

Pronostico MC

2025-01-16

En este caso, el objetivo es poder realizar pronosticos mediante el uso de simulaciones de Monte Carlo. Para ello, se utilizará un conjunto de datos históricos de precios de un activo financiero, con el fin de estimar posibles trayectorias futuras del precio.

```
# Cargar paquetes necesarios
library(readr)
library(xts)

## Cargando paquete requerido: zoo

##
## Adjuntando el paquete: 'zoo'

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

library(ggplot2)

NU_Historical_Data <- read_csv("NU Historical Data.csv")

## Rows: 718 Columns: 7

## -- Column specification -----
## Delimiter: ","
## chr (3): Date, Vol., Change %
## dbl (4): Price, Open, High, Low
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

data <- data.frame(Date = as.Date(NU_Historical_Data$Date, format = "%m/%d/%Y"),
                   Price = NU_Historical_Data$Price)
```

En este caso, al igual que en el pronostico ARIMA se ajusta una serie con xts y se recorta la serie a modo de conjunto de entrenamiento. En este caso, se calcularon los rendimientos diarios de forma logartimica

```
# Convertir a serie temporal
serie <- xts(data$Price, order.by = data$Date)
serie_recortada <- serie["2021-12-10/2024-10-08"]

# Calcular rendimientos diarios
rendimientos <- diff(log(serie_recortada))
```

Ahora, en este caso se estiman la media y la desviación estandar de los rendimientos, los cuales se usarán como parametros de la simulación de Monte Carlo. Además, se establecen los parámetros de simulación, como el número de simulaciones y el horizonte de pronóstico

```
mu <- mean(rendimientos, na.rm = TRUE)
sigma <- sd(rendimientos, na.rm = TRUE)

# Parámetros de simulación
n_sim <- 100000 # Número de simulaciones
horizonte <- 8 # Días de pronóstico
precio_inicial <- as.numeric(last(serie_recortada)) # Último precio disponible
```

En este caso, se generan las simulaciones de Monte Carlo para el precio futuro del activo financiero, utilizando la fórmula de precios geométricos Brownianos, el cual se expresa de la siguiente manera.

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

Se generan números aleatorios normales para cada día del horizonte de pronóstico y se calculan los precios simulados.

```
# Simular trayectorias futuras
set.seed(123) # Asegurar reproducibilidad
simulaciones <- matrix(NA, nrow = horizonte, ncol = n_sim)

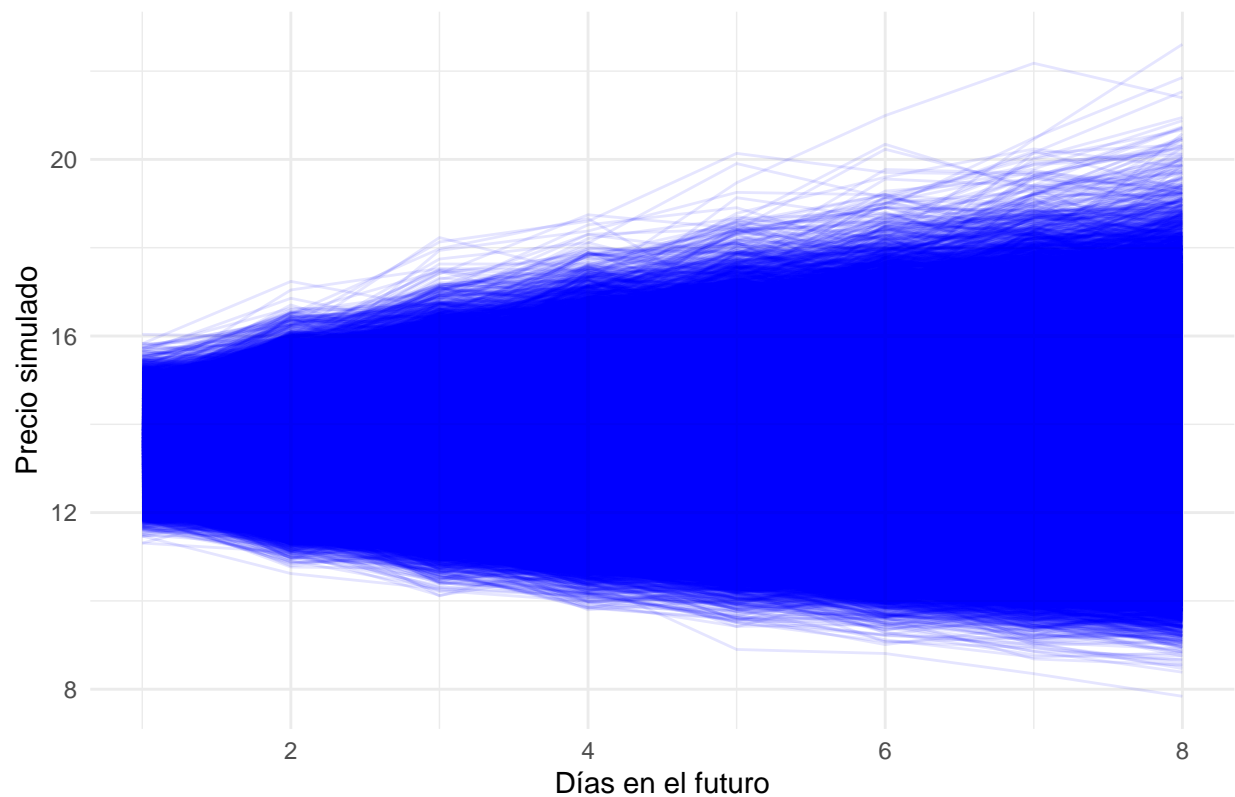
for (i in 1:n_sim) {
  z <- rnorm(horizonte) # Generar números aleatorios normales
  simulaciones[, i] <- precio_inicial * exp(cumsum((mu - 0.5 * sigma^2) + sigma * z))
}
```

