UNIVERSIDAD
COMPLUTENSE
MADRID

Facultad de Ciencias Económicas y Empresariales

Doble Grado en Administración y Dirección de Empresas e Ingeniería Informática

Trabajo de Fin de Grado

Análisis de criptomonedas usando R

Autor: Iván Hernández Roldán

Tutora: Zuleyka Díaz Martínez

Curso académico: 2022/023

Convocatoria: Junio

ÍNDICE

1. RESUMEN

Este trabajo final de grado aborda un estudio comparativo de dos criptomonedas prominentes: Bitcoin y Ethereum. A lo largo del análisis, se ha calibrado la distribución empírica de los retornos semanales de ambas criptomonedas a su correspondiente distribución teórica más afín, que podría ser la normal, la log-normal, la logística o la beta entre otras.

La finalidad de ajustar estas distribuciones es habilitar el cálculo de las probabilidades de ciertos eventos, en este caso, los retornos semanales de cada criptomoneda. Al determinar qué distribución ya estudiada se ajusta mejor a los datos empíricos, se pueden calcular estas probabilidades utilizando dicha distribución.

Finalmente, después de ajustar las distribuciones, se procedió a comparar las probabilidades de los retornos para cada criptomoneda. Además, se realizó un análisis exhaustivo de las conclusiones extraídas a partir de los hallazgos obtenidos.

2. INTRODUCCIÓN

Las criptomonedas son tipos de monedas digitales o virtuales. A diferencia del dinero físico emitido por los gobiernos (como los dólares o euros), las criptomonedas operan en una red descentralizada basada en la tecnología *blockchain*.

El blockchain es un tipo de base de datos distribuida que registra las transacciones en bloques que están vinculados entre sí en un "cadena". Esta tecnología permite que la información de las transacciones se comparta por toda la red de usuarios de la criptomoneda, lo que hace que las transacciones sean transparentes y casi imposibles de manipular o falsificar (Santander, 2023).

Las criptomonedas recurren a cifrados criptográficos para encriptar la información de las transacciones con el objetivo de garantizar su seguridad. Cuando se realiza una transacción, esta se cifra en un código que solo puede ser descifrado por la parte destinataria, asegurando que la información no pueda ser interceptada o modificada durante la transmisión.

Además, permiten controlar la creación de unidades adicionales. Este concepto se refiere a cómo las criptomonedas limitan la creación de nuevas instancias. Por ejemplo, en el caso de Bitcoin, la cantidad total que puede existir está limitada a 21 millones de monedas. Esto se logra a través de un proceso llamado "minado", donde los "mineros" usan poder computacional para resolver problemas matemáticos complejos y validar transacciones en la red blockchain. Como recompensa, crean nuevos Bitcoins, pero el diseño de la red hace que esta tarea sea cada vez más difícil a medida que se acerca al límite de 21 millones (LISA Institute, 2023).

Por último, las criptomonedas incluyen la verificación de la transferencia de activos. Cuando se realiza una transacción con criptomonedas, la red de usuarios verifica la transacción para asegurarse de que el remitente tiene la cantidad de criptomoneda que desea enviar y que no la ha enviado a otra persona al mismo tiempo. Esto se hace a través de un proceso de consenso en la red blockchain, donde varios participantes tienen que validar la transacción antes de que se pueda agregar al registro público de transacciones (el blockchain). Esto ayuda a prevenir el "gasto doble", que es un problema potencial en los sistemas digitales donde la misma moneda podría copiarse y gastarse más de una vez.

Bitcoin, lanzado en 2009, fue la primera criptomoneda y sigue siendo la más conocida. Desde entonces, se han creado muchas otras criptomonedas, cada una con sus propias características y usos. Algunas de las criptomonedas más populares son Bitcoin, Ethereum, Ripple, Litecoin, Cardano, etc.

Las criptomonedas pueden ser usadas para una variedad de propósitos, incluyendo compras en línea, inversión, remesas, y en algunos casos, incluso para evadir controles de capital y sanciones. Sin embargo, también han sido criticadas por su uso en actividades ilegales, su volatilidad en el mercado y los riesgos asociados a su seguridad y privacidad.

1.1 OBJETIVO

Hay muchas criptomonedas disponibles en el mercado, pero las dos más populares y con mayor capitalización de mercado son Bitcoin y Ethereum.

Particularmente, este trabajo se centra en el análisis de Bitcoin y Ethereum debido a la alta volatilidad que han experimentado en los últimos años para analizar los comportamientos de estas criptomonedas relativamente nuevas y compararlos entre sí.

El objetivo concreto de este análisis, realizado con R, es estudiar las fluctuaciones en la cotización de las criptomonedas seleccionadas durante un periodo de tiempo determinado y analizar las probabilidades de estas fluctuaciones. Además, se compararán las probabilidades de obtener determinados rendimientos en las cotizaciones de Bitcoin y Ethereum. Con esto, se consigue evaluar su grado de volatilidad y determinar cuáles son más riesgosas para invertir.

1.2 METODOLOGÍA

Para ello, en primer lugar, se recopilarán los datos históricos de cotizaciones de las criptomonedas elegidas para analizar. Se obtendrán estos datos del sitio web especializado en finanzas *Yahoo Finance*.

Una vez se tengan los datos, estos serán procesados con R para que puedan ser analizados. En esta etapa se estudiará la posibilidad de eliminar valores que distorsionan las conclusiones, se filtrarán los datos por rangos de fechas interesantes para el trabajo y se transformarán los datos en un formato adecuado para su análisis.

El análisis consistirá en dos fases. La primera fase se conoce como ajuste de las distribuciones empíricas y sirve para explorar metódicamente la distribución de los datos con el objetivo de identificar que distribución de las ya estudiadas, como por ejemplo la distribución normal o la distribución logística, se asemeja más a la distribución observada de los retornos. Concretamente, en esta primera fase, se comienza determinando cuáles son las distribuciones candidatas a ser la que mejor se ajuste, y por último, se hace un ajuste más preciso por estimación de máxima verosimilitud para definir finalmente cuál es la distribución que mejor se ajusta. Para todo esto, se utilizarán herramientas de visualización como gráficos de línea, histogramas, gráficos de dispersión, etc.

En la segunda fase, se analizarán la variabilidad y las probabilidades de que se den determinados retornos para cada una de las dos criptomonedas y se compararán. Con los resultados obtenidos, se realizarán unas conclusiones acerca de la idoneidad de la metodología utilizada.

Todo el trabajo de análisis se lleva a cabo mediante la implantación en el lenguaje de programación R (R Core Team, 2022) del código correspondiente en el entorno de desarrollo integrado (IDE) RStudio (Posit team, 2022). Todo lo relacionado con el código y algunos resultados obtenidos será explicado en a la sección 5 denominada "Programación con R".

Además, todos los desarrollos realizados en R han quedado reflejados en el repositorio GitHub del autor de este trabajo (Hernández Roldán, 2023).

3. ANÁLISIS DEL RETORNO DE CRIPTOMONEDAS

El *retorno* o rentabilidad en finanzas se refiere a la ganancia o pérdida realizada en una inversión en comparación con el precio de compra. Los retornos de un activo serían el cambio en su precio de un día para otro, y podrían expresarse en términos absolutos (la diferencia de precio) o en términos porcentuales (la diferencia de precio como porcentaje del precio inicial). Concretamente, en este trabajo se expresan en términos porcentuales.

La razón por la cual los retornos son a menudo el foco del análisis financiero, en lugar de los precios en sí, es que los retornos son generalmente más estables y comparables entre diferentes inversiones.

Para comprender esta afirmación, se plantea un ejemplo en el que se tienen dos acciones: acción A y acción B. La acción A está valorada en 1000€, y la acción B en 10€. Ambas suben 1€ en un día.

Si solo se contempla el cambio en el precio (1€ para ambas acciones), se podría concluir erróneamente que las dos acciones tuvieron el mismo retorno, pero esto no sería cierto. La acción A, que comenzó a 1000€ y subió 1€, tuvo un retorno del 0.1% (1€ / 1000€). En contraste, la acción B, que comenzó a 10€ y subió 1€, tuvo un retorno del 10% (1€ / 10€). Por lo tanto, aunque el cambio en el precio fue el mismo (1€), el retorno fue significativamente diferente entre estas dos acciones. Al analizar los retornos en lugar de los cambios en el precio, se compara de manera más justa y precisa el retorno de diferentes inversiones, independientemente de su precio original.

Por otra parte, en el contexto de los análisis estadísticos y financieros, las distribuciones de probabilidad se usan para modelar y entender la variabilidad, incertidumbre, o riesgo asociado con fenómenos o series de datos. Una distribución de probabilidad es una función matemática que describe el comportamiento de una variable, proporcionando las probabilidades de ocurrencia de todos los posibles valores que una variable puede tomar.

En este trabajo, se hará uso de las distribuciones de probabilidad de los retornos de diferentes criptomonedas para ayudar a entender, en primer lugar, la forma de dichos retornos. Es decir, las distribuciones pueden indicar si los datos están sesgados hacia un extremo, si están concentrados alrededor de una media, o si tienen una estructura más compleja.

