Analisis_de_acciones

Grupo de criptomonedas

2024-11-24

Librerías

Se cargan las librerías necesarias.

```
pacman::p_load(corrplot,
               readxl,
               tidyverse,
               xtable)
```

Descarga de datos

Se cargan los datos de los activos necesarios.

```
# Criptomonedas
datos.bitcoin.original <- read_excel("data/Datos históricos del Bitcoin.xlsx")
datos.ethereum <- read_excel("data/Datos históricos del Ethereum.xlsx")

# Acciones de empresas
datos.apple <- read_excel("data/Apple.xlsx")
datos.nvidia <- read_excel("data/NVIDIA.xlsx")
```

Correcciones en las bases de datos

Se corrigen algunas características de las bases de datos a utilizar.

```
# Formato de fecha
datos.bitcoin.original$Fecha <-
  as.Date(datos.bitcoin.original$Fecha, format = "%d.%m.%Y")

datos.ethereum$Fecha <-
  as.Date(datos.ethereum$Fecha, format = "%d.%m.%Y")

datos.nvidia$Date <-
  as.Date(datos.nvidia$Date, format = "%d.%m.%Y")

datos.apple$Date <-
  as.Date(datos.apple$Date, format = "%d.%m,%Y")
```

```r
# Se agregan columnas con el volumen en números para el bitcoin
datos.bitcoin.original <- datos.bitcoin.original %>%
  mutate(
    miles = as.numeric(gsub("K", "", datos.bitcoin.original$Vol)) * 1000,
    millones = as.numeric(gsub("M", "", datos.bitcoin.original$Vol)) * 1000000,
    billones = as.numeric(gsub("B", "", datos.bitcoin.original$Vol)) * 1000000000
  )

# Se juntan las columnas en una sola
datos.bitcoin.original$Vol <-
  ifelse(
    !is.na(datos.bitcoin.original$miles),
    datos.bitcoin.original$miles,
    ifelse(
      !is.na(datos.bitcoin.original$millones),
      datos.bitcoin.original$millones,
      ifelse(
        !is.na(datos.bitcoin.original$billones),
        datos.bitcoin.original$billones,
        NA
      )
    )
  )

# Se eliminan las columnas innecesarias
datos.bitcoin.original <- datos.bitcoin.original %>%
  select(-c("miles", "millones", "billones"))

# Se agregan columnas con el volumen en números para el ethereum
datos.ethereum <- datos.ethereum %>%
  mutate(
    miles = as.numeric(gsub("K", "", datos.ethereum$Vol)) * 1000,
    millones = as.numeric(gsub("M", "", datos.ethereum$Vol)) * 1000000,
    billones = as.numeric(gsub("B", "", datos.ethereum$Vol)) * 1000000000
  )

# Se juntan las columnas en una sola
datos.ethereum$Vol <-
  ifelse(
    !is.na(datos.ethereum$miles),
    datos.ethereum$miles,
    ifelse(
      !is.na(datos.ethereum$millones),
      datos.ethereum$millones,
      ifelse(!is.na(datos.ethereum$billones), datos.ethereum$billones, NA)
    )
  )

# Se eliminan las columnas innecesarias
datos.ethereum <- datos.ethereum %>%
  select(-c("miles", "millones", "billones"))

# Se rellenan los valores nulos con el último conocido
```

```r
datos.ethereum <- datos.ethereum %>%
  fill(Vol, .direction = "up")

# Nombres de las columnas
colnames(datos.bitcoin.original) <-
  c("fecha",
    "ultimo",
    "apertura",
    "maximo",
    "minimo",
    "volumen",
    "variacion")

colnames(datos.ethereum) <-
  c("fecha",
    "ultimo",
    "apertura",
    "maximo",
    "minimo",
    "volumen",
    "variacion")
```

```r
# Se elimina la columna del último precio ajustado de la base de NVIDIA
datos.nvidia <- datos.nvidia %>%
  select(-c("Adj Close"))


# Se agrega la columna de variación porcentual a la base de NVIDIA
datos.nvidia <- datos.nvidia %>%
  mutate(variacion = ((Close / lag(Close)) - 1))

# Se intercambian las columnas de la base de NVIDIA
datos.nvidia <- datos.nvidia %>%
  select(Date, Close, Open, High, Low, Volume, variacion)

# Se invierte el orden de las filas de la base de NVIDIA para que estén iguales
# las demás
datos.nvidia <- datos.nvidia[rev(rownames(datos.nvidia)), ]

# Nombres de las columnas
colnames(datos.nvidia) <-
  c("fecha",
    "ultimo",
    "apertura",
    "maximo",
    "minimo",
    "volumen",
    "variacion")
```

```r
# Se elimina la columna del último precio ajustado de la base de Apple
datos.apple <- datos.apple %>%
  select(-c("Adj Close"))
```

```r
# Se agrega la columna de variación porcentual a la base de NVIDIA
datos.apple <- datos.apple %>%
  mutate(variacion = ((Close / lag(Close)) - 1))

# Se intercambian las columnas de la base de Apple
datos.apple <- datos.apple %>%
  select(Date, Close, Open, High, Low, Volume, variacion)


# Se invierte el orden de las filas de la base de NVIDIA para que estén iguales
# las demás
datos.apple <- datos.apple[rev(rownames(datos.apple)), ]

#Nombre de las columnas
colnames(datos.apple) <-
  c("fecha",
    "ultimo",
    "apertura",
    "maximo",
    "minimo",
    "volumen",
    "variacion")
```

Comparación de criptomonedas

Se extienden las bases del bitcoin y del ethereum para tener algunas estadísticas descriptivas.

```r
# Se comparan la misma cantidad de fechas de los activos.
datos.bitcoin <- datos.bitcoin.original[1:nrow(datos.ethereum), ]

# Datos del bitcoin extendidos
resumen.btc <- datos.bitcoin %>%
  pivot_longer(-c(fecha))

# Datos del ethereum extendidos
resumen.etr <- datos.ethereum %>%
  pivot_longer(-c(fecha))
```

Se procede a realizar una tabla resumen de las criptomonedas.

```r
# Resumen del bitcoin
resumen.btc <- resumen.btc %>%
  group_by(name) %>%
  summarise(
    promedio = mean(value, na.rm = TRUE),
    minimo = min(value, na.rm = TRUE),
    mediana = quantile(value, 0.5),
    maximo = max(value, na.rm = TRUE),
    desviacion = sd(value, na.rm = TRUE)
  )
```

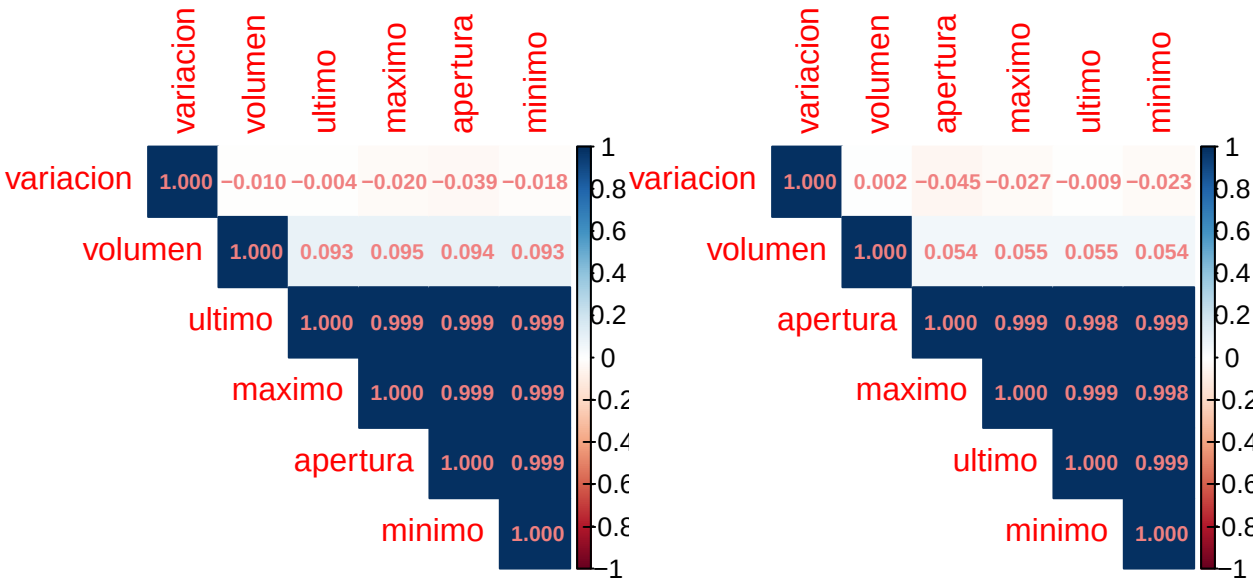
```r
# Resumen del ethereum
resumen.etr <- resumen.etr %>%
  group_by(name) %>%
  summarise(
    promedio = mean(value, na.rm = TRUE),
    minimo = min(value, na.rm = TRUE),
    mediana = quantile(value, 0.5),
    maximo = max(value, na.rm = TRUE),
    desviacion = sd(value, na.rm = TRUE)
  )
```

Hacemos una matriz de correlación para observar si el comportamiento de las 2 criptomonedas tiene relación.

