

Arima

Corona López José Luis

2025-02-27

Elección de variable

```
library(quantmod)
library(dynlm)
library(nlWaldTest)
library(zoo)
library(pdfetch)
library(lmtest)
library(broom)
library(car)
library(sandwich)
library(knitr)
library(forecast)
library(readxl)
library(systemfit)
library(AER)
library(Matrix)
library(vars)
library(plm)
library(tseries)
library(readxl)
library(ggplot2)
library(dplyr)
library(lubridate)

# Definir el ticker de NVIDIA y el rango de fechas
ticker <- "NVDA"
start_date <- "2015-01-01"
end_date <- Sys.Date()

# Descargar datos de Yahoo Finance
getSymbols(ticker, src = "yahoo", from = start_date, to = end_date, auto.assign = TRUE)

## [1] "NVDA"

nvda_data <- data.frame(Date = index(NVDA), coredata(NVDA))

nvda_monthly <- nvda_data %>%
  mutate(Month = floor_date(Date, "month")) %>%
  group_by(Month) %>%
  summarize(Close = last(NVDA.Close))

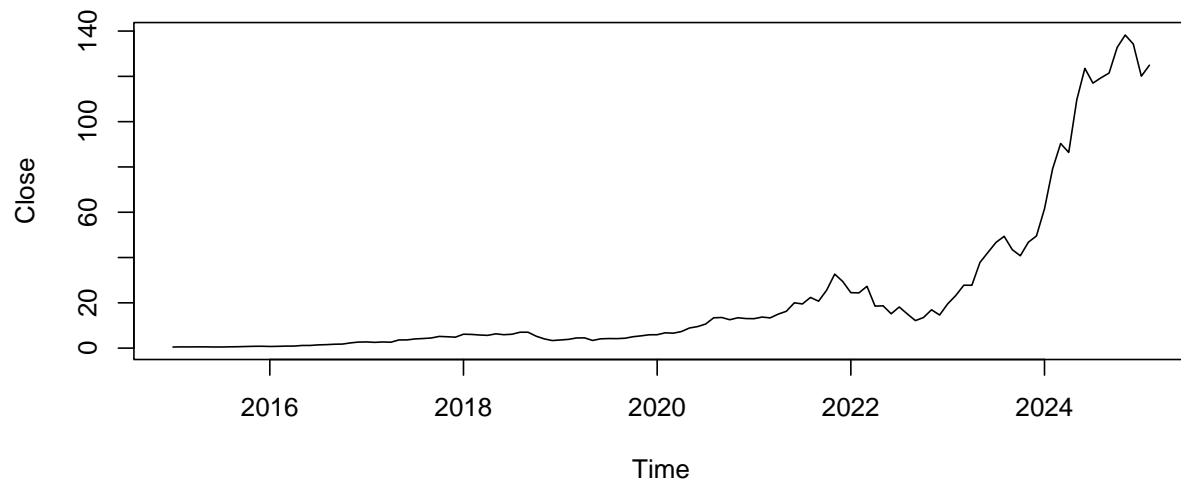
nvda.ts <- ts(nvda_monthly[,2], start = c(2015, 01), frequency = 12)
```

Justificación

Nvidia es una de las empresas más importantes tecnológicas, entonces precio de cierre representa el último precio al que se negoció una acción durante un día de negociación, por lo tanto junta todos los movimientos que existieron en el mercado durante el día.