Además, permiten calcular las probabilidades de los retornos futuros de una criptomoneda en el caso de que estos se ajusten a una distribución de probabilidad conocida y estudiada. Las distribuciones ya definidas (normal, logística, log-normal, beta, gamma, etc.) representan un comportamiento determinado y permiten calcular la probabilidad de cada suceso, bajo tales circunstancias. De esta forma, en el contexto de las inversiones, si se sabe que los retornos de una inversión siguen una distribución particular, esta se puede usar para calcular la probabilidad de obtener un cierto retorno en el futuro.

La frase “ajustar los datos a una distribución” hace referencia a la búsqueda de la distribución predefinida que más se parezca a la distribución real de dichos datos. Para aclarar este último punto, a continuación, se presenta una explicación acerca de cómo los comportamientos de dos variables que tengan la misma media aritmética y desviación típica o estándar pueden estar, a su vez, descritos por distribuciones muy diferentes.

Se consideran dos variables o conjuntos de datos diferentes, pero que tienen la misma media (0) y la misma desviación típica (1). Un conjunto de datos sigue una distribución normal, mientras que el otro sigue una distribución uniforme.

La distribución normal es la famosa "curva de campana" en la que los datos siguen una forma de campana, con la parte más alta de la campana en la media. Muchos fenómenos naturales siguen una distribución normal, como las alturas de las personas en una población. Si se tiene un conjunto de datos que sigue una distribución normal con media 0 y desviación estándar 1, la mayoría de los datos se agruparán alrededor de la media (0), y los valores se vuelven cada vez menos frecuentes a medida que te alejas de la media en cualquier dirección.

Por otro lado, la distribución uniforme es completamente diferente. En una distribución uniforme, cada valor dentro de un cierto rango tiene la misma probabilidad de ocurrir. Así que, si se tiene un conjunto de datos que sigue una distribución uniforme con media 0 y desviación estándar 1, los datos se distribuirán de manera uniforme a lo largo de todo el rango de valores. Por consiguiente, no habrá un agrupamiento de valores cerca de la media como lo hay en la distribución normal.

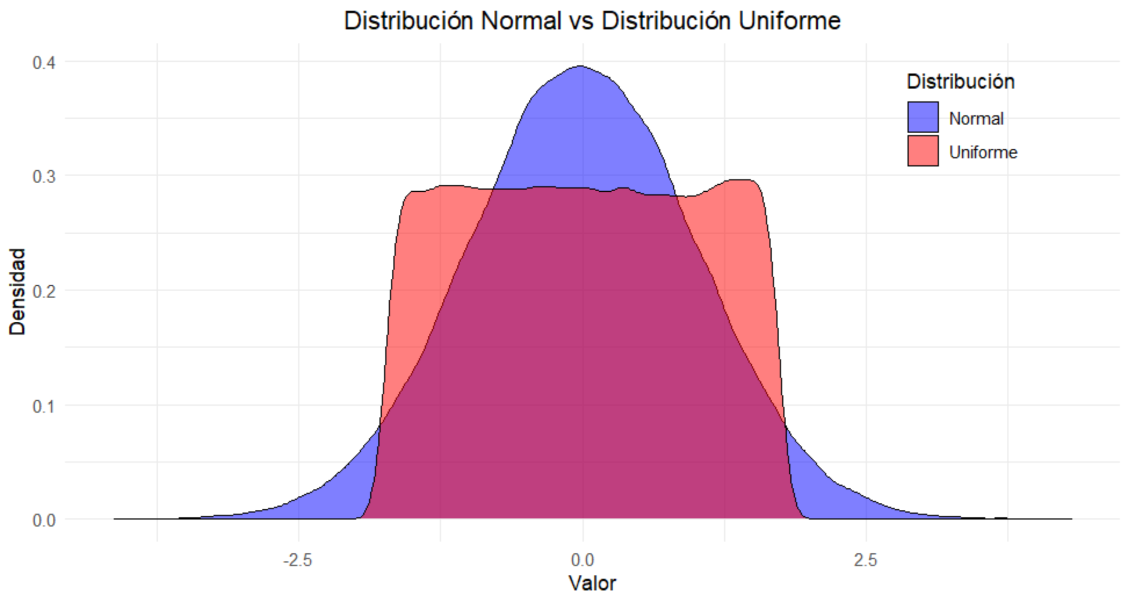

Figura 3.1

Las diferencias mencionadas se verificaron mediante la implementación descrita en la sección 4.1.1 y se reflejan en la figura 3.1, que representa los dos conjuntos de datos analizados en este ejemplo. En ambos casos, la media y la desviación estándar son 0 y 1, respectivamente. Sin embargo, las distribuciones de los datos tienen formas muy diferentes. El conjunto de datos, que sigue una distribución normal y se representa en color azul en la gráfica, tiene forma de campana, mientras que el conjunto que sigue una distribución uniforme, representado en color rojo, tiene forma de caja. Así que, aunque tienen la misma media y desviación estándar, representan fenómenos muy diferentes, lo que implica diferentes probabilidades de que la variable adquiera un valor futuro. Por tanto, esto explica la razón por la que, para analizar una variable y realizar predicciones correctas, hay que encontrar el tipo de distribución que más se ajuste a la distribución real.

La razón por la que los analistas intentan ajustar una distribución conocida a los datos, en lugar de simplemente usar la distribución empírica de los datos observados, es que las distribuciones de probabilidad conocidas tienen propiedades matemáticas bien establecidas que facilitan el análisis y la interpretación de los datos.

Por ejemplo, si se sabe que los retornos de una criptomoneda están descritos por una distribución normal con una media de 0.01 y una desviación típica de 0.02, y se quiere calcular la probabilidad de obtener un retorno (X) mayor que el 5%, el cálculo sería 1 – Probabilidad (X ≤ 5%). Este cálculo se puede realizar cómodamente utilizando R, tal y como queda expuesto en el apartado 4.1.2.

Es importante mencionar que ningún modelo de distribución de probabilidad captura perfectamente la realidad, y que siempre hay un grado de error en cualquier modelo. Es por eso que los analistas a menudo prueban varios modelos diferentes para ver cuál se ajusta mejor a los datos.

Por último, las distribuciones de probabilidad de los retornos de diferentes criptomonedas posibilitan la comparación de dichos retornos. Al ajustar las mismas distribuciones de probabilidad a diferentes conjuntos de datos, se pueden comparar las propiedades de estos conjuntos de datos, como por ejemplo la volatilidad.

3.1 AJUSTES DE LOS RETORNOS DE CRIPTOMONEDAS

Antes de comenzar a ajustar una distribución predefinida a los retornos de las diferentes criptomonedas que serán estudiadas en este trabajo, es importante determinar el horizonte temporal bajo el cual se quiere invertir. Esto se debe a que los retornos obtenidos por un determinado activo no son iguales si se analiza un periodo de un día, una semana, un mes, etc. Gracias al código reflejado en el apartado 4.2, se demuestra la afirmación anterior y se obtiene la figura 3.1.1 como resultado de analizar los retornos de Bitcoin durante los últimos dos años para diferentes rangos de tiempo.

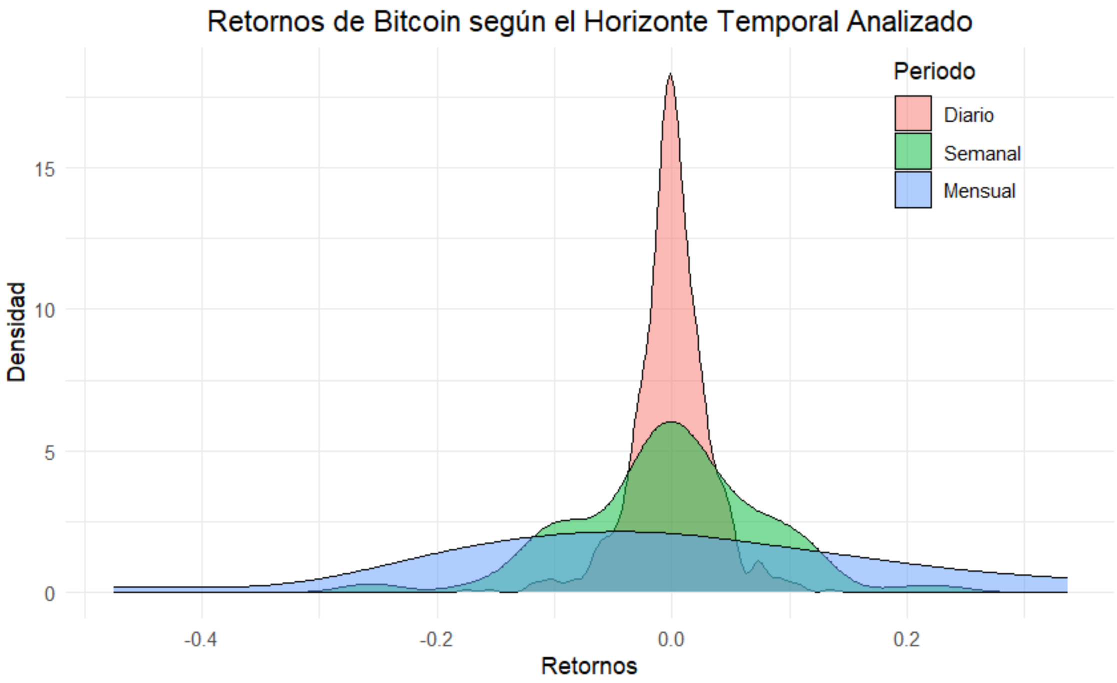

Figura 3.1.1

Con base en esto, queda evidenciado que no es lo mismo entrar a un mercado buscando una rentabilidad a corto plazo que buscarla a largo plazo. Por tanto, es necesario determinar el punto de vista que se va a plantear en el análisis que será realizado de ahora en adelante.

