Modelos Estructurales

OKONOMETRIC CISE

William X. Ramos Chucuri

2025-10-24

# **1. Modelos Estructurales**

* Modelos estructurales vs Modelos Estadísticos
* La estimación estructural tiene múltiples limitaciones.
* El uso de modelos parsimoniosos tiene un costo, podemos caer el modelos sub-parametrizados, es decir podemos estimar modelos sesgados e inconsistentes. (Condiciones fundamentales en econometría).
* La exogeneidad estricta es un supuesto muy restrictivo en un modelo multiecuacional.
* Los modelos VAR son modelos estadísticos.
  + El poner restricciones al modelo (Matrices o condiciones- nos ayudarían a transformarlo en un modelo estructural **SVAR.**
* Los modelos de Equilibrio general son modelos estructurales | Estos modelos me permiten modelar situaciones como la **pandemia COVID-19.**
  + Permiten modelar situaciones sobre las cuales no tenemos historia, es decir una estructura consistente.

## 1.2.Modelo VAR(p)

Un modelo autorregresivo vectorial de orden ( p ), denotado como ( VAR(p) ), se define de la siguiente forma:

donde:

* es un vector de variables endógenas de dimensión .
* es un vector de constantes de dimensión .
* son las matrices de coeficientes de dimensión para cada rezago .
* son los términos de error o innovaciones, con media cero y matriz de covarianza:

donde es simétrica y definida positiva.

En forma compacta, el modelo VAR(p) capta la interdependencia dinámica entre las variables endógenas en el tiempo.

## 1.3. Propiedades Básicas

* **Endogeneidad total:** todas las variables son explicadas por sus propios rezagos y los de las demás.
* **Identificación reducida:** los residuos pueden estar correlacionados, lo que impide inferencias estructurales.
* **Necesidad de estacionariedad:** Las series deben ser integradas de orden 0 o cointegradas (en cuyo caso se usa VECM).
* Hay que recordar que al ser integradas I (1) significa que tienen raíz unitaria y son procesos no estacionarios. Si queremos trabajar en modelos multiecuacionales debemos quitar esta raíz unitaria; por eso mencionamos que necesitamos que estas series sean procesos estacionarios.
* Es decir que su media y varianza esten mas o menos alrededor de cero y constantes en el tiempo.

## 1.4. Proceso de estimación

Cada ecuación del VAR se estima por Mínimos Cuadrados Ordinarios (OLS), dado que los regresores son idénticos entre ecuaciones y se espera que el sistema no presente endogeneidad contemporánea.

El VAR describe la **dinámica conjunta de las series** y permite análisis como:

* impulso-respuesta (IRF)- Una aproximación estructural
* descomposición de varianza (FEVD),
* pronósticos conjuntos.

## 1.5. El problema de identificación

Un VAR reducido no permite identificar los efectos contemporáneos ni distinguir causalidad estructural.

## 1.6. Identificación en un modelo VAR estructural

Un VAR con ( n ) variables contiene:

parámetros en las matrices ( A ) y ( B ).

La matriz de covarianza de los errores, denotada como , solo aporta:

ecuaciones independientes.

Por lo tanto, para lograr la identificación del modelo, se requieren al menos:

restricciones adicionales, que generalmente se imponen sobre las matrices ( A ) o ( B ) (dependiendo del tipo de identificación: contemporánea, de largo plazo, o basada en restricciones de signo).

En un modelo VAR estructural (SVAR), el objetivo es recuperar las relaciones estructurales contemporáneas entre las variables, es decir, cómo los choques estructurales (innovaciones puras) afectan simultáneamente a las variables del sistema.

Sin embargo, el modelo estimado directamente (el VAR reducido) solo nos proporciona información sobre la **covarianza observada de los errores**, .  
Esta matriz tiene elementos únicos, ya que es **simétrica**.

El problema surge porque las matrices ( A ) y ( B ) contienen en total ( 2n^2 ) parámetros desconocidos, mientras que solo contamos con ecuaciones provenientes de .  
Esto implica que el sistema está **subidentificado**: hay más parámetros desconocidos que ecuaciones disponibles.

Para resolver este problema, se deben imponer **restricciones adicionales** que permitan identificar cada parámetro estructural.  
En concreto, se necesitan al menos:

restricciones adicionales para que el número de ecuaciones iguale al número de parámetros y el sistema quede **justamente identificado**.

