Análisis de anomalías en los días de la semana, en precios del Bitcoin, con R Studio® y RapidMiner Studio®

Autores

Daniel Guillermo Cavaller Riva, Antonio Sottile Bordallo, Emiliano Andrés Dueñas, Cristian Darío Ortega Yubro

{daniel.cavaller; antonio.sottile; emiliano.dueñas; cristian.ortega}@fce.uncu.edu.ar
http://fce.uncuyo.edu.ar

Colaboradores

Héctor Nicolás Sosa, Diego Vidal Silva, Roberto Fermín Moreno, Pablo Ariel Masier, Matías Rodolfo Codorniú

{hector.sosa; diego.silva; roberto.moreno; pablo.masier; matias.codorniu}@fce.uncu.edu.ar
http://fce.uncuyo.edu.ar

Resúmen: el estudio de las anomalías en los días de la semana del mercado de las criptodivisas completa una vasta literatura que analiza anomalías en los calendarios. Los modelos generados para el estudio de la anomalía contempla el análisis con datos que no responden a una distribución normal. Se utiliza una transformación adecuada para ajustar un modelo normal a los datos y lograr los tres supuestos: normalidad, homocedasticidad e independencia, analizando si aún así se puede evidenciar el efecto del día de la semana, y se realiza un test ajustando el modelo para el no cumplimiento del supuesto normalidad, y se concluye que persisten en los modelos, las anomalías en los días de la semana.

Palabras claves: modelo, datos, conjunto de datos, test, precio, criptodivisas, bitcoin, API, test de Shapiro - Wilks, homocedasticidad, independencia, rendimiento, valor, factor, días de la semana, test de Durbin - Watson, transformación de Johnson, R Studio®, RapidMiner Studio®.

adfa, p. 1, 2011.

1 Introducción

Existe una vasta literatura que analiza anomalías de datos en los calendarios, algunos ejemplos son: el efecto del día de la semana, efecto del mes del año, el efecto enero, el efecto fiestas, etc., (Caporale & Plastun, 2018), sin embargo no se podía evidenciar ese efecto en la hipótesis de eficiencia del mercado (Foye, 2018), (de Groot & Huij, 2018).

El estudio de las anomalías en los días de la semana del mercado de las criptodivisas ofrece oportunidades calificadas como "ganancias anormales" que se originan por la adopción de estrategias de negocios orientadas a esas anomalías.

Si bien los rendimientos de los días lunes para el Bitcoin son significativamente más altos que los de otros días de la semana, la mayoría de los resultados no pueden considerarse pruebas concluyentes contra la eficiencia del mercado, (Caporale & Plastun, 2018).

El presente trabajo refuerza el estudio del efecto del día de la semana en el mercado de criptodivisas, en este caso Bitcoin (Caporale & Plastun, 2018), día que se definirá como "factor" y que se incluirá en el desarrollo del modelo lineal con un factor, teniendo en cuenta para ese análisis desarrollos como el Test de Tukey, ANOVA, Kruskall - Wallis, los cuales serán expuestos completamente en lenguaje R, y los datos necesarios del comercio de la criptodivisa Bitcoin se obtendrán a través de una API[1] de "CoinDesk". Como aporte adicional, se realiza una comprobación con la herramienta RapidMiner Studio® a través del algoritmo K-means, y la interpretación del resumen de los resultados obtenidos.

Los modelos generados para el estudio de la anomalía contempla el análisis con datos uniformes, es decir, datos que no responden a una distribución normal. En el presente trabajo se busca utilizar una transformación adecuada para ajustar un modelo normal a los datos y lograr los tres supuestos, analizando si aún así se puede evidenciar el efecto del día de la semana, y realizar test con el no cumplimiento del supuesto normalidad.

Para ello, como primera medida, se analiza independencia del conjunto de datos de los precios del Bitcoin. Esa independencia se logra con el análisis de los "rendimientos" diarios, entendiendo como rendimiento, para el presente trabajo, la diferencia de precios de un día con el día siguiente, obteniendo en consecuencia, un conjunto de datos con valores positivos y negativos.

Luego, se busca la normalidad del conjunto de datos del rendimiento diario del Bitcoin, llevando a cabo una transformación Johnson, la cual admite para su análisis datos positivos como negativos. Téngase presente que si se quisiera aplicar la transformación

[1] API (Interfaz de Programación de Aplicaciones). Es un conjunto de funciones y procedimientos que cumplen una o muchas funciones para ser utilizadas por algún programa o aplicación, es decir, se reutiliza código de programación que ha sido testeado y publicado.

Johnson al conjunto de datos “Price”, el resultado no será satisfactorio porque ese conjunto de datos no tiene independencia. Por ello, primero se trabaja la independencia de los datos, y luego la transformación de datos uniformes a datos normales.

Luego se realizan las comparaciones múltiples y se desarrolla una alternativa ante el no cumplimiento del supuesto normalidad.

De acuerdo a la literatura citada, el efecto del día de la semana, relacionado con las diferencias estadísticamente significativas entre los rendimientos en diferentes días de la semana, fue una de las primeras anomalías del calendario que se examinaron.

Tanto las pruebas paramétricas como no paramétricas se llevan a cabo con las 364 observaciones diarias, comprendidas entre el día 01-01-2018 al 30-12-2018. Pero para que sean pruebas estadísticas válidas, tienen que cumplirse los supuestos de normalidad, homocedasticidad e independencia.

Si bien los test utilizados son de facil interpretación, la desventaja de ellos es que no consideran las distribuciones no normales de los datos. Por tal motivo, se lleva a cabo una transformación de la distribución de los datos para lograr normalidad, la que se produce con una distribución Johnson, y tambien se realiza un test de Pairwise para pruebas no paramétricas.

Con relación al aporte adicional, el trabajar con un algoritmo parametrizable dentro de una herramienta como RapidMiner Studio® como segunda instancia, es decir luego de los análisis de los datos a través de la estadística, permite una mayor confianza e interpretación del resúmen de los resultados obtenidos.

Se destaca y agradece los aportes de la Doctora Ángela Diblasi y la colaboración del Doctorando Santiago Emiliano Eguren.

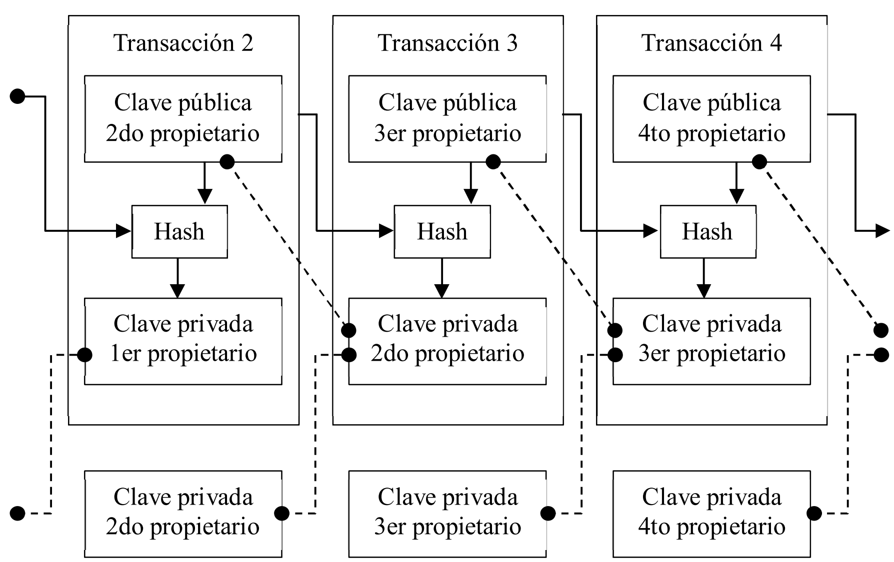
2 Criptodivisa y valuación

Se define criptodivisa como una cadena de firmas digitales (Nakamoto, 2008), lo que comunmente se conoce como cadenas de bloques - blockchain -. Un propietario de criptodivisa puede transferir o vender un Bitcoin, firmando digitalmente un “hash”. Se define como “hash” al algoritmo matemático que transforma arbitrariamente cualquier bloque de datos en una nueva serie de caracteres con una longitud fija, es decir siempre tendrá el mismo tamaño independientemente de la longitud de los datos de entrada.

Adicionalmente se necesita que el nuevo propietario de la criptodivisa tambien tenga la certeza de que los propietarios anteriores no realizan otras transacciones con esa unidad de criptodivisa que ha sido transferida. Por ello, la única forma de confirmar la ausencia de nuevas transacciones con divisas transferidas, es conocer todas las transacciones que se han realizado hasta el momento de la adquisición, lo que se logra con servidores denominados “de marca de tiempo”.

