Informe evaluación del módulo 3

1. **Modelos implementados**

En el presente informe se describe la implementación de dos modelos de pronóstico de series de tiempo aplicados a las series de tiempo de los activos RYLD, TSLA, TM y GM comprendido entre el 22 de Abril de 2019 y el 31 de Diciembre de 2022. Los modelos evaluados fueron: VAR y ARDL.

Selección del modelo: VAR vs. VECM

Para determinar el modelo adecuado entre VAR y VECM, se realizaron pruebas de estacionariedad (ADF) y de cointegración (Johansen). Los resultados fueron los siguientes:

- Las pruebas ADF indicaron que las series no son estacionarias en niveles.
- La prueba de Johansen no encontró evidencia de cointegración entre las variables.

Conclusión: Dado que las series no son estacionarias y no están cointegradas, el modelo VECM no es apropiado. En su lugar, se deben diferenciar las series para hacerlas estacionarias y luego aplicar un modelo VAR sobre las series diferenciadas.

1.1 **Modelo VAR**: El modelo VAR es una técnica estadística multivariada que modela la dinámica conjunta de varias series temporales al considerar cada variable como una función lineal de sus propios rezagos y de los rezagos de las demás variables del sistema.

Ventajas: Permite capturar relaciones dinámicas entre múltiples variables endógenas sin necesidad de especificar una variable dependiente explícita. Es útil para el análisis de causalidad, impulso-respuesta y descomposición de varianza.
Desventajas: Requiere que todas las series sean estacionarias, por lo que puede necesitar diferenciación previa. La interpretación puede volverse compleja cuando se incluyen muchas variables o rezagos. Además, puede ser sensible a la especificación del número de rezagos y a la multicolinealidad entre variables.

Para la aplicación del modelo **VAR** se utilizaron las librerías vars y MTS, obteniéndose resultados consistentes entre ambas. El modelo con 2 rezagos mostró patrones interesantes en las relaciones entre las variables.

RYLD: **Su rendimiento está fuertemente influido por sus** propios rezagos**. El** segundo rezago **tiene un efecto** positivo y altamente significativo**, mientras que el** primer rezago **tiene un efecto** negativo también significativo**. Además, el segundo rezago de** GM **(GM.l2) muestra un** efecto negativo marginalmente significativo **sobre RYLD.**

TSLA: Ninguna variable alcanzó significancia al 5%, aunque el segundo rezago de RYLD presenta un **efecto positivo cercano al 10%**, lo que sugiere una posible influencia débil.

TM: Se destaca el efecto **negativo y significativo** del primer rezago de RYLD, indicando una relación inversa entre el rendimiento pasado de RYLD y el actual de TM.

GM: En esta ecuación, los rezagos de **TSLA** y **RYLD** —particularmente RYLD.l2— muestran **efectos significativos**, lo que sugiere cierta dependencia de GM respecto al comportamiento pasado de estas variables.

VAR con librería MST.

```
Constant term:
Estimates:  -0.005413425 0.2545003 0.007388606 -0.001985235
Std.Error:  0.008886438 0.2742414 0.07765216 0.03683012
AR coefficient matrix
AR( 1 )-matrix
        [,1]      [,2]    [,3]     [,4]
[1,] -0.1405  0.000373 -0.0018  0.01528
[2,]  1.2525 -0.043873 -0.0131 -0.35200
[3,] -1.0907  0.014009  0.0122  0.09280
[4,] -0.0555  0.009925 -0.0102 -0.00975
standard error
       [,1]    [,2]    [,3]   [,4]
[1,] 0.0427 0.00120 0.00466 0.0104
[2,] 1.3184 0.03688 0.14391 0.3219
[3,] 0.3733 0.01044 0.04075 0.0911
[4,] 0.1771 0.00495 0.01933 0.0432
AR( 2 )-matrix
      [,1]     [,2]      [,3]    [,4]
[1,] 0.286 -0.00234 -0.000191 -0.0206
[2,] 2.394 -0.01228 -0.016173 -0.0696
[3,] 0.243 -0.00693  0.012395  0.1269
[4,] 0.442 -0.00867 -0.032857 -0.0179
```

VAR con librería VARS

```
Estimation results for equation RYLD:
=====================================
RYLD = RYLD.l1 + TSLA.l1 + TM.l1 + GM.l1 + RYLD.l2 + TSLA.l2 + T

          Estimate Std. Error t value Pr(>|t|)
RYLD.l1 -0.1404907  0.0427211  -3.289  0.00105 **
TSLA.l1  0.0003732  0.0011952   0.312  0.75490
TM.l1   -0.0017988  0.0046631  -0.386  0.69977
GM.l1    0.0152832  0.0104306   1.465  0.14322
RYLD.l2  0.2856285  0.0428700   6.663 4.75e-11 ***
TSLA.l2 -0.0023379  0.0011954  -1.956  0.05081 .
TM.l2   -0.0001906  0.0046566  -0.041  0.96736
GM.l2   -0.0206117  0.0104105  -1.980  0.04803 *
const   -0.0054134  0.0088864  -0.609  0.54256
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

1.2 **Modelo ARDL**: El modelo ARDL es una técnica de regresión que combina componentes autorregresivos con rezagos de variables explicativas, permitiendo modelar relaciones tanto de corto como de largo plazo entre una variable dependiente y una o más explicativas, aunque estas no estén integradas del mismo orden.

Ventajas: Es flexible al permitir que las variables estén integradas de orden 0 o 1 (I(0) o I(1)), sin requerir que todas sean estacionarias en el mismo nivel. Permite estimar efectos de corto y largo plazo de manera directa.Es útil para análisis con muestras pequeñas y permite identificar relaciones de cointegración mediante la prueba de límites (bounds test) permitiendo ver de forma mas clara las relaciones entre las variables a corto y largo plazo.
Desventajas: No puede aplicarse si alguna de las variables está integrada de orden 2 (I(2)). Requiere una correcta elección del número de rezagos, ya que un mal ajuste puede llevar a resultados sesgados.

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.1488489  0.0915044   1.627 0.104165
L(RYLD, 1)   0.8668074  0.0309039  28.049  < 2e-16 ***
L(RYLD, 2)   0.3373508  0.0390773   8.633  < 2e-16 ***
L(RYLD, 3)  -0.2217836  0.0306534  -7.235 1.02e-12 ***
TSLA         0.0062359  0.0008919   6.992 5.41e-12 ***
L(TSLA, 1)  -0.0066197  0.0008909  -7.430 2.59e-13 ***
TM           0.0274560  0.0034883   7.871 1.04e-14 ***
L(TM, 1)    -0.0262163  0.0035202  -7.447 2.29e-13 ***
GM           0.0944349  0.0074301  12.710  < 2e-16 ***
L(GM, 1)    -0.0781005  0.0104085  -7.504 1.53e-13 ***
L(GM, 2)    -0.0346547  0.0103583  -3.346 0.000856 ***
L(GM, 3)     0.0214165  0.0075761   2.827 0.004809 **
---
```

Efectos a corto plazo: Los resultados muestran que los activos TSLA**,** TM y GM tienen un impacto significativo en el índice RYLD en el corto plazo: Un aumento de 1 unidad en TSLA incrementa RYLD en aproximadamente 0.006 unidades. Un aumento de 1 unidad en TM incrementa RYLD en aproximadamente 0.027 unidades. Un aumento de 1 unidad en GM incrementa RYLD en aproximadamente 0.094 unidades.