Algunos inversores en criptomonedas buscan rentabilidad a corto plazo, aprovechando la volatilidad de dicho mercado para realizar operaciones rápidas, a veces llamadas operaciones de comercio diario. Este enfoque puede ser potencialmente rentable si se logra entender las tendencias y los patrones diarios o semanales, pero también es arriesgado, ya que los precios de las criptomonedas pueden variar rápidamente y de manera impredecible.

Otros inversores adoptan un enfoque a largo plazo, comprando criptomonedas con la esperanza de que aumenten de valor con el tiempo. Esto puede ser menos arriesgado que el comercio diario, ya que no es necesario predecir fluctuaciones de precios a corto plazo, pero aun así implica un cierto nivel de riesgo, ya que el mercado de las criptomonedas es bastante volátil.

En este caso, se adoptará la posición de un inversor que busca obtener rentabilidades al cabo de siete días. Por lo tanto, en este trabajo se asumirá el riesgo derivado del mercado de las criptomonedas tratando de beneficiarnos de un análisis de los retornos a corto plazo.

3.1.1 AJUSTE DEL RETORNO DE BITCOIN

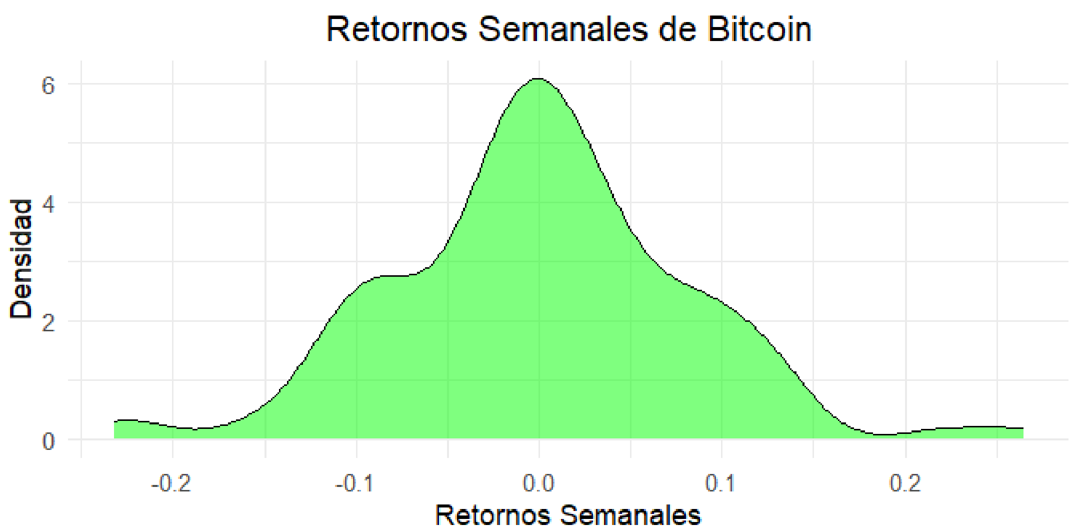

Figura 3.1.1.1

Los retornos semanales de Bitcoin durante los últimos dos años forman la distribución ilustrada en la figura 3.1.1.1. Esta imagen evidencia una media de retornos alrededor del 0% y una volatilidad muy grande del activo ya que sus retornos semanales varían en un rango aproximado desde -20% a 20%.

3.1.1.1 Elección de las distribuciones candidatas

Además de las métricas comunes como el valor máximo, valor mínimo, la media y la desviación típica, la asimetría y la curtosis son dos parámetros estadísticos que también se usan para describir y entender las características de la distribución de una variable aleatoria o una población.

La asimetría, o "skewness" en inglés, mide el grado de desviación de la simetría de una distribución. En una distribución simétrica, como la distribución normal, la asimetría es cero. Si la asimetría es positiva, la cola a la derecha de la distribución es más larga o gruesa que la izquierda. En cambio, si es negativa, la cola a la izquierda de la distribución es la más larga o gruesa. Por tanto, la asimetría es útil para entender la dirección y grado de la desviación de una distribución respecto a la simetría.

Por otra parte, la curtosis, o "kurtosis" en inglés, mide la concentración de los valores de la distribución en torno a su media. Compara la cantidad de datos cerca de la media y lejos de la media (en las colas). Una distribución normal tiene un coeficiente de curtosis de 3 y se denomina *mesocúrtica*. Si el coeficiente de curtosis es mayor que 3, los datos están muy concentrados en la media, formando una distribución *leptocúrtica*, más puntiaguda que una normal. En el caso opuesto, si la curtosis es menor que 3, la distribución se conoce *platicúrtica*, siendo esta más achatada que una distribución normal y con colas más gruesas (Sanjuán, 2022).

Con R, todos estos estadísticos se pueden obtener de una manera muy sencilla gracias a la función *descdist* de la biblioteca *fitdistrplus* (Delignette-Muller & Dutang, 2015). Para los datos reflejados en la figura 3.1.1.1, dicha función arroja los valores de los estadísticos mencionados anteriormente. Estos quedan presentados en sección 4.3.1, confirmando que sus valores describen correctamente la forma de la distribución de los retornos semanales de Bitcoin.

Además, esta función devuelve, de forma complementaria a estos valores, el gráfico de asimetría-curtosis expuesto por la figura 3.1.1.2. Este gráfico fue introducido por Cullen y Frey (1999). Este gráfico es una herramienta visual muy valiosa que muestra los valores de las distribuciones comúnmente utilizadas, facilitando la selección de las distribuciones más adecuadas para ajustarse a la distribución empírica de los retornos semanales de Bitcoin. Ciertas distribuciones, como la normal, uniforme, logística y exponencial, poseen solo un conjunto de valores posibles para asimetría y curtosis. Por ejemplo, en una distribución normal, la asimetría es 0 y la curtosis es 3, y por lo tanto, se representa como un solo punto en el gráfico. Sin embargo, hay distribuciones que ofrecen un rango de valores posibles para asimetría y curtosis. En estos casos, se dibujan áreas que representan estos rangos de valores posibles. Por ejemplo, las distribuciones gamma y log-normal se representan como líneas, mientras que la distribución beta se representa como un área más amplia. Cabe mencionar que la distribución de Weibull, aunque no se muestra en el gráfico, puede generar formas que se aproximan a las distribuciones log-normal y gamma, tal y como se señala en la leyenda del gráfico (Delignette-Muller & Dutang, 2015).