Los servidores “de marca de tiempo” funcionan tomando un hash de un bloque de transacciones, para marcar el momento en el que fue efectuada, y lo “publica”, es decir lo registra en el cadenas de bloques. Por lo tanto la marca de tiempo demuestra que las transacciones deben haber existido en ese momento para poder ingresar al hash del bloque, y cada marca del tiempo incluye una marca anterior en su hash, lo que se conoce como hash previo, formándose de esa forma, una cadena de registros de transacciones.

Esquema Nº 1 - Cadenas de Firmas Digitales.



Por lo tanto, las nuevas transacciones realizadas se transmiten a todos los servidores de la red de criptodivisas participantes quedando así registradas en todos ellos, y a su vez, cada servidor recibe nuevas transacciones, y las registra en un bloque de transacciones. Cuando el servidor valida la nueva transacción, transmite el bloque validado a todos los servidores de la red de criptodivisas, pero los servidores solamente aceptaran esa transacción, si todas las transacciones que contiene el bloque validado son válidas también, es decir, no solamente la última transacción, sino todas las transacciones contenidas en el cadenas de bloques.

Basicamente, cuando se hace la transacción el servidor que interviene en la misma automáticamente avisa a los servidores referentes - denominados mineros -, y ellos actualizan sus tablas - cadenas de bloques - teniendo todos asi las mismas transacciones.

Los servidores siempren consideran que la cadena de bloques con datos mas extensa es la cadena de bloques correcta. Si dos servidores transmiten versiones diferentes de cadenas al siguiente bloque simultáneamente, entonces los servidores trabajan sobre la primera cadena que reciben, pero guardan la segunda, y validaran solamente la mas extensa de las dos.

Por tal motivo, no todas las transacciones nuevas tienen que llegar a todos los servidores de la red de criptodivisas, porque con que esa nueva transacción llegue a varios servidores, indefectiblemente formará parte de una cadena de bloques, alargando esa cadena, y posteriormente será validada la nueva transacción.

Concluyendo, los datos de todas las transacciones se encuentran en los servidores de la red de criptodivisas cuando han sido validados por los propietarios. Ahora bien, con respecto a la valuación y la estimación de los valores futuros del Bitcoin (Blau, 2017) los inusuales niveles de valuación de la criptodivisa están relacionados con su comercio especulativo y es esa volatibilidad la que atenta a que el Bitcoin no se la considere aún una divisa propiamente dicha.

La dinámica en los precios del Bitcoin reveló la presencia de una burbuja especulativa y no se sabe si la misma ha explotado totalmente. Si bien hoy en día esa dinámica es menor por haberse producido una caida significativa del precio del Bitcoin, sigue persistiendo una volatibilidad muy marcada en los precios y una gran incertidumbre en su comportamiento futuro.

La cantidad actual de datos de las transacciones contenidos en la red de criptodivisas, registrados en las cadenas de bloques, y en este caso del Bitcoin, son suficientes como para analizar propiedades del mercado de criptodivisas, y lo más interesante es que los mismos están disponibles y pueden ser accedidos. Si bien el presente analisis a través de la estadística se concentra solamente a datos del precio del Bitcoin, no significa que las conclusiones a que se arriben se puedan extender al mercado de las criptodivisas, ya que existen multiples criptodivisas y cada una de ellas presenta un comportamiento particular de las características de esa criptodivisa.

Lo que la presente investigacion pretende, es reforzar la evidencia del efecto del día de la semana en este mercado, a través del análisis científico. Luego, validado un modelo estadístico de un conjunto de datos, a través de los tres supuestos básicos, normalidad, homocedasticidad e independencia, se estudia el mismo conjunto de datos a través de la aplicación del algoritmo k-means, de la aplicación RapidMiner Studio®.

3 Datos del precio del Bitcoin y metodología de análisis

Se analizan los datos del precio del Bitcoin tomando como muestra los datos de casi un año, es decir 364 observaciones, lo que se puede traducir a 52 días por cada uno de los siete días de la semana.

La fuente de los datos es tomada de la API de indexación del precio del Bitcoin llamada "Coindesk", desarrollada como paquete instalable para R Studio® de la plataforma Anaconda® Navigator. El paquete seleccionado se llama "coindeskr" y fue publicado el día 5 de Enero del 2018.

Esta API extrae en tiempo real el precio del Bitcoin del sitio https://www.coindesk.com. El paquete que se instala en R Studio® habilita cuatro comandos para ejecutar:

> get_currency_list
> get_current_price
> get_historic_price (este es el comando que se utilizará en el presente trabajo)
> get_last31days_price

Como metodología de análisis de los datos del precio del Bitcoin se utilizan algunas de las fases detalladas de CRISP - DM, particularmente la fase "comprensión de los datos" y la fase "preparación de los datos". Estas fases también se aplican en el uso de la aplicación RapidMiner Studio®.

La fase comprensión de los datos implica estudiar con mayor profundidad el conjunto de datos disponibles, y se la considera como una de las fases esenciales que permitirá no encontrarse con potenciales problemas en la fase preparación de los datos (Sottile Bordallo et al., 2019).

En la fase preparación de los datos se ajustará el primer modelo de análisis, para convertir los datos de los precios del Bitcoin, en rendimientos, lo que se traduce en la diferencia de precios diarios, en consecuencia para esta investigación, el rendimiento de un Bitcoin en un día determinado, es la diferencia del precio de un Bitcoin de ese día, con el día inmediato anterior. Este algoritmo se desarrollará más adelante.

El primer modelo que se desarrolla para análisis estadístico, es el conformado con dos variables, en función del intervalo de tiempo que va desde el día 01 de Enero del 2018 al día 30 de Diciembre del 2018, la variable dependiente "Price" (Precio) y la variable independiente "Day" (Dia).

4 Desarrollo en R Studio®

La aplicación que se utiliza es R Studio® versión 1.1.456, la cual esta integrada a Anaconda® Navigator versión 1.9.6.

4.1 Obtención y preparación de datos del precio del Bitcoin, API Coindeskr

Para extraer los datos del precio del Bitcoin en forma automatizada, se conecta la aplicación R Studio® a la API de Coindesk del sitio https://www.coindesk.com/API, instalando el paquete “coindeskr”, y activando la librería.

```
> install.packages(‘coindeskr’)      # paquetes de datos de la API Coindesk.
> library(coindeskr)                 # activación de la librería de datos.
```

Luego se procede a obtener un conjunto de datos con 364 observaciones de los valores del precio de un Bitcoin, tomando como intervalo, la fecha 01-01-2018 al 30-12-2018, mediante el comando get_historic_price, porque se quiere obtener como conjunto de datos, 52 lunes, 52 martes, 52 miércoles, 52 jueves, 52 viernes, 52 sábados y 52 domingos.

```
> ultimos364 <- get_historic_price (currency = "USD", start = '2018-01-01', end = '2018-12-30')
```

Este conjunto de datos obtenido tiene la particularidad de que los valores asignados para el nombre de las filas de la tabla resultante, son las fechas de cada una de las observaciones.

El desafio entonces es agregar una nueva columna de datos al conjunto de datos, que indique el dia de la semana de ese valor observado, lo que se utilizará como nivel para el desarrollo de un modelo con un factor, siendo cada nivel un día de la semana, resultando así los siete (7) niveles, lunes, martes, miercoles, jueves, viernes, sábado y domingo.

4.2 Preparación de los Datos

El paquete “lubridate” de R-Studio® habilita el trabajo de los datos del conjunto de datos con fechas, lo que permite agregar una columna con el día de la semana resultante de cada valor observado, dato que se debe tomar del nombre de las filas. Por lo tanto, primero se generan el conjunto de los valores correspondientes a la fecha, en virtud de los nombres de las filas del conjunto de datos que se ha obtenido de la API coindesk previamente.

Posteriormente, se incluye ese conjunto de valores "fecha", como una columna nueva dentro del conjunto de datos "últimos 364" y se le asigna a la nueva variable independiente el nombre "Day" que solo contendrá los nombres del día de la semana de los valores "fecha".

El conjunto de datos "ultimos364" generado contiene ahora dos variables, la variable dependiente "Price" y la variable independiente "Day", con 364 observaciones. Para poder visualizar el conjunto de datos "ultimos364" resultante se ejecuta el comando view().

```
> install.packages('lubridate')
> library(lubridate)
```

Creación de los datos fecha, con los nombres de las filas

```
> fecha = rownames(ultimos364)
```

Creación de la variable independiente Día "Day" que solo contiene los nombres de cada día de la semana, correspondiente a los datos fecha.

```
> ultimos364$Day = wday (fecha, label = TRUE)
```

4.3 Propuesta de un modelo lineal.

El efecto de analizar si existen diferencias significativas entre los valores del precio del Bitcoin, en virtud a un día de la semana, se ajusta a un modelo lineal de la forma:

$$Price_{ij} = \mu_i + \mathcal{E}_{ij} \qquad i = 1, \dots, 7; \quad j = 1, \dots, 52$$

Donde:

$Price_{ij}$ es la observación del precio del Bitcoin en el día de la semana i, j;
μ_i es el precio medio del Bitcoin en el día i
$\mathcal{E}_{ij}$ es la variable aleatoria que representa la oscilación - error - del $precio_{ij}$ del Bitcoin

Se supone que el modelo desarrollado presenta una distribución $\mathcal{E}_{ij} \overset{iid}{\sim} \mathrm{N}(0, \delta^2)$.
El modelo se ajusta a los datos mediante el comando lm de R.

