economía requería un coste de explotación minera para obtenerlos. Esto mismo ocurre con el bitcoin, puesto que si queremos crear nuevos se requiere también un proceso de minado. Si esto lo comparamos con la facilidad que tenemos hoy en día de imprimir nuevos billetes, entendemos por qué nuestros antepasados mantenían unos precios constantes durante toda su vida y actualmente nosotros sufrimos un constante aumento de estos.

Otro problema que nos encontramos actualmente con el dinero que usamos, llamado dinero FIAT, es que permite al los gobernantes reducir su déficit depreciando la moneda en circulación. Esto afecta gravemente a las personas más pobres y a la clase media, mientras que los más ricos pueden llegar a sacar provecho de esta situación. Esto podría ser solucionado adoptando una forma de dinero que mantenga su valor a largo plazo, tal y como aspira el bitcoin.

El bitcoin está limitado a 21 millones de unidades para evitar todos estos problemas de inflación que hemos visto que ha sufrido y sigue sufriendo las economías de todo el mundo. Pero, no todo es positivo respecto al Bitcoin, a parte del gasto energético que se produce, es una moneda que no es aceptada en todas partes y suele ser atesorada por la gente. Esto último es provocado por los llamados millonarios Bitcoin, que retienen millones en bitcoins, privando así al mercado de ese dinero.

[5] ver la página web de EL CEO
[6] ver la página web de la Wikipedia

3 Objetivos

El objetivo de este trabajo será el estudio del precio de opciones sobre el bitcoin. Para ello usaremos la fórmula de Black-Scholes (19), calculada en el capítulo 1.

Para la realización de esta estimación necesitaremos una base de datos con el histórico de precios del bitcoin, por lo que usaremos la obtenida por Zielak [2017]. Este histórico incluye el precio del bitcoin a cada minuto desde el 1 de enero de 2012 hasta 31 de marzo de 2021, teniendo el siguiente aspecto

	Open	High	Low	Close	Volume_(BTC)
0	4.39	4.39	4.39	4.39	0.455581
4857371	58742.18	58742.18	58714.31	58714.31	2.519999
4857372	58714.31	58714.31	58686.00	58686.00	1.384487
4857373	58683.97	58693.43	58683.97	58685.81	7.294848
4857374	58693.43	58723.84	58693.43	58723.84	1.705682
4857375	58742.18	58770.38	58742.18	58760.59	0.720415
4857376	58767.75	58778.18	58755.97	58778.18	2.712831

Ilustración 6: Tabla del histórico de precios del bitcoin.

dónde la primera columna representa cada minuto, "Open" indica el precio de apertura en ese minuto, "Close" el precio de cierre, "High" el máximo alcanzado, "Low" el mínimo alcanzado y "Volume_(BTC)" la cantidad de bitcoins tramitados.

En nuestro caso trabajaremos con que el precio del bitcoin en cada minuto es el precio de cierre, con lo que representando estos datos obtendríamos gráfica de la Ilustración 4.

Además de los precios del bitcoin, para aplicar la fórmula de Black-Scholes necesitaremos calcular la rentabilidad y la volatilidad del bitcoin, haciéndolo a través del *método de los momentos*.

Definición: El método de los momentos es un método que sirve para la estimación de parámetros poblacionales, que consiste en igualar momentos poblacionales a momentos muestrales obteniendo ecuaciones de esta forma:[1]

$$\begin{gathered} E[W] = \mu_1 \\ E[W^2] = \mu_2 \\ \vdots \\ E[W^k] = \mu_k \end{gathered} \tag{21}$$

dónde para $j = 1, ..., k$ tenemos que $\mu_k = \frac{1}{n}\sum_{i=1}^{n} w_i^2$.

La rentabilidad que nos ofrece el bitcoin a cada minuto la calcularemos a partir de la primera fórmula del método anterior, quedándonos de la siguiente forma:

[1]Obtenido este método en el libro de Rice

$$r = \frac{\sum_{i=1}^{n} R_{i-1,i}}{n} \tag{22}$$

donde n es el número de periodos que tenemos y $R_{i-1,i}$ es la rentabilidad entre el minuto $i-1$ y el i calculada de la siguiente forma:

$$R_{i-1,i} = \frac{C_i - C_{i-1}}{C_{i-1}} \tag{23}$$

siendo C_i el precio de cierre del bitcoin en el minuto i.

A partir de la segunda ecuación del método de los momentos obtenemos que la volatilidad o desviación típica se calcularía de la siguiente forma:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(R_{i-1,i} - r)^2}{n-1}} \tag{24}$$

donde $R_{i-1,i}$ y n son los mismos que antes, y la r es la rentabilidad anteriormente calculada.

Ejemplo: Creando un programa en Python que aplique las ecuaciones 22 y 24 a los datos que tenemos, obtenemos las rentabilidades y las volatilidades siguientes:

	Rentabilidad	Volatilidad
Minuto	0,0001249	0,2162
Diario	0,003546	0,03765
Semanal	0,02525	0.1078
Mensual	0.1237	0.3254

Otra forma de calcular estas constantes sería a través del *método de máxima verosimilitud*, pero antes veremos como definir el movimiento browniano. Para el cálculo de éste, tendremos que hacer una integral que represente el valor medio esperado, y para esto utilizaremos la *regla del trapecio*.

Definición: La regla del trapecio es un método de integración, es decir, un método para calcular aproximadamente el valor de una integral definida. La regla se basa en aproximar el valor de la integral de $f(x)$ por el de la función lineal, que pasa a través de los puntos $(a, f(a))$ y $(b, f(b))$. La integral de esta es igual al área del trapecio bajo la gráfica de la función lineal.[2]

$$\int_a^b f(x)dx = (b-a)\frac{f(b) - f(a)}{2} \tag{25}$$

Definición: El método de máxima verosimilitud es un método que sirve para ajustar un modelo y estimar sus parámetros. Consiste en obtener el valor que maximiza la función de verosimilitud:[3]

$$L_n(x|\theta) = f(x_1, ..., x_n|\theta), \tag{26}$$

donde se tiene una muestra $x = x_1, x_2, ..., x_n$ de n observaciones independientes e idénticamente distribuidas extraídas de una función de distribución desconocida con función de densidad f. Dadas estas condiciones, podemos expresar la función de verosimilitud de la siguiente forma:

[2]Obtenida esta regla en el artículo de Yeh et al.
[3]Obtenido este método en el libro de Rice

$$L_n(x|\theta) = \prod_{i=1}^{n} f(x_i|\theta) \tag{27}$$

Por lo tanto, lo que hay que resolver ahora es:

$$L_n(x|\hat{\theta}) = max_\theta(L_n(x|\theta)) \tag{28}$$

Para facilitar las cuentas se toma el logaritmo de la ecuación 28 quedándonos lo siguiente:

$$log(L_n(x|\theta)) = \sum_{i=1}^{n} log(f(x_i|\theta)) \tag{29}$$

con lo que derivando respecto de θ e igualando a 0 obtenemos que:

$$\frac{\partial log(L_n(x|\theta))}{\partial\theta} = \sum_{i=1}^{n} \frac{\partial log(f(x_i|\theta))}{\partial\theta} = 0 \tag{30}$$

Ejemplo: Veamos a hacer un ejemplo clásico en en que la función de densidad es una exponencial, en concreto con la que trabajaremos será:

$$f(t) = \mu \cdot exp\,(-\mu t)\,, t \in [0, +\infty).$$

A partir de muestras $t_1, \ldots, t_n$, vamos a estimar el parámetro μ, maximizando la siguiente función de verosimilitud:

$$L_n(\mu) = \mu \cdot exp\,(-\mu t_1)\ \cdot \ldots \cdot \mu \cdot exp\,(-\mu t_n) = \mu^n \cdot exp\left(-\sum_{i=1}^{n} \mu t_i\right) \Leftrightarrow$$

$$\Leftrightarrow \log(L_n(\mu)) = n \cdot \log(\mu) + -\sum_{i=1}^{n} \mu t_i$$

Por lo tanto tenemos que:

$$\frac{\partial log(L_n(\mu))}{\partial\mu} = \frac{n}{\mu} - \sum_{i=1}^{n} t_i = 0 \Leftrightarrow \mu = \frac{n}{\sum_{i=1}^{n} t_i}$$

Otros datos que usaremos serán el precio del instrumento financiero justo en el momento anterior del que queremos calcular la prima (P), el precio de ejecución (K) y la duración de la opción (T), que cómo no son fijos, se creará una función en Python tal que dados P, K y T te calcule la prima.