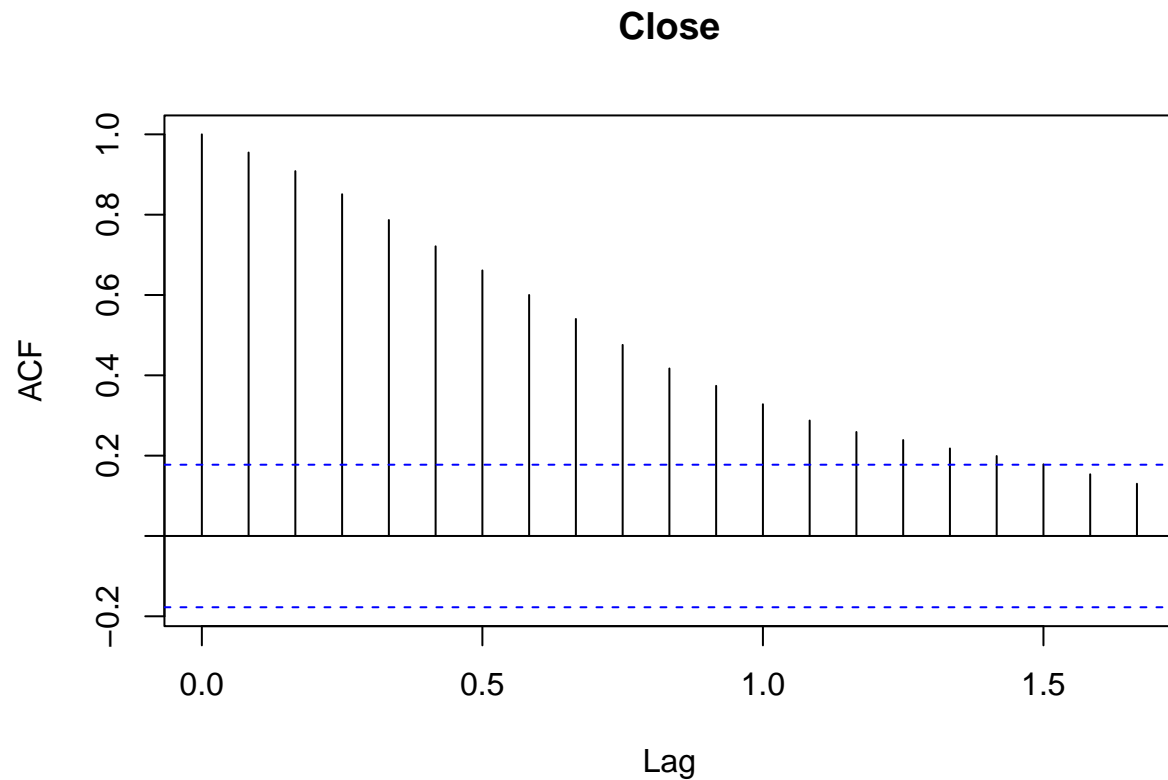
Estos datos nos permiten observar el comportamiento del precio, captar tendencias y detectar ciclos, para poder dar un pronóstico sobre el movimiento que tendrá, así se tomará mejores decisiones frente a posibles riesgos que se presenten.

```
plot(nvda.ts)
```



Observamos que el precio ha subido en los dos últimos años, de 2022 hacia atrás se mantuvo más o menos constantes, sin embargo ha crecido bastante, en los últimos meses ha tenido bajas significativas, esto responde a cierto tipo de eventos, podemos decir que esta caída fue de importante debido a que no se observa en ningún otro momento algo así. La serie observada no es estacionaria, es claro por los incrementos que se presentan.

```
acf(nvda.ts)
```



*# Notamos que no existe estacionariedad, debido a que varios de los rezagos
pasan las bandas de confianza al 95%*