Efectos a largo plazo: TSLA tiene un efecto negativo y significativo sobre RYLD. GM tiene un efecto positivo y significativo. TM también tiene un efecto positivo, aunque con una significancia marginal. En resumen, los activos TSLA y GM son los que más contribuyen a explicar el comportamiento de RYLD en el largo plazo.

2. Selección del modelo con mejor desempeño

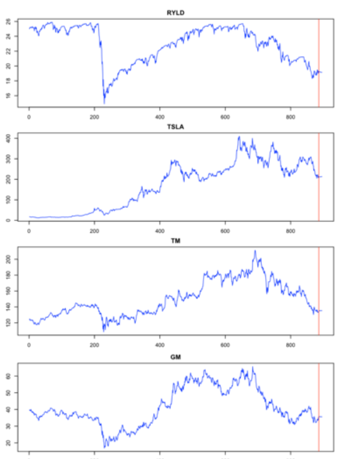
Tras la comparación de los modelos VAR y ARDL, se observó que el modelo VAR obtuvo un mejor desempeño en términos de error de predicción, con un RMSE de 0.1321 y un MAE de 0.1001. Estos valores reflejan una mayor precisión frente al modelo ARDL, que presentó un RMSE de 0.2003 y un MAE de 0.1418.

El modelo VAR se destaca por su capacidad para capturar relaciones dinámicas entre múltiples series temporales al considerar los rezagos de todas las variables del sistema. Esta estructura le permite adaptarse mejor a la naturaleza interdependiente de las variables económicas y financieras analizadas.

En cambio, aunque el modelo ARDL también ofrece una aproximación robusta —especialmente útil cuando algunas variables son estacionarias y otras no—, en este caso específico mostró un rendimiento inferior al no capturar con la misma eficacia la dinámica conjunta de las series.

Modelo	RMSE	MAE
ARDL	0.2003	0.1418
VAR	0.1321	0.1001

El modelo **VAR** es seleccionado como el más adecuado para este análisis, gracias a su mejor capacidad predictiva y su habilidad para representar de forma conjunta la evolución temporal de las variables involucradas. Su estructura multivariada lo convierte en una herramienta más precisa y eficiente en este contexto.



En la siguiente gráfica se muestra el pronóstico a 10 días del índice RYLD (y los activos TSLA, TM y GM), realizado utilizando el modelo VAR (Vector Autoregresivo). Se observa que la proyección sigue una trayectoria coherente con la tendencia reciente de la serie histórica, sin presentar valores atípicos ni cambios abruptos. Esto indica que el modelo ha logrado capturar adecuadamente la dinámica temporal del índice, basándose en la relación entre sus propios rezagos y los de las demás variables del sistema. La suavidad y continuidad de la curva pronosticada refuerzan la fiabilidad del modelo, ya que evita fluctuaciones extremas no respaldadas por los datos históricos.

Ambos modelos, VAR y ARDL, presentan fortalezas y limitaciones que dependen estrechamente de la naturaleza de las series temporales y las técnicas aplicadas. El modelo VAR, al requerir series estacionarias y diferenciar previamente las variables, es especialmente adecuado para capturar dinámicas multivariadas complejas y relaciones interdependientes entre todas las variables, lo que se refleja en su mejor desempeño predictivo en este caso. Sin embargo, su necesidad de estacionariedad y sensibilidad a la especificación de rezagos puede limitar su uso en series con propiedades heterogéneas. Por otro lado, el modelo ARDL ofrece una mayor flexibilidad al permitir variables integradas de distinto orden sin requerir diferenciación estricta, facilitando la estimación de efectos de corto y largo plazo, lo cual es útil en contextos con muestras pequeñas y series mixtas en integración. No obstante, su desempeño puede verse afectado cuando las relaciones dinámicas entre variables son complejas o cuando la dinámica conjunta es dominante, como ocurrió en este análisis. En resumen, la elección del modelo debe basarse en la estructura estadística de los datos y el objetivo del análisis, siendo VAR más idóneo para relaciones dinámicas multivariadas y ARDL para análisis más focalizados en relaciones a diferentes horizontes temporales con series de integración mixta.

Paso 1. Aplicación de los modelos

1. Descarga de YahooFinancelas series de precios de cierre de RYLD, TSLA, TM y GM en las fechas: 22-04-2019 al 31-12-2022

```
#Otros de manipulación de datos
library(forecast)
library(tseries)
library(tidyverse)
library(urca)
library(quantmod)
library(TSstudio)
library(dygraphs)
theme_set(theme_bw())
library(vars)
library(MTS)
library(ARDL)
library(Metrics)
library(zoo)


options(warn = - 1)

###Función para obtener datos:
start<-format(as.Date("2019-04-22"),"%Y-%m-%d")
end<-format(as.Date("2022-12-31"),"%Y-%m-%d")
#funcion para obtener los precios
precios <-function(simbolo)
{
  ##Obtener precios stocks de Yahoo FInance
  datos <- getSymbols(simbolo, auto.assign = FALSE, from=start,
to=end)
  ## Elimar faltantes:
  datos<-na.omit(datos)
  ##mantener columnas con precios maximo, minimo, de cierre y volumen
de mercado:
  datos <- datos[,4]
  ##Para hacerlo datos accesibles en el global environment:
  assign(simbolo, datos, envir = .GlobalEnv)
}
##Llamar el activo de interés, pueden ser varios:
precios("RYLD")
precios("TSLA")
precios("TM")
precios("GM")
```

2. Prepara los datos a manera que cuentes con un formato adecuado para la fecha.

```r
#Ya que tenemos que trabajar en formato ts
## Juntamos los datos, renombramos las columnas y las visualizamos:
prices<-merge.xts(`RYLD`, `TSLA`, `TM`, `GM`, join = "inner")
colnames(prices)<-c("RYLD", "TSLA", "TM", "GM")

s1=`RYLD`
s2=`TSLA`
s3=`TM`
s4=`GM`
```

3. Grafica las series.

```r
#Podemos  visualizar la serie de tiempo

dygraph(prices, main = c("RYLD", "TSLA", "TM", "GM")) %>%
  dyAxis("y", label = "Prices") %>%
  dyOptions(colors = RColorBrewer::brewer.pal(4, "Set1"))
```

```
HTML widgets cannot be represented in plain text (need html)
```

4. Parte la serie a un 5% de prueba y el restante para el entrenamiento.

```r
# Separar conjunto de entrenamiento y prueba (95% / 5%)
# Porcentaje de entrenamiento
n_total <- nrow(prices)
n_train <- floor(0.95 * n_total)

# División en conjuntos
train_prices <- prices[1:n_train, ]
test_prices <- prices[(n_train + 1):n_total, ]

dim(train_prices)  # Verifica las dimensiones
dim(test_prices)
```

```
[1] 886   4
```

```
[1] 47  4
```

5. Aplica los modelos VAR o VECM según corresponda (con ambas librerías, vars y mts) y ARDL al conjunto de series.

Analisisi VAR vs VECM

Prueba de estacionariedad

```
# Prueba ADF para cada serie en niveles
adf.test(prices$RYLD)
adf.test(prices$TSLA)
adf.test(prices$TM)
adf.test(prices$GM)


	Augmented Dickey-Fuller Test

data:  prices$RYLD
Dickey-Fuller = -1.9681, Lag order = 9, p-value = 0.5918
alternative hypothesis: stationary


	Augmented Dickey-Fuller Test

data:  prices$TSLA
Dickey-Fuller = -0.82177, Lag order = 9, p-value = 0.9597
alternative hypothesis: stationary


	Augmented Dickey-Fuller Test

data:  prices$TM
Dickey-Fuller = -1.3187, Lag order = 9, p-value = 0.8667
alternative hypothesis: stationary


	Augmented Dickey-Fuller Test

data:  prices$GM
Dickey-Fuller = -1.3763, Lag order = 9, p-value = 0.8423
alternative hypothesis: stationary
```