### 🧩 Tipos de restricciones comunes

1. **Restricciones contemporáneas (modelo tipo AB o recursivo - Cholesky):**  
   Se impone una estructura triangular en ( A ) o ( B ) asumiendo que ciertas variables no responden contemporáneamente a otras.
2. **Restricciones de largo plazo (modelo tipo Blanchard-Quah):**  
   Se imponen condiciones sobre la respuesta acumulada a los choques, por ejemplo, que ciertos choques no afectan permanentemente algunas variables.
3. **Restricciones de signo (modelo de Uhlig):**  
   En lugar de ceros, se imponen signos (+/–) sobre las respuestas o sobre los impactos contemporáneos.

### 🧠 Ejemplo ilustrativo

Para un VAR de **2 variables (n = 2)**:

* Parámetros totales:
* Ecuaciones de la covarianza:
* Restricciones necesarias:

Por tanto, se necesita **al menos una restricción adicional** (por ejemplo, asumir que una variable no reacciona contemporáneamente a otra) para lograr la identificación.

💡 **Conclusión:**

* La identificación en los modelos VAR estructurales es esencial para poder interpretar los choques estimados como “causales” o “estructurales”.
* Sin restricciones suficientes, solo observamos correlaciones reducidas, no relaciones estructurales.

**Ejemplo empíco**

# =========================================================  
# 0) Paquetes y setup  
# =========================================================  
# install.packages(c("zoo","ggplot2","tseries","vars","urca","forecast","dplyr","tidyr"))  
library(zoo)

Adjuntando el paquete: 'zoo'

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

library(ggplot2)  
library(tseries)

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

library(vars)

Cargando paquete requerido: MASS

Cargando paquete requerido: strucchange

Cargando paquete requerido: sandwich

Cargando paquete requerido: urca

Cargando paquete requerido: lmtest

library(urca)  
library(forecast)  
library(dplyr)

Adjuntando el paquete: 'dplyr'

The following object is masked from 'package:MASS':  
  
 select

The following objects are masked from 'package:stats':  
  
 filter, lag

The following objects are masked from 'package:base':  
  
 intersect, setdiff, setequal, union

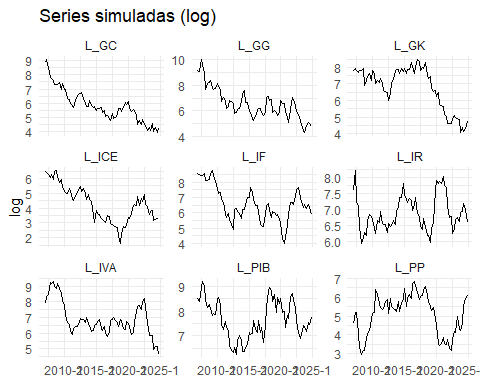
library(tidyr)  
  
set.seed(42)  
  
# =========================================================  
# 1) Calendario y variables (72 trimestres 2007Q1–2024Q4)  
# =========================================================  
T <- 72  
fq <- 4  
fechas <- as.yearqtr(seq(from = as.Date("2007-01-01"),  
 by = "quarter", length.out = T))  
vars <- c("L\_PP","L\_PIB","L\_GG","L\_GC","L\_GK","L\_IF","L\_IVA","L\_ICE","L\_IR")  
k <- length(vars)  
  
# =========================================================  
# 2) DGP: I(1) con 2 relaciones de cointegración plausibles  
# =========================================================  
# β (columnas = vectores cointegrados)  
beta <- matrix(0, nrow = k, ncol = 2, dimnames = list(vars, c("CI\_gasto","CI\_tributos")))  
beta["L\_GG","CI\_gasto"] <- 1; beta["L\_GC","CI\_gasto"] <- -0.7; beta["L\_GK","CI\_gasto"] <- -0.3  
beta["L\_IF","CI\_tributos"] <- 1; beta["L\_IVA","CI\_tributos"] <- -0.5; beta["L\_IR","CI\_tributos"] <- -0.3; beta["L\_ICE","CI\_tributos"] <- -0.2  
  