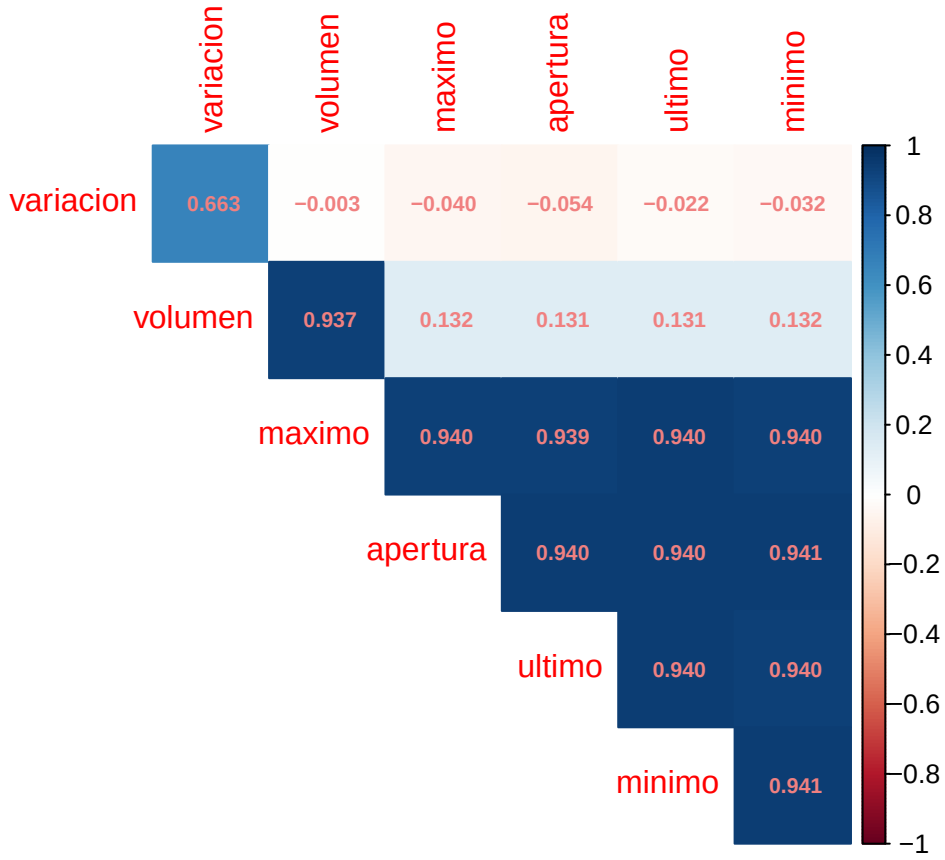
```r
# Correlaciones individuales
correlaciones.btc <- cor(datos.bitcoin[, -1], datos.bitcoin[, -1])
correlaciones.etr <- cor(datos.ethereum[, -1], datos.ethereum[, -1])

# Correlaciones cruzadas
correlaciones.cruz <- cor(datos.bitcoin[, -1], datos.ethereum[, -1])
```

Se realizan los gráficos de las matrices de correlación.

```r
# Output de los gráficos
par(mfrow = c(1, 2))

# Correlación del bitcoin
corrplot(
  correlaciones.btc,
  method = "color",
  type = "upper",
  order = "hclust",
  addCoef.col = "lightcoral",
  number.cex = 0.7,
  number.digits = 3
)

# Correlación de ethereum
corrplot(
  correlaciones.etr,
  method = "color",
  type = "upper",
  order = "hclust",
  addCoef.col = "lightcoral",
  number.cex = 0.7,
  number.digits = 3
)
```



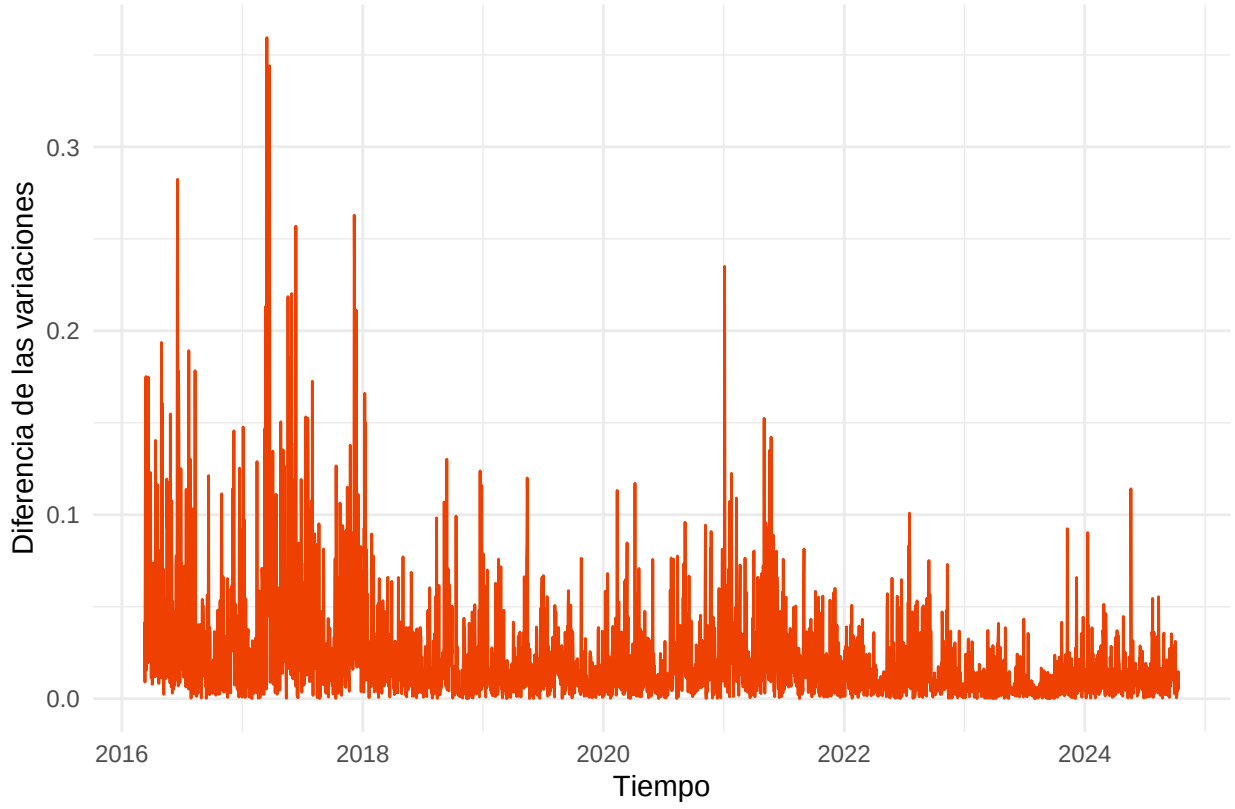

```
# Correlaciones cruzadas
corrplot(
  correlaciones.cruz,
  method = "color",
  type = "upper",
  order = "hclust",
  addCoef.col = "lightcoral",
  number.cex = 0.7,
  number.digits = 3
)
```



Se realiza un gráfico comparativo de las variaciones diarias del bitcoin y de ethereum.

```r
# Dataframe auxiliar
df.auxiliar <-
  as.data.frame(cbind(
    datos.bitcoin$fecha,
    datos.bitcoin$variacion,
    datos.ethereum$variacion
  ))

# Se corrigen los formatos de columna
colnames(df.auxiliar) <- c("fecha", "cambio_btc", "cambio_etr")

# Formato de fecha
df.auxiliar$fecha <- as.Date(df.auxiliar$fecha, format = "%d.%m.%Y")

# Diferencias de las variaciones
df.auxiliar <- df.auxiliar %>%
  mutate(diferencia = abs(cambio_btc - cambio_etr))

# Gráfico comparativo de varianzas diarias
ggplot(df.auxiliar, aes(x = fecha)) +
  geom_line(aes(y = diferencia),
            color = "orangered2",
            linewidth = 0.5) +
  labs(x = "Tiempo", y = "Diferencia de las variaciones",
```

```
      caption = "Fuente: Elaboración propia") +
  theme_minimal()
```