Prueba de Cointegración (Johansen Test)

Primero identificar numero optimo de rezagos

```
nivelk=VARselect(prices, lag.max = 7, type = "const")
nivelk$selection

AIC(n)  HQ(n)  SC(n) FPE(n)
     3      1      1      3
```

Usaremos nivel de rezago K=3. Ahora si aplicamos nuestra test de Johansen.

```
johansen_test <- ca.jo(prices, type = "trace", K=3, ecdet ="none",
spec = "longrun")
# Resumen del test
summary(johansen_test)


######################
# Johansen-Procedure #
######################

Test type: trace statistic , with linear trend

Eigenvalues (lambda):
[1] 0.025178838 0.013551622 0.005454097 0.001665807

Values of teststatistic and critical values of test:

          test 10pct  5pct  1pct
r <= 3 |  1.55  6.50  8.18 11.65
r <= 2 |  6.64 15.66 17.95 23.52
r <= 1 | 19.33 28.71 31.52 37.22
r = 0  | 43.04 45.23 48.28 55.43

Eigenvectors, normalised to first column:
(These are the cointegration relations)

            RYLD.l3     TSLA.l3        TM.l3      GM.l3
RYLD.l3  1.00000000  1.00000000  1.000000000 1.00000000
TSLA.l3  0.04047053  0.01420505 -0.006160056 0.07517971
TM.l3   -0.19656578  0.11367107 -0.016571013 0.04088028
GM.l3   -0.13677494 -0.47698175 -0.007522554 0.33936225

Weights W:
(This is the loading matrix)

           RYLD.l3      TSLA.l3        TM.l3         GM.l3
RYLD.d -0.00167326 -0.005669915 -0.003600594 -0.0005984274
TSLA.d -0.15022102  0.046282162  0.103842662 -0.0217639916
TM.d    0.06401477 -0.083909134  0.019062380 -0.0028190164
GM.d    0.05691598 -0.002735887 -0.006252483 -0.0023303854
```

Conclusión:

- De acuerdo a la prueba ADF Las series no son estacionarias en niveles
- De acuerdo a la prueba Johansen no hay cointegración detectada

Por tanto, NO debes aplicar un modelo VECM, sino que aplicaremos el modelo VAR. Para esto debemos primero diferenciar las series para hacerlas estacionarias y luego si aplicar el modelo VAR.

5.1 Aplicacion modelo VAR

```
# Diferenciación de las series
diff_prices <- diff(prices)[-1,]  # eliminamos el NA inicial

# Verificamos estacionariedad de las series diferenciadas
adf.test(diff_prices$RYLD)
adf.test(diff_prices$TSLA)
adf.test(diff_prices$TM)
adf.test(diff_prices$GM)
```

```

	Augmented Dickey-Fuller Test

data:  diff_prices$RYLD
Dickey-Fuller = -8.0814, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary


	Augmented Dickey-Fuller Test

data:  diff_prices$TSLA
Dickey-Fuller = -9.2478, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary


	Augmented Dickey-Fuller Test

data:  diff_prices$TM
Dickey-Fuller = -10.768, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary


	Augmented Dickey-Fuller Test

data:  diff_prices$GM
Dickey-Fuller = -9.895, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary
```

Los resultados muestran que todas las series diferenciadas son estacionarias, ya que:

En todos los casos, el p-value < 0.01.

Esto nos permite rechazar la hipótesis nula de no estacionariedad.

5.1.1 Estimar modelo VAR con libreria VARS

```
# Selección de orden VAR óptimo con criterio AIC o BIC
var_selection <- VARselect(diff_prices, lag.max=10, type="const")
var_selection$selection
```

```
AIC(n)  HQ(n)  SC(n) FPE(n) 
     2      2      1      2 
```

Usaremos nivel de rezago K=2

```
# Asegurémonos que es data.frame numérico
diff_train <- diff(train_prices)[-1, ]
diff_train <- as.data.frame(diff_train)
#Podemos volver a llamar la librería de vars y aplicar el regresión
habiendo encontrado que p=2.
m0=vars::VAR(diff_train, p=2)
summary(m0)
```

```
VAR Estimation Results:
=========================
Endogenous variables: RYLD, TSLA, TM, GM
Deterministic variables: const
Sample size: 883
Log Likelihood: -6062.921
Roots of the characteristic polynomial:
0.5707 0.4178 0.2799 0.2799 0.2761 0.2761 0.1395 0.09844
Call:
vars::VAR(y = diff_train, p = 2)


Estimation results for equation RYLD:
=====================================
RYLD = RYLD.l1 + TSLA.l1 + TM.l1 + GM.l1 + RYLD.l2 + TSLA.l2 + TM.l2 +
GM.l2 + const

          Estimate Std. Error t value Pr(>|t|)
RYLD.l1 -0.1404907  0.0427211  -3.289  0.00105 **
TSLA.l1  0.0003732  0.0011952   0.312  0.75490
TM.l1   -0.0017988  0.0046631  -0.386  0.69977
GM.l1    0.0152832  0.0104306   1.465  0.14322
RYLD.l2  0.2856285  0.0428700   6.663 4.75e-11 ***
TSLA.l2 -0.0023379  0.0011954  -1.956  0.05081 .
TM.l2   -0.0001906  0.0046566  -0.041  0.96736
GM.l2   -0.0206117  0.0104105  -1.980  0.04803 *
const   -0.0054134  0.0088864  -0.609  0.54256
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 0.2634 on 874 degrees of freedom
Multiple R-Squared: 0.08033,     Adjusted R-squared: 0.07191
F-statistic: 9.543 on 8 and 874 DF,  p-value: 1.013e-12


Estimation results for equation TSLA:
=====================================
TSLA = RYLD.l1 + TSLA.l1 + TM.l1 + GM.l1 + RYLD.l2 + TSLA.l2 + TM.l2 +
GM.l2 + const
```

```
        Estimate Std. Error t value Pr(>|t|)
RYLD.l1  1.25252    1.31840   0.950   0.3424
TSLA.l1 -0.04387    0.03688  -1.189   0.2346
TM.l1   -0.01308    0.14391  -0.091   0.9276
GM.l1   -0.35200    0.32189  -1.094   0.2745
RYLD.l2  2.39364    1.32300   1.809   0.0708 .
TSLA.l2 -0.01228    0.03689  -0.333   0.7394
TM.l2   -0.01617    0.14371  -0.113   0.9104
GM.l2   -0.06962    0.32128  -0.217   0.8285
const    0.25450    0.27424   0.928   0.3537
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 8.128 on 874 degrees of freedom
Multiple R-Squared: 0.007947,    Adjusted R-squared: -0.001134
F-statistic: 0.8751 on 8 and 874 DF,  p-value: 0.537


Estimation results for equation TM:
===================================
TM = RYLD.l1 + TSLA.l1 + TM.l1 + GM.l1 + RYLD.l2 + TSLA.l2 + TM.l2 +
GM.l2 + const

         Estimate Std. Error t value Pr(>|t|)
RYLD.l1 -1.090654   0.373309  -2.922  0.00357 **
TSLA.l1  0.014009   0.010444   1.341  0.18014
TM.l1    0.012203   0.040747   0.299  0.76464
GM.l1    0.092795   0.091145   1.018  0.30891
RYLD.l2  0.242946   0.374610   0.649  0.51681
TSLA.l2 -0.006930   0.010446  -0.663  0.50726
TM.l2    0.012395   0.040691   0.305  0.76073
GM.l2    0.126924   0.090970   1.395  0.16330
const    0.007389   0.077652   0.095  0.92422
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 2.301 on 874 degrees of freedom
Multiple R-Squared: 0.01836,     Adjusted R-squared: 0.009372
F-statistic: 2.043 on 8 and 874 DF,  p-value: 0.03895


Estimation results for equation GM:
===================================
GM = RYLD.l1 + TSLA.l1 + TM.l1 + GM.l1 + RYLD.l2 + TSLA.l2 + TM.l2 +
GM.l2 + const

         Estimate Std. Error t value Pr(>|t|)
```

```
RYLD.l1 -0.055505   0.177059  -0.313   0.7540
TSLA.l1  0.009925   0.004953   2.004   0.0454 *
TM.l1   -0.010229   0.019326  -0.529   0.5967
GM.l1   -0.009746   0.043230  -0.225   0.8217
RYLD.l2  0.442106   0.177676   2.488   0.0130 *
TSLA.l2 -0.008667   0.004954  -1.749   0.0806 .
TM.l2   -0.032857   0.019300  -1.702   0.0890 .
GM.l2   -0.017924   0.043147  -0.415   0.6779
const   -0.001985   0.036830  -0.054   0.9570
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 1.092 on 874 degrees of freedom
Multiple R-Squared: 0.01674,     Adjusted R-squared: 0.007735
F-statistic: 1.859 on 8 and 874 DF,  p-value: 0.06313



Covariance matrix of residuals:
        RYLD    TSLA      TM     GM
RYLD 0.06937  0.8099 0.2963 0.1628
TSLA 0.80986 66.0628 4.9555 2.7252
TM   0.29629  4.9555 5.2966 1.2769
GM   0.16278  2.7252 1.2769 1.1915

Correlation matrix of residuals:
       RYLD   TSLA     TM     GM
RYLD 1.0000 0.3783 0.4888 0.5662
TSLA 0.3783 1.0000 0.2649 0.3072
TM   0.4888 0.2649 1.0000 0.5083
GM   0.5662 0.3072 0.5083 1.0000
```