Ahora, se guardan las simulaciones en un data frame y se grafican las trayectorias simuladas para el precio futuro del activo financiero

```
sim_data <- data.frame(Day = 1:horizonte, simulaciones)
sim_data_long <- reshape2::melt(sim_data, id.vars = "Day")

ggplot(sim_data_long, aes(x = Day, y = value, group = variable)) +
  geom_line(alpha = 0.1, color = "blue") +
  labs(title = "Simulaciones Monte Carlo para el precio futuro",
       x = "Días en el futuro",
       y = "Precio simulado") +
  theme_minimal()
```

Simulaciones Monte Carlo para el precio futuro



En este caso, se calculan los intervalos de confianza del para las simulaciones de Monte Carlo y se grafican junto con los datos reales para comparar

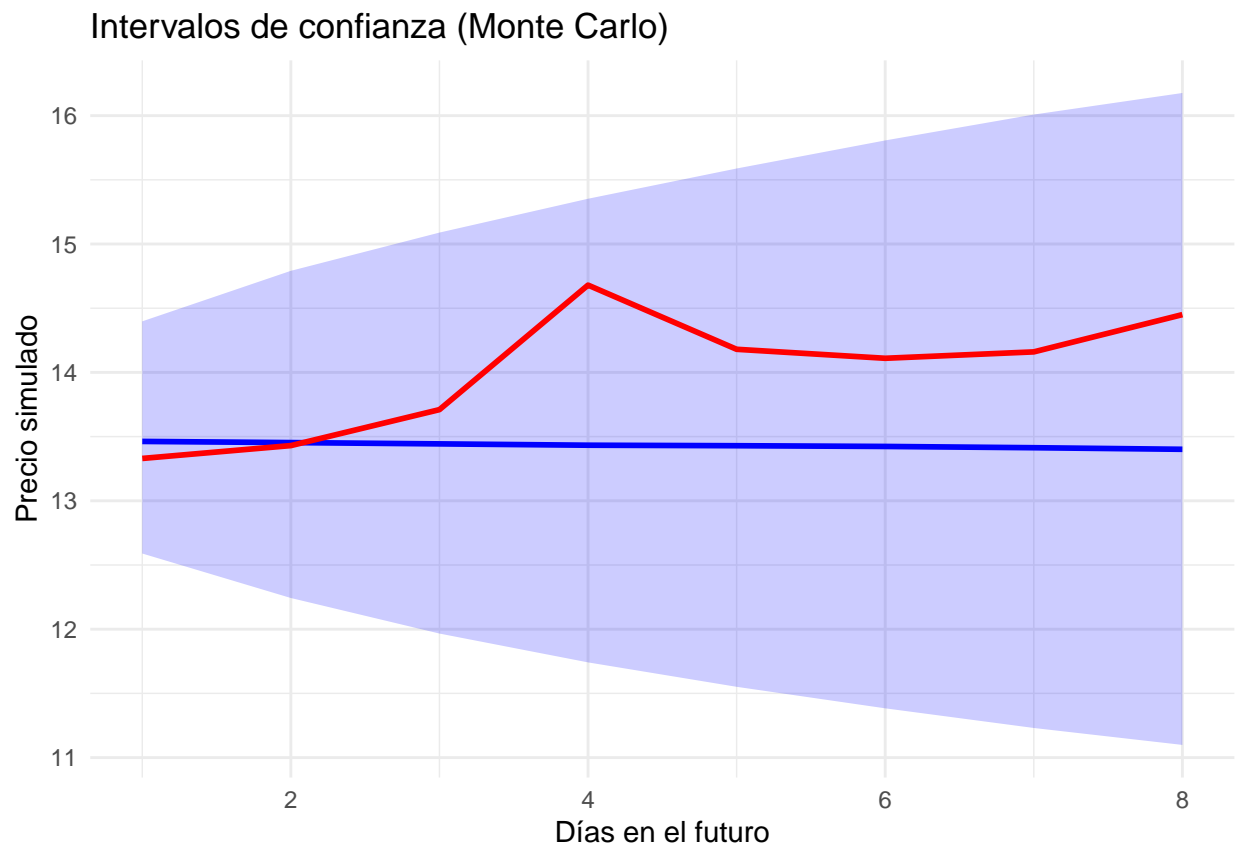
```
percentiles <- apply(simulaciones, 1, quantile, probs = c(0.05, 0.5, 0.95))
ic_data <- data.frame(
  Day = 1:horizonte,
  P5 = percentiles[1, ],
  P50 = percentiles[2, ],
  P95 = percentiles[3, ]
)

datos_reales = serie["2024-10-09/2024-10-18"]

ggplot(ic_data, aes(x = Day)) +
  geom_ribbon(aes(ymin = P5, ymax = P95), fill = "blue", alpha = 0.2) +
  geom_line(aes(y = P50), color = "blue", size = 1) +
  geom_line(aes(y = ((datos_reales))), color = "red", size = 1) +
  labs(title = "Intervalos de confianza (Monte Carlo)",
       x = "Días en el futuro",
       y = "Precio simulado") +