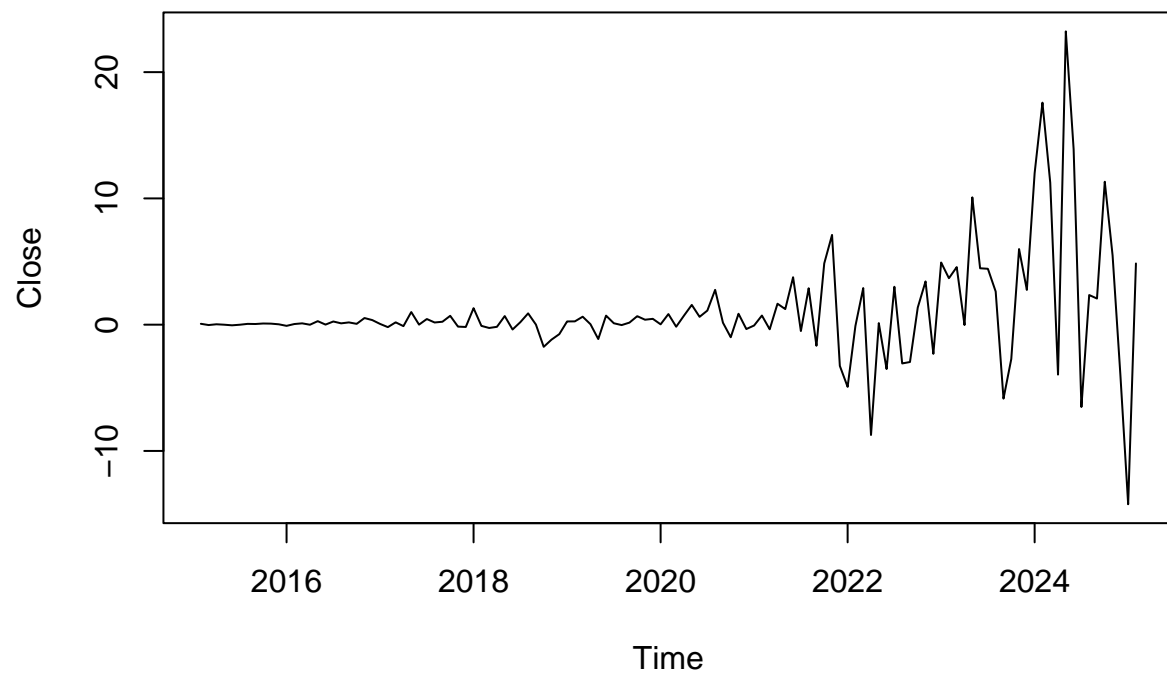
```
adf.test(nvda.ts)
```

```
##
## Augmented Dickey-Fuller Test
##
## data:  nvda.ts
## Dickey-Fuller = -1.2161, Lag order = 4, p-value = 0.9008
## alternative hypothesis: stationary
```

*# Como p-value = 0.8788 > 0.05, no rechazamos Hipótesis nula
es decir, la serie no es estacionaria*

Probando con la diferencia

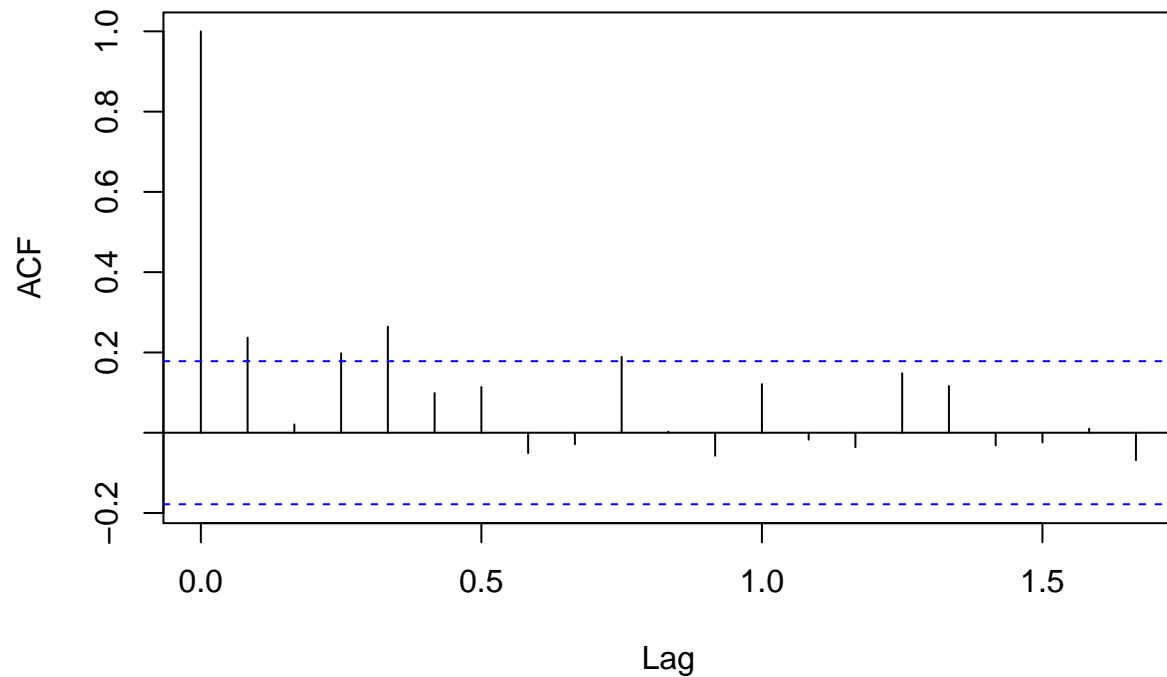
```
Dnvda <- diff(nvda.ts)
plot(Dnvda)
```