Para la fijación del precio de ejecución en el tiempo t para una opción de compra europea seguiremos dos estrategias:

1. $K = C_{t-1} \cdot 1{,}05$
2. $K = C_{t-1} \cdot 0{,}95$

Nota: En el caso del estudio de las ganancias por minuto sólo nos centraremos en una estrategia. Esta consistirá en que el precio de ejecución será $K = C_{t-1} \cdot (1 + r)$, donde r es la rentabilidad del tramo en el que estemos trabajando. Además, el programa del cálculo de la prima será ligeramente diferente, en el que tendremos que darle como argumentos los mencionados anteriormente más la rentabilidad y la volatilidad del tramo en el que estemos trabajando.

4 Desarrollo

4.1. Herramientas

Esta sección la dedicaremos a la explicación y exposición de todas las herramientas de software que han sido necesarias para la realización de este trabajo, además de cuál a sido la función de cada una de ellas.

4.1.1. Git

En primer lugar nos encontramos con un software ya incluido en el propio ordenador y que se accede a él a través de la terminal. Su utilidad es la de tener un registro de los cambios que se han hecho en unos determinados archivos, además de poder coordinar el trabajo que realizan varias personas de forma que se comparten las modificaciones a través de un repositorio online. En este caso, ese repositorio escogido para la sincronización de los datos fue Bitbucket. La opción más popular de repositorios online es GitHub, pero esta opción no era viable, puesto que las normas de la universidad estipulan que los repositorios deben de ser privados, y para obtener un repositorio privado aquí tienes que pagar. Este es el motivo por el que escogimos Bitbucket, puesto que ofrece repositorios privados de forma gratuita.

Para aprender a usar este software hice el curso de Version Control (Git)[1] y además utilicé una herramienta muy intuitiva para aprender los comandos de Git, de forma que se mostraban los cambios de una forma muy visual.[2]

4.1.2. Python

En segundo lugar nos encontramos con un lenguaje de programación cuya funcionalidad es la de leer el código que se ha programado, y que además posee una licencia de código abierto. Python fue creado a finales de los ochenta por Guido van Rossum y actualmente es uno de los lenguajes más usados por todo el mundo.[3]

Este lenguaje no era conocido para mi, puesto que durante la carrera los únicos dos lenguajes que se dan son el de MATLAB en Cálculo Numérico I y II y en Optimización; y el de Java en Programación. Actualmente en esta última asignatura mencionada se enseña Python, pero yo pertenecía al último grupo que dio Java, cambiando justo al año siguiente. Por lo tanto, para poder desarrollar los programas necesarios para el cálculo de una prima de una opción necesitaba aprender este lenguaje. El curso de Practical Python Programming[4] fue el elegido para aprenderlo, puesto que no comenzaba explicando lo que es programar como si no supieses nada, sino que asumía que ya sabías programar en otro lenguaje y lo tomaba como base para empezar.

Pero no sólo utilizamos el software libre de Python, sino que fueron necesaria varias librerías para el desarrollo del programa.

[1] ver el tutorial de Git
[2] ver el tutorial más visual de Git
[3] ver la página web de Kaggle
[4] ver la página web de Practical Python Programming

4.1.3. Librería math

Esta es una librería incluida en Python cuyo objetivo es el poder usar las funciones matemáticas básicas con escalares, ya sea el logaritmo, la raíz cuadrada y la exponencial; que son las usadas en este trabajo. Esta librería es muy sencilla de utilizar y no requería de ningún tutorial o curso para aprender a utilizarla, ya que sólo basta con importarla y a la hora de llamar por ejemplo a la función logaritmo hacerlo de la siguiente forma: math.log().

4.1.4. Librería quadpy

Esta también es una librería incluida en Python, que para usarla basta sólo con importarla, cuyo objetivo es el poder calcular integrales de funciones. Para aprender a utilizar esta librería vi un tutorial de YouTube[5], que con un par de ejemplos muy sencillos explicaba el funcionamiento y la forma de llamar a todos los elementos. La librería de NumPy, posteriormente introducida, también permite el cálculo de integrales, pero no está tan especializada como esta. Es por esto que se prefiere en este caso el uso de quadpy para el cálculo de integrales.

Su uso fue necesario para el cálculo de la integral de la fórmula de Black-Scholes (19).

4.1.5. Librería datetime

Esta es una librería, como las dos anteriores, incluida en Python, que para usarla basta sólo con importarla. Su función principal es la de poder hacer operaciones con fechas y con el tiempo. Para aprender a utilizar esta librería vi un tutorial con ejemplos[6] y otro en el que aprendí el método concreto para poder avanzar el tiempo en meses[7].

Su uso fue necesario para transformar el vector de minutos que teníamos en el histórico de precios a fecha y hora.

4.1.6. Librería NumPy

Esta es una librería que, no bastaba sólo con importarla, sino que requería descargársela para poder utilizarla. La función de esta es la creación de vectores y matrices además de que incorpora unas operaciones muy útiles entre estos. Para aprender a usar esta librería vi un tutorial de la empresa Enthought[8], impartido por Alex Chabot-Leclerc.

Su uso fue necesario para el tratamiento de los datos del histórico de precios del bitcoin, puesto que se requería poner estos datos en una matriz, tal y como vimos en la Ilustración 6 pero sin el encabezado, y a la hora de definir vectores o extraer de esa matriz el vector de precios de cierre. Además se utilizan las funciones de numpy.nanmean(), que calcula la media de los datos de un vector omitiendo los NaN; y la de numpy.empty(), para la creación de un vector de la dimensión que se quiera pero vacío. Esta librería también me sirvió para que a la hora de llamar a los elementos de una matriz o un vector me fuese lo más familiar posible, puesto que se hace de forma muy similar en MATLAB.

4.1.7. Librería Pandas

Esta es una librería que también requiere descargársela como la anterior y además es una ampliación de la anterior. La utilidad que tiene es la de creación de matrices con encabezado. Para aprender

[5] ver el tutorial de quadpy
[6] ver la página web de Stack
[7] ver la página web de Grepper
[8] ver el tutorial de NumPy

a usar esta librería vi un tutorial de la empresa Enthought[9], impartido por Daniel Chen.

Su uso fue necesario para la creación de la matriz de la Ilustración 6, para lo cual se usó el código pandas.read_csv(), que lo que hace es que a través del archivo .csv descargado[10] que contiene todos los datos del bitcoin separados por comas crea una matriz en la que cada celda la ocupa cada dato que está entre las comas.