Se activa la librería "lmtest" en R Studio® para el modelo lineal de un factor con múltiples niveles, por lo que se define la variable independiente que será el factor. Para verificar que las variables están bien configuradas desde el punto de vista en la puntuación de los datos, se solicita un resumen del conjunto de datos.

```
> library(lmtest)
```

```
> summary(ultimos364)
```

	Price	Day
Min. :	3214	Sun:52
1st Qu. :	6366	Mon:52
Median :	6902	Tue:52
Mean :	7540	Wed:52
3rd Qu. :	8614	Thu:52
Max. :	17136	Fri:52
		Sat:52

Se puede constatar que de las 364 observaciones, 52 observaciones corresponden a cada uno de los días de la semana.

```
> attach(ultimos364)
```

Se adjunta el conjunto de datos para llamar las variables directamente, la variable dependiente “Price” y la variable independiente “Day”, y poder acceder a los datos que ellas contienen. Se define a la variable independiente “Day” como factor. Cada nivel del factor es entonces, el día de la semana.

```
> Day = factor(Day)
> modelo = lm(Price~Day-1)
> summary(modelo)
```

Se utiliza el comando lm() de R Studio® para desarrollar el modelo. Se le resta -1 para que no aparezca el valor “intercept”, y si aparezcan cada uno de los días de la semana.

Call:
lm(formula = Price ~ Day - 1)

Residuals:

Min	1Q	Median	3Q	Max
-4350.3	-1157.8	-690.2	1023.3	9571.2

Coefficients:

	Estimate	Std. Error.	t value.	Pr(>|t|)
DaySun	7505.2	337.5	22.24	<2e-16 ***
DayMon	7629.0	337.5	22.61	<2e-16 ***
DayTue	7581.0	337.5	22.46	<2e-16 ***
DayWed	7515.6	337.5	22.27	<2e-16 ***
DayThu	7471.9	337.5	22.14	<2e-16 ***
DayFri	7511.2	337.5	22.26	<2e-16 ***
DaySat	7564.6	337.5	22.41	<2e-16 ***

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2434 on 357 degrees of freedom
Multiple R-squared:  0.9073,	Adjusted R-squared:  0.9055
F-statistic: 499.1 on 7 and 357 DF,  p-value: < 2.2e-16
```

Sin embargo, este modelo no es válido si no se comprueban los supuestos considerados básicos, es decir: el supuesto de normalidad, el de homocedasticidad e el de independencia.

A simple vista se observa que el coeficiente del valor “t” cambia en el día lunes, el día martes y el día sábado, para un conjunto de datos compuesto por 364 observaciones, cambio consecuente con el coeficiente estimado que posee mayores valores para el día lunes, el día martes y el día sábado, ya que el valor del error estándar es el mismo para todos los días de la semana, por ser el tamaño de las observaciones el equivalente a siete (7) días por cincuenta y dos (52) semanas, es decir 364 observaciones en el conjunto de datos.

4.4 Análisis de los supuestos del modelo propuesto

Se divide el análisis de los supuestos del modelo en: el supuesto de normalidad, el de homocedasticidad e el de independencia.

4.4.1 Normalidad

Para comprobar la normalidad de los errores se utiliza el Test de Shapiro - Wilks, test utilizado usualmente, aplicado a los residuos y a los valores estándar del modelo propuesto.

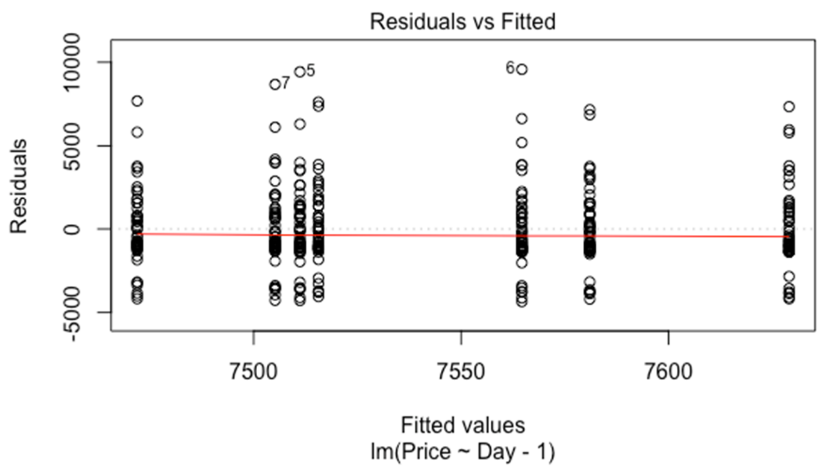
```
> shapiro.test(rstandard(modelo))

Shapiro-Wilk normality test
data:  rstandard(modelo)
W = 0.91391, p-value = 1.445e-13

> shapiro.test(residuals(modelo))