*# En algunos puntos parece ser estacionaria, sin embargo,
como notamos en el primer gráfico, hay mucha más volatilidad en los últimos años*

`acf(Dnvda)`

Close



Notamos que los rezagos no rebasan las bandas de confianza, a excepción de unos cuantos

```
adf.test(Dnvda, k = 4)
```

```
##  
## Augmented Dickey-Fuller Test  
##  
## data: Dnvda  
## Dickey-Fuller = -3.4193, Lag order = 4, p-value = 0.05467  
## alternative hypothesis: stationary
```

*# En este caso p-value = 0.05467 > 0.05, por lo que no rechazamos Hipótesis nula
la serie con una diferencia aún no es estacionaria*

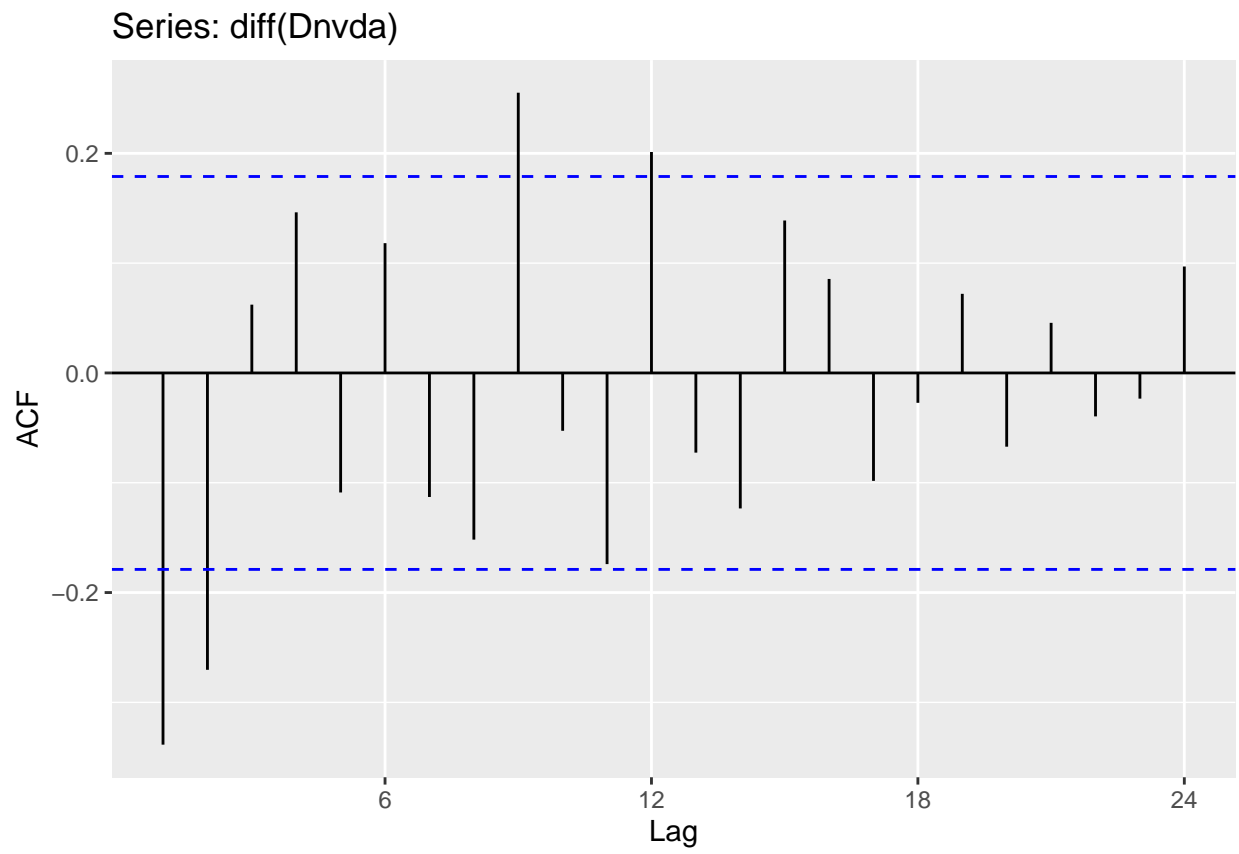
```
ndiffs(nvda.ts)
```

```
## [1] 2
```