4.1.8. Librería Matplotlib

Esta es una librería que también requiere descargársela como las dos anteriores. La utilidad que tiene es la de creación de gráficas en Python a través de dos vectores. Para aprender a usar esta librería vi un tutorial de la empresa Enthought[11], impartido por Hannah Aizenman y Thomas Caswell.

Su uso fue necesario para hacer la representación de la gráfica de la Ilustración 4, que contiene los datos de los precios de cierre del bitcoin. También usaremos esta librería para la representación de los resultados obtenidos de este trabajo.

4.2. Programas

A continuación explicaremos como se han generado y el funcionamiento de los programas de Python que se han entregado junto con esta memoria. Podemos distinguir en la carpeta de Programas otras cuatro carpetas llamadas Minuto, Diario, Semanal y Mensual, con tres programas Python en cada una, llamados calculo_prima_****.py, grafica_ganancias_****.py y grafica_precios_****.py, dónde las estrellas representan si lo calcula al minuto, al día, a la semana o al mes.

4.2.1. calculo_prima_****.py

Para empezar a programar necesitaremos importar la librería Pandas, la cual importamos como pd, para que a la hora de llamar a un método incluido en esta librería sólo haya que escribir esta abreviación y no pandas por completo. Posteriormente nos encontramos con la librería NumPy importada con la abreviación np, la librería math y por último la de de quadpy.

Una vez que todo está importado se empieza a programar este código, que consta de dos partes:

Primera Parte

En esta parte del programa se crean las siguientes variables:

- bitcoin: Es una matriz creada con la librería Pandas, la cual incluye todos los datos del histórico descargado de Internet.
- bitcoin2: Es la matriz anterior transformada a partir de la librería NumPy, para poder utilizar posteriormente las funciones de esta librería con la matriz. Lo único que perdemos con esta transformación es el encabezado que tenía la matriz.
- C: Es un vector extraído de la matriz anterior, el cual está formado por la cuarta columna de esta. Este vector contiene todos los precios de cierre a cada minuto del bitcoin. Este, dependiendo de la versión en la que nos situemos, lo encontraremos dividido en minutos, días, semanas o meses. En el caso de los tres últimos, se divide el vector en grupos de ese tiempo y se crea uno nuevo que tendrá como componentes la media que omite los valores NaN de los precios de cada grupo.

[9] ver el tutorial de Pandas
[10] ver la página web de Kaggle
[11] ver el tutorial de Matplotlib

- n: Es la longitud del vector C, calculado con un método de la librería NumPy.
- r: Es la rentabilidad del bitcoin, calculada a través de la ecuación 22. Este cálculo ha requerido varias partes:
  1. Se crea un vector vacío cuya longitud sea n-1, para que pueda contener las rentabilidades obtenidas entre cada periodo.
  2. Se genera un bucle for que lo que va a hacer es calcular cada una de esas rentabilidades anteriormente mencionadas usando la ecuación 23 y posteriormente las almacena en ese vector vacío que creamos.
  3. Se hace el cálculo de la rentabilidad como la media de todas las rentabilidades entre cada periodo.
- sigma: Es la volatilidad del bitcoin, calculada a través de la ecuación 24. Para este cálculo se ha requerido la librería math y el calculo de la media omitiendo los valores NaN a través del método de la librería NumPy y la rentabilidad calculada anteriormente.

Segunda Parte

Esta segunda parte es en la que se define la función prima(P,K,T), la cual recibe como argumentos el precio del instrumento financiero justo en el momento anterior del que queremos calcular la prima, el precio de ejecución de la opción y la madurez o tiempo que esta va durar. Esta función lo que contiene es básicamente el método de Black-Scholes expuesto en la ecuación 19, y calculado por partes.

- Primero nos encontramos con parte de la función $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} exp(-t^2/2)dt$, en concreto la función $exp(-t^2/2)$, que es la parte que está dentro de la integral, con el objetivo de ser definida para su posterior uso y cálculo.
- Segundo nos encontramos dos bloques de cuentas similares en los que ambos empiezan con el cálculo de d_1 y de d_2 respectivamente, definidas estas variables en la ecuación 20.
- Seguido de esto tenemos el cálculo de la integral a través de la librería quadpy de la función definida en el primer paso, poniendo en cada bloque como límite superior d_1 y d_2 respectivamente.
- Cada bloque termina multiplicando el resultado obtenido de la integral por la parte de la función $\Phi(x)$ que no definimos anteriormente, siendo esto $\frac{1}{\sqrt{2\pi}}$.
- Por último, cómo ya se tiene todos los valores necesarios, se hace el cálculo de la prima a través de la ecuación 19.

Con esto finalizamos la explicación del programa que calcula la prima de una opción de compra, la cuál utilizaremos e importaremos en en el siguiente programa.

Nota: Como ya mencionamos anteriormente, el programa del cálculo de la prima de opciones que se ejecutan al minuto es ligeramente diferente. Este no contará con la primera parte, es decir con el cálculo de la rentabilidad y la volatilidad, si no que serán argumentos de la función prima.

4.2.2. grafica_ganancias_****.py

Para este programa necesitaremos importar las librerías Pandas, NumPy, math, matplotlib, datetime y la función prima del programa anterior. Este programa lo que hará será hacer una simulación como si hiciésemos una opción a cada periodo de tiempo y representar la ganancias obtenidas a lo largo del tiempo en una gráfica.

Una vez que todo está importado se empieza a programar este código, que consta también de dos partes:

Primera Parte

Esta parte es prácticamente igual que la primera del programa anterior, destinada a la creación de las variables, teniendo exactamente las mismas que antes además de estas:

- x: Es un vector, que representa el intervalo tiempo del cual vamos a calcular las ganancias que nos dan las sucesivas opciones.
- t: Es la longitud del vector x mencionado anteriormente.
- y: Es el eje de ordenadas, el cual contendrá las ganancias acumuladas y se establece que su primer valor es 0.
- inicio: Es la fecha de inicio del intervalo de datos del vector x.
- d: Es el eje de abscisas, el cual contará con los minutos del vector x expresados en fecha y hora, partiendo de la fecha calculada anteriormente.

Segunda Parte

Esta parte está dedicada al cálculo de de las ganancias para ir completando el vector y. Para esto, crearemos un bucle for que consta se los siguientes cálculos:

- Primero calcularemos el precio de ejecución de las formas que vimos en el capítulo anterior, es decir, como el anterior $\pm$ 5 %.
- Luego calculamos la prima de la opción a través del precio anterior al que queremos calcular la opción, el precio de ejecución previamente calculado y una madurez de 1 plazo de tiempo. Esta puede ser de un minuto, un día, una semana o un mes, según el programa en el que nos encontremos.
- Posteriormente calculamos si el precio de ejecución está por encima del precio real o no. Esto lo hacemos para saber si la opción se ejecutará o no y poder establecer las ganancias.
- Para finalizar el bucle establecemos que la siguiente componente del vector y será la anterior más las ganancias que hemos calculado.

Por ultimo vemos unos comandos que lo que hacen es dibujar la gráfica y que modifican la fecha para que se muestre hasta el dato que nos interese. Esto quiere decir que cuando trabajemos en minutos se mostrarán los datos de la fecha y hora hasta los minutos, mientras que si trabajamos en meses, sólo se mostrarán el mes y el año.