# α (velocidades de ajuste)  
alpha <- matrix(0, nrow = k, ncol = 2, dimnames = list(vars, c("CI\_gasto","CI\_tributos")))  
alpha["L\_GG","CI\_gasto"] <- -0.25  
alpha["L\_GC","CI\_gasto"] <- -0.10  
alpha["L\_GK","CI\_gasto"] <- -0.05  
alpha["L\_IF","CI\_tributos"] <- -0.20  
alpha["L\_IVA","CI\_tributos"] <- -0.10  
alpha["L\_IR","CI\_tributos"] <- -0.08  
alpha["L\_ICE","CI\_tributos"] <- -0.05  
alpha["L\_PIB","CI\_gasto"] <- -0.02  
alpha["L\_PIB","CI\_tributos"] <- -0.01  
alpha["L\_PP",] <- c(0,0) # petróleo no corrige directamente  
  
# Γ1 (dinámica de corto plazo sobre ΔY\_{t-1})  
Gamma1 <- matrix(0, nrow = k, ncol = k, dimnames = list(vars, vars))  
Gamma1["L\_PIB","L\_PP"] <- 0.10  
Gamma1["L\_IF","L\_PP"] <- 0.08  
Gamma1["L\_IF","L\_IVA"] <- 0.10; Gamma1["L\_IF","L\_IR"] <- 0.07; Gamma1["L\_IF","L\_ICE"] <- 0.05  
Gamma1["L\_GG","L\_PIB"] <- 0.06  
Gamma1["L\_GC","L\_GG"] <- 0.10; Gamma1["L\_GK","L\_GG"] <- 0.06  
  
# Σ (varianzas-covarianzas de shocks)  
Sigma <- diag(c(0.20, 0.18, 0.15, 0.12, 0.12, 0.18, 0.15, 0.12, 0.12))  
dimnames(Sigma) <- list(vars, vars)  
Sigma["L\_PIB","L\_PP"] <- Sigma["L\_PP","L\_PIB"] <- 0.05  
Sigma["L\_IF","L\_PP"] <- Sigma["L\_PP","L\_IF"] <- 0.04  
Sigma["L\_IF","L\_IVA"] <- Sigma["L\_IVA","L\_IF"] <- 0.06  
Sigma["L\_GG","L\_PIB"] <- Sigma["L\_PIB","L\_GG"] <- 0.04  
Sigma <- (Sigma + t(Sigma))/2  
C <- t(chol(Sigma))  
  
# =========================================================  
# 3) Simulación VECM (ΔY\_t = Γ1 ΔY\_{t-1} + α β' Y\_{t-1} + ε\_t)  
# =========================================================  
Y <- matrix(0, nrow = k, ncol = T, dimnames = list(vars, NULL))  
dY <- matrix(0, nrow = k, ncol = T, dimnames = list(vars, NULL))  
Y[,1] <- c(4.6, 8.5, 9.2, 8.9, 7.8, 8.6, 7.9, 6.5, 7.6)  
  
for (t in 2:T) {  
 eps\_t <- C %\*% rnorm(k)  
 EC\_lag <- t(beta) %\*% Y[, t-1] # (2x9)\*(9x1) = (2x1)  
 dY[,t] <- Gamma1 %\*% dY[,t-1] + alpha %\*% EC\_lag + eps\_t  
 Y[,t] <- Y[,t-1] + dY[,t]  
}

**Series a modelar**

Identificación económica que usaré (ajústala si quieres):

* **Petróleo (L\_PP)** es “exógeno contemporáneo”: no recibe shocks contemporáneos de otras variables (solo su propio shock).
* **PIB (L\_PIB)** puede reaccionar contemporáneamente a **petróleo**.
* **Gasto total (L\_GG)** reacciona contemporáneamente al **PIB** (reglas/prociclicidad).
* **Gasto corriente / de capital (L\_GC, L\_GK)** reaccionan a **G\_G**.
* **Ingresos (L\_IF)** reaccionan a **petróleo** (vía actividad/sector petrolero) y a **impuestos indirectos** (**L\_IVA, L\_IR, L\_ICE**).
* Para el resto, dejamos parámetros “libres” (NA) solo donde tiene sentido; el resto se fijan en 0.