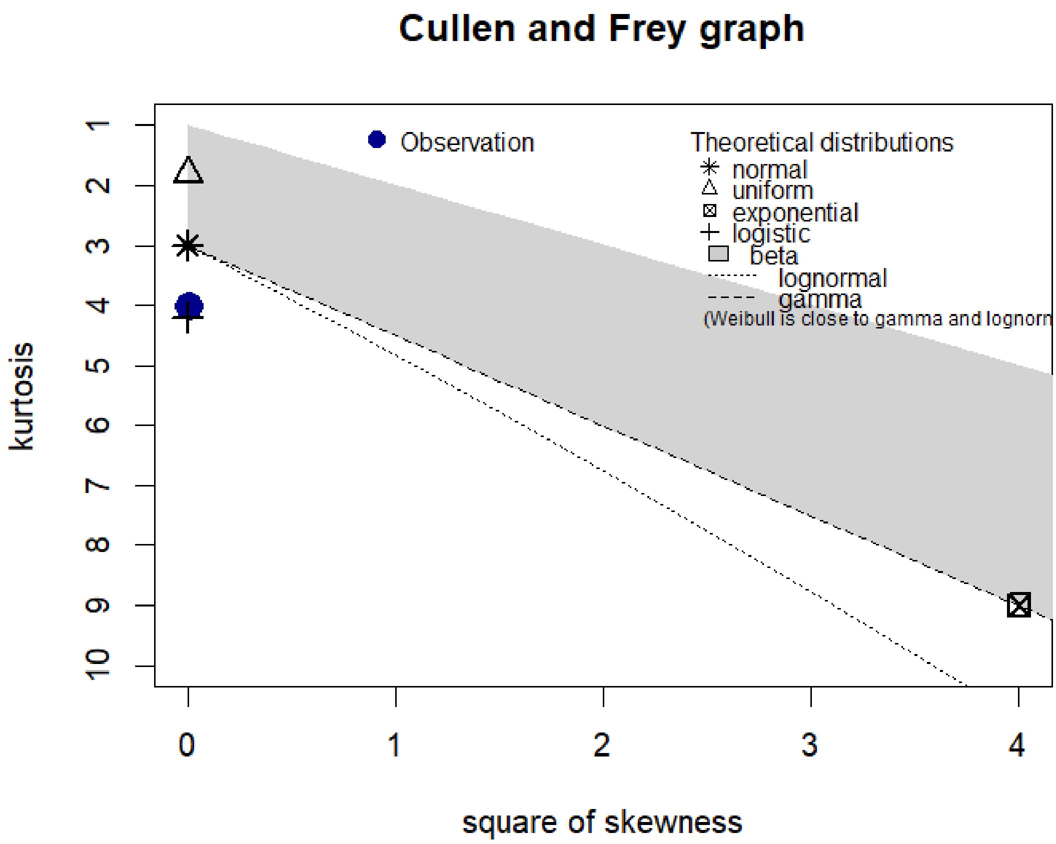

Figura 3.1.1.1.1

En este caso, el gráfico de asimetría-curtosis indica que, según los valores de simetría y curtosis de la distribución empírica, la distribución que mejor se ajusta a los datos observados es la distribución logística. Esta distribución es una distribución de probabilidad continua, que ha sido utilizada ampliamente en estadísticas, economía, medicina, entre otros campos de estudio. Es muy similar a la distribución normal (o gaussiana), pero tiene colas más pesadas. Esto significa que es más probable obtener valores que se desvían considerablemente de la media en comparación con la distribución normal.

Por tanto, en principio, la distribución que mejor se ajusta a los retornos semanales de Bitcoin es la distribución logística. De todas formas, para confirmar esta hipótesis se ha realizado el ajuste de distribuciones por estimación de máxima verosimilitud.

3.1.1.2 Ajuste de distribuciones por estimación de máxima verosimilitud

Este ajuste ha sido muy sencillo de implementar con R, tal y como demuestra el apartado 4.3.2, haciendo uso de la función *fitdist,* que también pertenece a la biblioteca *fitdistrplus* (Delignette-Muller & Dutang, 2015). En la llamada a *fitdist*, se debe especificar la distribución, de la cual se desea valorar su ajuste, indicando en el argumento "dist", la cadena de caracteres correspondiente a su nombre comúnmente usado. Para este análisis, solo podremos obtener el ajuste con respecto a la distribución normal y la logística, ya que el resto solo aceptan valores positivos, y si se produce una bajada en el precio los retornos semanales de Bitcoin pueden ser negativos.

Al llamar a la función, esta proporciona las cuatro gráficas clásicas de bondad de ajuste según Cullen y Frey (1999):

- Una gráfica de densidad, reflejada en la figura 3.1.1.2.1, que representa la función de densidad, que describe la probabilidad relativa de que una variable aleatoria tome un valor dado, de la distribución ajustada junto con el histograma de la distribución empírica.
- Una gráfica de las funciones de distribución acumulada, o "cumulative distribution function" (CDF) en inglés, tanto de la distribución empírica como de la distribución ajustada, descrita por la figura 3.1.1.2.2. En términos simples, la función de distribución acumulada es una función que describe la probabilidad de que una variable aleatoria X asuma un valor menor o igual a x. Es decir, describe la acumulación de las probabilidades a medida que se avanza a lo largo del eje x.
- Una gráfica Q-Q, ilustrada en la figura 3.1.1.2.3, que representa los cuantiles empíricos (eje y), que dividen a la distribución en partes iguales, contra los cuantiles teóricos (eje x).

- Una gráfica P-P que representa la función de distribución empírica evaluada en cada punto de datos (eje y) contra la función de distribución ajustada (eje x), y que queda expuesta por la figura 3.1.1.2.4.

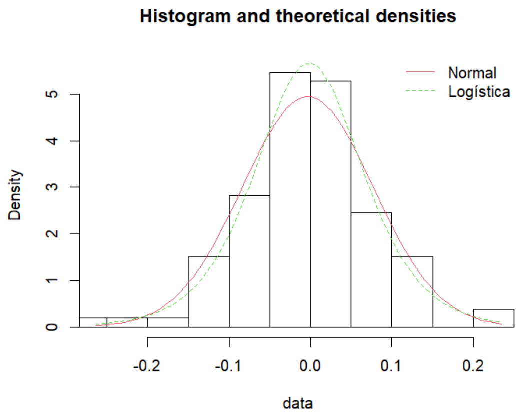

Figura 3.1.1.2.1

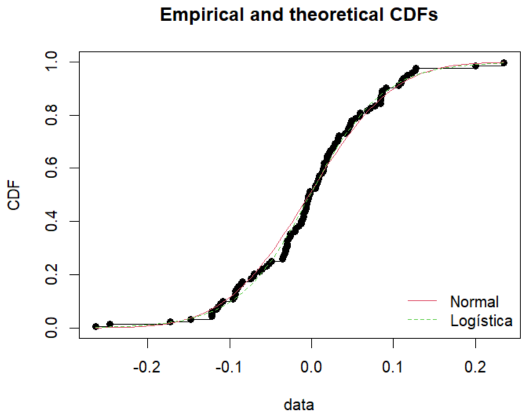

Figura 3.1.1.2.2

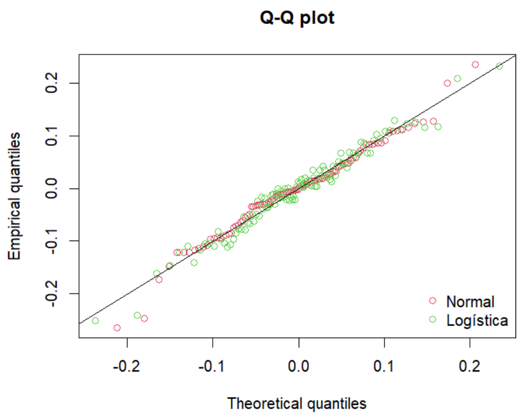

Figura 3.1.1.2.3

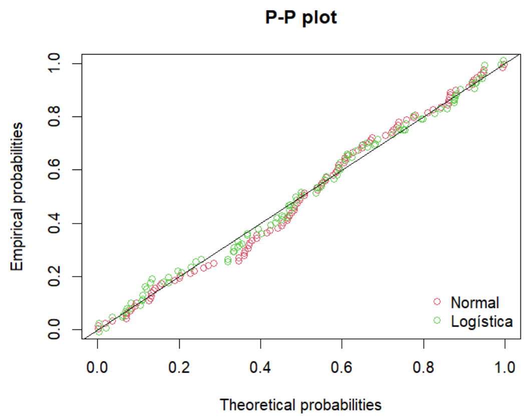

Figura 3.1.1.2.4

Entre todas las gráficas, las representadas en las figuras 3.1.1.2.1 y 3.1.1.2.2 son las que mejor representan el ajuste de la distribución logística con respecto en la normal. De las figuras 3.1.1.2.3 y 3.1.1.2.4 es muy difícil sacar alguna conclusión clara, debido a que los ajustes parecen ser muy parecidos en los términos que desean describir dichas gráficas.

Asimismo, la función *fitdist* devuelve varios resultados numéricos, como las estimaciones de los parámetros, los errores estándar estimados, la log-verosimilitud, los criterios de información de Akaike y Bayesianos, conocidos como AIC y BIC, y la matriz de correlación entre las estimaciones de los parámetros. Concretamente, en este trabajo, para analizar los ajustes de las distintas distribuciones se van a tomar como referencias la log-verosimilitud y los criterios de información AIC y BIC.

La log-verosimilitud es una medida de cuán probable es que los datos observados hayan sido generados por el modelo propuesto. Cuanto más grande sea el valor de la log-verosimilitud, mejor se ajusta el modelo a los datos.

El criterio de Información de Akaike (AIC) toma en cuenta tanto la bondad de ajuste del modelo como la complejidad de este. Modelos más complejos pueden ajustarse mejor a los datos, pero también pueden sobreajustarse, es decir, capturar ruido en lugar de señal. El AIC penaliza la complejidad del modelo para evitar el sobreajuste. Cuanto más pequeño sea el valor del AIC, mejor será el modelo.

Por último, el criterio de Información Bayesiano (BIC) es similar al AIC, pero penaliza la complejidad del modelo de forma más severa. Al igual que con el AIC, los modelos con menores valores de BIC son preferibles.

Según queda expuesto en los resultados 4.3.2.1 de la sección 4.3.2, los valores de la log-verosimilitud son más altos y los valores de AIC y BIC son más bajos para la distribución logística. Por tanto, parece que la distribución logística se ajusta mejor a los datos que la distribución normal, confirmando la hipótesis inicial planteada tras observar la figura 3.1.1.2.