Nota: En el caso de los minutos, este programa sería casi idéntico, salvo por que la rentabilidad y la volatilidad son calculadas solamente del tramo que usaremos, además de que son introducidas en la función prima como argumentos.

4.2.3. grafica_precios_****.py

Para este programa necesitaremos importar las librerías Pandas, NumPy, matplotlib y datetime, con la finalidad de dibujar la gráfica con el tramo de precios que se ha usado en el programa anterior. Esto será usado en el capítulo siguiente para comparar los resultados obtenidos con respecto a las ganancias y como está evolucionando el precio del bitcoin en ese intervalo. Para ello programaremos que el eje de abscisas sea el vector x del programa anterior transformado a fecha y hora con datetime y que el eje de coordenadas sean los precios de cierre en ese tramo.

5 Resultados

En este capítulo veremos y analizaremos los resultados obtenidos por los programas expuestos en el capítulo anterior, viendo como evolucionarían nuestras ganancias si hiciésemos una opción en los distinto periodos de tiempo.

Lo primero de todo, analizaremos la evolución la volatilidad a lo largo de la historia del bitcoin:

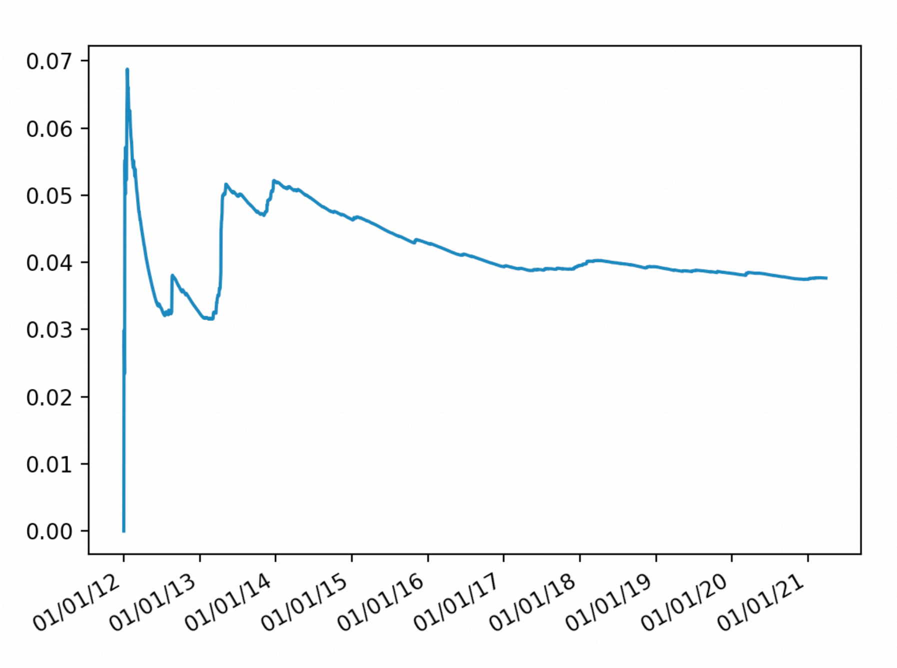

Ilustración 7: Evolución de la volatilidad.

Trabajar con Bitcoin nos lleva aplicar un algoritmo con un estilo agresivo. Cuanto mayor es la volatilidad, más agresivo será este, y mayores riesgos correremos. Estos algoritmos buscan obtener beneficios, por lo tanto son del tipo “Profit Seeking Algorithms”.

5.1. Minuto

Comenzaremos analizando las ganancias que obtenemos si comprásemos una opción al minuto. En este caso no es posible analizar toda la gráfica al mismo tiempo, puesto que habría que ejecutar algo más de 4,8 millones de puntos, además del hecho de que muchos de esos datos son NaN y en cuanto se topa el programa con uno de ellos este se detiene. Lo primero de todo será introducir al lector unos conceptos con los que vamos a trabajar:[1]

[1]Obtenidas estas definiciones en los apuntes de la asignatura de Análisis de los Mercados de Valores del Grado en Administración de Empresas

- **Tendencia Alcista:** Los máximos y los mínimos alcanzados por el precio cada vez son más altos.

- **Tendencia Bajista:** Los máximos y los mínimos alcanzados por el precio cada vez son más bajos.

- **Tendencia Lateral:** El precio se mueve de una forma lateral, sin una tendencia definida.

A continuación plantearemos las distintas situaciones con las que nos podemos encontrar:

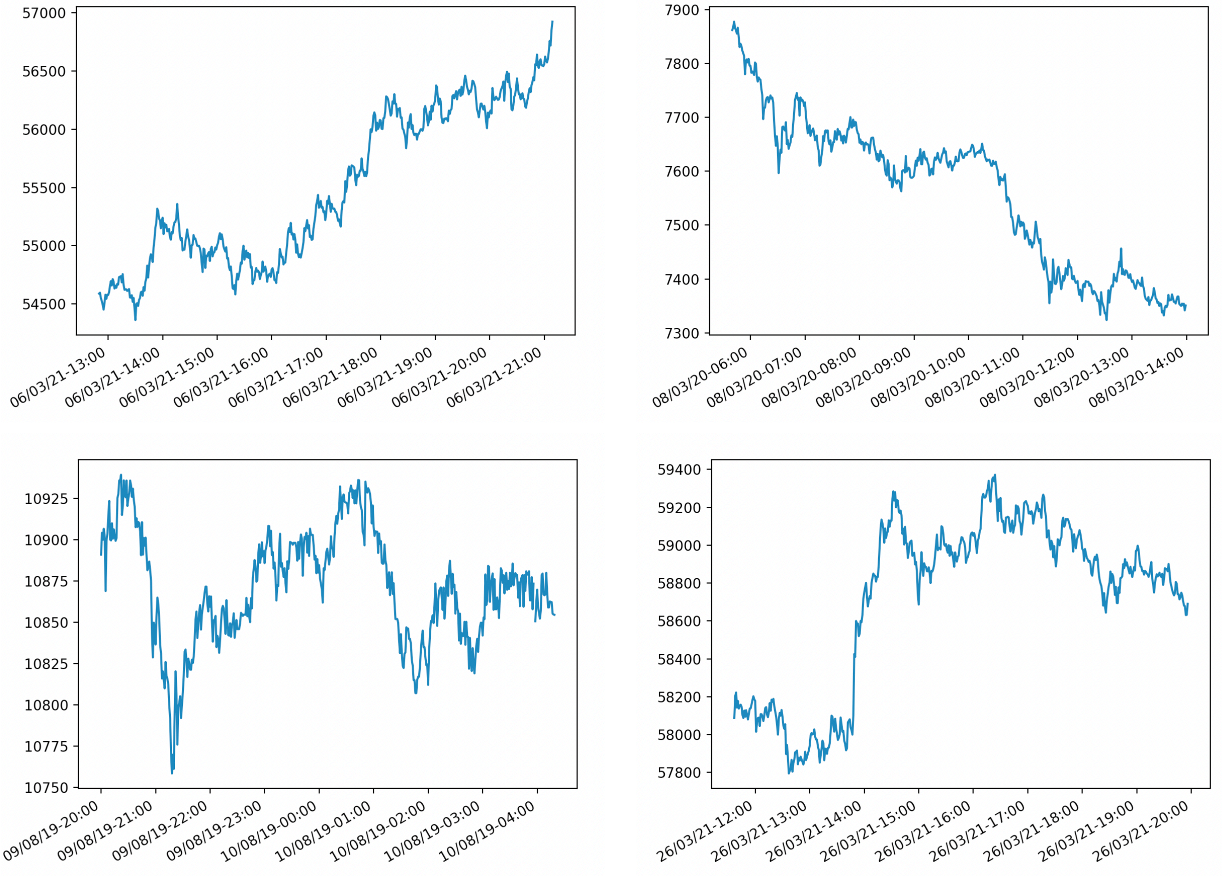

Ilustración 8: Distintos tramos de precios al minuto del bitcoin.