Shapiro-Wilk normality test
data:  residuals(modelo)
W = 0.91391, p-value = 1.445e-13
```

La hipotesis nula (H0) es que los datos de los residuos son normales, el rechazo de la hipótesis nula (H1) sugiere la presencia de una desviación de la distribución, respecto de la normal.

Siendo el “p” valor < 0,05, es decir = 1.445e-13, se rechaza la hipótesis nula (H0), con lo que se sugiere la presencia de la anomalía en los datos de los precios de Bitcoin en relación con la normalidad, con lo que se refuerza el concepto del “efecto del día de la semana”.

En el gráfico siguiente se observa que los residuos no parecieran tener una distribución simétrica:

Imágen Nº 1 - Distribución de los residuos.



4.4.2 Homocedasticidad

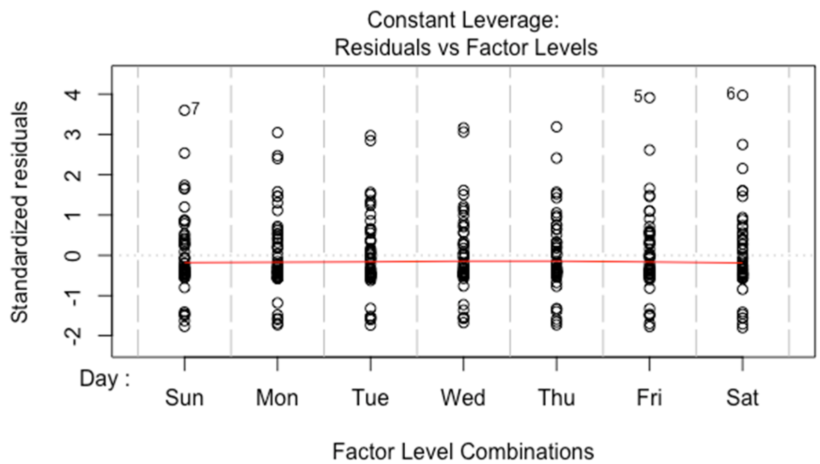
La igualdad de las varianzas de la variable dependiente precio - “Price” - por cada día de la semana se puede probar con el test de Breusch - Pagan.

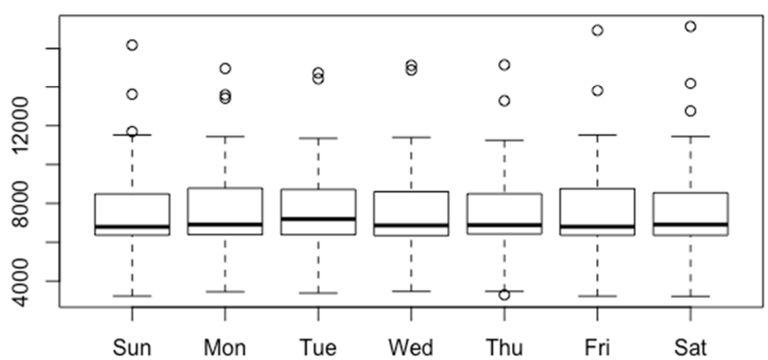
> bptest(modelo)

studentized Breusch-Pagan test
data: modelo
BP = 0.48106, df = 6, p-value = 0.9981

Este test arrojó un “p” valor de 0.9981. Dado este “p” valor no se rechaza la hipótesis nula (H0) de varianzas no significativamentes diferentes. El gráfico siguiente ilustra esta situación. En efecto, no parece haber diferencia significativa en los medios de la caja, para cada día de la semana.

Imágen Nº 2 - Residuos estándar en los niveles del factor.





4.4.3 Independencia

Otro de los supuestos planteados para los errores del modelo desarrollado, es el supuesto de independencia estotástica o probabilística. Debido a la forma de los datos del conjunto de datos, esta independencia parece difícil de asumir debido a que dos filas consecutivas de la tabla de los datos significan dos días consecutivos para el precio de un Bitcoin.

Para comprobar el supuesto de independencia se utiliza el Test de Durbin - Watson.

> dwtest(modelo)

Durbin-Watson test

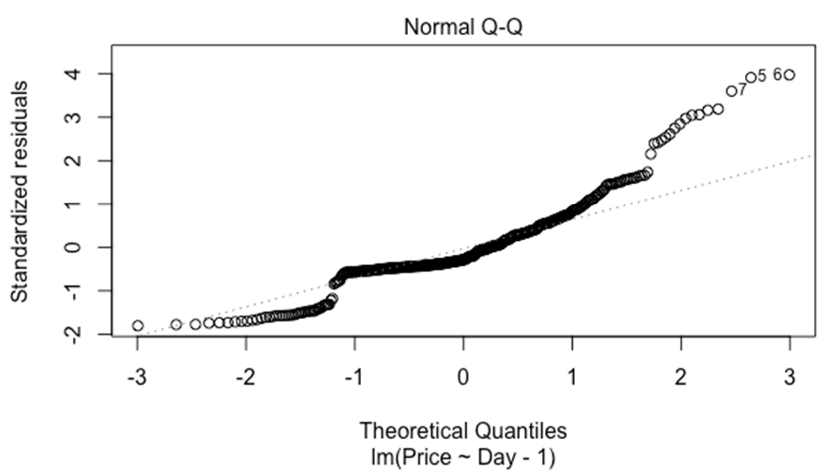
data: modelo
DW = 0.025888, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0

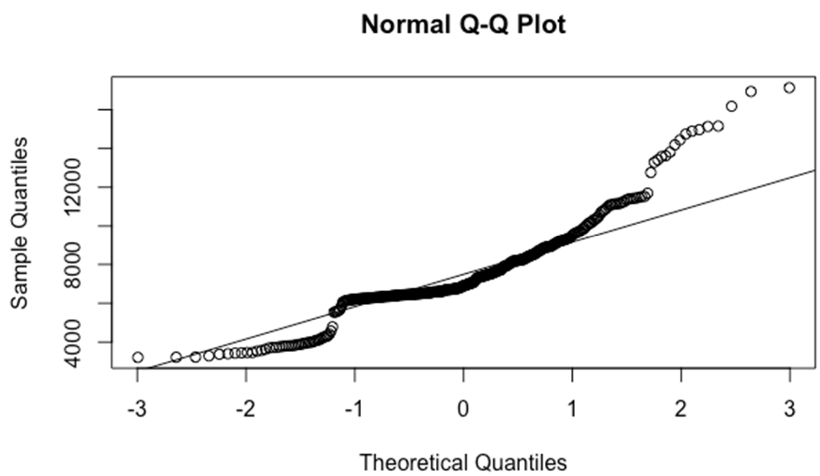
En efecto, el "p" valor asociado al test de Durbin - Watson resulta menor que 2.2e-16, se rechaza la hipótesis nula (H0), y se puede aseverar que no hay independencia entre los datos del precio de un Bitcoin, en el conjunto de datos bajo análisis.

Una de las posibles causas de la falta de independencia es la temporalidad del precio del Bitcoin, y que los mismos no son tomados aleatoriamente, sino que son los resultantes en un intervalo de tiempo que comprende desde el día 01-01-2018 al 30-12-2018, es decir, el valor diario de un Bitcoin en ese lapso de tiempo.

Por lo tanto, pareciera existir una dependencia en el precio de un día determinado de un Bitcoin, con respecto al precio del mismo Bitcoin del día anterior.

Imágen Nº 3 - Cuantíles del modelo lineal.




Normal Q-Q Plot
Sample Quantiles
4000
8000
12000
-3
-2
-1
0
1
2
3
Theoretical Quantiles

5 Cambiando la variable respuesta

Ahora bien, si una de las causas de la dependencia de los datos del precio de un Bitcoin en un día determinado con respecto al precio del mismo Bitcoin del día anterior podría ser explicada en virtud del supuesto de que el precio del dia "n" depende del precio del dia "n-1" más un rendimiento - performance -, el que puede ser positivo o negativo incrementado o disminuyendo su valor, es decir lo que responde a la variabilidad del precio, entonces se puede armar un conjunto de datos integrado solamente por esos rendimientos diarios, y de esa forma tratar de alcanzar el supuesto de la independencia de los datos del rendimiento diario de un Bitcoin.

De tal forma que para obtener el rendimiento de un Bitcoin correspondiente al día 01-01-2018, se le resta al precio de un Bitcoin de ese dia, el día "n", el precio del mismo Bitcoin del día "n-1" correspondiendo entonces el de la fecha 31-12-2017. Por lo tanto, el rendimiento del día "n" es:

$$R(d)_n = P(d)_n - P(d)_{n-1}$$

Donde:

$R(d)_n$ es el rendimiento de un Bitcoin de un día determinado.
$P(d)_n$ es el precio de un Bitcoin de un día determinado.
$P(d)_{n-1}$ es el precio del mismo Bitcoin del día inmediato anterior.

Con el algoritmo detallado anteriormente se ajusta el conjunto de datos reemplazando la columna Precios - Price - por la columna Rendimientos - Performance -.

Con ese ajuste detallado, se construye un nuevo modelo lineal con la forma del modelo anterior, ver punto 4.3 Propuesta de un modelo lineal., pero con una variable respuesta"el rendimiento" entre un día y el día inmediato anterior, en lugar del precio de un Bitcoin de un día determinado. Es decir:

$$R_{ij} = \mu_i + \mathcal{E}_{ij} \qquad i = 1, \dots, 7; \quad j = 1, \dots, 52$$

Donde:

R_{ij} es la observación del rendimiento de un Bitcoin en el día de la semana i, j respecto del dia anterior;
μ_i es el rendimiento de un Bitcoin del día i respecto del día anterior, es decir $i - 1$
$\mathcal{E}_{ij}$ es la variable aleatoria que representa la oscilación - error - del R_{ij} del Bitcoin

En este modelo también se supone una distribución del tipo $\mathcal{E}_{ij} \overset{iid}{\sim} \mathrm{N}(0, \delta^2)$, supuesto que se divide en: normalidad, homocedasticidad e independencia.

5.1 Ajuste del modelo lineal

El ajuste de este modelo a los datos se realiza con la funcion “lm” de R Studio®. Se puede observar que el “p” valor para el día lunes, es el valor más bajo de todos los días de la semana, siendo el mismo = 0.204.

Se adquieren 365 observaciones de los datos del precio del Bitcoin, para lograr los rendimientos diarios.

```
> datos <- get_historic_price(currency = "USD", start = '2017-12-31', end = '2018-12-30')
```

Obtenidos los datos, los mismos se exportan como archivo .csv, para poder trabajar el campo “Price” en una planilla de cálculos y restar los valores de cada día con el del día inmediato anterior.

```
> write.csv(datos, file = "datos.csv")
```

Se abre el archivo con una planilla de cálculos, y se procede a restar los valores de los precios del Bitcoin, para lograr los rendimientos diarios. Luego se deja solamente la columna “Performance” (Rendimientos).

Se graban los cambios en un archivo de texto denominado “datos364.txt”, y se importan los datos a R Studio®. En este nuevo conjunto de datos, se trabajará sin decimales, para simplificar el análisis.

```
> datos364 <- read.table("datos364.txt", header = TRUE)
> modelo = lm(Performance~Day-1)
```

Call:
lm(formula = Performance ~ Day - 1)

Residuals:

Min	1Q	Median	3Q	Max
-2189.98	-141.62	26.58	147.39	1742.73

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
DaySun	-59.48	53.51	-1.112	0.267
DayMon.	-68.13	53.51	-1.273	0.204
DayTue	-48.02	53.51	-0.897	0.370
DayWed	-65.38	53.51	-1.222	0.223
DayThu	-43.69	53.51	-0.817	0.415
DayFri	39.27	53.51	0.734	0.463
DaySat	53.52	53.51	1.000	0.318

Residual standard error: 385.9 on 357 degrees of freedom
Multiple R-squared: 0.0202, Adjusted R-squared: 0.0009918
F-statistic: 1.052 on 7 and 357 DF, p-value: 0.3946

5.2 Análisis de los supuestos del modelo con variable respuesta el rendimiento

Análisis de los supuestos de normalidad, de homocedasticidad e de independencia

5.2.1 Normalidad

Mediante el test de Shapiro - Wilks aplicado a los residuos del modelo con variable respuesta el rendimiento.