En el modelo VAR con rezagos p=2, observamos que el rendimiento de RYLD está significativamente influido por sus propios rezagos: el segundo rezago tiene un impacto positivo y altamente significativo, mientras que el primero es negativo y también significativo. Además, GM.l2 presenta un efecto negativo marginalmente significativo sobre RYLD. En el caso de TSLA, ningún predictor alcanza significancia al 5%, aunque el segundo rezago de RYLD muestra un efecto positivo cercano al 10%. Para TM, destaca el efecto negativo y significativo del primer rezago de RYLD. Finalmente, en la ecuación de GM, los rezagos de TSLA y RYLD (especialmente RYLD.l2) muestran una influencia significativa. En general, RYLD parece tener un papel importante en explicar la dinámica de varias series, mientras que TSLA, TM y GM presentan relaciones menos consistentes entre sí.

5.1.2 Estimar modelo VAR con libreria MTS

```
# En la librería de MTS, la función que permite la identificación del nivel regresivo es la de VARorder().
VARorder(diff_prices)
```

```
selected order: aic =  2
selected order: bic =  0
selected order: hq =  2
Summary table:
       p    AIC    BIC     HQ    M(p) p-value
 [1,]  0 2.4738 2.4738 2.4738  0.0000  0.0000
 [2,]  1 2.4539 2.5369 2.4856 49.5841  0.0000
 [3,]  2 2.4059 2.5720 2.4692 74.9077  0.0000
 [4,]  3 2.4311 2.6802 2.5261  8.2821  0.9400
 [5,]  4 2.4365 2.7687 2.5632 26.0077  0.0539
 [6,]  5 2.4510 2.8663 2.6094 17.8070  0.3353
 [7,]  6 2.4647 2.9629 2.6547 18.5089  0.2949
 [8,]  7 2.4577 3.0390 2.6793 36.7762  0.0023
 [9,]  8 2.4651 3.1294 2.7184 23.8195  0.0935
[10,]  9 2.4620 3.2094 2.7471 32.9700  0.0075
[11,] 10 2.4752 3.3056 2.7919 18.5853  0.2908
[12,] 11 2.4854 3.3989 2.8338 21.0232  0.1776
[13,] 12 2.5054 3.5019 2.8854 12.5226  0.7073
[14,] 13 2.5261 3.6057 2.9379 11.7476  0.7612
```

Usaremos nivel de rezago K=2

```
# Convertir a matriz
diff_train_mat <- as.matrix(diff_train)


# Ajuste del modelo VAR desde la librería MTS
var_model_mts <- MTS::VAR(diff_train_mat, p = 2)

# Visualizar resumen
summary(var_model_mts)

Constant term:
Estimates:  -0.005413425 0.2545003 0.007388606 -0.001985235
Std.Error:  0.008886438 0.2742414 0.07765216 0.03683012
AR coefficient matrix
AR( 1 )-matrix
        [,1]      [,2]    [,3]     [,4]
[1,] -0.1405  0.000373 -0.0018  0.01528
[2,]  1.2525 -0.043873 -0.0131 -0.35200
[3,] -1.0907  0.014009  0.0122  0.09280
[4,] -0.0555  0.009925 -0.0102 -0.00975
standard error
       [,1]    [,2]    [,3]   [,4]
[1,] 0.0427 0.00120 0.00466 0.0104
[2,] 1.3184 0.03688 0.14391 0.3219
[3,] 0.3733 0.01044 0.04075 0.0911
[4,] 0.1771 0.00495 0.01933 0.0432
AR( 2 )-matrix
```

```
      [,1]     [,2]      [,3]    [,4]
[1,] 0.286 -0.00234 -0.000191 -0.0206
[2,] 2.394 -0.01228 -0.016173 -0.0696
[3,] 0.243 -0.00693  0.012395  0.1269
[4,] 0.442 -0.00867 -0.032857 -0.0179
standard error
       [,1]    [,2]    [,3]   [,4]
[1,] 0.0429 0.00120 0.00466 0.0104
[2,] 1.3230 0.03689 0.14371 0.3213
[3,] 0.3746 0.01045 0.04069 0.0910
[4,] 0.1777 0.00495 0.01930 0.0431

Residuals cov-mtx:
           [,1]       [,2]      [,3]      [,4]
[1,] 0.06865896  0.8016045 0.2932716 0.1611224
[2,] 0.80160454 65.3894485 4.9050138 2.6974189
[3,] 0.29327159  4.9050138 5.2426260 1.2639048
[4,] 0.16112235  2.6974189 1.2639048 1.1793644

det(SSE) =  10.81617
AIC =  2.453359
BIC =  2.626397
HQ  =  2.519514

          Length Class  Mode
data      3540   -none- numeric
cnst         1   -none- logical
order        1   -none- numeric
coef        36   -none- numeric
aic          1   -none- numeric
bic          1   -none- numeric
hq           1   -none- numeric
residuals 3532   -none- numeric
secoef      36   -none- numeric
Sigma       16   -none- numeric
Phi         32   -none- numeric
Ph0          4   -none- numeric
fixed        0   -none- NULL
```

El modelo VAR estimado con la librería MTS confirma relaciones dinámicas moderadas entre las variables, destacando una autocorrelación significativa en RYLD (especialmente en el segundo rezago) y cierta influencia de RYLD y TSLA sobre GM. Las matrices AR(1) y AR(2) muestran patrones similares a los obtenidos con la librería vars, lo que refuerza la consistencia de los resultados. Las métricas de ajuste (AIC = 2.45, BIC = 2.63) indican un modelo razonablemente parsimonioso, y la matriz de covarianza de los residuos sugiere correlaciones positivas entre los errores, especialmente entre RYLD y TSLA.