En la Ilustración 8 podemos observar tramos de 500 minutos todos ellos, en los que vemos uno con tendencia alcista, otro con tendencia bajista, otro con tendencia lateral y por último un tramo general escogido de forma aleatoria que son casi los últimos datos de precios que disponemos.

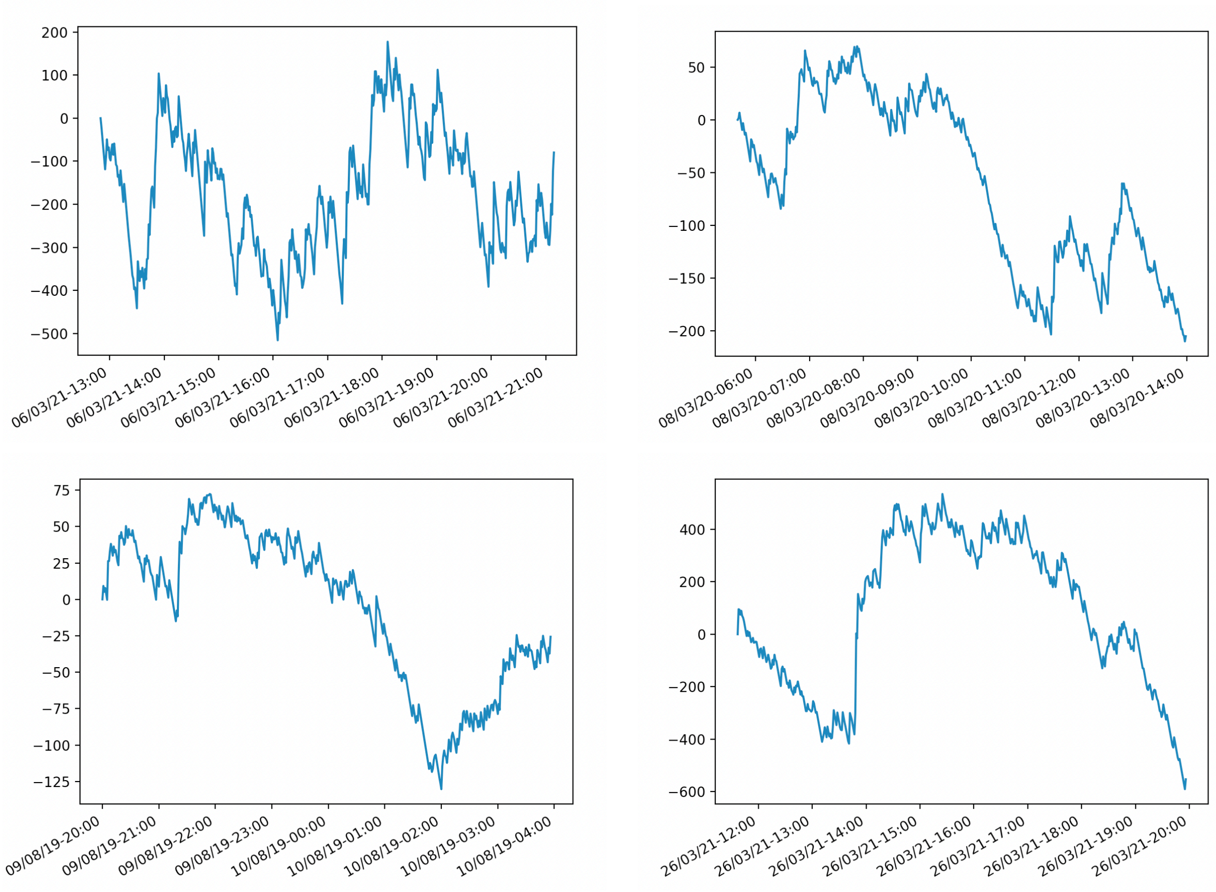

Ilustración 9: Evolución de las ganancias acumuladas al minuto en los tramos de la Ilustración 8.

En la Ilustración 9 vemos las ganancias que obtendríamos, correspondiéndose cada una de estas a la gráfica de precios que ocupa su mismo lugar. Observamos los siguiente:

- En el caso de la tendencia alcista vemos como las ganancias y la pérdidas tienden a estabilizarse.
- En el caso de la tendencia bajista vemos como se ven mermadas nuestras ganancias, pero en cuanto hay una pequeña recuperación del precio las ganancias se disparan.
- En el caso de la tendencia lateral también vemos como las ganancias y la pérdidas tienden a estabilizarse.
- En el caso aleatorio hay una primera parte de tendencia lateral, en la que las ganancias y las pérdidas se compensan. Posteriormente se observa un fuerte crecimiento de las ganancias debido a un gran aumento de los precios. Por último, estas ganancias tenderán otra vez al equilibrio sobre los 400 dólares, que terminarán cayendo debido a la caída que se observa al final del tramo.

5.2. Diario

En esta sección analizaremos lo que ocurre si compramos una opción al día, siguiendo unas expectativas bajistas y unas alcistas. Ahora, y en las secciones sucesivas, si que podremos analizar toda la gráfica a la vez.

Primero de todo, para poder evaluar los resultados obtenidos expondremos la Ilustración 10, la cuál contiene los precios del bitcoin al día. Se ha transformado de minutos a días siguiendo lo expuesto en el capítulo anterior.

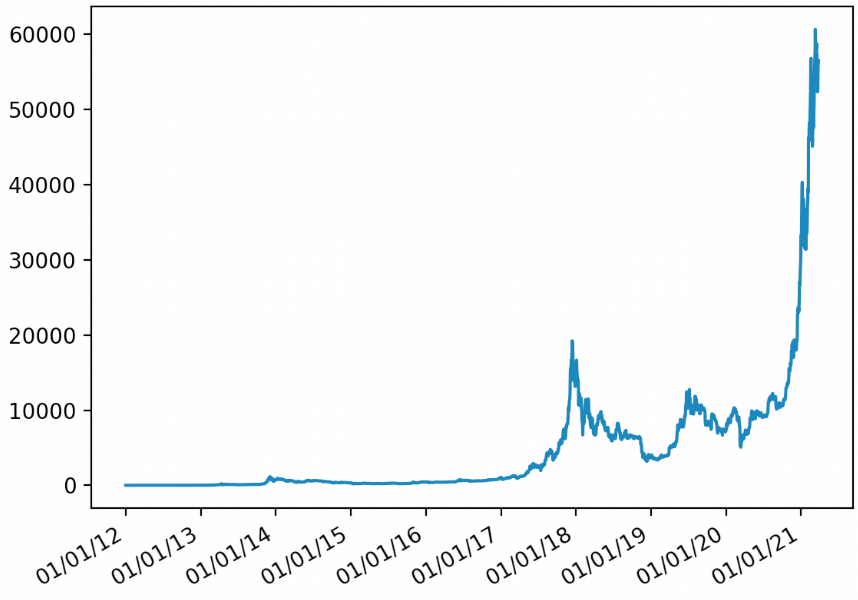

Ilustración 10: Precio diario del bitcoin.