Fuente: Elaboración propia

Se imprimen las tablas correspondientes.

```
# Se corrigen los nombres
colnames(resumen.btc)[1] <- "variable"
colnames(resumen.etr)[1] <- "variable"

# Creamos las tablas
tabla.btc <- xtable(resumen.btc)
tabla.etr <- xtable(resumen.etr)

# Las imprimimos
print(tabla.btc, type = "latex", digits = 4)
```

```
## % latex table generated in R 4.4.1 by xtable 1.8-4 package
## % Sun Nov 24 00:20:11 2024
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrrrrr}
##   \hline
##  & variable & promedio & minimo & mediana & maximo & desviacion \\
##   \hline
## 1 & apertura & 21158.15 & 408.20 & 11244.50 & 73066.70 & 19998.08 \\
```

```
##   2 & maximo & 21657.07 & 410.50 & 11539.00 & 73740.90 & 20443.63 \\
##   3 & minimo & 20624.78 & 402.10 & 10878.40 & 71338.40 & 19518.64 \\
##   4 & ultimo & 21178.05 & 408.20 & 11268.00 & 73066.30 & 20008.69 \\
##   5 & variacion & 0.00 & -0.39 & 0.00 & 0.26 & 0.04 \\
##   6 & volumen & 19662591.09 & 260.00 & 120380.00 & 4470000000.00 & 195109406.19 \\
##    \hline
## \end{tabular}
## \end{table}
```

```
print(tabla.etr, type = "latex", digits = 4)
```

```
## % latex table generated in R 4.4.1 by xtable 1.8-4 package
## % Sun Nov 24 00:20:11 2024
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrrrrr}
##   \hline
##  & variable & promedio & minimo & mediana & maximo & desviacion \\
##   \hline
## 1 & apertura & 1175.03 & 6.68 & 539.90 & 4808.34 & 1206.07 \\
##   2 & maximo & 1208.77 & 7.32 & 566.63 & 4864.06 & 1238.10 \\
##   3 & minimo & 1137.68 & 5.86 & 514.44 & 4715.43 & 1169.71 \\
##   4 & ultimo & 1175.81 & 6.70 & 543.42 & 4808.38 & 1206.11 \\
##   5 & variacion & 0.00 & -0.45 & 0.00 & 0.30 & 0.05 \\
##   6 & volumen & 9649290.63 & 0.00 & 827200.00 & 1790000000.00 & 88024929.38 \\
##    \hline
## \end{tabular}
## \end{table}
```

Comparación del Bitcoin con NVIDIA

Se tienen que guardar las observaciones de ambos activos en las fechas en común.

```
# Se calcula la diferencia de días entre la última fecha del bitcoin y la
# última fecha de NVIDIA
diferencia_dias_nvidia <-
  as.numeric(datos.bitcoin.original$fecha[1] - datos.nvidia$fecha[1]) + 1

# Seleccionamos las observaciones del bitcoin usando la diferencia de días
datos.bitcoin <-
  datos.bitcoin.original[diferencia_dias_nvidia:nrow(datos.bitcoin.original), ]

# Filtramos los datos del bitcoin para que coincidan con la misma fecha que los
# de NVIDIA
datos.bitcoin <- datos.bitcoin[datos.bitcoin$fecha %in% datos.nvidia$fecha, ]

# Seleccionamos los datos de NVIDIA a partir de la fecha del primer dato del bitcoin
datos.nvidia <- datos.nvidia[1:nrow(datos.bitcoin), ]
```

Pivoteamos los datos del Bitcoin y de las acciones.

```r
# Datos del bitcoin extendidos
resumen.btc <- datos.bitcoin %>%
  pivot_longer(-c(fecha))

# Datos de nvidia extendidos
resumen.nvidia <- datos.nvidia %>%
  pivot_longer(-c(fecha))
```

Calculamos algunas estadísticas descriptivas de ambos instrumentos.

```r
# Resumen del bitcoin
resumen.btc <- resumen.btc %>%
  group_by(name) %>%
  summarise(
    promedio = mean(value, na.rm = TRUE),
    minimo = min(value, na.rm = TRUE),
    mediana = quantile(value, 0.5),
    maximo = max(value, na.rm = TRUE),
    desviacion = sd(value, na.rm = TRUE)
  )

# Resumen de NVIDIA
resumen.nvidia <- resumen.nvidia %>%
  group_by(name) %>%
  summarise(
    promedio = mean(value, na.rm = TRUE),
    minimo = min(value, na.rm = TRUE),
    mediana = quantile(value, 0.5),
    maximo = max(value, na.rm = TRUE),
    desviacion = sd(value, na.rm = TRUE)
  )
```

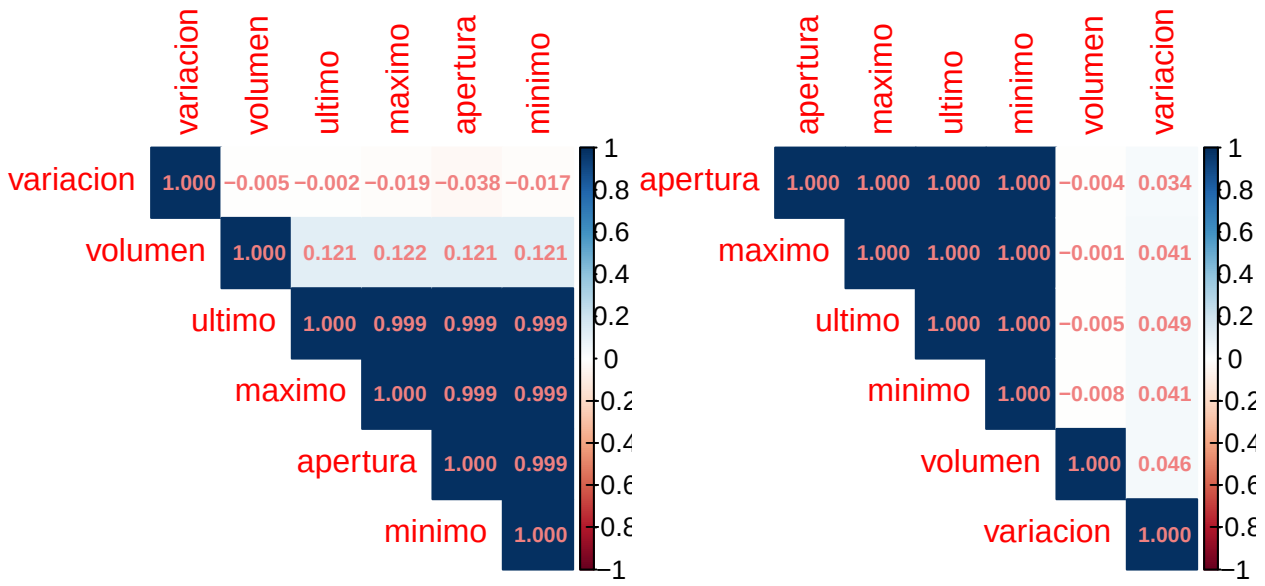
Calculamos las correlaciones individuales y conjunta.

```r
# Correlaciones individuales
correlaciones.btc <- cor(datos.bitcoin[, -1], datos.bitcoin[, -1])
correlaciones.nvidia <- cor(datos.nvidia[, -1], datos.nvidia[, -1])

# Correlaciones cruzadas
correlaciones.cruz <- cor(datos.bitcoin[, -1], datos.nvidia[, -1])
```

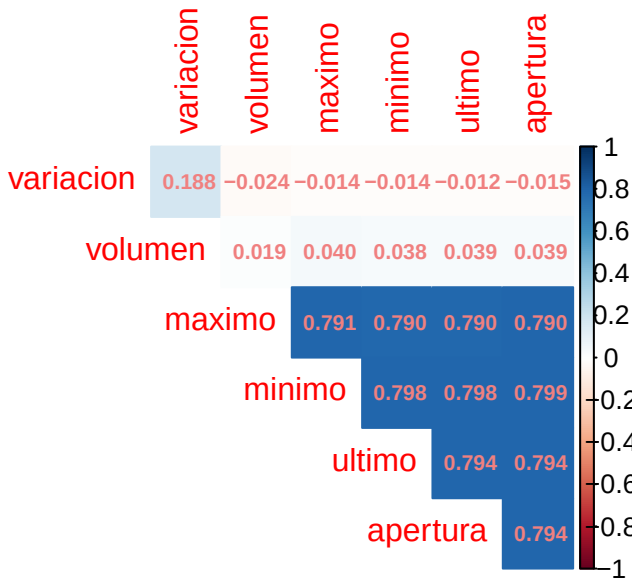
Graficamos las correlaciones.

```r
# Output de los gráficos
par(mfrow = c(1, 2))

# Correlación del bitcoin
corrplot(
  correlaciones.btc,
  method = "color",
  type = "upper",
  order = "hclust",
  addCoef.col = "lightcoral",
```

```
  number.cex = 0.7,
  number.digits = 3
)

# Correlación de NVIDIA
corrplot(
  correlaciones.nvidia,
  method = "color",
  type = "upper",
  order = "hclust",
  addCoef.col = "lightcoral",
  number.cex = 0.7,
  number.digits = 3
)
```