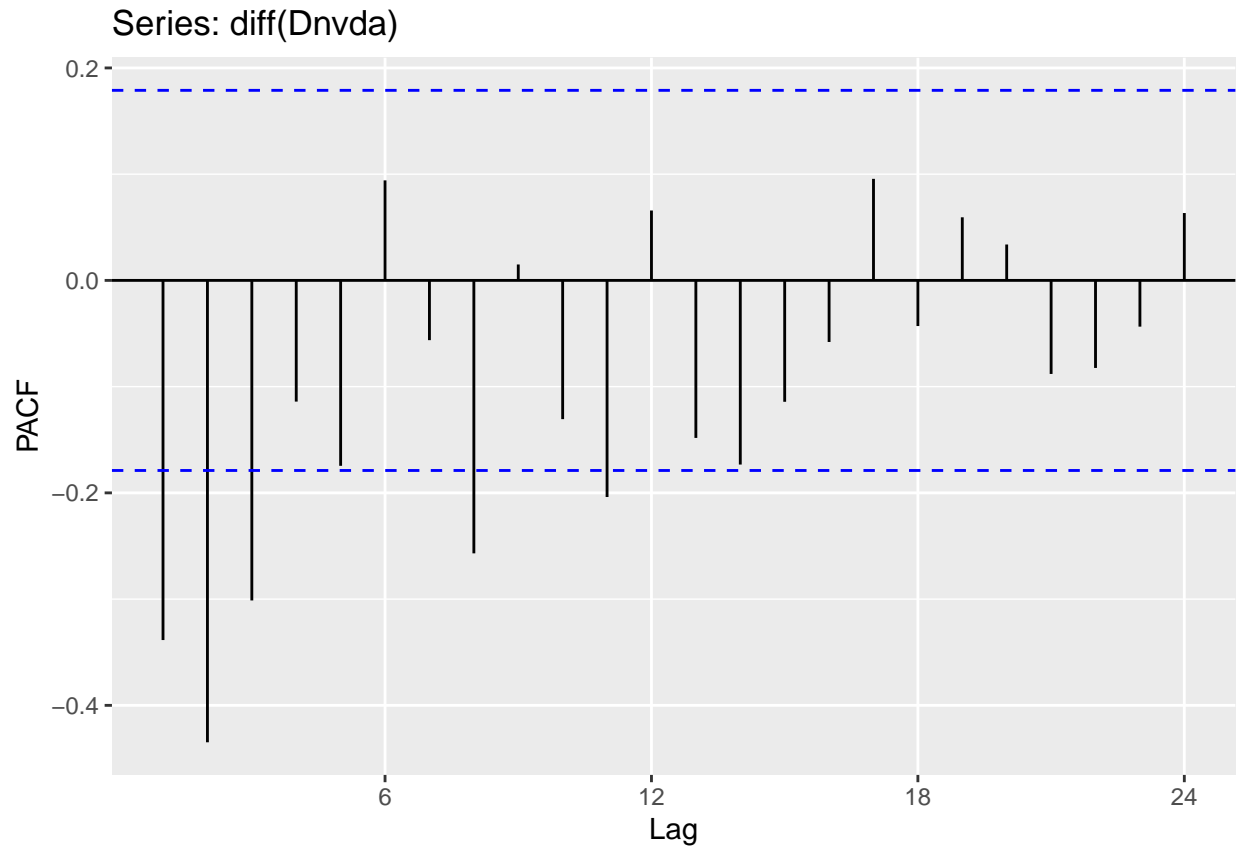
Se necesitan diferenciar la serie 2 veces para hacerla estacionaria