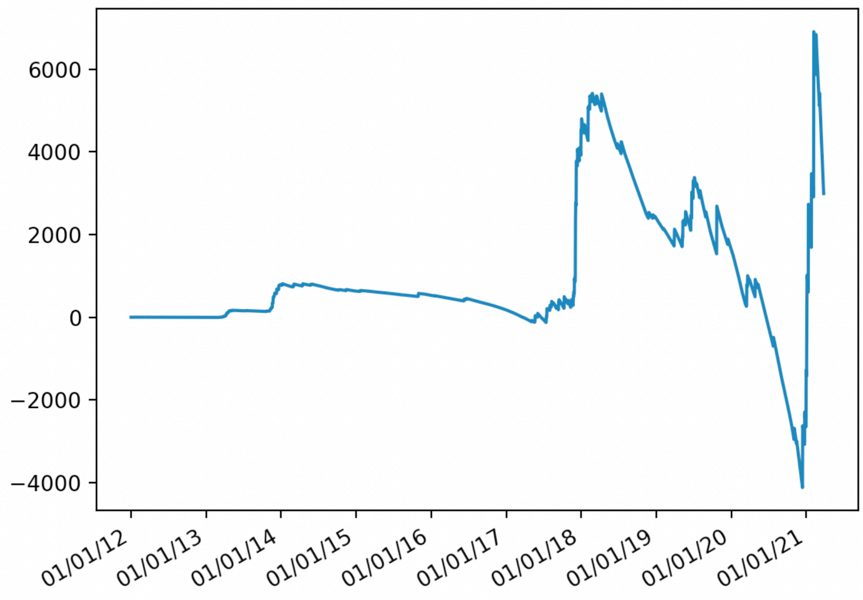

Ilustración 11: Evolución de las ganancias acumuladas al día según expectativas alcistas.

En la Ilustración 11 vemos como las ganancias y las pérdidas tienden a estabilizarse. Durante la primera mitad de la historia de precios del bitcoin, las ganancias se mantienen estables. No es hasta entorno al 1 de enero de 2018 cuando el crecimiento del precio del bitcoin se hace más fuerte, y por lo tanto, esas opciones negociadas con precios de ejecución de un 5 % superior al precio anterior empiezan a ejecutarse, resultando en beneficios. Una vez que estos precios tienden a caer las ganancias también lo hacen, puesto que el precio de ejecución será superior al precio del mercado, no ejecutándose esta y resultado en pérdidas sólo pagar la prima.

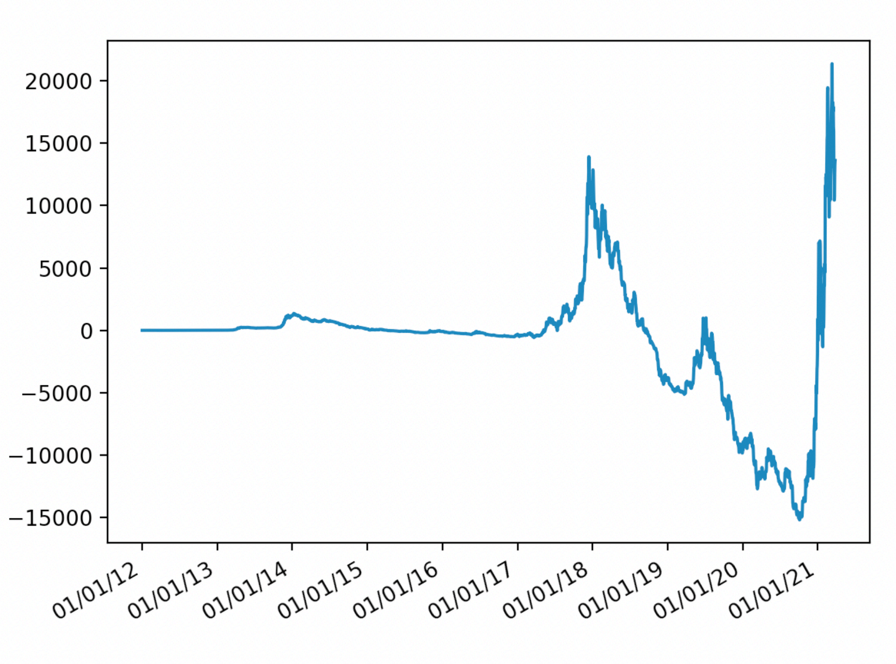

Ilustración 12: Evolución de las ganancias acumuladas al día según expectativas bajistas.

En la Ilustración 12 vemos una situación similar a la anterior, pero en la que las ganancias y las pérdidas máximas obtenidas se han disparado. Ahora lo que estamos haciendo es negociar opciones teniendo expectativas bajistas sobre el precio de bitcoin, en concreto se negocian a un precio de ejecución de un 5 % menos que el precio anterior. Esto provoca que cuando el precio del bitcoin siga una tendencia alcista obtengamos muchos beneficios, pero la prima a pagar también sea alta; y que cuando el precio siga una tendencia bajista, si este cae más de un 5 % resultará en pérdidas de sólo pagar esa alta prima.

Teniendo en cuenta la información actual del precio del bitcoin vista en la Ilustración 5, estos beneficios siguiendo ambas estrategias pueden que se hayan visto mermados debido a la fuerte del precio.

5.3. Semanal

En esta sección también analizaremos de dos formas lo que ocurre si compramos una opción a la semana, siguiendo unas expectativas bajistas y unas alcistas. Como en las secciones anteriores, lo primero de todo será exponer la Ilustración 13, la cuál contiene los precios del bitcoin a la semana, para poder evaluar los resultados obtenidos.

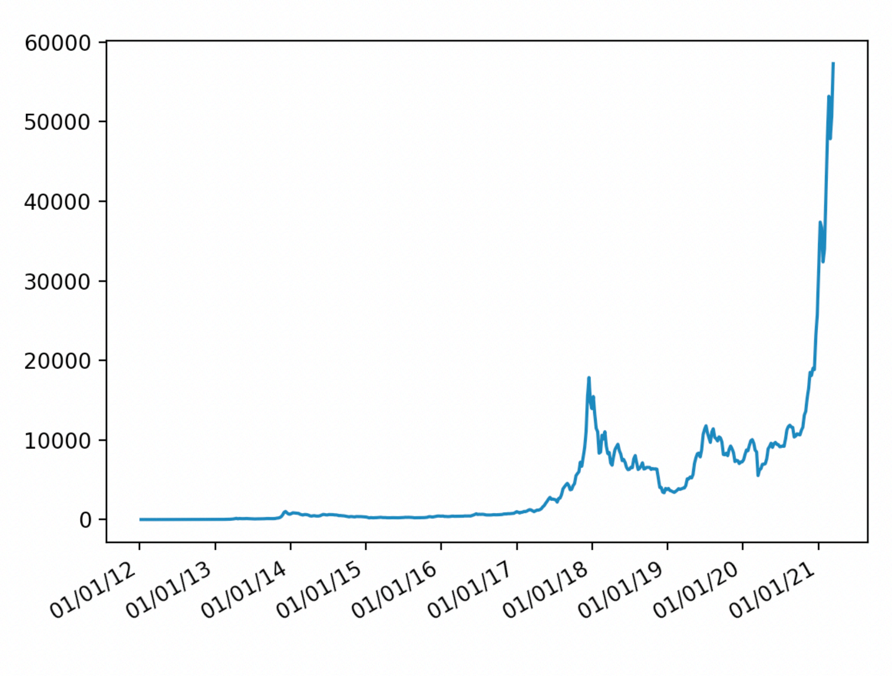

Ilustración 13: Precio semanal del bitcoin.