Y\_ts <- ts(t(Y), start = c(2007,1), frequency = fq); colnames(Y\_ts) <- vars  
base\_wide <- data.frame(  
 fecha\_q = fechas,  
 year = as.integer(format(fechas, "%Y")),  
 quarter = as.integer(cycle(Y\_ts)),  
 as.data.frame(Y\_ts),  
 check.names = FALSE  
)  
  
base\_long <- base\_wide |>  
 pivot\_longer(cols = all\_of(vars), names\_to = "variable", values\_to = "valor")  
  
ggplot(base\_long, aes(x = fecha\_q, y = valor)) +  
 geom\_line() + facet\_wrap(~ variable, scales = "free\_y", ncol = 3) +  
 labs(title = "Series simuladas (log)", x = NULL, y = "log") +  
 theme\_minimal()



**Estacionariedad en niveles y en diferencias**

adf\_levels <- sapply(vars, function(v) adf.test(Y\_ts[,v])$p.value)  
adf\_diffs <- sapply(vars, function(v) adf.test(diff(Y\_ts[,v])[-1])$p.value)

Warning in adf.test(diff(Y\_ts[, v])[-1]): p-value smaller than printed p-value  
Warning in adf.test(diff(Y\_ts[, v])[-1]): p-value smaller than printed p-value

cat("\nADF p-valores en niveles:\n"); print(round(adf\_levels,4))

ADF p-valores en niveles:

L\_PP L\_PIB L\_GG L\_GC L\_GK L\_IF L\_IVA L\_ICE L\_IR   
0.5077 0.4831 0.1429 0.2622 0.8133 0.3314 0.1580 0.6756 0.1775

cat("\nADF p-valores en diferencias:\n"); print(round(adf\_diffs,4))

ADF p-valores en diferencias:

L\_PP L\_PIB L\_GG L\_GC L\_GK L\_IF L\_IVA L\_ICE L\_IR   
0.0302 0.0100 0.0100 0.0162 0.1010 0.1381 0.1824 0.0965 0.0104

## 📘 **1️⃣ Hipótesis del test ADF**

* **H₀ (nula):** La serie tiene una raíz unitaria → no estacionaria.
* **H₁ (alternativa):** La serie es estacionaria.

## 📊 **2️⃣ Resultados por niveles**

| Variable | p-valor | Interpretación |
| --- | --- | --- |
| L\_PP | 0.5077 | No se rechaza H₀ → no estacionaria |
| L\_PIB | 0.4831 | No estacionaria |
| L\_GG | 0.1429 | No estacionaria |
| L\_GC | 0.2622 | No estacionaria |
| L\_GK | 0.8133 | No estacionaria |
| L\_IF | 0.3314 | No estacionaria |
| L\_IVA | 0.1580 | No estacionaria |
| L\_ICE | 0.6756 | No estacionaria |
| L\_IR | 0.1775 | No estacionaria |

🔹 **Conclusión en niveles:**  
Todas las variables tienen p-valores > 0.05 → **no se rechaza la hipótesis nula de raíz unitaria**.  
➡️ **Ninguna serie es estacionaria en niveles.**

## 📈 **3️⃣ Resultados en primeras diferencias**

| Variable | p-valor | Interpretación |
| --- | --- | --- |
| L\_PP | 0.0302 | Estacionaria |
| L\_PIB | 0.0100 | Estacionaria |
| L\_GG | 0.0100 | Estacionaria |
| L\_GC | 0.0162 | Estacionaria |
| L\_GK | 0.1010 | No estacionaria al 5% (pero casi) |
| L\_IF | 0.1381 | No estacionaria |
| L\_IVA | 0.1824 | No estacionaria |
| L\_ICE | 0.0965 | No estacionaria (marginal al 10%) |
| L\_IR | 0.0104 | Estacionaria |

🔹 **Conclusión en diferencias:**  
La mayoría de las series

(L\_PP, L\_PIB, L\_GG, L\_GC, L\_IR)

Son estacionarias en primeras diferencias (p < 0.05).  
Algunas (L\_GK, L\_IF, L\_IVA, L\_ICE) presentan p-valores entre 0.09–0.18, lo que podría considerarse estacionarias al 10% o requerir una segunda diferencia o un ajuste de rezagos.