3.1.2 AJUSTE DEL RETORNO DE ETHEREUM

Con el ánimo de comparar estadísticamente las probabilidades de los mismos sucesos en diferentes criptomonedas, en este apartado se busca ajustar la distribución de los retornos semanales de Ethereum durante los últimos dos años. De este modo, serán posibles ciertas comparaciones entre Bitcoin y Ethereum. La distribución empírica de los retornos semanales de Ethereum está representada en la figura 3.1.2.1. Igual que la de Bitcoin, esta distribución expone una media de retornos cercana al 0% y una volatilidad alta del activo, presentando retornos semanales de entre casi el -30% y el 20%. Cabe destacar que la forma de la distribución empírica de Ethereum varía con respecto a la de Bitcoin, con una cola izquierda más larga pero menos pesada o congestionada.

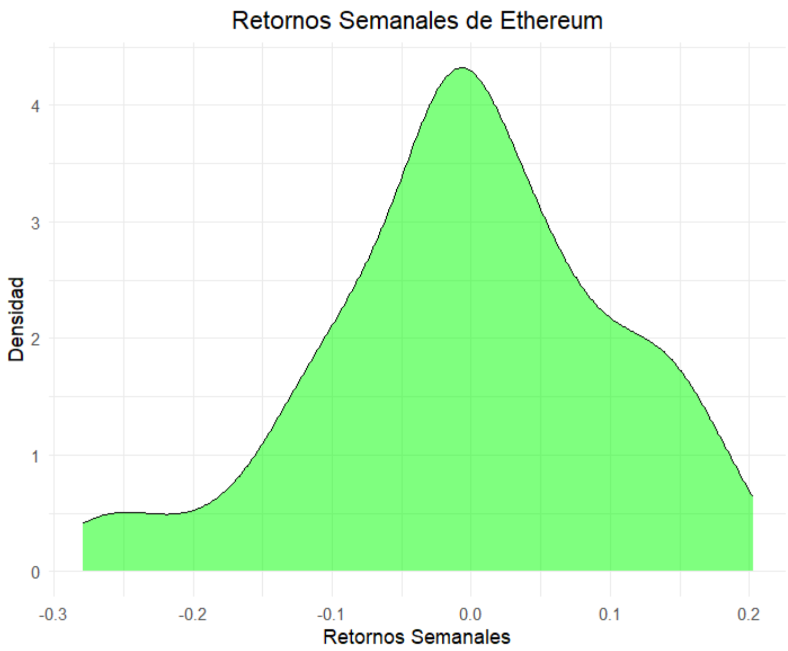

Figura 3.1.2.1

Todo el código necesario para el ajuste de los retornos semanales de Ethereum, comentado a continuación, queda expresado por el bloque de código 4.4.1 ubicado en la sección 4.4.

3.1.2.1 Elección de las distribuciones candidatas

El proceso para la elección de las distribuciones candidatas es equivalente al realizado en la sección 3.1.1.1. Por tanto, en este caso, no se realiza una explicación extensa del procedimiento y solo se comentan los resultados obtenidos.

En primer lugar, tras la ejecución de la función *descdist* se han obtenido el gráfico de asimetría-curtosis expresado por la figura 3.1.2.1.1 y los valores de los estadísticos especificados por los resultados 4.4.1 del apartado 4.4. Estos valores reflejan que la asimetría es negativa, lo que confirma que la cola de la izquierda es más larga que la de la derecha, y que la curtosis es muy cercana a 3, el valor que presentaría una distribución normal.

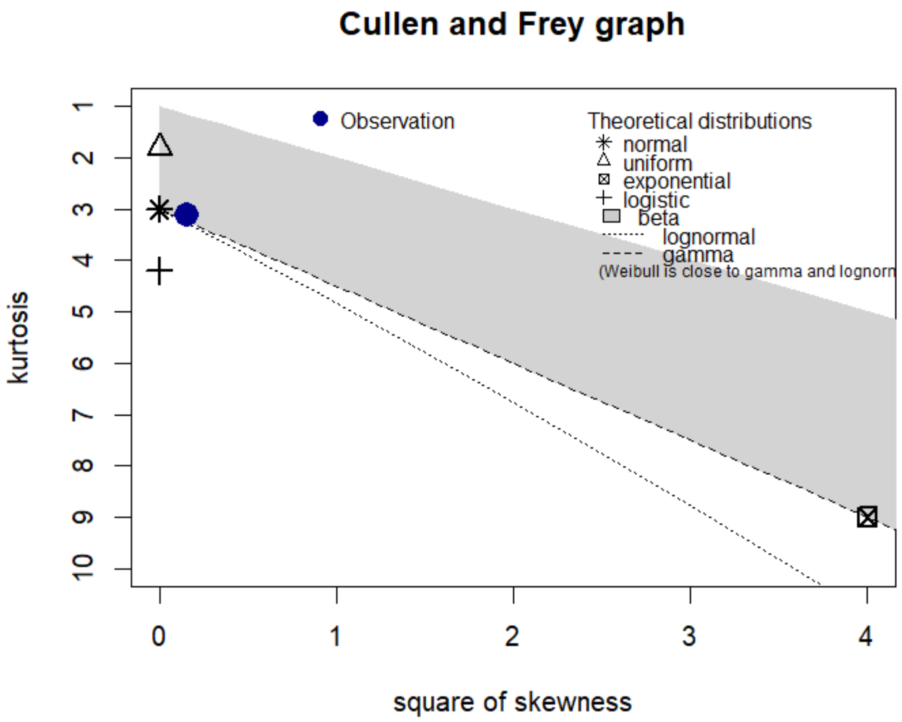

Figura 3.1.2.1.1

Esta figura parece indicar que las distribuciones candidatas que podrían proporcionar el mejor ajuste a la distribución empírica de los retornos semanales de Ethereum son la normal, la log-normal, la gamma y la beta.

3.1.2.2 Ajuste de distribuciones por estimación de máxima verosimilitud

Como ya se comentó durante el ajuste de la distribución empírica de Bitcoin, la existencia de retornos negativos impedía el análisis en profundidad de las distribuciones candidatas a través de la función *fitdist*, si estas eran distribuciones log-normal, gamma o beta. Por tanto, ha sido necesario optar por una conversión a positivo de los retornos que no alterase el análisis.

Para asegurar que todos los retornos sean positivos manteniendo su distribución relativa, se han desplazado todos los valores añadiendo la magnitud del retorno más pequeño (es decir, el retorno más negativo) más una pequeña constante. Este enfoque cambia la media y la varianza de los datos, por lo que se debe tener cuidado al interpretar los resultados del análisis. Queda constancia de esta modificación en el bloque de código 4.4.1. Los resultados obtenidos quedan reflejados en las siguientes figuras:

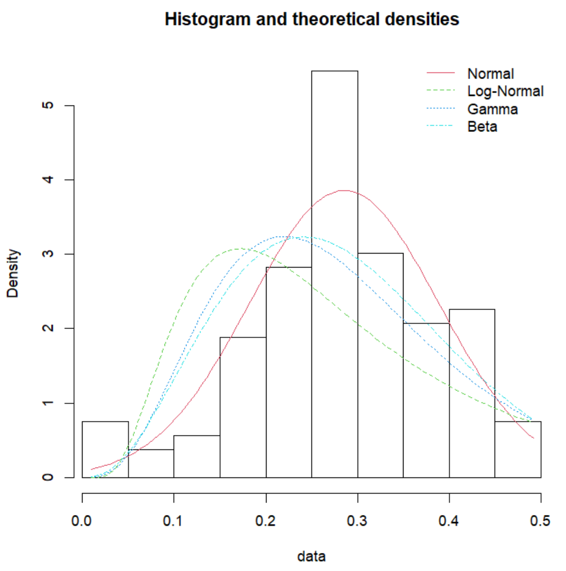

Figura 3.1.2.2.1

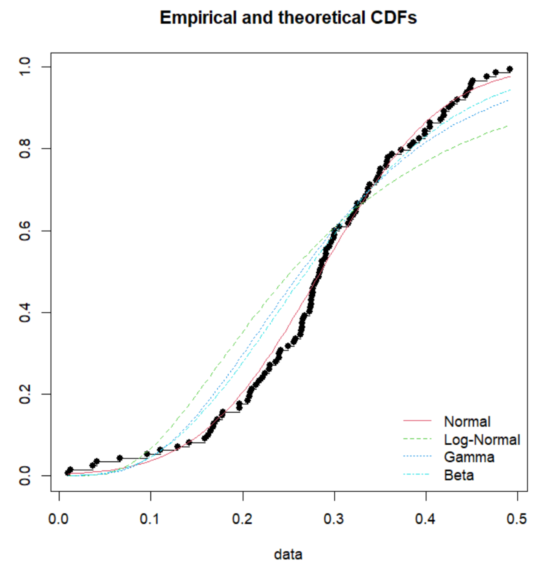

Figura 3.1.2.2.2

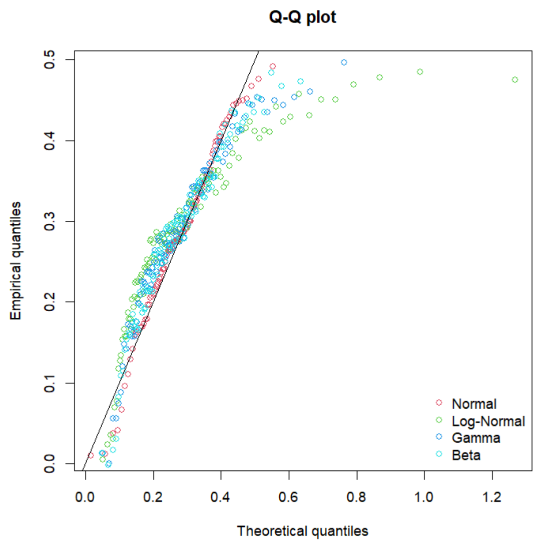

Figura 3.1.2.2.3

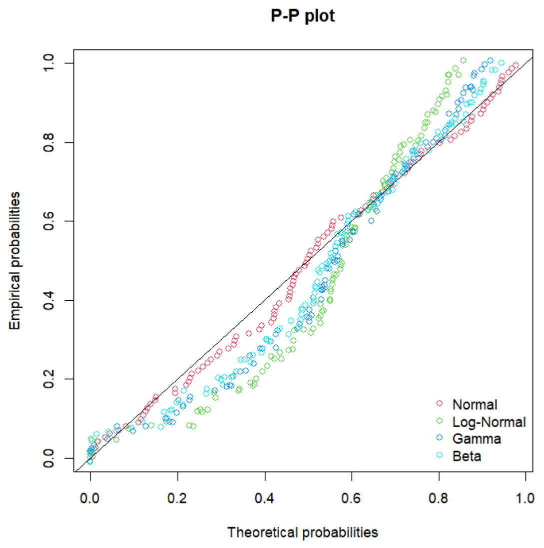

Figura 3.1.2.2.4

De forma visual, en todas las gráficas, exceptuando la de la figura 3.1.2.2.3, se puede detectar como la línea asociada a la distribución normal, de color rojo, es la que mejor se ajusta a los valores empíricos de los retornos semanales de Ethereum. También se puede comprobar en la figura 3.1.2.2.1 como la distribución se ha deslazado hacia la derecha para tomar valores positivos, pero sin perder la forma.

Además, los valores de la log-verosimilitud, el AIC y el BIC, reflejados en los resultados 4.4.2, confirman la afirmación anterior. Es decir, la distribución normal presenta la mayor log-verosimilitud, 90.22847, y los menores AIC y BIC, -176.4569 y -171.1301, respectivamente.

Por tanto, se concluye que la distribución que mejor se ajusta a la distribución empírica de los retornos semanales de Ethereum es la distribución normal. Para la siguiente sección, esta situación presenta una ventaja ya que la distribución normal si acepta valores negativos, por lo que no será necesario ningún reajuste para restablecer los valores iniciales de los retornos que han sido trasladados a valores positivos.

3.2 COMPARACIÓN DE LOS RETORNOS DE CRIPTOMONEDAS

Como se ha podido determinar anteriormente, las distribuciones que más se ajustan a los retornos semanales de Bitcoin y Ethereum son la distribución logística y la distribución normal, respectivamente.

Lo que hasta ahora queda claro es que los operadores que utilicen una distribución logística para fijar el precio de los derivados del Bitcoin, por ejemplo, probablemente tengan una ventaja sobre los que utilicen un modelo distinto. Lo mismo sucede con aquellos operadores que sigan una distribución normal para prever los retornos futuros de Ethereum (Parker, 2021).

Sin embargo, se puede ir un paso más allá e intentar comparar las probabilidades de que un determinado retorno ocurra tanto en Bitcoin como en Ethereum. Para hacer esto, es necesario utilizar las funciones de densidad de probabilidad de las distribuciones que hemos ajustado previamente para cada criptomoneda. Este proceso con R es muy sencillo de realizar, tal y como queda reflejado en el bloque de código 4.5.1 de la sección 4.5.

Se podría calcular, por ejemplo, ciertas magnitudes positivas de retorno, como un retorno mayor al 5%, al 10% o al 20%, y comparar estas probabilidades. La probabilidad de obtener un retorno mayor a un cierto valor se puede calcular utilizando la función de distribución acumulativa (CDF) de la distribución ajustada a tus datos. En R, las funciones *pnorm* y *plogis* permiten obtener la CDF de las distribuciones normal y logística, respectivamente. Estas funciones devuelven la posibilidad de que una variable aleatoria con la distribución especificada tenga un valor menor o igual al valor dado. Por lo tanto, para calcular la probabilidad de obtener un retorno mayor a un cierto valor, se debe restar el resultado de estas funciones de 1. Los resultados han sido los siguientes:

```
Probabilidad de obtener un retorno mayor al 5%, 10%, 15% y 20% para Bitcoin:
[0.2941, 0.1498, 0.0693, 0.0305] (respectivamente)
Probabilidad de obtener un retorno mayor al 5%, 10%, 15% y 20% para Ethereum:
[0.3011, 0.1573, 0.0681, 0.0242] (respectivamente)
```

Resultados 3.2.1

Según los cálculos realizados utilizando las distribuciones ajustadas a los retornos de Bitcoin y Ethereum, parece que ambos tienen probabilidades similares de obtener un retorno mayor al 5%, 10% y 15%, pero la probabilidad de obtener un retorno mayor al 20% es ligeramente mayor para Bitcoin.

Para Bitcoin, las posibilidades de obtener un retorno mayor al 5%, 10%, 15% y 20% son aproximadamente 29.41%, 14.98%, 6.93% y 3.05%, respectivamente. Mientras que, para Ethereum, estas cifras son aproximadamente 30.11%, 15.73%, 6.81% y 2.42%, respectivamente.

Esto sugiere que, aunque ambos tienen un riesgo similar (es decir, la posibilidad de obtener un alto retorno), Bitcoin podría tener un poco más de potencial de rendimiento, ya que tiene una mayor posibilidad de obtener un retorno mayor al 20%.

Por otro lado, es interesante realizar la misma operación para calcular las probabilidades de que se den retornos semanales menores de -5%, de -10%, de -15%, de -20% y de -25%, para ambas criptomonedas. A continuación, se muestran los resultados:

```
Probabilidad de obtener un retorno menor a -5%, -10%, -15%, -20% y -25% para Bitcoin:
[0.3003, 0.1536, 0.0713, 0.0314, 0.0135] (respectivamente)
Probabilidad de obtener un retorno menor a -5%, -10%, -15%, -20% y -25% para Ethereum:
[0.3272, 0.1757, 0.0783, 0.0287, 0.0085] (respectivamente)
```

Resultados 3.2.2

Los resultados obtenidos indican las probabilidades estimadas de que los retornos semanales de Bitcoin y Ethereum caigan por debajo de ciertos valores negativos, es decir, las pérdidas.

Para Bitcoin, hay alrededor de un 30% de posibilidad de que los retornos semanales sean menores a -5%, 15.4% para -10%, 7.1% para -15%, 3.1% para -20% y 1.4% para -25%. Para Ethereum, las cifras son aproximadamente 32.7% para -5%, 17.6% para -10%, 7.8% para -15%, 2.9% para -20% y 0.85% para -25%.

4. Programación con R

4.1 ANÁLISIS DE DATOS MEDIANTE DISTRIBUCIONES

4.1.1 Distribución Normal VS Distribución Uniforme

Para construir gráficos de densidad con la librería *ggplot2* (Wickham, 2009) resulta mucho más fácil trabajar con números aleatorios que intentar trazar la fórmula real (Parker, 2021). Así, el bloque de código 4.1.1 genera 100.000 números aleatorios basados en los parámetros de cada distribución.

```
install.packages("ggplot2")

library(ggplot2)

citation("ggplot2")


set.seed(123)

normal_data <- rnorm(100000, mean=0, sd=1)
uniform_data <- runif(100000, min=-sqrt(3), max=sqrt(3))

normal_df <- data.frame(Value = normal_data, Distribution = "Normal")
uniform_df <- data.frame(Value = uniform_data, Distribution = "Uniforme")

combined_df <- rbind(normal_df, uniform_df)

ggplot(combined_df, aes(x=Value, fill=Distribution)) +
  geom_density(alpha=0.5) +
  theme_minimal() +
  labs(title="Distribución Normal vs Distribución Uniforme", x="Valor", y="Densidad", fill="Distribución") +
  scale_fill_manual(values = c("Normal" = "blue", "Uniforme" = "red")) +
  theme(legend.position = c(.85, .85),
        plot.title = element_text(hjust = 0.5))
```

Bloque de código 4.1.1.1

En este bloque de código, primero se generan cien mil puntos de datos a partir de una distribución normal con una media de 0 y una desviación típica de 1, esto es relativamente sencillo.