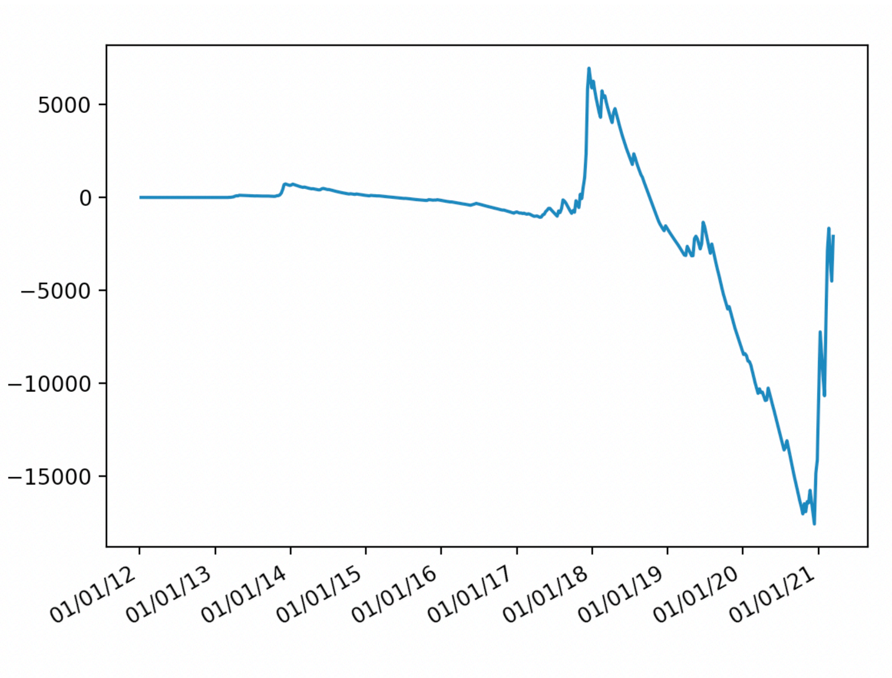

Ilustración 14: Evolución de las ganancias acumuladas a la semana según expectativas alcistas.

En la Ilustración 14, como en el caso anterior, vemos como las ganancias y las pérdidas tienden a estabilizarse. El comportamiento de las ganancias también es similar al del caso anterior, siguiendo las mismas tendencias, sólo que en el caso de las ganancias lo hacen a un ritmo inferior, y en el caso de las pérdidas a un ritmo superior. Esto se debe a que la prima a pagar es mayor, puesto que cuanto más largo es el periodo de tiempo más fácil es que ocurra ese crecimiento esperado de un 5 %.

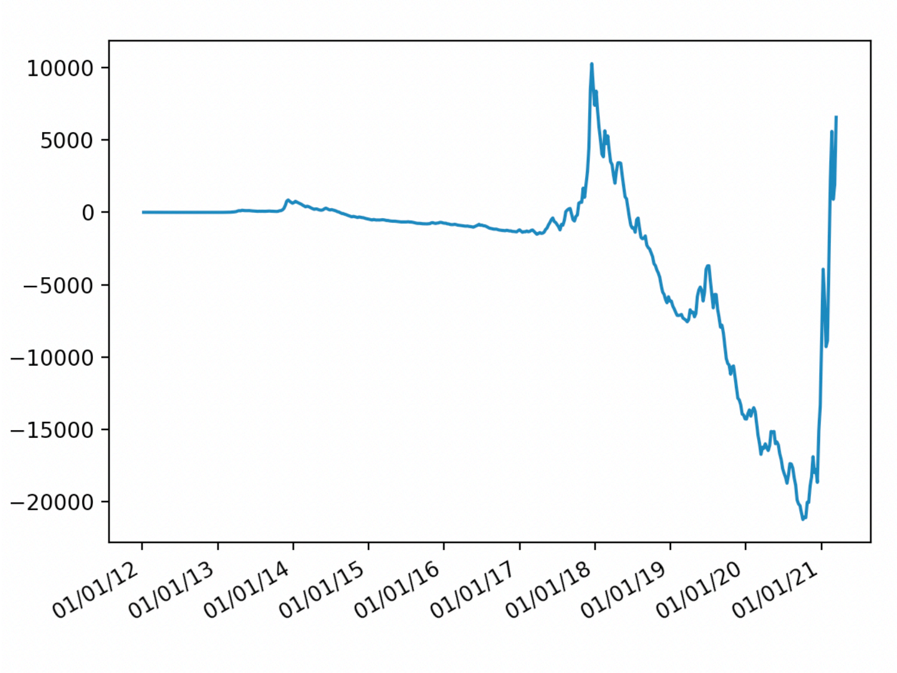

Ilustración 15: Evolución de las ganancias acumuladas a la semana según expectativas bajistas.

En la Ilustración 15 vemos una situación similar a la anterior, pero en la que las ganancias y las pérdidas máximas obtenidas también se han disparado como en el caso diario. Como en la sección anterior, esto implica beneficios más altos, en el caso de haberlos, pero también primas más altas.

En este caso también veríamos mermadas estas pocas ganancias debido al hecho de que el precio del bitcoin no pasa por su mejor momento.

5.4. Mensual

Esta sección la dedicaremos a la compra de una opción al mes, siguiendo también las mismas estrategias que en las secciones anteriores. Lo primero de todo será exponer la Ilustración 16, la cuál contiene los precios del bitcoin al mes, para poder evaluar los resultados obtenidos.

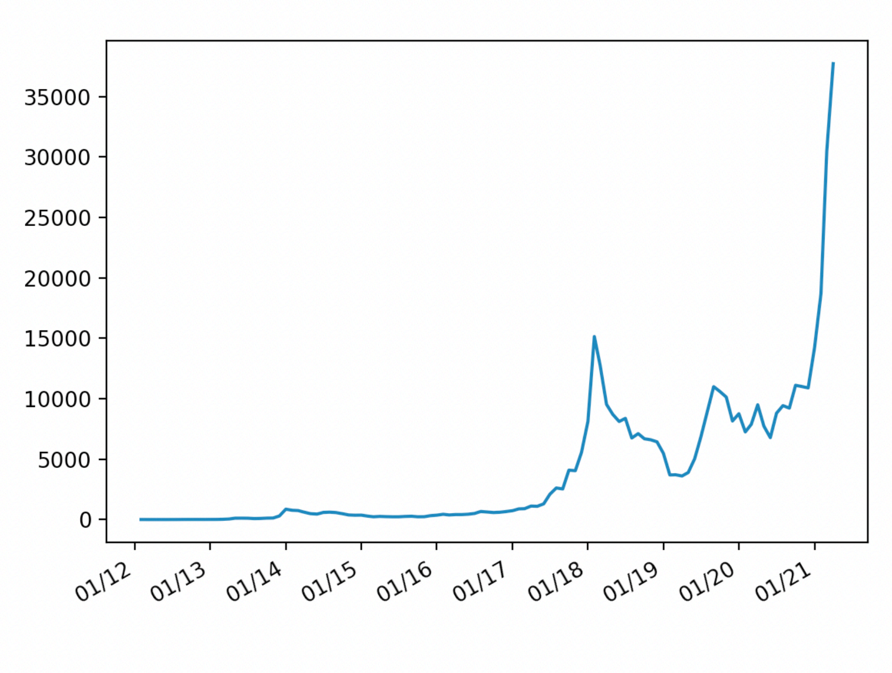

Ilustración 16: Precio mensual del bitcoin.