```
> shapiro.test(rstandard(modelo))

Shapiro-Wilk normality test
data:  rstandard(modelo)
W = 0.90537, p-value = 2.671e-14

> shapiro.test(residuals(modelo))

Shapiro-Wilk normality test
data:  residuals(modelo)
W = 0.90537, p-value = 2.671e-14
```

Se rechaza la H0, los datos de los rendimientos diarios del Bitcoin en el conjunto de datos bajo análisis sigue sin tener un comportamiento normal, arrojando un "p" valor = 2.671e-14, es decir, muy inferior a 0.05

5.2.2 Homocedasticidad

La igualdad de las varianzas en homocedasticidad de los rendimientos de un Bitcoin por día de la semana se analiza con el test de Breusch - Pagan.

El "p" valor asociado es = 0.5518, con lo cual se garantiza que los datos del conjunto de datos bajo análisis son homocedasticos.

```
> bptest(modelo)
studentized Breusch-Pagan test
data:  modelo
BP = 4.9383, df = 6, p-value = 0.5518
```

5.2.3 Independencia

Mediante el test de Durbin - Watson, obtenemos un “p” valor = 0.7035 la cual nos permite asumir la independencia entre los datos de los rendimientos medios del Bitcoin.

```
> dwtest(modelo)
Durbin-Watson test
data:  modelo
DW = 2.0509, p-value = 0.7035
alternative hypothesis: true autocorrelation is greater than 0
```

Si bien se logra la independencia en los datos del conjunto de datos bajo análisis, todavía no se obtiene el supuesto de normalidad.

6 En búsqueda de un modelo que cumpla los supuestos

Se ha analizado un modelo lineal con la variable dependiente "Rendimiento" de un Bitcoin para los distintos días de la semana. Esta variable respuesta, sin embargo, no resulta normal.

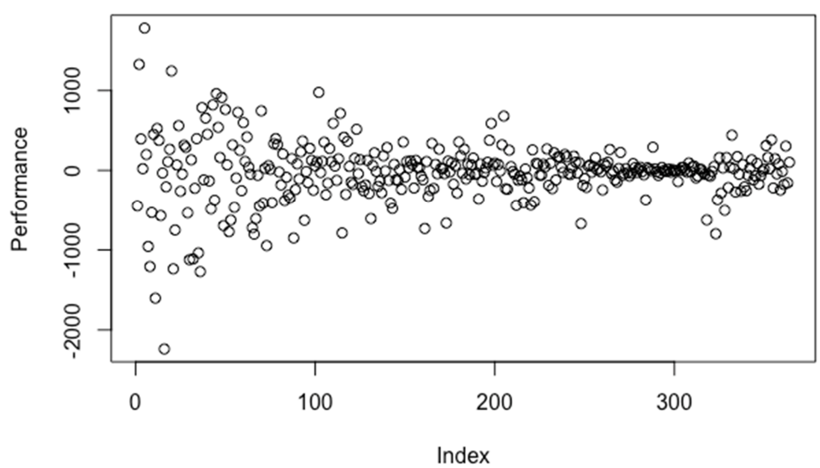
En este punto, y para continuar el análisis estadístico, se tienen dos posibles caminos:

1. Realizar una transformación adecuada para alcanzar el supuesto de normalidad, o
2. Recurrir al análisis de un modelo que no exija tal normalidad.

Se desarrolla a continuación la primera opción. En el gráfico siguiente se observa que los valores del Rendimiento - Performance - diario de un Bitcoin presentan una menor dispersión de los datos.

Dispersión del conjunto de datos de la variable "Performance".

Imágen Nº 4 - Dispersión de la variable dependiente Performance.



> plot(Performance)

Donde dice "Performance" debe entenderse Rendimientos.
Donde dice "Index" debe entenderse Indices

7 Transformación de la variable respuesta rendimiento

Para transformar los datos de la variable Rendimiento diario de un Bitcoin, siendo estos datos del conjunto de datos bajo análisis no normales, en datos normales, se emplea una distribución Johnson. Esta distribución tiene la forma:

$$Z = \gamma + \eta \sinh^{-1}\left(\frac{X - \varepsilon}{\lambda}\right)$$

donde X es la variable a transformar y Z la variable transformada. Los valores $\gamma, \eta, \epsilon, \lambda$ son parámtros a estimar mediante un algorítmo matemático complejo.

En primer lugar, se activa la librería “Johnson” en R Studio®, siempre y cuando ese paquete se encuentre instalado. De no ser así, previamente se deberá instalar el paquete correspondiente.

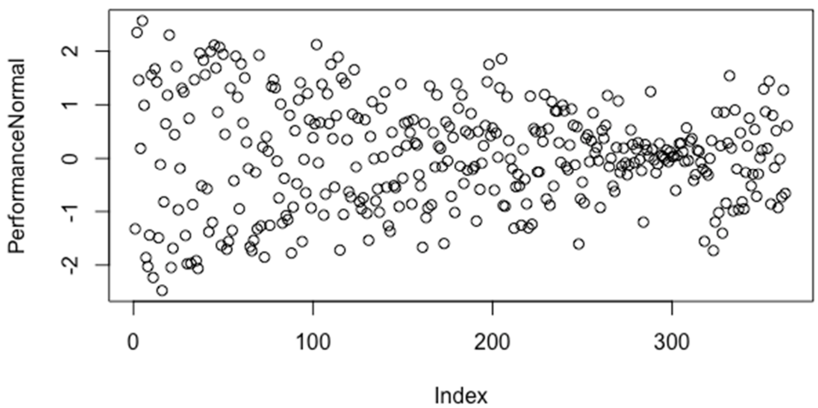
```
> library(Johnson)

> Z <- RE.Johnson(Performance)
> Z

[1] "Johnson Transformation"
$`function`
[1] "SU"
$p            [1] 0.3131561
$f.gamma      [1] 0.1326709    $f.lambda     [1] 128.8424
$f.epsilon    [1] 11.54548     $f.eta        [1] 0.7339281

> PerformanceNormal <- Z$transformed
> plot(PerformanceNormal)
```

Con los valores obtenidos en el resúmen del paquete “Johnson”, el rendimiento diario de un Bitcoin transformado “Rt” resulta en la siguiente ecuación:

$$Rt(d)_n = 0{,}1326709 + 0{,}7339281 \sinh^{-1}\left(\frac{R(d)_n - 11.54548}{128.8424}\right)$$

En el siguiente gráfico se puede observar la dispersión del conjunto de datos de la variable “Performance” diaria de un Bitcoin, luego de la ejecución de la transformación Johnson en R Studio®.

Imágen Nº 5 - Dispersión de la variable dependiente Performance normalizada.



Donde dice "PerformanceNormal" debe entenderse Rendimientos Normales.
Donde dice "Index" debe entenderse Indices

7.1 Modelo transformado

Con esta nueva variable respuesta, ajustamos el modelo lineal transformado:

$Rt_{ij} = \mu_i + \mathcal{E}_{ij} \qquad i = 1, \ldots, 7; \quad j = 1, \ldots, 52$

Donde:

Rt_{ij} es la observación del rendimiento transformado del Bitcoin en el día de la semana i, j respecto del dia anterior;
μ_i es el rendimiento transformado medio del Bitcoin del día i respecto del día anterior, es decir $i - 1$
$\mathcal{E}_{ij}$ es la variable aleatoria que representa la oscilación - error - del R_{ij} del Bitcoin.

En este caso también se presume una distribución del tipo $\mathcal{E}_{ij} \overset{iid}{\sim} \mathrm{N}(0, \delta^2)$, supuesto que se divide en: normalidad, homocedasticidad e independencia.

En este modelo transformado se observa que el "p" valor correspondiente al día sábado es el menor siendo = 0.0592, luego le sigue el "p" valor del día viernes = 0.2997 y en tercer lugar el "p" valor del día lunes = 0.4771.

Si bien aparecen en el análisis del "p" valor otros días de la semana, sigue siendo observado el día lunes como uno de los días que obtiene un valor significativo al resto de los días de la semana, si se tiene en cuenta además el valor "t" estadístico, siendo para el día lunes el valor mas pequeño = -0.712. Se ejecuta el comando "lm" de R Studio®, y se analizan los datos de su resumen.

```
> modelo1 = lm(PerformanceNormal~Day-1)
> summary(modelo1)

Call:
lm(formula = PerformanceNormal ~ Day - 1)

Residuals:
Min          1Q           Median       3Q           Max
-2.45736     -0.73895     -0.00152     0.61952      2.42342

Coefficients:
             Estimate     Std. Error   t value      Pr(>|t|)
DaySun        0.023250    0.136854      0.170       0.8652
DayMon       -0.097402    0.136854     -0.712       0.4771
DayTue       -0.018251    0.136854     -0.133       0.8940
DayWed       -0.049259    0.136854     -0.360       0.7191
DayThu        0.008148    0.136854     0.060        0.9526
DayFri        0.142129    0.136854     1.039        0.2997
DaySat        0.259079    0.136854     1.893        0.0592 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9869 on 357 degrees of freedom
Multiple R-squared:  0.01476,     Adjusted R-squared:  -0.004557
F-statistic: 0.7641 on 7 and 357 DF,  p-value: 0.6178
```

7.2 Análisis de los supuestos del modelo transformado

Análisis de los supuestos de normalidad, de homocedasticidad e de independencia.

7.2.1 Normalidad

El test de Shapiro - Wilk arroja un "p" valor = 0.1235, con el cual no se descarta la normalidad del modelo transformado. Adicionalmente se realizan otros test, como el test de normalidad de Jarque - Bera, el test de normalidad de Anderson - Darling y el test de normalidad de Kolmogorov - Smirnov, arrojando todos ellos un "p" valor > 0,05.