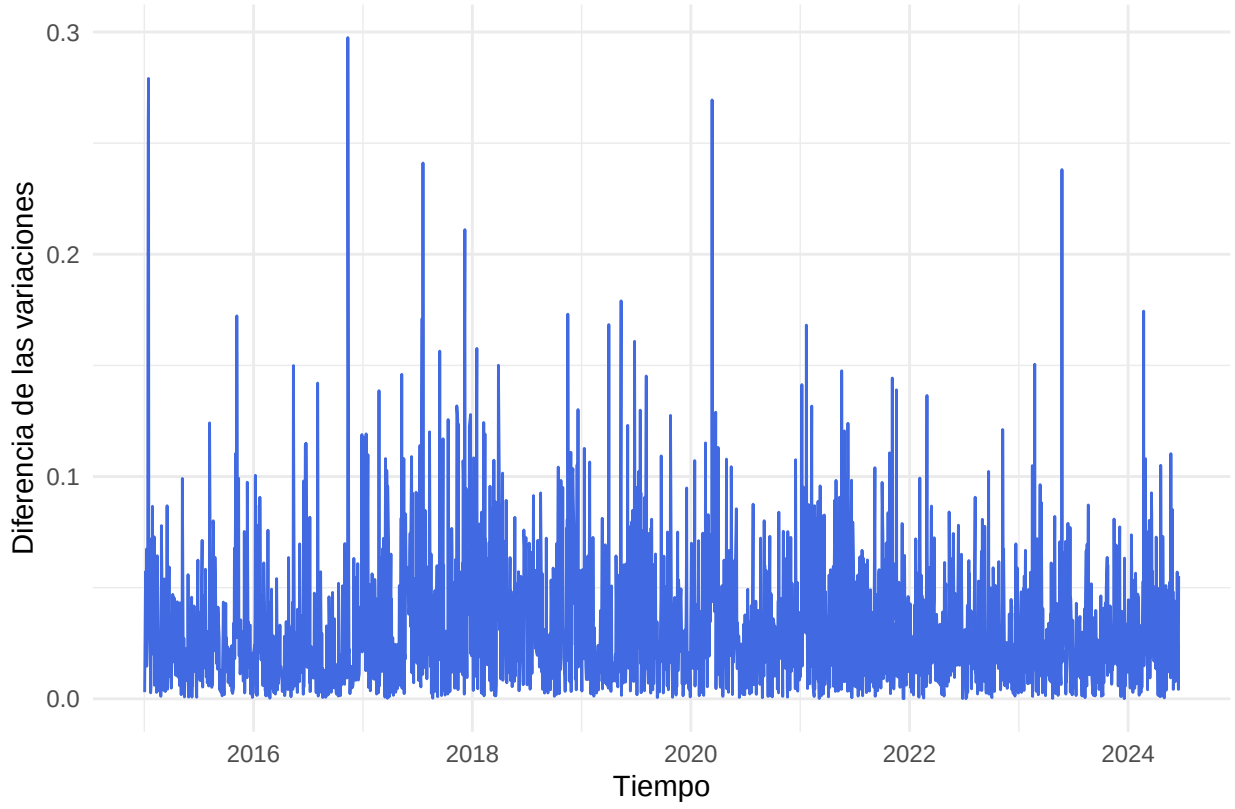

```
# Correlaciones cruzadas
corrplot(
  correlaciones.cruz,
  method = "color",
  type = "upper",
  order = "hclust",
  addCoef.col = "lightcoral",
  number.cex = 0.7,
  number.digits = 3
)
```



Seguidamente realizamos el gráfico de la diferencia de las variaciones de cada instrumento.

```r
# Dataframe auxiliar
df.auxiliar <-
  as.data.frame(cbind(
    datos.bitcoin$fecha,
    datos.bitcoin$variacion,
    datos.nvidia$variacion
  ))

# Se corrigen los formatos de columna
colnames(df.auxiliar) <- c("fecha", "cambio_btc", "cambio_nvidia")

# Formato de fecha
df.auxiliar$fecha <- as.Date(df.auxiliar$fecha, format = "%d.%m.%Y")

# Diferencias de las variaciones
df.auxiliar <- df.auxiliar %>%
  mutate(diferencia = abs(cambio_btc - cambio_nvidia))

# Gráfico comparativo de varianzas diarias
ggplot(df.auxiliar, aes(x = fecha)) +
  geom_line(aes(y = diferencia),
            color = "royalblue",
            linewidth = 0.5) +
  labs(x = "Tiempo", y = "Diferencia de las variaciones",
```

```
      caption = "Fuente: Elaboración propia") +
  theme_minimal()
```



Fuente: Elaboración propia

Por último, imprimimos las tablas correspondientes.

```
# Se corrigen los nombres
colnames(resumen.btc)[1] <- "variable"
colnames(resumen.nvidia)[1] <- "variable"

# Creamos las tablas
tabla.btc <- xtable(resumen.btc)
tabla.nvidia <- xtable(resumen.nvidia)

# Las imprimimos
print(tabla.btc, type = "latex", digits = 4)
```

```
## % latex table generated in R 4.4.1 by xtable 1.8-4 package
## % Sun Nov 24 00:20:11 2024
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrrrrr}
##   \hline
##  & variable & promedio & minimo & mediana & maximo & desviacion \\
##   \hline
## 1 & apertura & 17210.86 & 164.90 & 9194.90 & 73066.70 & 18690.13 \\
```

```
##   2 & maximo & 17665.91 & 216.60 & 9363.00 & 73740.90 & 19177.05 \\
##   3 & minimo & 16716.78 & 157.30 & 8962.10 & 71338.40 & 18147.78 \\
##   4 & ultimo & 17225.86 & 164.90 & 9185.40 & 73066.30 & 18694.17 \\
##   5 & variacion & 0.00 & -0.39 & 0.00 & 0.27 & 0.04 \\
##   6 & volumen & 21198260.74 & 260.00 & 117510.00 & 4470000000.00 & 213388409.32 \\
##    \hline
## \end{tabular}
## \end{table}
```

```
print(tabla.nvidia, type = "latex", digits = 4)
```

```
## % latex table generated in R 4.4.1 by xtable 1.8-4 package
## % Sun Nov 24 00:20:11 2024
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrrrrr}
##   \hline
##  & variable & promedio & minimo & mediana & maximo & desviacion \\
##   \hline
## 1 & apertura & 14.74 & 0.48 & 6.20 & 132.99 & 20.18 \\
##   2 & maximo & 15.01 & 0.49 & 6.28 & 136.33 & 20.54 \\
##   3 & minimo & 14.46 & 0.47 & 6.10 & 130.69 & 19.79 \\
##   4 & ultimo & 14.75 & 0.48 & 6.19 & 135.58 & 20.21 \\
##   5 & variacion & 0.00 & -0.19 & 0.00 & 0.30 & 0.03 \\
##   6 & volumen & 478202198.87 & 52448000.00 & 427700000.00 & 3692928000.00 & 255604462.01 \\
##    \hline
## \end{tabular}
## \end{table}
```

Comparación del Bitcoin con Apple

```
diferencia_dias_apple <-
  as.numeric(datos.bitcoin.original$fecha[1] - datos.apple$fecha[1]) + 1

# Seleccionamos las observaciones del bitcoin usando la diferencia de días
datos.bitcoin.apple <-
  datos.bitcoin.original[diferencia_dias_apple:nrow(datos.bitcoin.original), ]

# Filtramos los datos del bitcoin para que coincidan con la misma fecha que los
# de NVIDIA
datos.bitcoin.apple <- datos.bitcoin.apple[datos.bitcoin.apple$fecha %in% datos.apple$fecha, ]

# Seleccionamos los datos de NVIDIA a partir de la fecha del primer dato del bitcoin
datos.apple <- datos.apple[1:nrow(datos.bitcoin.apple), ]
```

Pivoteamos los datos del Bitcoin y de las acciones.

```
# Datos del bitcoin extendidos
resumen.btc.y.apple. <- datos.bitcoin.apple %>%
  pivot_longer(-c(fecha))
```

```r
# Datos de apple extendidos
resumen.apple <- datos.apple %>%
  pivot_longer(-c(fecha))
```