A continuación, se generan otros cien mil puntos de datos de una distribución uniforme con media 0 y desviación estándar 1. Para una distribución uniforme, se calcula media = (min + max) / 2, y desviaciónTípica = (max - min) / sqrt(12). Resolviendo este sistema de ecuaciones para min y max con media = 0 y desviación estándar = 1, se obtienen: min = - sqrt(3) y max = sqrt(3).

Luego, se crean dos estructuras de datos en R conocidas como *data frame*, que se asemejan a las tablas de una base de datos, una para cada conjunto de datos. Posteriormente, ambas estructuras se combinan en uno solo por filas, con el objetivo de poder representar ambas distribuciones en un mismo gráfico.

Finalmente, se utiliza el paquete ggplot2 para crear los gráficos de densidad de los datos de las dos distribuciones. En el grafico resultante, incluido en la figura 3.1.1, el eje x representa los valores de los datos y el eje y representa la densidad de los datos. El gráfico resultante muestra las dos distribuciones, normal y uniforme, superpuestas para una comparación directa.

4.1.2 Cálculo de Probabilidad de Ejemplo

En una distribución normal, la probabilidad de que una observación sea mayor (o menor) que un cierto valor se puede calcular utilizando la función de distribución acumulativa, o CDF. En R, la función *pnorm* proporciona la probabilidad de que la variable aleatoria tome un valor igual o menor que cierto valor especificado para una distribución normal.

```
media <- 0.01
desviacion <- 0.02
retorno_objetivo <- 0.05

probabilidad_menor_igual <- pnorm(retorno_objetivo, mean = media, sd = desviacion)

probabilidad_mayor <- 1 - probabilidad_menor_igual

probabilidad_mayor
```

Bloque de código 4.1.2.1

En el bloque de código 4.1.2.1, dicho valor especificado se guarda en la variable "retorno_objetivo". Tras la ejecución del código, la "probabilidad_menor_igual" calculada es 2.28%, y por consiguiente la "probabilidad_mayor" es 97.72%.

4.2 ANÁLISIS DEL HORIZONTE TEMPORAL

Para obtener los datos de los retornos de Bitcoin en distintos periodos de tiempo (un día, una semana y un mes), se recomienda usar la librería *quantmod* (Ryan & Ulrich, 2023) de R para descargar los datos de precios de Bitcoin, y luego calcular los retornos para cada periodo de tiempo.

```
install.packages("quantmod")

library(quantmod)
library(ggplot2)

getSymbols("BTC-USD", src = "yahoo", from = Sys.Date()-365*2, to = Sys.Date())
btc <- Cl(`BTC-USD`)
nrow(btc)

daily_returns <- dailyReturn(btc, type="log")
weekly_returns <- weeklyReturn(btc, type="log")
monthly_returns <- monthlyReturn(btc, type="log")

daily_df <- data.frame(Returns = coredata(daily_returns))
names(daily_df)[1] <- "Returns"
daily_df$Period <- "Diario"

weekly_df <- data.frame(Returns = coredata(weekly_returns))
names(weekly_df)[1] <- "Returns"
weekly_df$Period <- "Semanal"

monthly_df <- data.frame(Returns = coredata(monthly_returns))
names(monthly_df)[1] <- "Returns"
monthly_df$Period <- "Mensual"

combined_df <- rbind(daily_df, weekly_df, monthly_df)
combined_df$Period <- factor(combined_df$Period, levels = c("Diario", "Semanal", "Mensual"))

ggplot(combined_df, aes(x=Returns, fill=Period)) +
  geom_density(alpha=0.5) +
  theme_minimal() +
  labs(title="Retornos de Bitcoin según el Horizonte Temporal Analizado", x="Retornos", y="Densidad", fill="Periodo") +
  theme(legend.position = c(.85, .85),
        plot.title = element_text(hjust = 0.5))
```

Bloque de código 4.2.1

El bloque de código 4.2.1 muestra un ejemplo de cómo hacer esto. Primero, se obtienen los datos diarios de los precios de cierre de Bitcoin de *Yahoo Finance*, una fuente de información financiera ampliamente conocida.

Después, se calculan los retornos logarítmicos diarios, semanales y mensuales. El valor del campo "type", tal y como indica la documentación de la función *periodReturn* que es padre de la función *weeklyReturn* del paquete *quantmod* (Ryan & Ulrich, 2023), debe ser establecido a "arithmetic" si los retornos son discretos o a "log" si los retornos son continuos. En este caso, los retornos semanales de Bitcoin se tratan como una variable continua.

Tras el cálculo de los retornos logarítmicos, estos se almacenan en estructuras de datos, que posteriormente se combinan en una sola. Finalmente, este código generará tres gráficas de densidad, una para los retornos diarios de Bitcoin, otra para los retornos semanales y otra para los retornos mensuales. Estas gráficas muestran cómo cambia la distribución de los retornos de Bitcoin en diferentes periodos de tiempo.

4.3 AJUSTE DE LOS RETORNOS DE BITCOIN

4.3.1 Aproximación mediante la función *descdistr*

La función *descdist* proporciona los estadísticos descriptivos clásicos (mínimo, máximo, mediana, media, desviación típica), asimetría y curtosis. Por defecto, se proporcionan estimaciones insesgadas de los tres últimos estadísticos. No obstante, el argumento "method" puede cambiarse de "unbiased" (por defecto) a "sample" para obtenerlos sin corrección por sesgo (Delignette-Muller & Dutang, 2015). Esta función se ha aplicado a los retornos semanales de Bitcoin, tal y como muestra el bloque de código 4.3.1.1.

```r
library(quantmod)
library(fitdistrplus)
library(logspline)

getSymbols("BTC-USD", src = "yahoo", from = Sys.Date()-365*2, to = Sys.Date())
btc <- Cl(`BTC-USD`)

weekly_returns <- weeklyReturn(btc, type="log")
weekly_returns_vector <- as.numeric(coredata(btc_weekly_returns))

descdist(weekly_returns_vector, discrete = FALSE, method = "sample")
```

Bloque de código 4.3.1.1

Los valores arrojados tras la ejecución de esta implementación son los siguientes:

```
Mínimo:  -0.2320165    Máximo:  0.2650578    Mediana:  -0.002672157
Media:  0.000583863    Desviación típica estimada:  0.08103466
Asimetría estimada:  0.1004982    Curtosis estimada:  4.126363
```

Resultados 4.3.1.1

Estos valores describen la forma de la distribución de los retornos semanales de Bitcoin, ilustrada en la figura 3.1.1.1. De esta forma, queda comprobada la simetría que presenta y la notable concentración de los datos en torno a la media característica de las distribuciones leptocúrticas.

4.3.2 Aproximación mediante la función *fitdist*

La función *fitdist* devuelve un objeto que, si se pasa como argumento a la función *summary,* esta devuelve los resultados numéricos del ajuste de distribuciones por estimación de máxima verosimilitud. Concretamente, a partir de la ejecución del bloque de código 4.3.2.1, el cual incorpora nuevas líneas para implementar el código completo que realiza el ajuste de los retornos semanales de Bitcoin en R, se obtienen los resultados 4.3.2.1.

```
library(quantmod)
library(fitdistrplus)
library(logspline)
library(extraDistr)

getSymbols("BTC-USD", src = "yahoo", from = Sys.Date()-365*2, to = Sys.Date())
btc <- Cl(`BTC-USD`)

weekly_returns <- weeklyReturn(btc, type="log")
weekly_returns_vector <- as.numeric(coredata(weekly_returns))

df <- data.frame(Returns = weekly_returns_vector)
ggplot(df, aes(x=Returns)) +
  geom_density(alpha=0.5, fill="green") +
  theme_minimal() +
  labs(title="Retornos Semanales de Bitcoin", x="Retornos Semanales", y="Densidad") +
  theme(plot.title = element_text(hjust = 0.5))

descdist(weekly_returns_vector, discrete = FALSE, method = "sample")

fn <- fitdist(weekly_returns_vector, "norm")
fl <- fitdist(weekly_returns_vector, "logis")

summary(fn)
summary(fl)

plot.legend <- c("Normal", "Logística")
denscomp(list(fn, fl), legendtext = plot.legend)
cdfcomp(list(fn, fl), legendtext = plot.legend)
qqcomp(list(fn, fl), legendtext = plot.legend)
ppcomp(list(fn, fl), legendtext = plot.legend)
```

Bloque de código 4.3.2.1

```
> summary(fn)
Log-verosimilitud:  90.24378    AIC:  -176.4876    BIC:  -171.1607
> summary(fl)
Log-verosimilitud:  90.03072    AIC:  -176.0614    BIC:  -171.7346
```

Resultados 4.3.2.1

Las 4 últimas líneas del código completo, mostrado anteriormente, sirven para la obtención de las figuras presentadas en la sección 3.1.1.2.