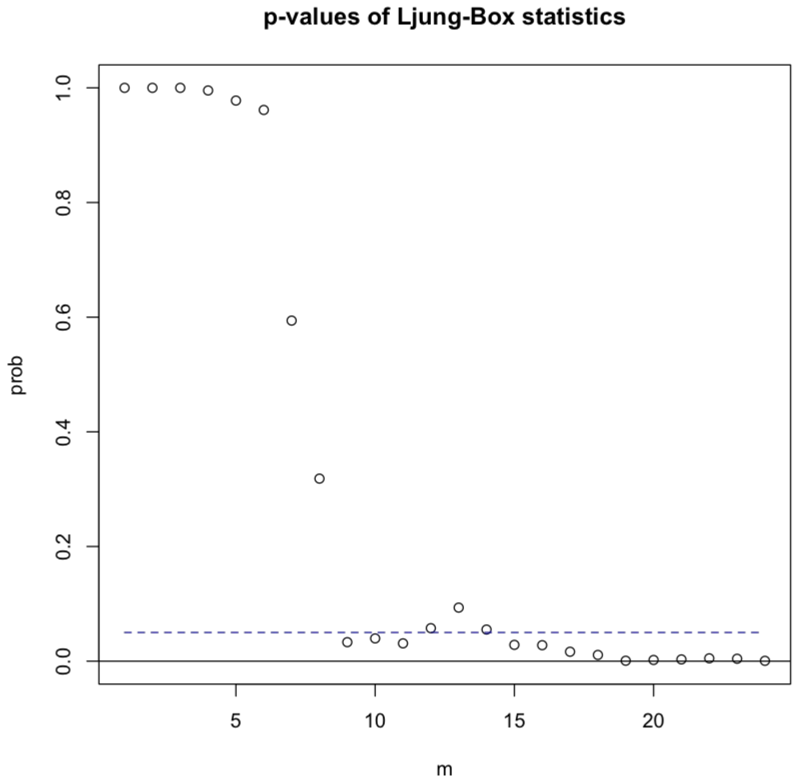
```
# Realizamos modelo VAR refinado
m3_mts=refVAR(var_model_mts,thres = 1.96)
```

```
Constant term:
Estimates:  0 0 0 0
Std.Error:  0 0 0 0
AR coefficient matrix
AR( 1 )-matrix
       [,1] [,2] [,3] [,4]
[1,] -0.110    0    0    0
[2,]  0.000    0    0    0
[3,] -0.712    0    0    0
[4,]  0.000    0    0    0
standard error
       [,1] [,2] [,3] [,4]
[1,] 0.0329    0    0    0
[2,] 0.0000    0    0    0
[3,] 0.2832    0    0    0
[4,] 0.0000    0    0    0
AR( 2 )-matrix
      [,1]     [,2]    [,3]    [,4]
[1,] 0.272  0.00000  0.0000 -0.0243
[2,] 0.000  0.00000  0.0000  0.0000
[3,] 0.000  0.00000  0.0000  0.1554
[4,] 0.439 -0.00962 -0.0361  0.0000
standard error
       [,1]   [,2]   [,3]    [,4]
[1,] 0.0396 0.0000 0.0000 0.00979
[2,] 0.0000 0.0000 0.0000 0.00000
[3,] 0.0000 0.0000 0.0000 0.07076
[4,] 0.1612 0.0049 0.0183 0.00000

Residuals cov-mtx:
                      resi
     0.06916046  0.7964633 0.2955452 0.1612436
resi 0.79646333 65.9615077 4.8661707 2.6771854
     0.29554517  4.8661707 5.2722566 1.2711235
     0.16124361  2.6771854 1.2711235 1.1854212

det(SSE) =  11.20644
AIC =  2.434568
BIC =  2.477827
HQ  =  2.451107
```

Contamos grados de libertad (Coeficientes distintos de cero): 8

```
#Separemos primero los residuales del modelo de regresión y apliquemos la función de mq() para la revisión de la calidad del modelo.
#Indicamos los grados de libertad del modelo, que son 8.
resi=var_model_mts$residuals
mq(resi, adj=8)
```

```
Ljung-Box Statistics:
          m       Q(m)      df    p-value
 [1,]   1.000     0.193   8.000     1.00
 [2,]   2.000     0.890  24.000     1.00
 [3,]   3.000     6.124  40.000     1.00
 [4,]   4.000    32.267  56.000     1.00
 [5,]   5.000    49.949  72.000     0.98
 [6,]   6.000    66.022  88.000     0.96
 [7,]   7.000    99.954 104.000     0.59
 [8,]   8.000   126.774 120.000     0.32
 [9,]   9.000   167.835 136.000     0.03
[10,]  10.000   183.934 152.000     0.04
[11,]  11.000   203.825 168.000     0.03
[12,]  12.000   215.183 184.000     0.06
[13,]  13.000   226.861 200.000     0.09
[14,]  14.000   250.189 216.000     0.06
[15,]  15.000   274.746 232.000     0.03
[16,]  16.000   292.363 248.000     0.03
[17,]  17.000   315.377 264.000     0.02
[18,]  18.000   337.159 280.000     0.01
[19,]  19.000   379.438 296.000     0.00
[20,]  20.000   388.513 312.000     0.00
[21,]  21.000   402.650 328.000     0.00
[22,]  22.000   415.505 344.000     0.00
[23,]  23.000   434.643 360.000     0.00
[24,]  24.000   471.315 376.000     0.00
```



Revisando los rezagos de los residuos y considerando que este es un modelo AR(2) podemos evidenciar que los dos primeros puntos se encuentran por encima del umbral de la linea azul, indicando que no estan correlacionados entre ellos.