## **4️⃣ Implicación para el modelo VAR**

* Dado que las series no son estacionarias en niveles pero sí lo son en diferencias, se clasifican como I(1) (integradas de orden uno).
* Esto sugiere que podría existir cointegración entre ellas, por lo cual el paso siguiente sería aplicar el test de Johansen:

**Seleccionamos los rezagos en niveles (para el Test de Johansen)**

sel\_lvl <- VARselect(Y\_ts, lag.max = 4, type = "const")  
cat("\nCriterios de información (niveles):\n"); print(sel\_lvl$criteria)

Criterios de información (niveles):

1 2 3 4  
AIC(n) -1.864410e+01 -1.800915e+01 -1.739430e+01 -1.776323e+01  
HQ(n) -1.748014e+01 -1.579762e+01 -1.413521e+01 -1.345658e+01  
SC(n) -1.570651e+01 -1.242774e+01 -9.169070e+00 -6.894172e+00  
FPE(n) 8.153935e-09 1.731364e-08 4.454123e-08 6.316023e-08

p\_opt <- as.integer(sel\_lvl$selection["AIC(n)"])  
Kj <- max(2, p\_opt)  
cat("\nUsando K =", Kj, "para Johansen.\n")

Usando K = 2 para Johansen.

j\_tr <- ca.jo(Y\_ts, type = "trace", K = Kj, ecdet = "const", spec = "transitory")  
j\_eig <- ca.jo(Y\_ts, type = "eigen", K = Kj, ecdet = "const", spec = "transitory")  
cat("\n--- Johansen TRACE ---\n"); print(summary(j\_tr))

--- Johansen TRACE ---

######################   
# Johansen-Procedure #   
######################   
  
Test type: trace statistic , without linear trend and constant in cointegration   
  
Eigenvalues (lambda):  
 [1] 5.909955e-01 4.423184e-01 3.813286e-01 3.171301e-01 2.314686e-01  
 [6] 2.030888e-01 1.567262e-01 9.683625e-02 4.903297e-02 -8.006596e-17  
  
Values of teststatistic and critical values of test:  
  
 test 10pct 5pct 1pct  
r <= 8 | 3.52 7.52 9.24 12.97  
r <= 7 | 10.65 17.85 19.96 24.60  
r <= 6 | 22.58 32.00 34.91 41.07  
r <= 5 | 38.47 49.65 53.12 60.16  
r <= 4 | 56.90 71.86 76.07 84.45  
r <= 3 | 83.60 97.18 102.14 111.01  
r <= 2 | 117.22 126.58 131.70 143.09  
r <= 1 | 158.09 159.48 165.58 177.20  
r = 0 | 220.68 196.37 202.92 215.74  
  
Eigenvectors, normalised to first column:  
(These are the cointegration relations)  
  
 L\_PP.l1 L\_PIB.l1 L\_GG.l1 L\_GC.l1 L\_GK.l1  
L\_PP.l1 1.0000000 1.00000000 1.00000000 1.0000000 1.0000000  
L\_PIB.l1 0.7324992 -2.08073054 0.06556848 -0.8190391 0.7237758  
L\_GG.l1 -0.2778916 3.36639420 -1.31418394 -0.2134218 -0.8948099  
L\_GC.l1 -0.1847404 -1.88785850 -0.48758497 1.3354645 -1.3605263  
L\_GK.l1 -0.2480989 -1.37287595 0.01989781 -1.0902060 0.4782761  
L\_IF.l1 -0.1274489 0.06782783 0.80838209 -0.3852260 -1.6148731  
L\_IVA.l1 0.5791197 1.20062298 1.23863346 1.4955178 1.5141218  
L\_ICE.l1 0.7588437 -1.09644442 0.41950093 -1.0926320 2.1507986  
L\_IR.l1 1.3174608 -0.32179846 -0.11028251 0.4897667 0.6610095  
constant -21.3593180 7.20167336 -9.16030864 -4.6071778 -14.4253446  
 L\_IF.l1 L\_IVA.l1 L\_ICE.l1 L\_IR.l1 constant  
L\_PP.l1 1.00000000 1.0000000 1.0000000 1.0000000 1.000000000  
L\_PIB.l1 2.31180164 2.1965497 0.8382612 -0.4410457 0.441582292  
L\_GG.l1 -0.55387332 0.1661096 -0.1437685 0.1185211 -0.563672905  
L\_GC.l1 -0.54347756 -0.4730255 1.6080807 -0.8404113 -0.240905965  
L\_GK.l1 -0.11323347 1.0887838 -1.1336760 0.9718469 -1.549307168  
L\_IF.l1 -0.96615237 1.3870214 -0.8986143 -1.3342308 -0.491574728  
L\_IVA.l1 0.97208201 -4.9683112 -0.3383076 1.9828351 -0.690720871  
L\_ICE.l1 0.53517813 1.8773423 1.0909467 -0.1795450 0.840946996  
L\_IR.l1 -0.01142896 -0.8260108 -3.7907235 0.2523096 0.009541541  
constant -18.09336666 -5.5963588 18.1598105 -9.0688320 12.989079589  
  