Comparacion de ambos activos

```r
# Resumen del bitcoin
resumen.btc.y.apple. <- resumen.btc.y.apple. %>%
  group_by(name) %>%
  summarise(
    promedio = mean(value, na.rm = TRUE),
    minimo = min(value, na.rm = TRUE),
    mediana = quantile(value, 0.5),
    maximo = max(value, na.rm = TRUE),
    desviacion = sd(value, na.rm = TRUE)
  )

# Resumen de Apple
resumen.apple <- resumen.apple %>%
  group_by(name) %>%
  summarise(
    promedio = mean(value, na.rm = TRUE),
    minimo = min(value, na.rm = TRUE),
    mediana = quantile(value, 0.5),
    maximo = max(value, na.rm = TRUE),
    desviacion = sd(value, na.rm = TRUE)
  )
```

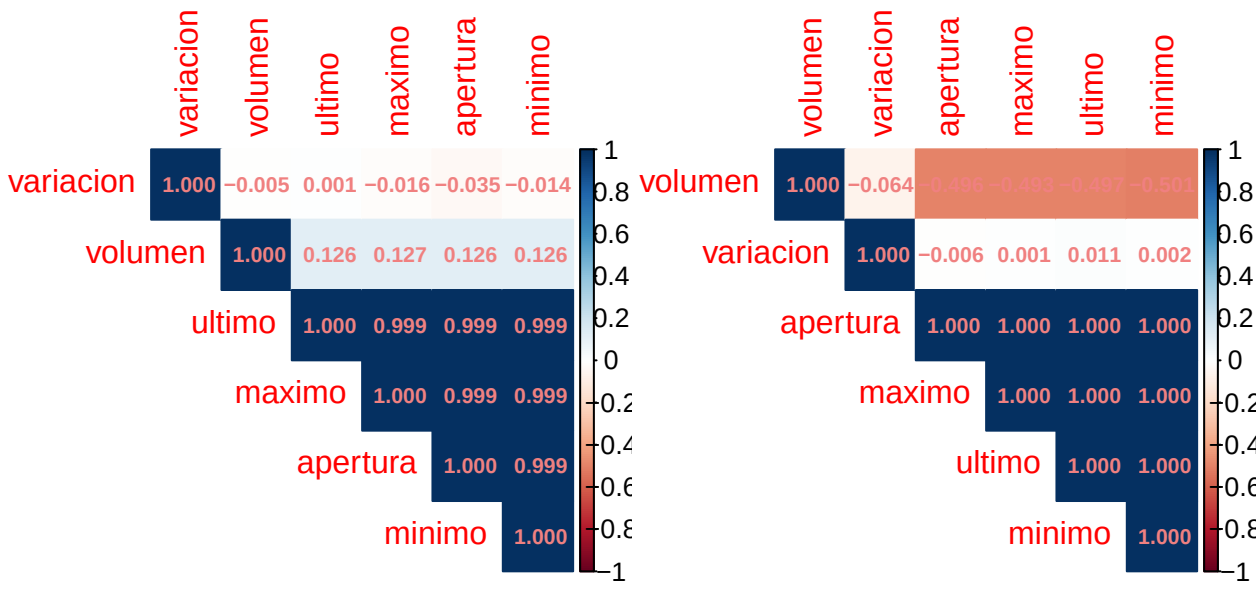
Calculamos las correlaciones individuales y conjunta.

```r
# Correlaciones individuales
correlaciones.btc.con.apple <- cor(datos.bitcoin.apple[, -1], datos.bitcoin.apple[, -1])

correlaciones.apple <- cor(datos.apple[, -1], datos.apple[, -1])

# Correlaciones cruzadas
correlaciones.cruzadas <- cor(datos.bitcoin.apple[, -1], datos.apple[, -1])
```

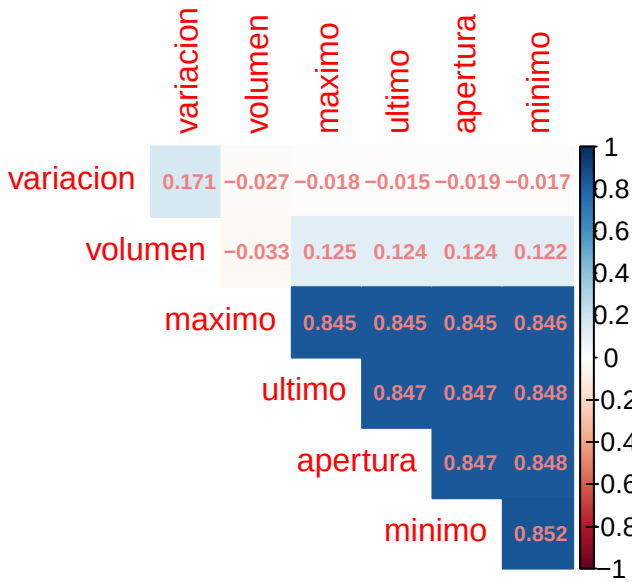
Graficamos las correlaciones.

```r
# Output de los gráficos
par(mfrow = c(1, 2))

# Correlación del bitcoin
corrplot(
  correlaciones.btc.con.apple,
  method = "color",
  type = "upper",
  order = "hclust",
  addCoef.col = "lightcoral",
  number.cex = 0.7,
  number.digits = 3
```

```
)

# Correlación de Apple
corrplot(
  correlaciones.apple,
  method = "color",
  type = "upper",
  order = "hclust",
  addCoef.col = "lightcoral",
  number.cex = 0.7,
  number.digits = 3
)
```



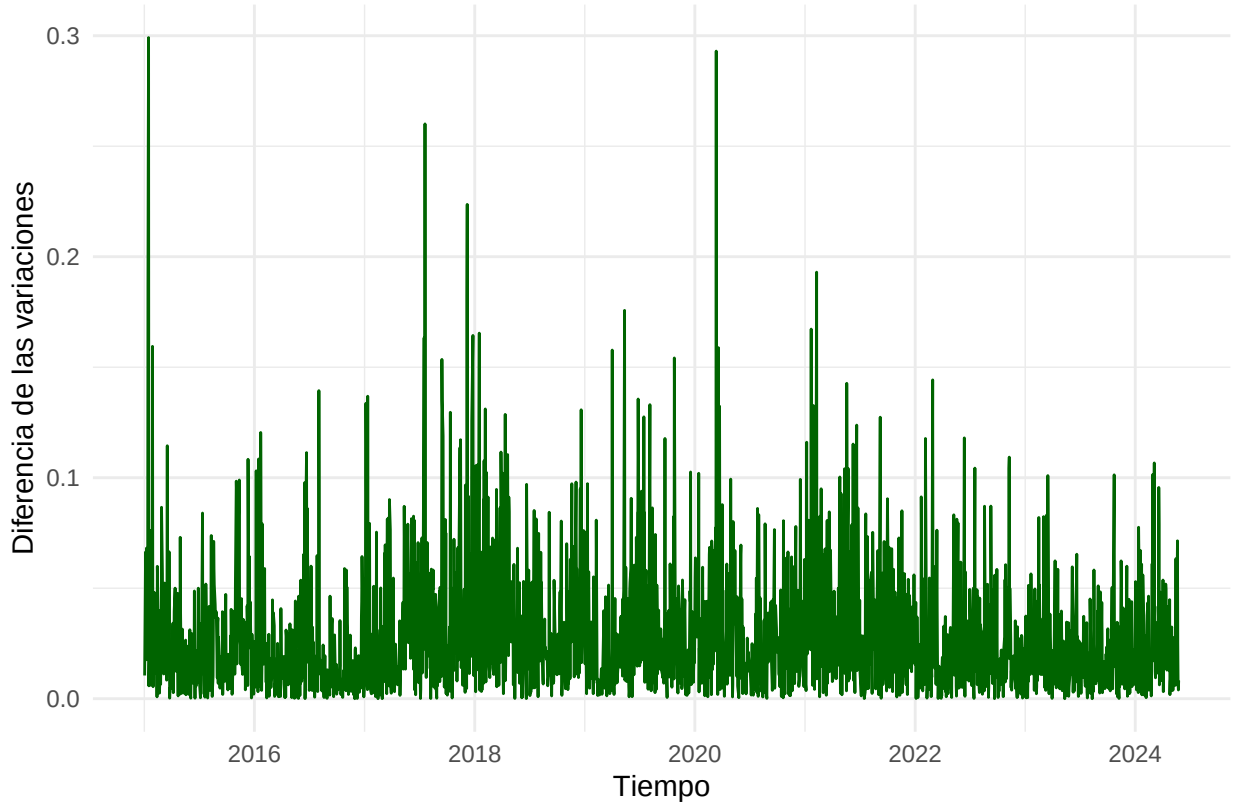

```
# Correlaciones cruzadas
corrplot(
  correlaciones.cruzadas,
  method = "color",
  type = "upper",
  order = "hclust",
  addCoef.col = "lightcoral",
  number.cex = 0.7,
  number.digits = 3
)
```