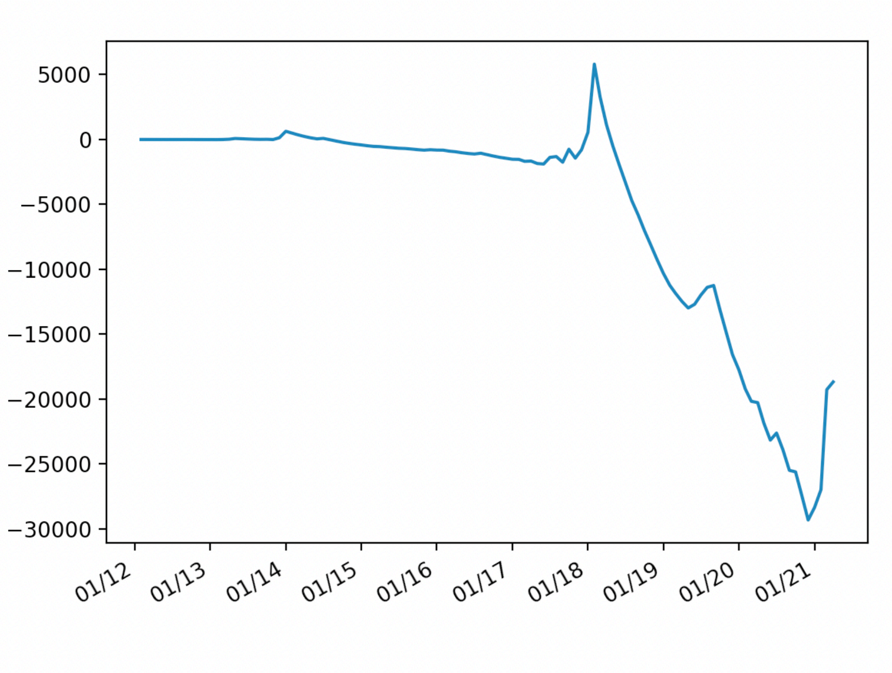

Ilustración 17: Evolución de las ganancias acumuladas al mes según expectativas alcistas.

En la Ilustración 17, como en los dos casos anteriores, vemos que el comportamiento de las ganancias es similar, siguiendo las mismas tendencias, sólo que en el caso de las ganancias lo hacen a un ritmo aún menor, y en el caso de las pérdidas a un ritmo aún mayor. Esto se debe a que la prima a pagar vuelve a ser mayor, puesto que el plazo de tiempo a vuelto a crecer.

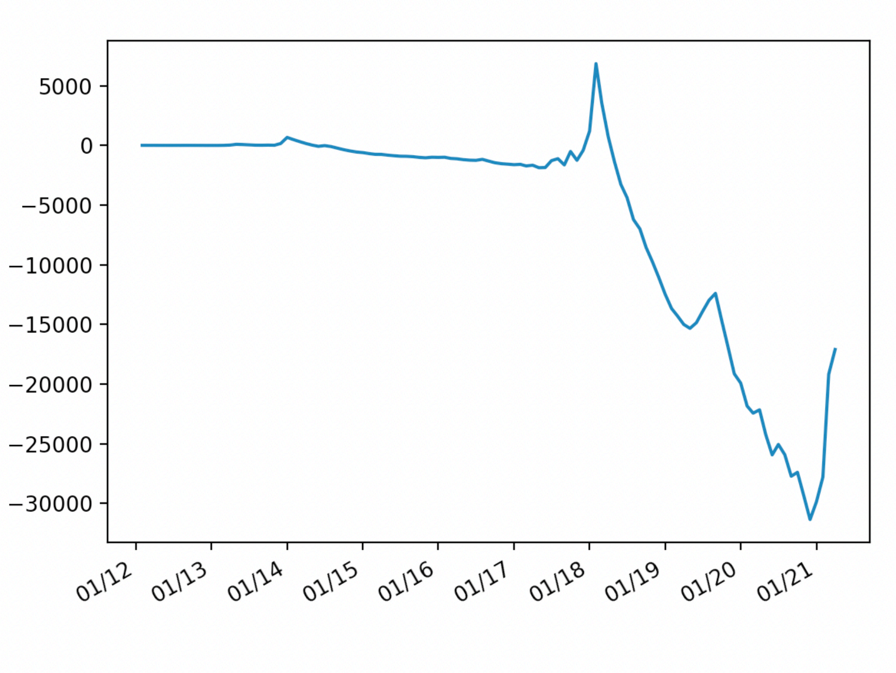

Ilustración 18: Evolución de las ganancias acumuladas al mes según expectativas bajistas.

Entre la Ilustración 17 y la Ilustración 18 no hay grandes diferencias y por lo tanto obtendríamos los mismos resultados si seguimos una estrategia o la otra.

En conclusión, una vez vistos todos estos resultados podemos afirmar que la forma menos arriesgada y más recomendable para comprar acciones sería negociarlas diariamente con expectativas alcistas. En el caso de querer obtener más beneficios habría que tomar más riesgos, por lo tanto sería recomendable comprar acciones también diariamente pero con expectativas bajistas. Aun así, ahora no es el mejor momento para invertir de ninguna forma, puesto que tras la gran caída que sufrió el bitcoin sólo nos llevaría a obtener pérdidas. Lo más recomendable ahora mismo sería invertir directamente en bitcoin y esperar a venderlo cuando el precio de este haya vuelto a subir.

Referencias

R. B. Ash y C. A. Doléans-Dade. *Probability and Measure Theory*, chapter 2 and 9. Academic Press, 2 edition, 2000.

P. Champagne. *El Libro de Satoshi*, chapter 1, 2 and 5. Blockchain España, 2008.

P. Glasserman. *Monte Carlo Methods in Financial Engineering*, chapter Appendix B. Springer, 2003.

R. Kissel. *Algorithmic Trading Methods*, chapter 1. Academic Press, 2 edition, 2020.

G. Leobacher y F. Pillichshammer. *Introduction to Quasi-Monte Carlo Integration and Applications, Compact Textbooks in Mathematics*, chapter 7 and 8. Birkhäuser, 2014.

A. L. Molina. Medidas invariantes absolutamente continuas, 2020. https://www.um.es/documents/118351/14467409/TFMAlejoLópezMolina.pdf/b66984fa-fd19-482c-a36f-1b7a2f538eec

S. Nakamoto. Bitcoin, a peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, 4, 2008.

J. A. Rice. *Mathematical statistics and data analysis*. Cengage Learning, 2006.

J. M. Uribe Gil y I. M. Ulloa Villegas. Revisando la hipótesis de los mercados eficientes: nuevos datos, nuevas crisis y nuevas estimaciones. *Cuadernos de Economía*, 30(55):127–154, 2011.

Varios. *Blockchain, bitcoin y criptomonedas*, pages 12–20. Profit Editorial, 2018.

S.-T. Yeh et al. Using trapezoidal rule for the area under a curve calculation. *Proceedings of the 27th Annual SAS® User Group International (SUGI'02)*, 2002.

Zielak. Bitcoin historical data, 2017. https://www.kaggle.com/mczielinski/bitcoin-historical-data