  theme_minimal()
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
```

```
## generated.
```



Finalmente, se calculan los errores de pronóstico para cada simulación y se promedian para obtener una medida general de precisión

```
mape <- function(sim, real) mean(abs((sim - real) / real))
mse <- function(sim, real) mean((sim - real)^2)
mae <- function(sim, real) mean(abs(sim - real))

# Calcular errores para cada simulacion
errores_mape <- apply(simulaciones, 2, function(sim) mape(sim, datos_reales))
errores_mse <- apply(simulaciones, 2, function(sim) mse(sim, datos_reales))
errores_mae <- apply(simulaciones, 2, function(sim) mae(sim, datos_reales))

# Promediar errores por cada metrica
promedio_mape <- mean(errores_mape)
promedio_mse <- mean(errores_mse)
promedio_mae <- mean(errores_mae)

# Resultados
list(MAPE = promedio_mape, MSE = promedio_mse, MAE = promedio_mae)
```

```
## $MAPE
## [1] 0.07425932
##
```

```
## $MSE
## [1] 1.832041
##
## $MAE
## [1] 1.050096
```

En este caso no es tan importante basarse en la media de los pronósticos sino en los intervalos de confianza, ya que estos nos permiten tener una idea de la variabilidad de los precios futuros. En este caso, se observa que los intervalos de confianza son bastante amplios, lo cual refleja la incertidumbre asociada con los pronósticos de precios financieros