Era de esperarse un resultado de este tipo, por la naturaleza de la serie, tuvo un crecimiento casi exponencial en los últimos dos años, por lo que una diferencia no bastaría para poder hacerla estacionaria, aunque la diferencia es muy poca, o sea 0.00467, entonces en los modelos siguientes tomaremos 1 y 2 diferencias para evaluar el mejor.

```
ggAcf(diff(Dnvda))
```



```
ggPacf(diff(Dnvda))
```



Modelos

```
# Modelo con 1 diferencia
fit1 <- auto.arima(nvda.ts, d = 1)
fit1

## Series: nvda.ts
## ARIMA(0,1,1)(0,0,1)[12] with drift
##
## Coefficients:
##          ma1      sma1    drift
##          0.3110  0.3706  1.0569
## s.e.  0.0884  0.1346  0.6396
##
## sigma^2 = 16.61:  log likelihood = -341.13
## AIC=690.26   AICc=690.61   BIC=701.44

# Modelo con 2 diferencias
fit2 <- auto.arima(nvda.ts, d = 2)
fit2

## Series: nvda.ts
## ARIMA(0,2,2)(0,0,1)[12]
##
## Coefficients:
##          ma1      ma2      sma1
```

```
##      -0.6682 -0.2843  0.3351
## s.e.   0.0926   0.0937  0.1440
##
## sigma^2 = 16.91: log likelihood = -340.05
## AIC=688.1   AICc=688.45   BIC=699.25
```

```
# Modelo sin estacionalidad
```

```
fit3 <- auto.arima(nvda.ts, seasonal = F)
fit3
```

```
## Series: nvda.ts
## ARIMA(4,2,1)
##
## Coefficients:
##      ar1      ar2      ar3      ar4      ma1
##      0.2182 -0.1035  0.1801  0.2018 -0.9805
## s.e.  0.0956   0.1041  0.1051  0.1036   0.0397
##
## sigma^2 = 16.99: log likelihood = -338.87
## AIC=689.74   AICc=690.48   BIC=706.46
```

```
# Modelo sin estacionalidad y forzando a la función
```

```
fit4 <- auto.arima(nvda.ts, , seasonal = F, stepwise = F, approximation = F)
fit4
```

```
## Series: nvda.ts
## ARIMA(0,2,5)
##
## Coefficients:
##      ma1      ma2      ma3      ma4      ma5
##      -0.7433 -0.2350  0.1360  0.3181 -0.4302
## s.e.   0.0834   0.1041  0.1269  0.1741   0.1328
##
## sigma^2 = 16.16: log likelihood = -336.06
## AIC=684.12   AICc=684.86   BIC=700.84
```

```
# Modelo con estacionalidad y forzando a la función
```

```
fit5 <- auto.arima(nvda.ts, , seasonal = T, stepwise = F, approximation = F)
fit5
```

```
## Series: nvda.ts
## ARIMA(0,2,5)
##
## Coefficients:
##      ma1      ma2      ma3      ma4      ma5
##      -0.7433 -0.2350  0.1360  0.3181 -0.4302
## s.e.   0.0834   0.1041  0.1269  0.1741   0.1328
##
## sigma^2 = 16.16: log likelihood = -336.06
## AIC=684.12   AICc=684.86   BIC=700.84
```

```
# Modelo quitandole la estacionalidad manualmente
```

```
des <- decompose(nvda.ts, type = "m")
desnvda <- nvda.ts - des$seasonal
fit6 <- Arima(nvda.ts, order = c(3,2,5))
fit6
```

```
## Series: nvda.ts
```



```
## ARIMA(3,2,5)
##
## Coefficients:
##          ar1      ar2      ar3      ma1      ma2      ma3      ma4      ma5
##      -0.2085  0.0953  0.4512 -0.5000 -0.5208 -0.3245  0.6916 -0.3233
## s.e.   0.1895  0.2026  0.1826   0.2014   0.1666   0.1712  0.1487  0.1630
##
## sigma^2 = 15.64: log likelihood = -333.25
## AIC=684.49   AICc=686.13   BIC=709.58
```