5.2 Aplicacion modelo ARDL

```
#Validacion de si se puede usar ARDL
adf.test(diff(s1)%>%na.omit())
adf.test(diff(s2)%>%na.omit())
adf.test(diff(s3)%>%na.omit())
adf.test(diff(s4)%>%na.omit())


     Augmented Dickey-Fuller Test

data:  diff(s1) %>% na.omit()
Dickey-Fuller = -8.0814, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary


     Augmented Dickey-Fuller Test

data:  diff(s2) %>% na.omit()
Dickey-Fuller = -9.2478, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary
```

```

	Augmented Dickey-Fuller Test

data:  diff(s3) %>% na.omit()
Dickey-Fuller = -10.768, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary


	Augmented Dickey-Fuller Test

data:  diff(s4) %>% na.omit()
Dickey-Fuller = -9.895, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary
```

Validacion de si se puede usar ARDL: En este caso al aplicar diferenciacion a las series podemos evidenciar que son I1 por lo que podemos aplicar el modelo de ARDL.

```
# Selección automática del modelo ARDL
train_ts <- zoo(train_prices)
auto_model <- auto_ardl(RYLD ~ TSLA + TM + GM, data = train_ts,
max_order = 6)

#Revisemos el top 20 de los mejores modelos según su critrio de
información de Akaike
auto_model$top_orders
```

	RYLD	TSLA	TM	GM	AIC
1	3	1	1	3	-307.4130
2	3	1	2	3	-305.8611
3	3	2	1	3	-305.5586
4	3	1	1	4	-304.8608
5	4	1	1	3	-304.0696
6	3	2	2	3	-303.9719
7	4	1	1	4	-303.0162
8	4	1	2	3	-302.5227
9	4	2	1	3	-302.2173
10	3	2	3	3	-302.0901
11	3	2	2	4	-301.3247
12	3	3	3	3	-301.2631
13	4	2	2	3	-300.6353
14	4	2	2	4	-299.4420
15	4	2	3	3	-298.7485
16	5	5	1	5	-298.2758
17	4	3	3	3	-297.9065
18	5	5	1	4	-297.6391
19	5	4	1	5	-297.1265
20	4	3	3	4	-296.8666

```
#Procedemos a construir el modelo de regresión con la mejor
combinación.
```

```r
mod1 <- ardl(RYLD ~ TSLA + TM + GM, data = train_ts, order =
c(3,1,1,3))
summary(mod1)
```

```

Time series regression with "zoo" data:
Start = 2019-04-25, End = 2022-10-24

Call:
dynlm::dynlm(formula = full_formula, data = data, start = start,
    end = end)

Residuals:
     Min       1Q   Median       3Q      Max
-1.67649 -0.10110  0.00956  0.11214  0.73684

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.1488489  0.0915044   1.627 0.104165
L(RYLD, 1)   0.8668074  0.0309039  28.049  < 2e-16 ***
L(RYLD, 2)   0.3373508  0.0390773   8.633  < 2e-16 ***
L(RYLD, 3)  -0.2217836  0.0306534  -7.235 1.02e-12 ***
TSLA         0.0062359  0.0008919   6.992 5.41e-12 ***
L(TSLA, 1)  -0.0066197  0.0008909  -7.430 2.59e-13 ***
TM           0.0274560  0.0034883   7.871 1.04e-14 ***
L(TM, 1)    -0.0262163  0.0035202  -7.447 2.29e-13 ***
GM           0.0944349  0.0074301  12.710  < 2e-16 ***
L(GM, 1)    -0.0781005  0.0104085  -7.504 1.53e-13 ***
L(GM, 2)    -0.0346547  0.0103583  -3.346 0.000856 ***
L(GM, 3)     0.0214165  0.0075761   2.827 0.004809 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2017 on 871 degrees of freedom
Multiple R-squared:  0.9934,	Adjusted R-squared:  0.9933
F-statistic: 1.187e+04 on 11 and 871 DF,  p-value: < 2.2e-16
```

```r
# Para la interpretación, podemos imprimir los rezagos
correspondientes de cada variable que explican la respuesta.
mod1$full_formula
```

```
RYLD ~ L(RYLD, 1) + L(RYLD, 2) + L(RYLD, 3) + TSLA + L(TSLA,
    1) + TM + L(TM, 1) + GM + L(GM, 1) + L(GM, 2) + L(GM, 3)
```

5.2.1. Bound test para la verificacion de relaciones a largo término - ARDL

```r
#Guardamos el mejor modelo obtenido en la sección anterior
modelo <- auto_model$best_model
#Realizamos la prueba de hipotesis: Hipotesis nula: no cointegración
bounds_f_test(modelo, case = 2) # el parametro "case" igual a 2
verifica si existe relaciones a largo término, con la combinación de
```

```
(restricted constant o intercepto cte, no linear trend o sin tendencia
determinista).


     Bounds F-test (Wald) for no cointegration

data:  d(RYLD) ~ L(RYLD, 1) + L(TSLA, 1) + L(TM, 1) + L(GM, 1) +
d(L(RYLD,     1)) + d(L(RYLD, 2)) + d(TSLA) + d(TM) + d(GM) + d(L(GM,
1)) +     d(L(GM, 2))
F = 2.5919, p-value = 0.2228
alternative hypothesis: Possible cointegration
null values:
   k    T
   3 1000
```

Hipotesis nula es que no hay cointegracion y la alternativa es que haya cointegracion. Con el p-value de 0.2228, no rechazo la hipotesis nula. Por lo que no hay evidencia estadisticamente significativa de cointegracion a largo plazo entre las variables del modelo especificado.

5.2.2 Mutiplicadores de largo y corto termino

```
#Multiplicadores a corto plazo
multipliers(modelo, type = "sr")
#Son los coeficientes que representan el impacto inmediato o dentro de
unos pocos períodos (pero antes de alcanzar el equilibrio de largo
plazo)

  Term        Estimate    Std. Error   t value   Pr(>|t|)
1 (Intercept) 0.148848875 0.0915043640  1.626686 1.041655e-01
2 TSLA        0.006235931 0.0008919174  6.991602 5.408892e-12
3 TM          0.027456034 0.0034882965  7.870900 1.043152e-14
4 GM          0.094434868 0.0074301356 12.709710 4.508574e-34
```

Comportamiento del índice RYLD en función de los activos: Los p-value de TSLA, TM y GM son significativos, por lo que estos activos generan un efecto en el corto plazo sobre el indice RYLD.

- Un cambio unitario en TSLA estaria generando un incremento de 0.006 en el indice RYLD.
- Un cambio unitario en TM estaria generando un incremento de 0.027 en el indice RYLD.
- Un cambio unitario en GM estaria generando un incremento de 0.094 en el indice RYLD.

```
#Multiplicadores a largo plazo
multipliers(modelo)

  Term        Estimate    Std. Error  t value   Pr(>|t|)
1 (Intercept)  8.44514390 4.021866309  2.099807 0.0360329457
2 TSLA        -0.02177608 0.005783516 -3.765197 0.0001775872
3 TM           0.07034011 0.037839392  1.858912 0.0633766642
4 GM           0.17566844 0.054624276  3.215941 0.0013481119
```

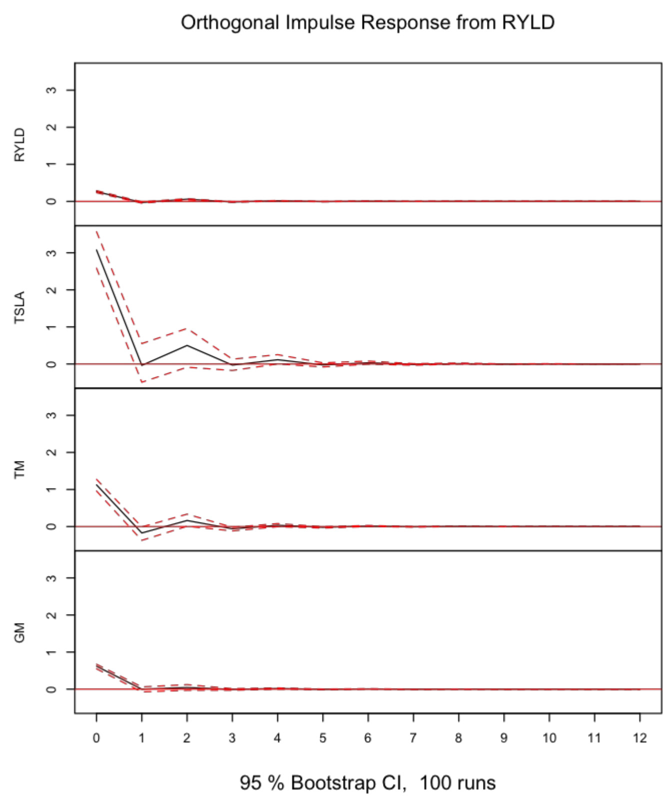
Analizando la respuesta a largo plazo, podemos evidenciar:

- TSLA tiene un efecto negativo y significativo a largo plazo sobre RYLD.
- GM tiene un efecto positivo y significativo.
- TM tiene un efecto positivo pero marginalmente significativo.