Seguidamente realizamos el gráfico de la diferencia de las variaciones de cada instrumento.

```
# Dataframe auxiliar
df.aux.btc.apple <-
  as.data.frame(cbind(
    datos.bitcoin.apple$fecha,
    datos.bitcoin.apple$variacion,
    datos.apple$variacion
  ))

# Se corrigen los formatos de columna
colnames(df.aux.btc.apple) <- c("fecha", "cambio_btc", "cambio_apple")

# Formato de fecha
df.aux.btc.apple$fecha <- as.Date(df.aux.btc.apple$fecha, format = "%d.%m.%Y")

# Diferencias de las variaciones
df.aux.btc.apple <- df.aux.btc.apple %>%
  mutate(diferencia = abs(cambio_btc - cambio_apple))

# Gráfico comparativo de varianzas diarias
ggplot(df.aux.btc.apple, aes(x = fecha)) +
  geom_line(aes(y = diferencia),
            color = "darkgreen",
            linewidth = 0.5) +
  labs(x = "Tiempo", y = "Diferencia de las variaciones",
```

```
      caption = "Fuente: Elaboración propia") +
  theme_minimal()
```



Fuente: Elaboración propia

Por último, imprimimos las tablas correspondientes.

```
# Se corrigen los nombres
colnames(resumen.btc.y.apple.)[1] <- "variable"
colnames(resumen.apple)[1] <- "variable"

# Creamos las tablas
tabla.btc.apple <- xtable(resumen.btc.y.apple.)
tabla.apple <- xtable(resumen.apple)

# Las imprimimos
print(tabla.btc.apple, type = "latex", digits = 4)
```

```
## % latex table generated in R 4.4.1 by xtable 1.8-4 package
## % Sun Nov 24 00:20:12 2024
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrrrrr}
##   \hline
##  & variable & promedio & minimo & mediana & maximo & desviacion \\
##   \hline
## 1 & apertura & 16863.24 & 164.90 & 9152.70 & 73066.70 & 18266.95 \\
```

```
##   2 & maximo & 17314.92 & 216.60 & 9296.50 & 73740.90 & 18758.77 \\
##   3 & minimo & 16374.64 & 157.30 & 8920.20 & 71338.40 & 17723.54 \\
##   4 & ultimo & 16880.65 & 164.90 & 9152.60 & 73066.30 & 18277.76 \\
##   5 & variacion & 0.00 & -0.39 & 0.00 & 0.27 & 0.04 \\
##   6 & volumen & 21341204.93 & 260.00 & 118260.00 & 4470000000.00 & 214102214.24 \\
##    \hline
## \end{tabular}
## \end{table}
```

```
print(tabla.apple, type = "latex", digits = 4)
```

```
## % latex table generated in R 4.4.1 by xtable 1.8-4 package
## % Sun Nov 24 00:20:12 2024
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrrrrr}
##   \hline
##  & variable & promedio & minimo & mediana & maximo & desviacion \\
##   \hline
## 1 & apertura & 88.05 & 22.50 & 56.20 & 198.02 & 58.13 \\
##   2 & maximo & 88.99 & 22.92 & 56.61 & 199.62 & 58.74 \\
##   3 & minimo & 87.16 & 22.37 & 55.33 & 197.00 & 57.55 \\
##   4 & ultimo & 88.12 & 22.58 & 55.99 & 198.11 & 58.17 \\
##   5 & variacion & 0.00 & -0.13 & 0.00 & 0.12 & 0.02 \\
##   6 & volumen & 121068734.59 & 24048300.00 & 103718400.00 & 648825200.00 & 68055983.66 \\
##    \hline
## \end{tabular}
## \end{table}
```

VaR MonteCarlo para Bitcoin

Vamos a descargar el precio de cotización de Bitcoin, utilizando función Ad, podemos cargar los datos desde la fecha máxima hasta el día del análisis, con fuente Yahoo Finance, esta función genera el vector en un objeto “xts”, por lo que los convertiremos en un vector númerico.

```
library(quantmod)
```

```
## Warning: package 'quantmod' was built under R version 4.4.2
```

```
## Cargando paquete requerido: xts
```

```
## Warning: package 'xts' was built under R version 4.4.2
```

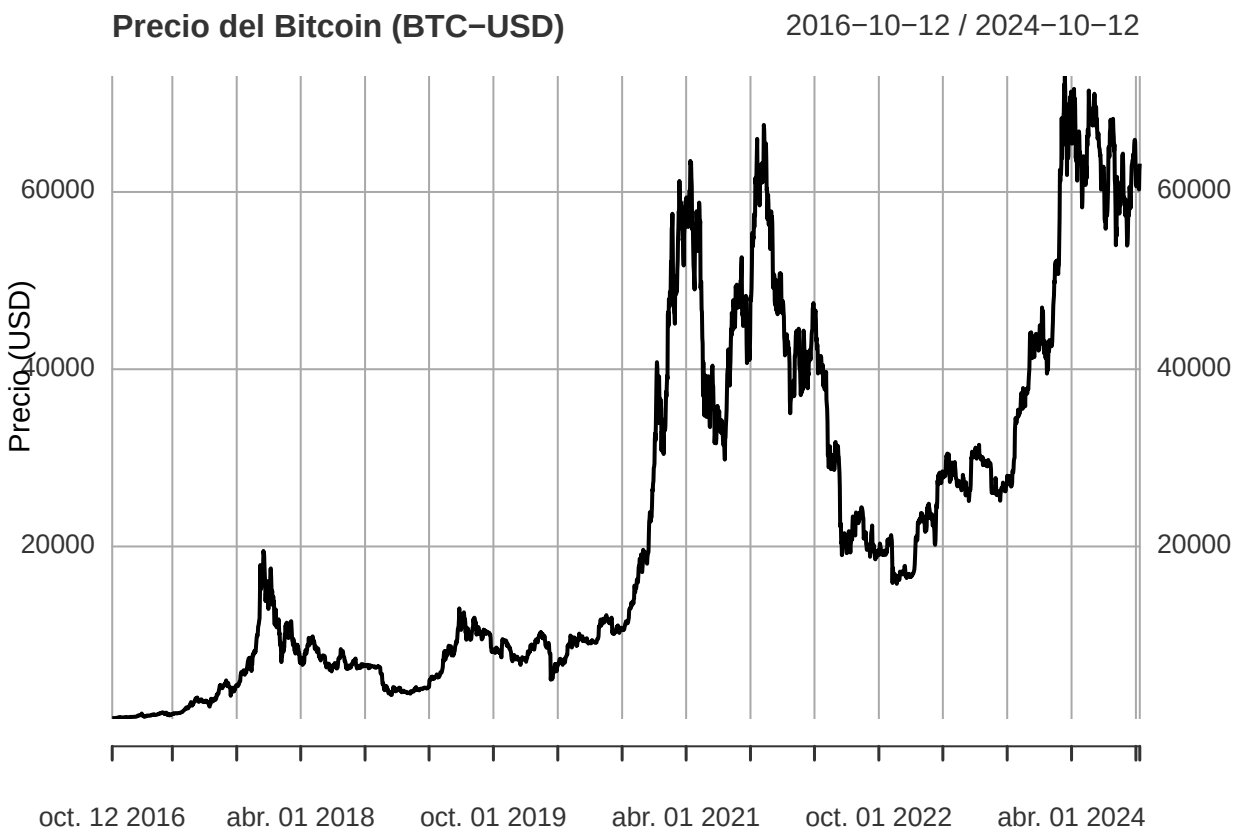
```
## Cargando paquete requerido: zoo
```

```
## Warning: package 'zoo' was built under R version 4.4.2
```

```
##
## Adjuntando el paquete: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## ########################## Warning from 'xts' package ##########################
## #                                                                              #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                            #
## #                                                                              #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop           #
## # dplyr from breaking base R's lag() function.                                #
## #                                                                              #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                              #
## ################################################################################

##
## Adjuntando el paquete: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

## Cargando paquete requerido: TTR

## Warning: package 'TTR' was built under R version 4.4.2

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
# Utilizamos las fechas que nos interesa para poder hacer una simulación
maxDate <- "2024-10-12"
minDate <- "2016-10-12"

# Símbolo de Bitcoin en Yahoo Finance
tickBTC <- "BTC-USD"

# Obtener datos de Bitcoin
p.bitcoin <- Ad(getSymbols(tickBTC, auto.assign = FALSE, from = minDate, to = maxDate))

# Graficar precios ajustados
plot(p.bitcoin, main = "Precio del Bitcoin (BTC-USD)", ylab = "Precio (USD)", xlab = "Fecha")
```



Una vez calculado el precio de las acciones, podemos ver como ha tenido una alta volatilidad, más en el periodo de del 2021 al 2024, ha presentado un comportamiento errático.

```
p.bitcoin <- as.numeric(p.bitcoin)
```

Ahora daremos paso al cálculo de los retornos, los cuales son calculados mediante la fórmula.

$$r_t = Ln(\frac{P_t}{P_{t-1}})$$

De donde tenemos que P_t es el precio en el tiempo t.