Revisando los criterios de información tenemos:

```
comparaciones <- data.frame(
  Model = c("fit1", "fit2", "fit3", "fit4", "fit5", "fit6"),
  AIC = c(AIC(fit1), AIC(fit2), AIC(fit3), AIC(fit4), AIC(fit5), AIC(fit6)),
  BIC = c(BIC(fit1), BIC(fit2), BIC(fit3), BIC(fit4), BIC(fit5), BIC(fit6)))
```

```
comparaciones
```

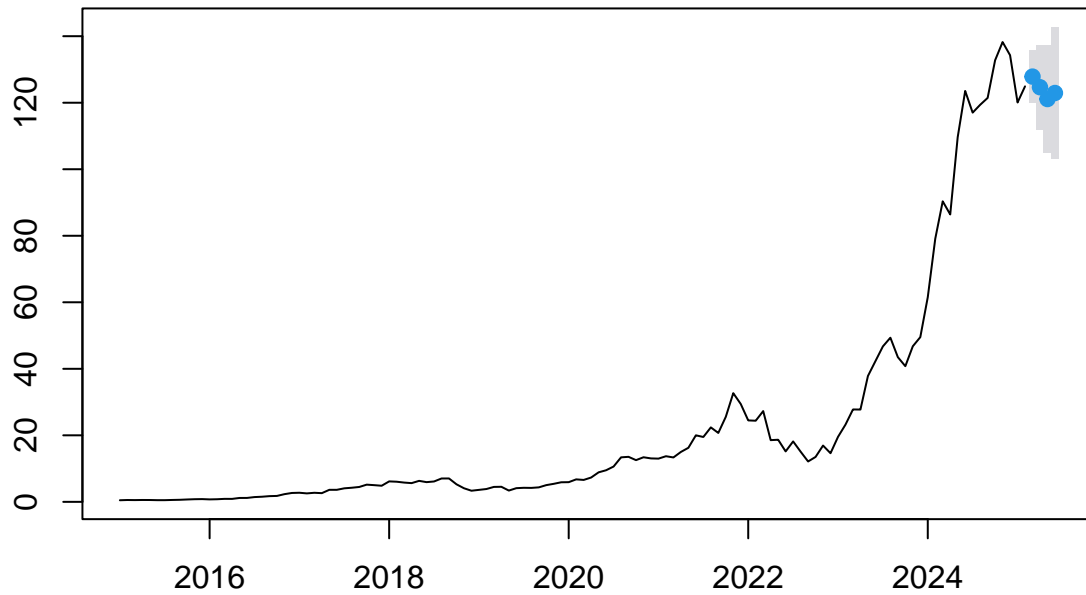
```
##   Model      AIC      BIC
## 1  fit1 690.2610 701.4442
## 2  fit2 688.1028 699.2528
## 3  fit3 689.7388 706.4637
## 4  fit4 684.1165 700.8414
## 5  fit5 684.1165 700.8414
## 6  fit6 684.4929 709.5803
```

Comparando los dos primeros modelos donde diferenciamos 1 y 2 veces respectivamente nos damos cuenta que es necesario y mejor hacerlo 2 veces, porque nuestros criterios de información son más bajos. Sin embargo, hay dos modelos muy similares que tienen AIC y BIC más bajos todavía, se trata del 4 y del 5, realmente son lo mismo, en las líneas del código la distinción es que uno está tomando en cuenta la estacionalidad y en otro no, pero al verlo detenidamente no incluye este factor. Entonces para escoger nuestro mejor modelo podríamos tomar el 4 o 5, sin distinción.

Pronóstico

```
Pro <- forecast(fit4, 4, 95)
plot(Pro)
```

Forecasts from ARIMA(0,2,5)