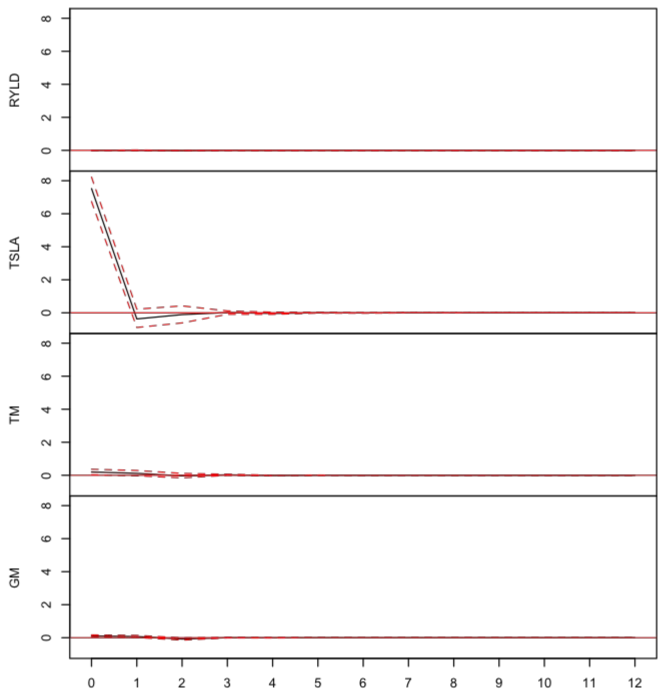
El modelo sugiere que los movimientos en TSLA y GM son los más relevantes para explicar cambios a largo plazo en RYLD.

7. Realiza un análisis de impulso-respuesta sobre el modelo vectorial.

```
# La función de impulso respuesta se lleva a cabo mediante la librería
de vars. La función es la irf(). Apliquemos la función al primer
modelo.
m1irf = irf(m0, n.ahead = 12, boot = TRUE)
plot(m1irf)
```

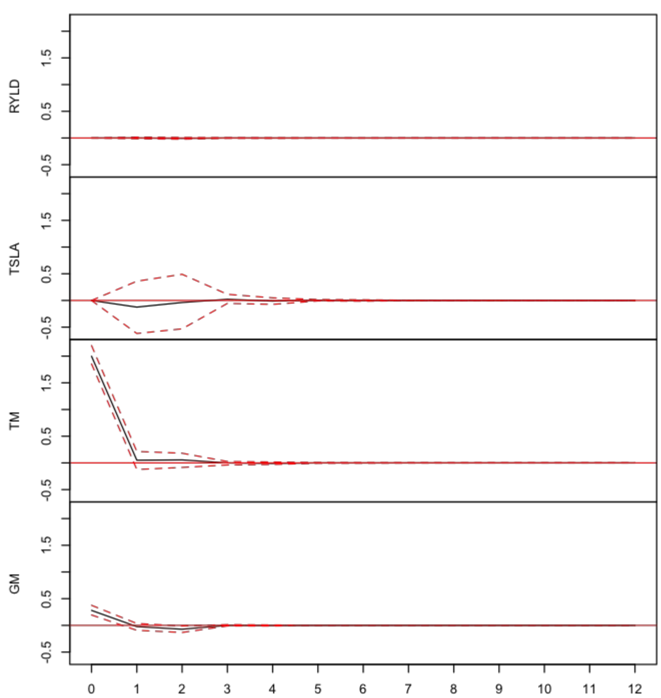


Orthogonal Impulse Response from TSLA

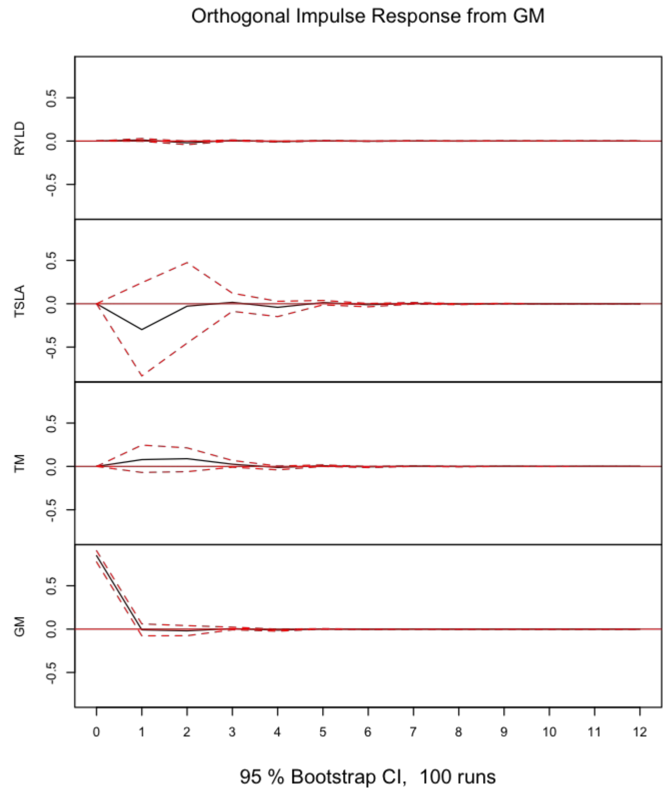


95 % Bootstrap CI, 100 runs

Orthogonal Impulse Response from TM



95 % Bootstrap CI, 100 runs



De la funcion de impulso respuesta (irf) Podemos evidenciar que un impuso a la variable RYLD genera que esta caiga en el primer periodo, en el segundo sbe nuevamente y luego se estabiliza directamente. El precio de TSLA cae y se recupera en el 5 periodo. Por otro lado el precio de TM igualmente cae y se recupera en el periodo 4. Para la variable GM, auqnue tambien cae, se recupera mas pronto, en el periodo tres.

Cuando aplicamos un cambio en TSLA podemos ver que afecta fuertemente su precio y no genera una afectacion significativa en las otras variables.

Cuando aplicamos un cambio en TM podemos ver que afecta fuertemente su precio y afecta en menor medida los precios de TSLA y GM. Los precios de RYLD no se ven afectados.

Cuando aplicamos un cambio en GM podemos ver que afecta fuertemente su precio y afecta en menor medida los precios de TM (que supen un poco) y TSLA (que cae). Los precios de RYLD tienne una pequeña afectacion.