4.4 AJUSTE DE LOS RETORNOS DE ETHEREUM

```
library(quantmod)
library(fitdistrplus)
library(logspline)
library(extraDistr)

getSymbols("ETH-USD", src = "yahoo", from = Sys.Date()-365*2, to = Sys.Date())
eth <- Cl(`ETH-USD`)

weekly_returns <- weeklyReturn(eth, type="log")
weekly_returns_vector <- as.numeric(coredata(weekly_returns))

df <- data.frame(Returns = weekly_returns_vector)
ggplot(df, aes(x=Returns)) +
  geom_density(alpha=0.5, fill='green') +
  theme_minimal() +
  labs(title="Retornos Semanales de Ethereum", x="Retornos Semanales", y="Densidad") +
  theme(plot.title = element_text(hjust = 0.5))

descdist(weekly_returns_vector, discrete = FALSE, method = "sample")

min_return <- min(weekly_returns_vector)

if (min_return < 0) {
  weekly_returns_vector <- weekly_returns_vector + abs(min_return) + 0.01
}

fn <- fitdist(weekly_returns_vector, "norm")
fln <- fitdist(weekly_returns_vector, "lnorm")
fg <- fitdist(weekly_returns_vector, "gamma")
fb <- fitdist(weekly_returns_vector, "beta")

summary(fn)
summary(fln)
summary(fg)
summary(fb)

plot.legend <- c("Normal", "Log-Normal", "Gamma", "Beta")
denscomp(list(fn, fln, fg, fb), legendtext = plot.legend)
cdfcomp(list(fn, fln, fg, fb), legendtext = plot.legend)
qqcomp(list(fn, fln, fg, fb), legendtext = plot.legend)
ppcomp(list(fn, fln, fg, fb), legendtext = plot.legend)
```

Bloque de código 4.4.1

```
Mínimo:  -0.2790264     Máximo:  0.2027018     Mediana:  -0.004223241
Media:  -0.003576923    Desviación típica estimada:  0.1032966
Asimetría estimada:  -0.3920269     Curtosis estimada:  3.106138
```

Resultados 4.4.1

```
> summary(fn)
Log-verosimilitud:  90.22847   AIC:  -176.4569   BIC:  -171.1301
> summary(fln)
Log-verosimilitud:  45.78174   AIC:  -87.56347   BIC:  -82.23659
> summary(fg)
Log-verosimilitud:  68.94914   AIC:  -133.8983   BIC:  -128.5714
> summary(fb)
Log-verosimilitud:  76.78142   AIC:  -149.5628   BIC:  -144.236
```

Resultados 4.4.2

4.5 CÁLCULO Y COMPARACIÓN DE PROBABILIDADES DE RETORNOS

En el bloque de código 4.5.1, "fl_btc$estimate['location']" y "fl_btc$estimate['scale']" son los parámetros estimados de la distribución logística ajustada a los retornos de Bitcoin, y "fn$estimate['mean']" y "fn$estimate['sd']" son los parámetros estimados de la distribución normal ajustada a los retornos de Ethereum.

Este código primero define los valores de retorno en los que se está interesado. Luego, utiliza las funciones *plogis* y *pnorm* para calcular la probabilidad acumulada hasta esos valores para las distribuciones ajustadas de los retornos de Bitcoin y Ethereum, respectivamente. Finalmente, el código imprime los resultados.

```r
library(quantmod)
library(fitdistrplus)
library(logspline)
library(extraDistr)

getSymbols("BTC-USD", src = "yahoo", from = Sys.Date()-365*2, to = Sys.Date())
btc <- Cl(`BTC-USD`)
getSymbols("ETH-USD", src = "yahoo", from = Sys.Date()-365*2, to = Sys.Date())
eth <- Cl(`ETH-USD`)

weekly_returns_btc <- weeklyReturn(btc, type="log")
weekly_returns_vector_btc <- as.numeric(coredata(weekly_returns))
weekly_returns_eth <- weeklyReturn(eth, type="log")
weekly_returns_vector_eth <- as.numeric(coredata(weekly_returns))

fl_btc <- fitdist(weekly_returns_vector_btc, "logis")
fn <- fitdist(weekly_returns_vector, "norm")

returns <- c(0.05, 0.10, 0.15, 0.20)

p_return_greater_btc <- 1 - plogis(returns, location = fl_btc$estimate['location'], scale = fl_btc$estimate['scale'])
p_return_greater_eth <- 1 - pnorm(returns, mean = fn$estimate['mean'], sd = fn$estimate['sd'])

print("Probabilidad de obtener un retorno mayor al 5%, 10%, 15% y 20% para Bitcoin, respectivamente: ")
print(p_return_greater_btc)
print("Probabilidad de obtener un retorno mayor al 5%, 10%, 15% y 20% para Ethereum:, respectivamente: ")
print(p_return_greater_eth)

returns <- c(-0.05, -0.10, -0.15, -0.20, -0.25)

p_return_less_btc <- plogis(returns, location = fl_btc$estimate['location'], scale = fl_btc$estimate['scale'])
p_return_less_eth <- pnorm(returns, mean = fn$estimate['mean'], sd = fn$estimate['sd'])

print("Probabilidad de obtener un retorno menor a -5%, -10%, -15%, -20% y -25% para Bitcoin, respectivamente: ")
print(p_return_less_btc)
print("Probabilidad de obtener un retorno menor a -5%, -10%, -15%, -20% y -25% para Ethereum, respectivamente: ")
print(p_return_less_eth)
```

Bloque de código 4.5.1

5. CONCLUSIONES

Se ha comprobado que R es un lenguaje de programación efectivo en cuanto a su misión, el análisis de datos. De forma sencilla, permite llevar a cabo todas las fases de este tipo de análisis: la adquisición de los datos, su preparación, su análisis, su comunicación a través de sus excelentes gráficos y su aplicación. Además, destaca por la gran variedad de posibilidades que ofrece en este ámbito.

Por otro lado, las probabilidades obtenidas en la sección 3.2 son consistentes con la volatilidad observada en el mercado de las criptomonedas. En términos generales, las pérdidas esperadas de Ethereum son ligeramente mayores que las de Bitcoin para todos los niveles de pérdida que se consideraron.

Como conclusión, ambas criptomonedas reflejan un comportamiento muy similar, lo que impide que este análisis sea realmente determinante para una posible decisión de inversión. Pero lo realmente importante es que el procedimiento de ajustar las distribuciones es efectivo y muy fácil de implementar con R. Por tanto, este análisis podría ser muy útil para comprobar rendimientos entre otros activos que no estén tan relacionados. Es importante destacar, para concluir, que todo lo comentado ha sido solo una estimación basada en los datos históricos y no garantiza que las futuras rentabilidades sigan estas distribuciones.

Además, no deben ser el único factor para tener en cuenta al decidir invertir en Bitcoin o Ethereum. Siempre es importante realizar una evaluación completa de los riesgos y beneficios potenciales antes de hacer una inversión. Sin embargo, este tipo de análisis puede proporcionar una visión útil para la toma de decisiones en inversiones y estrategias de gestión de riesgos.

6. BIBLIOGRAFÍA

Cullen, A.C., & Frey, H.C. (1999). *Probabilistic Techniques in Exposure Assessment. A Handbook for Dealing with Variability and Uncertainty in Models and Inputs*. Plenum Press, EE. UU.

Delignette-Muller, M. L., & Dutang, C. (2015). fitdistrplus: An R Package for Fitting Distributions. *Journal of Statistical Software*, *64*(4), 1–34. https://doi.org/10.18637/jss.v064.i04

Hernández Roldán, I. (2023). *GitHub - ivanhernandezroldan/business-management-and-administration-final-degree-project-2023: Repository for the source code of my Business Management and Administration Final Degree Project.* https://github.com/ivanhernandezroldan/business-management-and-administration-final-degree-project-2023

LISA Institute. (2023). *Qué son las criptomonedas: usos, ventajas y futuro*. https://www.lisainstitute.com/blogs/blog/criptomonedas-usos-ventajas-futuro

Parker, F. (2021). The Return Distribution of Bitcoin. *R-bloggers*. https://www.r-bloggers.com/2021/11/the-return-distribution-of-bitcoin/

Posit team (2022). RStudio: Integrated Development Environment for R. Posit Software, PBC, Boston, MA. http://www.posit.co/

R Core Team (2022). R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria. https://www.R-project.org/

Ryan, J.A., & Ulrich, J.M. (2023). quantmod: Quantitative Financial Modelling Framework [R package version 0.4.22]. https://CRAN.R-project.org/package=quantmod

Sanjuán, F. J. M. (2022). Curtosis. *Economipedia*. https://economipedia.com/definiciones/curtosis.html

Santander. (2022). ¿Qué son las criptomonedas y cómo funcionan? https://www.santander.com/es/stories/guia-para-saber-que-son-las-criptomonedas

Wickham, H. (2023). *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics [R package ggplot2 version 0.4.22]*.
https://cran.r-project.org/web/packages/ggplot2