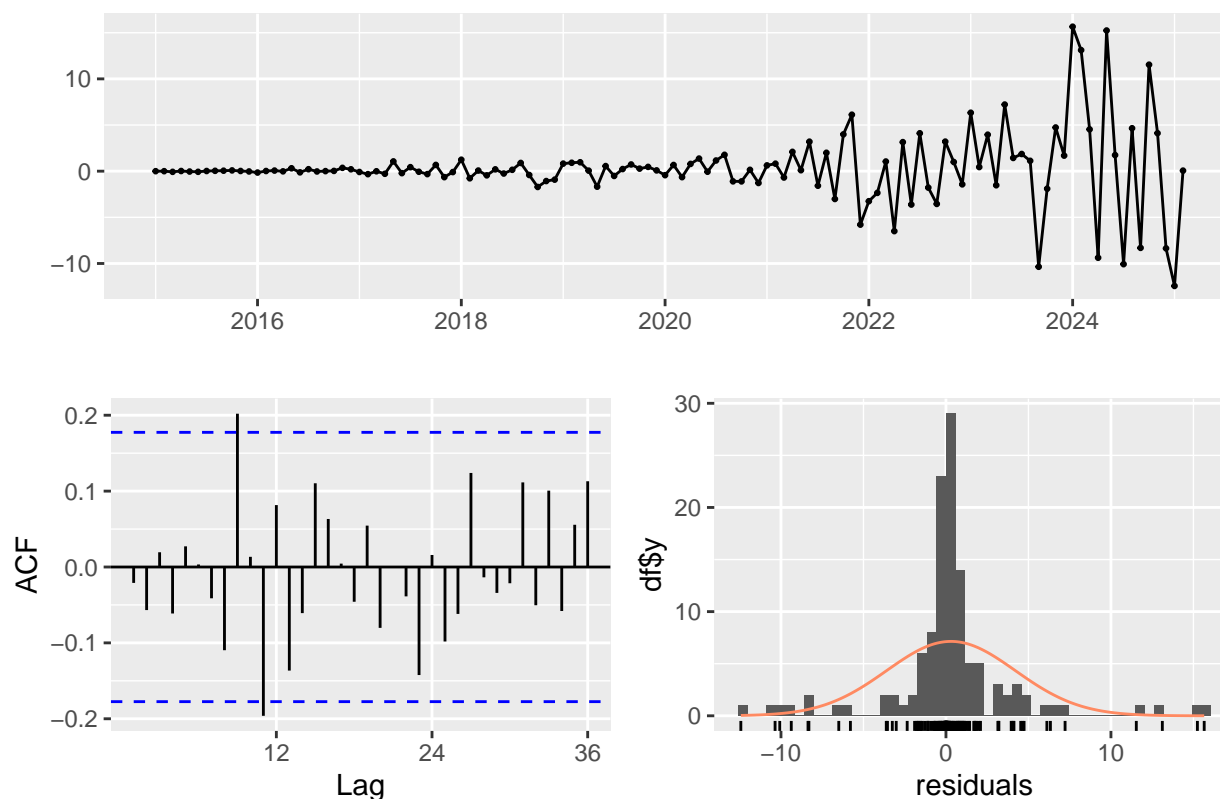


Vemos que las proyecciones que hace hacia 5 períodos hacia adelante varía y parece tener sentido con la naturaleza de los datos, podemos esperar los meses necesarios y comprobar si el probóstico realizado se asemeja a la realidad.

Revisaremos los residuos ahora

```
checkresiduals(fit4)
```

Residuals from ARIMA(0,2,5)



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,2,5)
## Q* = 25.008, df = 19, p-value = 0.1603
##
## Model df: 5.    Total lags used: 24
```

En este caso como $p\text{-value} = 0.1603 > 0.05$ consideramos los errores ruido blanco por lo que asumimos que no hay autocorrelación

Conclusión

El mejor modelo es el 4to por sus criterios de información, como es una serie compleja no es tan factible incluirle una estacionalidad determinada, por eso es que no se incluye, además el pronóstico realizado es coherente con lo visto, hay que tomar en cuenta que el precio de las acciones de NVIDIA experimentó una caída sin precedentes el lunes 27 de enero de 2025, cayendo un 17% y borrando casi \$600 mil millones en valor de mercado, esto fue provocada por el anuncio de la startup china DeepSeek sobre su modelo de inteligencia artificial.

Referencia

https://www.ig.com/latam/ideas-de-trading-y-noticias/por_que_nvidia_cayodespuesdenoticiasdeepseek-250128#:~:text=%C2%BFQu%C3%A9%20caus%C3%B3%20la%20ca%C3%ADda%20de,millones%20en%20valor%20de%20me