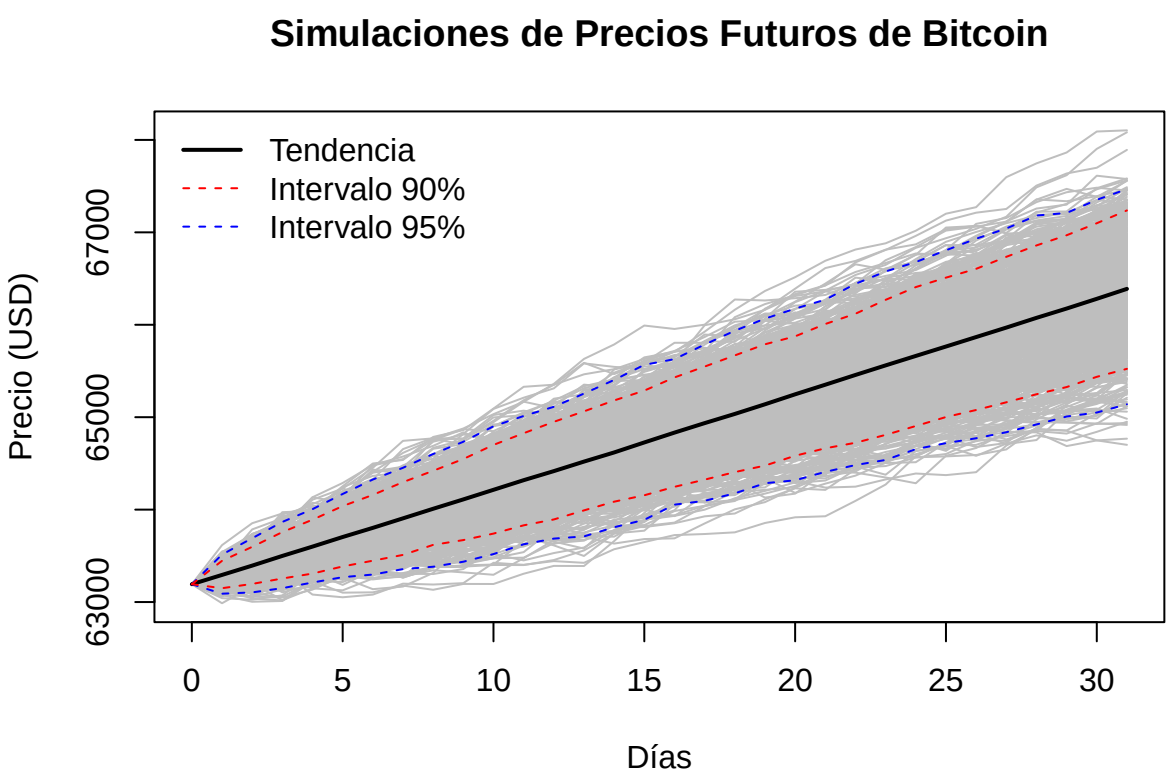
```
# Calculamos los retornos y lo guardamos en un data frame.
retorno_bitcoin <- data.frame("r.bitcoin"=diff(log(p.bitcoin)))

# Calculamos la media de los retornos
media_r<-colMeans(retorno_bitcoin)

# Visualizamos la media de los retornos
media_r
```

```
##   r.bitcoin
## 0.001573733
```

```r
# Calculamos la varianza de los retornos
varianza_r <- var(retorno_bitcoin)

# Visualizamos la varianza de los retornos
varianza_r
```

```
##             r.bitcoin
## r.bitcoin 0.001411426
```

```r
library(quantmod)

# Definimos los parámetros de la simulación
n_sims <- 1000  # Número de simulaciones
n_periods <- 31 # Número de días a proyectar
start_date <- "2024-10-12"  # Fecha inicial para el precio inicial

# Precio inicial del 12-Octubre-2024
S0 <- as.numeric(Ad(getSymbols("BTC-USD", auto.assign = FALSE, from = start_date, to = start_date)))

# Creamos una matriz para almacenar las simulaciones
precios_bitcoin <- matrix(, n_periods + 1, n_sims)
precios_bitcoin[1, ] <- S0

# Simulación de precios usando GBM
for (i in 1:n_sims) {
  for (j in 2:(n_periods + 1)) {
    precios_bitcoin[j, i] <- precios_bitcoin[j - 1, i] * exp((media_r - (varianza_r^2) / 2) + varianza_
  }
}

# Generar el vector de días
dias <- seq(0, n_periods)

# Calculamos la tendencia (drift) y los intervalos de confianza
Tendencia <- apply(precios_bitcoin, 1, mean)
IC <- t(apply(precios_bitcoin, 1, quantile, probs = c(0.01, 0.05, 0.95, 0.99)))

# Graficamos las simulaciones
matplot(dias, precios_bitcoin, type = "l", col = "gray", lty = 1,
        main = "Simulaciones de Precios Futuros de Bitcoin",
        ylab = "Precio (USD)", xlab = "Días")
lines(dias, Tendencia, col = "black", lwd = 2)
lines(dias, IC[, 1], col = "blue", lwd = 1, lty = 2)
lines(dias, IC[, 4], col = "blue", lwd = 1, lty = 2)
lines(dias, IC[, 2], col = "red", lwd = 1, lty = 2)
lines(dias, IC[, 3], col = "red", lwd = 1, lty = 2)
legend("topleft",
       legend = c("Tendencia", "Intervalo 90%", "Intervalo 95%"),
       lwd = c(2, 1, 1),
       col = c("black", "red", "blue"),
       lty = c(1, 2, 2),
       bty = "n")
```



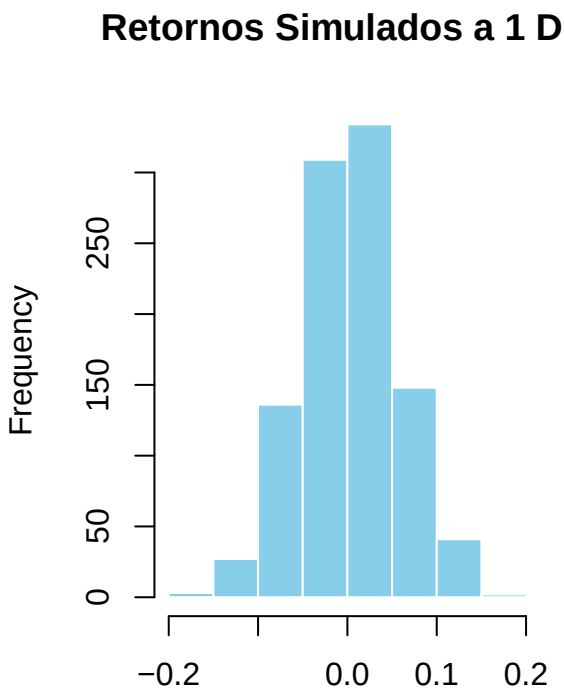

```r
library(quantmod)
library(knitr)

# Parámetros iniciales
n_sims <- 1000
n_periods <- 2

# Fecha específica para el precio inicial
start_date <- "2024-10-12"

# Obtener el precio ajustado del Bitcoin para el 12 de octubre de 2024
S0 <- as.numeric(Ad(getSymbols("BTC-USD", auto.assign = FALSE, from = start_date, to = start_date)))

media_btc <- media_r
volatilidad_btc <- sqrt(varianza_r)

# Inicializamos los vectores para precios simulados y retornos
precio_bitcoin <- numeric(n_periods + 1)
precio_bitcoin[1] <- S0
retornos_btc <- numeric(n_sims)

# Simulación de Monte Carlo para el Bitcoin
for (j in 1:n_sims) {
  for (i in 2:(n_periods + 1)) {
    Z <- rnorm(1)
    precio_bitcoin[i] <- precio_bitcoin[i - 1] * exp((media_btc - (volatilidad_btc^2) / 2) + volatilidad
```