```
> shapiro.test(rstandard(modelo1))

Shapiro-Wilk normality test
data:  rstandard(modelo1)
```

W = 0.99356, p-value = 0.1235

```
> shapiro.test(residuals(modelo1))
Shapiro-Wilk normality test
data:  residuals(modelo1)
W = 0.99356, p-value = 0.1235

> jarque.test(PerformanceNormal)

Jarque-Bera Normality Test
data:  PerformanceNormal
JB = 3.0758, p-value = 0.2148
alternative hypothesis: greater

> ad.test(PerformanceNormal)

Anderson-Darling normality test
data:  PerformanceNormal
A = 0.42612, p-value = 0.3132

> lillie.test(PerformanceNormal)

Lilliefors (Kolmogorov-Smirnov) normality test
data:  PerformanceNormal
D = 0.031255, p-value = 0.5243

> lillie.test(rstandard(modelo1))
Lilliefors (Kolmogorov-Smirnov) normality test
data:  rstandard(modelo1)
D = 0.028635, p-value = 0.6634
```

7.2.2 Homocedasticidad

El test de Brush - Pagan produce un “p” valor = 0.2914. De esta manera no se rechaza el supuesto de homocedasticidad.

```
> bptest(modelo1)
studentized Breusch-Pagan test
data:  modelo1
BP = 7.3306, df = 6, p-value = 0.2914
```

7.2.3 Independencia

El test de Durbin - Watson arroja un “p” valor = 0.8729, con el que no se descarta la independencia.

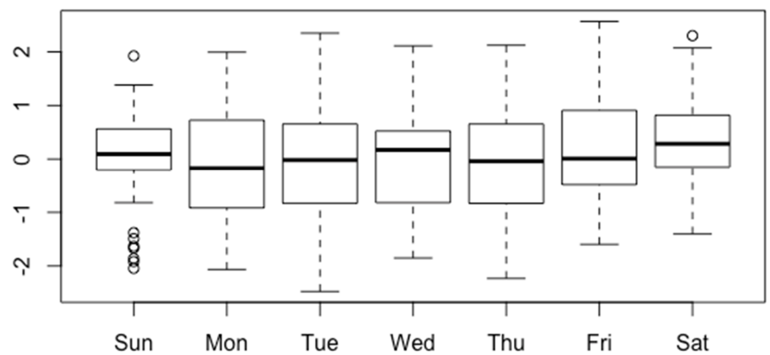
```
> dwtest(modelo1)

Durbin-Watson test
data:  modelo1
DW = 2.1149, p-value = 0.8729
alternative hypothesis: true autocorrelation is greater than 0
```

Por lo tanto, en el modelo con la variable respuesta “Rendimiento”, transformado a través de un algoritmo de la transformación Johnson, se alcanzan todos los supuestos, brindando así robustez al modelo resultante, y confianza en las conclusiones a las que se arriban.

8 Rendimiento transformado

Se analiza si existen diferencias significativas entre los rendimientos medios transformados de un Bitcoin, por cada día de la semana.

Imágen Nº 6 - Boxplot de los rendimientos medios transformados.



El primer paso es probar las hipótesis:
H(0): $\delta_1 = \delta_2 = \delta_3 = \delta_4 = \delta_5 = \delta_6 = \delta_7$ versus H(1): $\delta_i \neq \delta_j$ para al menos un par $i \neq j$. $\quad i, j = 1, \dots, 7$

El segundo paso es realizar un test de comparaciones múltiples como el que se logra con el test de Tukey.

9 Análisis de la Varianza de un factor y test de Tukey

Se realiza el ANOVA y el test de Tukey con la variable respuesta “Rendimiento” diario del Bitcoin transformado, que responde a una distribución correspondiente al tipo $\mathcal{E}_{ij} \overset{iid}{\sim} \mathrm{N}(0, \delta^2)$, alcanzando ese modelo los supuestos de normalidad, de homocedasticidad e de independencia:

$$Rt_{ij} = \mu_i + \mathcal{E}_{ij} \qquad i = 1, \dots, 7; \quad j = 1, \dots, 52$$

Donde:

Rt_{ij} es la observación del rendimiento transformado del Bitcoin en el día de la semana i, j respecto del dia anterior;
μ_i es el rendimiento transformado medio del Bitcoin del día i respecto del día anterior, es decir $i - 1$
$\mathcal{E}_{ij}$ es la variable aleatoria que representa la oscilación - error - del R_{ij} del Bitcoin.

9.1 ANOVA

Para realizar el análisis de la varianza de un factor, los datos deben proceder de una distribución normal, las varianzas de la variable dependiente en cada nivel del factor deben ser aproximadamente iguales y las observaciones deben ser independientes en los grupos que crea el factor.

En otras palabras, deben darse los supuestos de normalidad, de homocedasticidad e de independencia, supuestos que se lograron luego de la transformación de la variable respuesta.

Se prueba la hipótesis: H(0): $\delta_1 = \delta_2 = \delta_3 = \delta_4 = \delta_5 = \delta_6 = \delta_7$ versus H(1): $\delta_i \neq \delta_j$ para al menos un par $i \neq j$. $\quad i, j = 1, \dots, 7$ a través del análisis de la varianza de un factor, arrojando un “p” valor = 0.6178, con lo que no se rechaza la H(0).

```
> options(contrasts = c("contr.sum","contr.poly"))
> anova(modelo1)
```

Analysis of Variance Table

Response:	PerformanceNormal				
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Day	7	5.21	0.74417	0.7641	0.6178
Residuals	357	347.69	0.97391		

El resultado obtenido significa que el factor Day - Día - no ejerce ningún efecto sobre la variable dependiente Performance - Rendimiento -, no obstante ello puede analizarse el comportamiento entre los distintos niveles del factor, con el test de Tukey.

9.2 Comparaciones múltiples del modelo transformado

Se realizan comparaciones múltiples entre los rendimientos medios transformados del Bitcoin por cada día de la semana, a través del test de Tukey, arrojando un "p" valor de 0.521 para la dupla sábado - lunes.

```
> library(multcomp)
> ajuste1=aov(PerformanceNormal~Day-1)
> compara1=glht(ajuste1,mcp(Day="Tukey"))
> summary(compara1)
```

Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts
Fit: aov(formula = PerformanceNormal ~ Day - 1)

Linear Hypotheses:

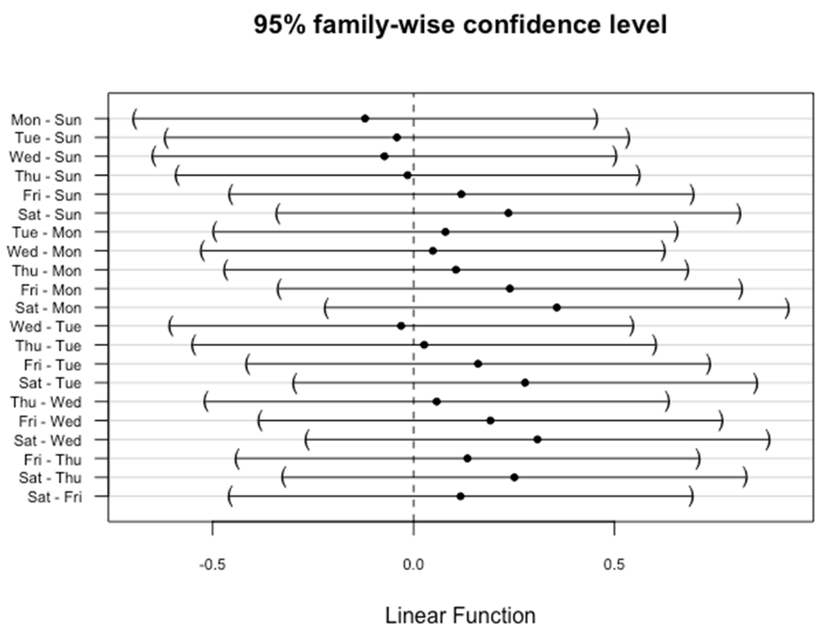
	Estimate	Std. Error	t value	Pr(>|t|)
Mon - Sun == 0	-0.12065	0.19354	-0.623	0.996
Tue - Sun == 0	-0.04150	0.19354	-0.214	1.000
Wed - Sun == 0	-0.07251	0.19354	-0.375	1.000
Thu - Sun == 0	-0.01510	0.19354	-0.078	1.000
Fri - Sun == 0	0.11888	0.19354	0.614	0.996
Sat - Sun == 0	0.23583	0.19354	1.218	0.887
Tue - Mon == 0	0.07915	0.19354	0.409	1.000
Wed - Mon == 0	0.04814	0.19354	0.249	1.000
Thu - Mon == 0	0.10555	0.19354	0.545	0.998
Fri - Mon == 0	0.23953	0.19354	1.238	0.879
Sat - Mon == 0	0.35648	0.19354	1.842	0.521
Wed - Tue == 0	-0.03101	0.19354	-0.160	1.000
Thu - Tue == 0	0.02640	0.19354	0.136	1.000
Fri - Tue == 0	0.16038	0.19354	0.829	0.982
Sat - Tue == 0	0.27733	0.19354	1.433	0.784
Thu - Wed == 0	0.05741	0.19354	0.297	1.000
Fri - Wed == 0	0.19139	0.19354	0.989	0.956
Sat - Wed == 0	0.30834	0.19354	1.593	0.687
Fri - Thu == 0	0.13398	0.19354	0.692	0.993
Sat - Thu == 0	0.25093	0.19354	1.297	0.853
Sat - Fri == 0	0.11695	0.19354	0.604	0.997

(Adjusted p values reported -- single-step method)

En el gráfico siguiente se observan las comparaciones múltiples de los rendimientos medios transformados del Bitcoin, la cual destaca la dupla sábado - lunes.