8.Empleando tus resultados, calcula las métricas para la medición de asertividad de un pronóstico, al menos debes calcular dos métricos de error de pronóstico, ejemplo: MAPE, RMSE, MAE, entre otros.

```
diff_test <- diff(test_prices)[-1, ]  # igual que hiciste con el train
# Número de pasos a predecir = longitud del conjunto de prueba
h <- nrow(diff_test)
```

```r
# Predicción con el modelo VAR
forecast_var <- predict(m0, n.ahead = h)

# Extraer las predicciones para cada variable y combinarlas
pred_matrix <- sapply(forecast_var$fcst, function(x) x[, "fcst"])
pred_df <- as.data.frame(pred_matrix)
colnames(pred_df) <- colnames(diff_test)  # para alinear

# Convertir ambas a data.frame numérico
pred_df <- as.data.frame(lapply(pred_df, as.numeric))
diff_test <- as.data.frame(lapply(diff_test, as.numeric))

# Calcular RMSE por cada serie
rmse <- sqrt(colMeans((pred_df - diff_test)^2))

# Calcular MAE por cada serie
mae <- colMeans(abs(pred_df - diff_test))

print("RMSE modelo VAR:")
print(rmse)

print("MAE modelo VAR:")
print(mae)
```

```
[1] "RMSE modelo VAR:"
     RYLD      TSLA        TM        GM
0.1320597 7.4167455 1.8893860 0.8188753
[1] "MAE modelo VAR:"
     RYLD      TSLA        TM        GM
0.1001395 5.6215463 1.4480250 0.6387399
```

```r
# ARDL RMSE - MAE

pred <- mod1$fitted.values

# Valores predichos y reales
pred_vec <- coredata(pred)
real_vec <- coredata(train_prices$RYLD[index(pred)])

# RMSE
rmse <- sqrt(mean((pred_vec - real_vec)^2, na.rm = TRUE))

# MAE
mae <- mean(abs(pred_vec - real_vec), na.rm = TRUE)

print(paste("RMSE modelo ARDL:", rmse))
print(paste("MAE modelo ARDL:", mae))
```

```
[1] "RMSE modelo ARDL: 0.200341050952005"
[1] "MAE modelo ARDL: 0.141832546232689"
```

Paso 2. Comparación de los modelos y selección del mejor

1 Compara y sustenta qué modelo presentó un mejor desempeño de acuerdo a los criterios de selección pertinentes.

Conclusión Tras comparar el desempeño de los modelos VAR y ARDL en la predicción de la variable dependiente RYLD, se concluye lo siguiente:

El modelo VAR obtiene mejores métricas de error (menores valores de RMSE y MAE) para RYLD, lo cual indica una mayor precisión predictiva respecto al modelo ARDL.

Por lo tanto, el modelo VAR es el más adecuado para este conjunto de datos y objetivo de pronóstico.

Comparación de desempeño de modelos (Variable RYLD)

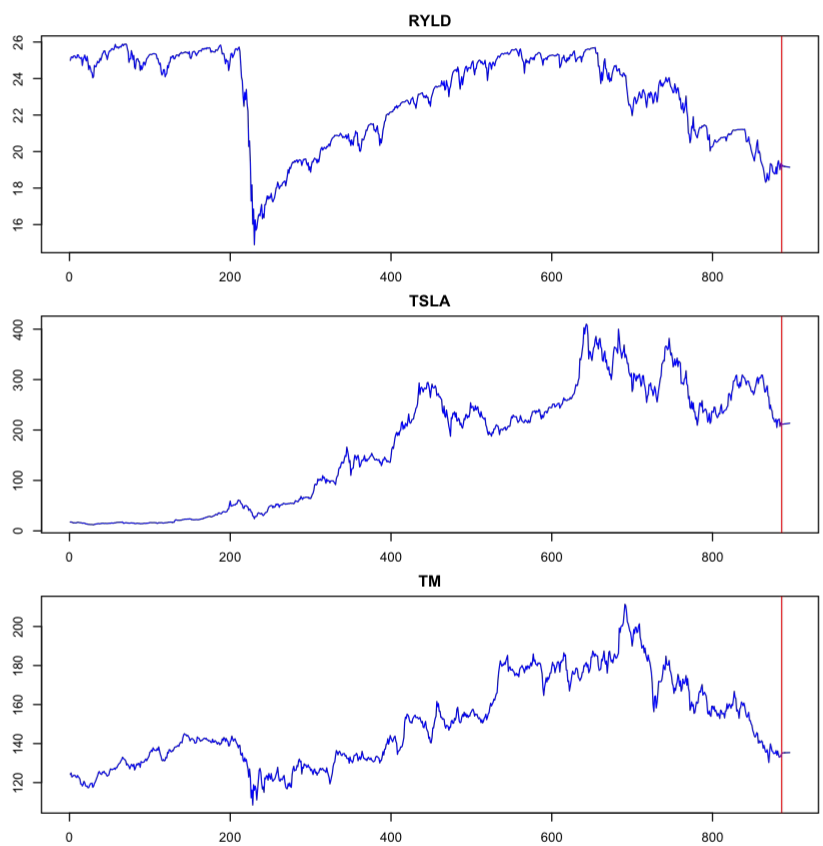
Modelo	RMSE	MAE
VAR	0.1321	0.1001
ARDL	0.2003	0.1418

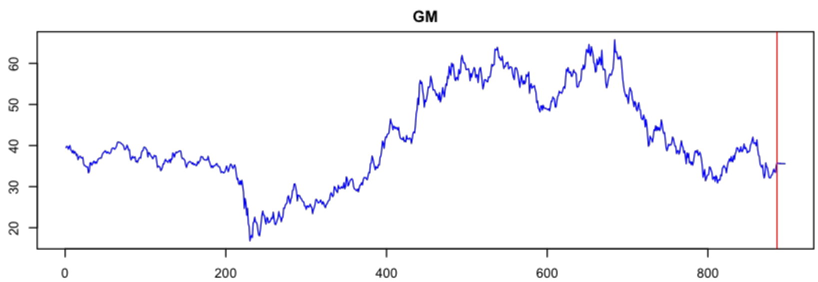
2. Genera un pronóstico de 10 días hacia adelante y una visualización del mismo.

```r
forecast_var <- predict(m0, n.ahead = 10)
#Como diferenciamos, recuperamos el nivel.
nhor=10 #pasos en el pronóstico.
nr_lev <- nrow(train_prices)
mr_lev= as.matrix(train_prices)
str(mr_lev)
m.varf_lev_ft <- rbind(mr_lev, matrix(NA, nhor, ncol(mr_lev)))
m.ft_df <- do.call(cbind,lapply(forecast_var$fcst,
                                function(x) x[,"fcst"]))
```

```
 num [1:886, 1:4] 25 25.2 25.2 25.1 25.2 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:886] "2019-04-22" "2019-04-23" "2019-04-24" "2019-04-25" ...
  ..$ : chr [1:4] "RYLD" "TSLA" "TM" "GM"
```

```r
for(h in (nr_lev+1):(nr_lev+nhor)) {
  hf <- h - nr_lev
  m.varf_lev_ft[h,] <- m.varf_lev_ft[h-1,] + m.ft_df[hf,]
}

str.main=c("RYLD", "TSLA", "TM", "GM")
par(mfrow=c(3,1), mar=c(2,2,2,2))

for(i in 1:4) {
  df <- m.varf_lev_ft[,i]
  matplot(df, type=c("l"), col = c("blue"),
          main = str.main[i])
  abline(v=nr_lev, col="red")
}
```




GM
20
30
40
50
60
0
200
400
600
800