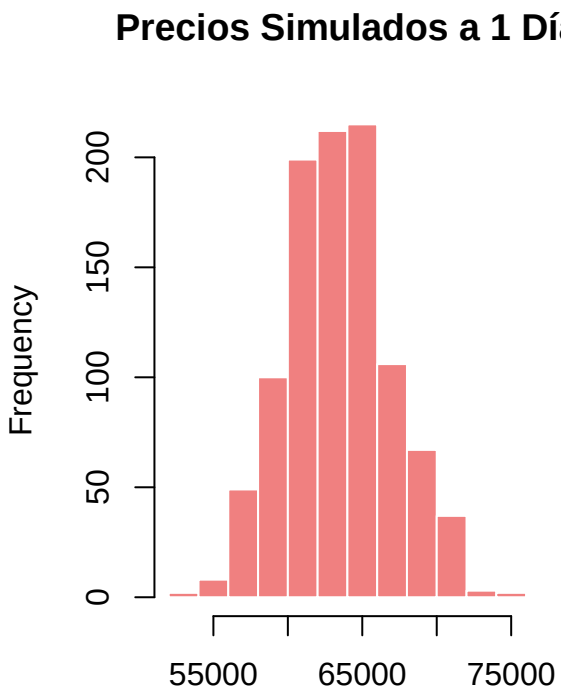
```
  }
  retornos_btc[j] <- log(precio_bitcoin[n_periods + 1] / S0)
}

# Calcular el VaR al 95% de confianza
VaR_MC_Bitcoin <- quantile(retornos_btc, 0.05)
kable(VaR_MC_Bitcoin, caption = "VaR Monte Carlo para Bitcoin")
```

Table 1: VaR Monte Carlo para Bitcoin

	x
5%	-0.082036

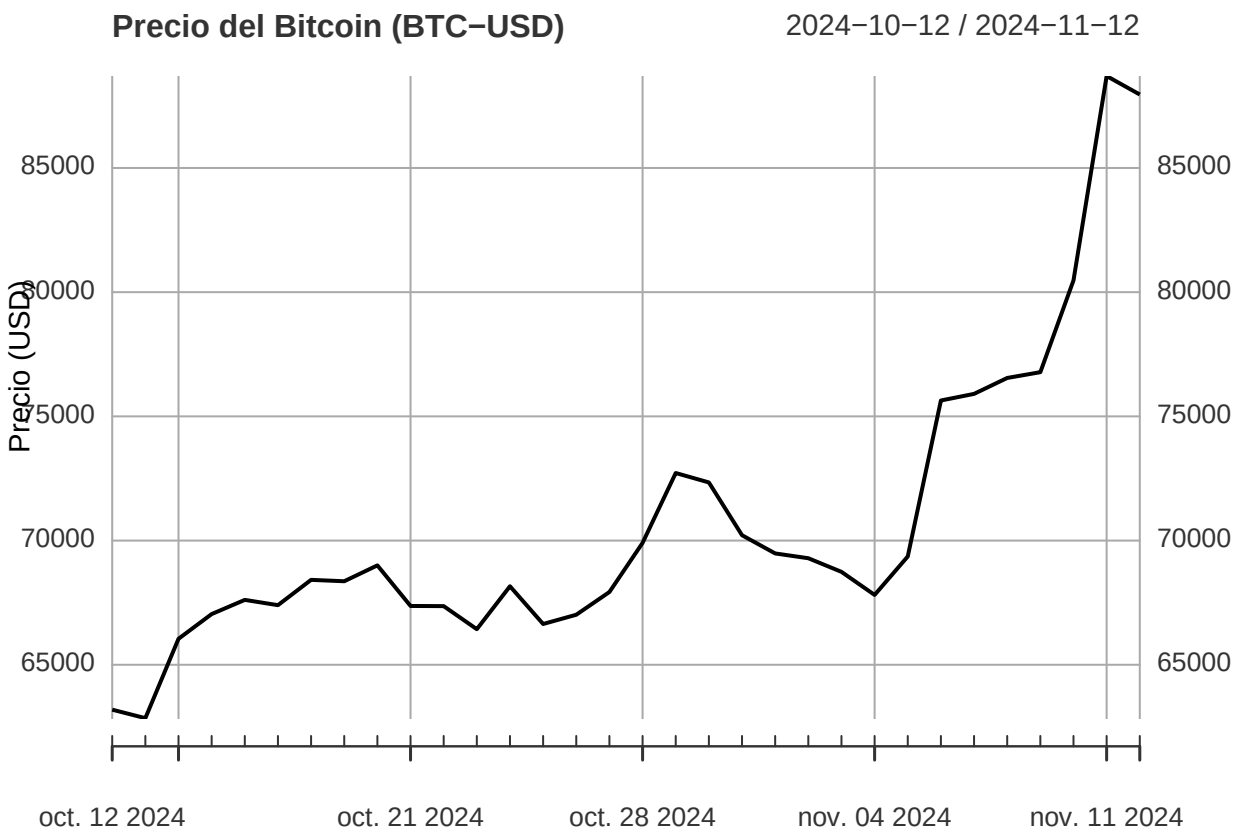
```
media_btc <- mean(retorno_bitcoin$r.bitcoin) # Media de los retornos de Bitcoin
varianza_btc <- var(retorno_bitcoin$r.bitcoin) # Varianza de los retornos de Bitcoin
volatilidad_btc <- sqrt(varianza_btc) # Volatilidad de los retornos de Bitcoin

# Inicializamos precios simulados
n_sims <- 1000
n_periods <- 2
precios_simulados <- matrix(, n_periods + 1, n_sims)
precios_simulados[1, ] <- S0

# Simulación de Monte Carlo
for (j in 1:n_sims) {
  for (i in 2:(n_periods + 1)) {
    Z <- rnorm(1)
    precios_simulados[i, j] <- precios_simulados[i - 1, j] *
      exp((media_btc - (volatilidad_btc^2) / 2) + volatilidad_btc * Z)
  }
}

# Calculamos los retornos simulados
retornos_simulados <- log(precios_simulados[n_periods + 1, ] / S0)


par(mfrow = c(1, 2))
hist(retornos_simulados,
     main = "Retornos Simulados a 1 Día",
     xlab = "Retornos",
     col = "skyblue",
     border = "white")
hist(precios_simulados[n_periods + 1, ],
     main = "Precios Simulados a 1 Día",
     xlab = "Precios",
     col = "lightcoral",
     border = "white")
```





A continuación se visualiza como se ve el gráfico de las acciones para las fechas reales, las cuales se habían intentado simular.

```r
library(quantmod)

# Utilizamos las fechas que nos interesa para poder hacer una simulación
maxDate <- "2024-11-12"
minDate <- "2024-10-12"

# Símbolo de Bitcoin en Yahoo Finance
tickBTC <- "BTC-USD"

# Obtener datos de Bitcoin
p.bitcoin <- Ad(getSymbols(tickBTC, auto.assign = FALSE, from = minDate, to = maxDate))

# Graficar precios ajustados
plot(p.bitcoin, main = "Precio del Bitcoin (BTC-USD)", ylab = "Precio (USD)", xlab = "Fecha")
```




```r
# Crear rango de días con fechas reales
fechas_simuladas <- seq(as.Date("2024-10-12"), by = "day", length.out = n_periods + 1)
fechas_reales <- index(p.bitcoin)

# Crear índice para etiquetar cada 5 días y el 5 de noviembre
indices_etiquetas <- c(seq(1, length(fechas_simuladas), by = 5), which(fechas_simuladas == "2024-11-05"))
indices_etiquetas <- sort(unique(indices_etiquetas))

# Graficar solo las simulaciones (sin las líneas grises)
plot(dias, Tendencia, type = "l", col = "black", lwd = 2, lty = 1,
     ylim = range(c(precios_bitcoin, p.bitcoin)),  # Rangos en la misma escala
     main = "Comparación de Simulaciones de Precios con Precios Reales",
     ylab = "Precio (USD)", xlab = "Fecha", xaxt = "n")

lines(dias, IC[, 1], col = "blue", lwd = 1, lty = 2)
lines(dias, IC[, 4], col = "blue", lwd = 1, lty = 2)
lines(dias, IC[, 2], col = "red", lwd = 1, lty = 2)
lines(dias, IC[, 3], col = "red", lwd = 1, lty = 2)

# Agregar precios reales
lines(seq_along(fechas_reales), as.numeric(p.bitcoin), col = "green", lwd = 2, lty = 1)

# Agregar línea vertical para el 5 de noviembre
dia_vertical <- which(fechas_reales == "2024-11-05")
abline(v = dia_vertical, col = "purple", lwd = 2, lty = 2)
axis(1, at = indices_etiquetas, labels = FALSE)
```

```r
text(x = indices_etiquetas,
     y = par("usr")[3] - diff(par("usr")[3:4]) * 0.05,
     labels = format(fechas_simuladas[indices_etiquetas], "%d-%b"),
     srt = 45, adj = 1, xpd = TRUE, cex = 0.8)


legend("topleft",
       legend = c("Tendencia (Simulaciones)", "Intervalo 90%", "Intervalo 95%", "Precios Reales", "5 Nov
       lwd = c(2, 1, 1, 2, 2),
       col = c("black", "red", "blue", "green", "purple"),
       lty = c(1, 2, 2, 1, 2),
       bty = "n")
```