```
> par(cex.axis=0.7)
> plot(compara1)
```

Imágen Nº 7 - Test de Tukey de los rendimientos medios transformados.



10 Alternativas ante el no cumplimiento de los supuestos

En el modelo con la variable respuesta "rendimiento" sin transformar, en donde se supone que responde a una distribución $\mathcal{E}_{ij} \overset{iid}{\sim} \mathrm{N}(0, \varsigma^2)$, supuesto que se divide en los supuestos de: normalidad, homocedasticidad e independencia, y que se expresa de la siguiente manera:

$$R_{ij} = \mu + \mathcal{E}_{ij} \qquad i = 1, \ldots, 7; \quad j = 1, \ldots, 52$$

Donde:

R_{ij} es la observación del rendimiento del Bitcoin en el día de la semana i, j respecto del dia anterior;
μ_i es el rendimiento del Bitcoin del día i respecto del día anterior, es decir $i - 1$
$\mathcal{E}_{ij}$ es la variable aleatoria que representa la oscilación - error - del R_{ij} del Bitcoin

ya se comprobó que el supuesto normalidad no se alcanza, por lo que se realiza la comparación Pairwise usando el ajuste Rfit, arrojando un "p" valor = 0.53697, con lo cual, no se rechaza la H(0) lo que implica que no existen oscilaciones significativas en los rendimientos medios en el modelo.

```
> oneway.rfit(Performance, Day)

Call:
oneway.rfit(y = Performance, g = Day)

Overall Test of All Locations Equal

Drop in Dispersion Test
F-Statistic   p-value
   0.84349    0.53697

Pairwise comparisons using Rfit
data:  Performance and Day
```

	Mon	Tue	Wed	Thu	Fri	Sat
Tue	0.292	-	-	-	-	-
Wed	0.641	0.557	-	-	-	-
Thu	0.683	0.518	0.954	-	-	-
Fri	0.766	0.449	0.866	0.911	-	-
Sat	0.847	0.213	0.510	0.548	0.624	-
Sun	0.340	0.045	0.156	0.173	0.211	0.446

P value adjustment method: none

No obstante, cuando se comparan las variables Rendimientos - Performance - y Día - Day - la dupla martes - domingo arroja un “p” valor < 0,05. En todas las demás combinaciones de días, el “p” valor es > 0,05

10.1 Comparaciones múltiples ante el no cumplimiento de los supuestos

Si no se rechaza la H(0) nula, aún así podemos ver cual es el día de la semana con mayor oscilación.

```
> modelo.rfit=oneway.rfit(Performance, Day)
> summary.oneway.rfit(modelo.rfit,method = "tukey")
```

Multiple Comparisons
Method Used tukey

	I	J	Estimate	St Err	Lower Bound CI	Upper Bound CI
1	Mon	Tue	-49.71928	47.12357	-189.45201	90.01345
2	Mon	Wed	-21.99779	47.12357	-161.73051	117.73494
3	Mon	Thu	-19.25699	47.12357	-158.98972	120.47574
4	Mon	Fri	-14.01301	47.12357	-153.74574	125.71972
5	Mon	Sat	9.08253	47.12357	-130.65020	148.81525
6	Mon	Sun	45.03146	47.12357	-94.70127	184.76419
7	Tue	Wed	-27.72149	47.12357	-167.45422	112.01124

Esa mayor oscilación se dá entre el día lunes y martes.

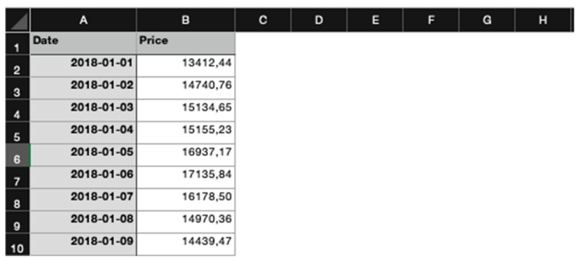
11 Desarrollo en RapidMiner Studio®

La aplicación que se utiliza es RapidMiner Studio® versión 9.2 con licencia educativa.

11.1 Preparación de datos del precio del Bitcoin.

Se importan los datos contenidos en el archivo "datos 364.xlsx" a la aplicación RapidMiner Studio®. La planilla de cálculos posee las columnas "Date" con las fechas de cada registración del valor del precio de un Bitcoin, y la columna "Price" con el precio resultante en ese día.

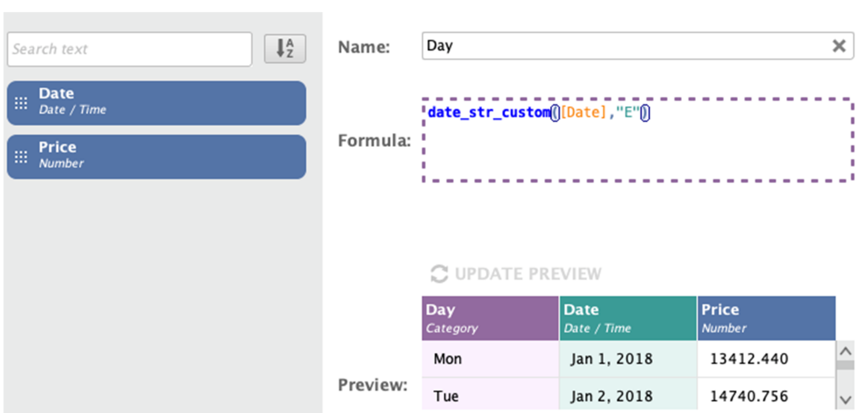
Imagen Nº 8 - Planilla de cálculos con las columnas Date y Price



	A	B	C	D	E	F	G	H
1	Date	Price						
2	2018-01-01	13412,44						
3	2018-01-02	14740,76						
4	2018-01-03	15134,65						
5	2018-01-04	15155,23						
6	2018-01-05	16937,17						
7	2018-01-06	17135,84						
8	2018-01-07	16178,50						
9	2018-01-08	14970,36						
10	2018-01-09	14439,47						

Una vez importados los datos del conjunto de datos contenidos en el archivo de la planilla de cálculos, correspondiente a los precios de un Bitcoin desde el día 01-01-2018 al 30-12-2018, se genera una nueva columna llamada "Day" con el nombre correspondiente al día de la semana, de acuerdo a los datos de la columna "Date".

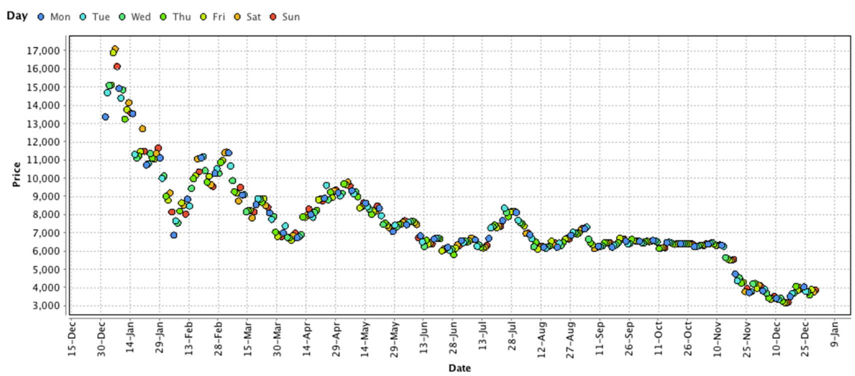
Imagen Nº 9 - Generación de la columna Day



El comando utilizado es date_str_custom([Date],"E")

Ahora, el conjunto de datos posee tres variables, la variable dependiente “Price” con atributo numérico, y las variables independientes “Day” y “Date”, con atributos categoría y fecha respectivamente. Se visualiza la variable “Day” en el siguiente gráfico.

Imagen Nº 10 - Gráfico de las variables Price y Date, agrupadas por el valor Day



11.2 Aplicación algoritmo K-means

Se ejecuta la modelización del conjunto de datos, teniendo en cuenta las tres variables, y parametrizando como dos (2) la cantidad de clúster para el algoritmo, de modo tal que el agrupamiento sea por la variable dependiente “Price”. Se selecciona como forma de medir la distancia, la euclidiana, por ser la mas estandarizada.

En el resumen de la ejecución del modelo definido se conforman los dos clústers de la siguiente manera, el primero agrupando al precio del Bitcoin en promedio mayor, y el segundo agrupando al precio del Bitcoin en promedio menor. En ambas agrupaciones, el día lunes es el día que destaca, junto con el día sábado, tal como se lo constata en la siguiente imagen.

Imagen Nº 11 - Resumen algoritmo K-means

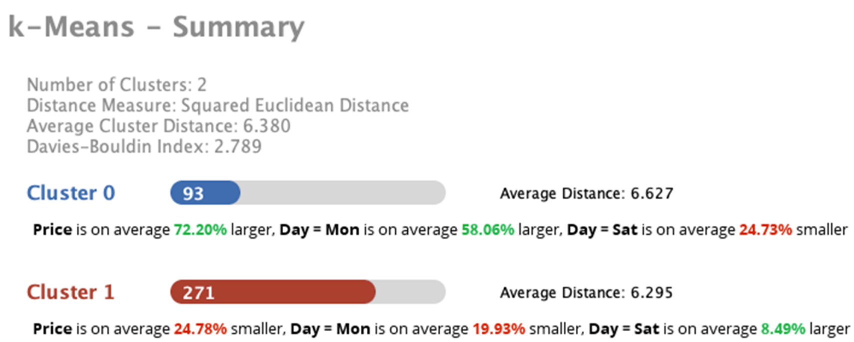
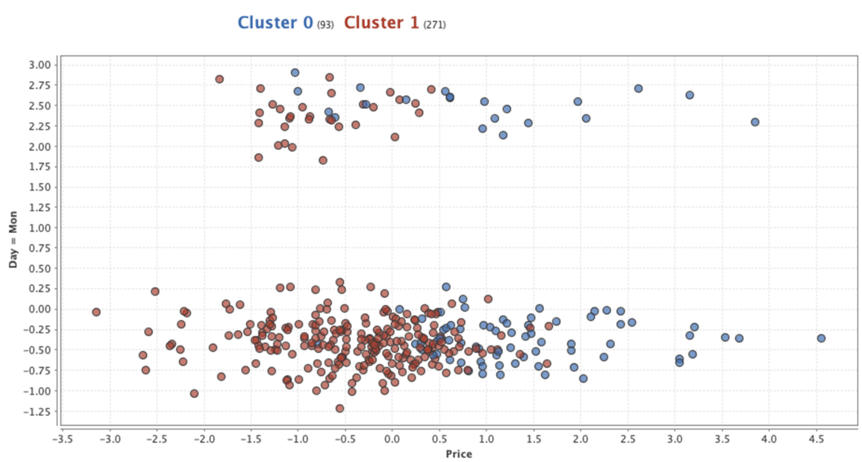


Imagen Nº 12 - Clusters K-means para el día Lunes



12 Cambiando la variable respuesta

Si la variable respuesta es ahora los rendimientos, tal como se ha desarrollado en los puntos anteriores, ajustando los datos a ser importados a la herramienta RapidMiner Studio®, se obtiene la siguiente tabla resumen de la aplicación del algoritmo K-means con valores similares para los distintos niveles del factor, ya que la muestra contiene la misma cantidad de días para cada uno de ellos.

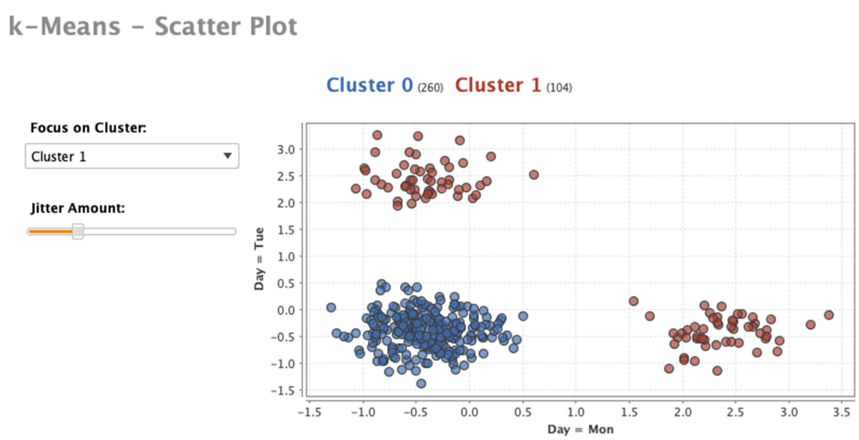
Se observa que para los dos clusters, los valores obtenidos en el día lunes, se repiten para el día martes, y el resto de los días tienen los mismos valores. Es decir los días lunes y los días martes presentan un comportamiento similar, y ese comportamiento es diferente con el resto de los días de la semana. Se constata que aún así, el día lunes sigue siendo un día que presenta un efecto en la semana del mercado de la criptodivisa Bitcoin. En el siguiente gráfico se puede observar la agrupación inusual (Baig, Blau, & Sabah, 2019) de los rendimientos del Bitcoin.

Imagen Nº 13 - Tabla K-means para los niveles del factor

k-Means - Centroid Table

Cluster	Day = Mon	Day = Sat	Day = Sun	Day = Thu	Day = Tue	Day = Wed	Performance
Cluster 0	-0.408	0.163	0.163	0.163	-0.408	0.163	0.032
Cluster 1	1.019	-0.408	-0.408	-0.408	1.019	-0.408	-0.079

Imagen Nº 13 - Clusters K-means para el día Lunes y el día Martes



13 Conclusión y proyección

Del conjunto de análisis realizados puede inferirse el efecto del día de la semana para el mercado de la criptodivisa Bitcoin, no solo en los precios, sino también en los rendimientos diarios del mismo.

La criptodivisa Bitcoin posee una gran independencia y las variables especulativas de los mercados financieros solamente se limitan a pronosticar el precio (Aharon & Qadan, 2018).

Teniendo en cuenta que el Bitcoin fue la primera criptodivisa en operar, se deberá analizar si ese efecto del día de la semana afecta a otras criptodivisas del mercado como las denominadas altcoins, entendiendo que la volatilidad en el mercado del Bitcoin no se debe al comercio especulativo de la criptodivisa, ya que el comercio especulativo no es significativo comparado con el total de las transacciones (Aharon & Qadan, 2018), sino que responde a otros factores del mercado, tales como el anonimato en las transacciones y la falta de regulación de la criptodivisa que la habilite como moneda.

Referencias

Aharon, D. Y., & Qadan, M. (2018). Bitcoin and the day-of-the-week effect. *Finance Research Letters*. doi: 10.1016/j.frl.2018.12.004

Baig, A., Blau, B. M., & Sabah, N. (2019). Price clustering and sentiment in bitcoin. *Finance Research Letters*, *29*, 111-116. doi: 10.1016/j.frl.2019.03.013

Blau, B. M. (2017). Price dynamics and speculative trading in bitcoin. *Research in International Business and Finance*, *41*, 493-499. doi: 10.1016/j.ribaf.2017.05.010

Caporale, G. M., & Plastun, A. (2018). The day of the week effect in the cryptocurrency market. *Finance Research Letters*. doi: 10.1016/j.frl.2018.11.012

de Groot, W., & Huij, J. (2018). Are the Fama-French factors really compensation for distress risk? *Journal of International Money and Finance*, *86*, 50-69. doi: 10.1016/j.jimonfin.2018.03.002

Foye, J. (2018). A comprehensive test of the Fama-French five-factor model in emerging markets. *Emerging Markets Review*. doi: 10.1016/j.ememar.2018.09.002

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. *www.bitcoin.org*, 9-9. doi: 10.1007/s10838-008-9062-0

Sottile Bordallo, A., Cavaller Riva, D., Ortega Yubro, C. D., Silva, D. V., Sosa, H. N., & Dueñas, E. A. (2019). *Análisis de causales de datos con anomalías*. Recuperado de http://itunes.apple.com/us/book/id1451742749