Weights W:  
(This is the loading matrix)  
  
 L\_PP.l1 L\_PIB.l1 L\_GG.l1 L\_GC.l1 L\_GK.l1  
L\_PP.d -0.170837587 -0.003662898 0.0575157278 -0.131035864 0.013545373  
L\_PIB.d -0.005989066 -0.017212711 0.0395195792 0.093970593 -0.008844093  
L\_GG.d -0.014904228 -0.131448226 0.0984319174 0.031597461 0.065662423  
L\_GC.d -0.202032138 0.037061936 0.0239556762 -0.053822660 0.023277256  
L\_GK.d 0.107879149 0.040323523 0.0399504908 0.097749889 0.035514573  
L\_IF.d -0.130986197 0.022476659 -0.0809102756 -0.027650974 0.078040213  
L\_IVA.d 0.034755025 -0.021334929 -0.1195008720 -0.006213521 0.037895083  
L\_ICE.d -0.086456592 -0.026877165 -0.0523795313 0.110418160 -0.018931774  
L\_IR.d -0.244164374 -0.033087565 -0.0007176711 0.055140201 -0.006378733  
 L\_IF.l1 L\_IVA.l1 L\_ICE.l1 L\_IR.l1 constant  
L\_PP.d -0.065563423 -0.012443347 -0.015033901 -0.0218652619 7.964410e-16  
L\_PIB.d -0.057023595 0.001508587 -0.029896084 0.0119994823 -2.777417e-17  
L\_GG.d 0.024103879 -0.007603056 -0.010517381 0.0042684620 2.576200e-15  
L\_GC.d 0.025564965 0.013161977 -0.008089357 0.0011980029 5.434942e-16  
L\_GK.d -0.033141892 0.009359561 0.011561300 -0.0161955978 5.953449e-16  
L\_IF.d -0.050478050 -0.015256264 -0.014706610 0.0046688332 -2.897984e-16  
L\_IVA.d -0.023889710 0.023095483 -0.008423701 -0.0001829324 -2.094494e-16  
L\_ICE.d 0.069818273 -0.004381045 -0.008119242 -0.0154794202 -1.182713e-15  
L\_IR.d -0.007205277 0.003809481 0.018052746 0.0066262475 -6.074391e-16

cat("\n--- Johansen EIGEN ---\n"); print(summary(j\_eig))

--- Johansen EIGEN ---

######################   
# Johansen-Procedure #   
######################   
  
Test type: maximal eigenvalue statistic (lambda max) , without linear trend and constant in cointegration   
  
Eigenvalues (lambda):  
 [1] 5.909955e-01 4.423184e-01 3.813286e-01 3.171301e-01 2.314686e-01  
 [6] 2.030888e-01 1.567262e-01 9.683625e-02 4.903297e-02 -8.006596e-17  
  
Values of teststatistic and critical values of test:  
  
 test 10pct 5pct 1pct  
r <= 8 | 3.52 7.52 9.24 12.97  
r <= 7 | 7.13 13.75 15.67 20.20  
r <= 6 | 11.93 19.77 22.00 26.81  
r <= 5 | 15.89 25.56 28.14 33.24  
r <= 4 | 18.43 31.66 34.40 39.79  
r <= 3 | 26.70 37.45 40.30 46.82  
r <= 2 | 33.61 43.25 46.45 51.91  
r <= 1 | 40.88 48.91 52.00 57.95  
r = 0 | 62.58 54.35 57.42 63.71  
  
Eigenvectors, normalised to first column:  
(These are the cointegration relations)  
  
 L\_PP.l1 L\_PIB.l1 L\_GG.l1 L\_GC.l1 L\_GK.l1  
L\_PP.l1 1.0000000 1.00000000 1.00000000 1.0000000 1.0000000  
L\_PIB.l1 0.7324992 -2.08073054 0.06556848 -0.8190391 0.7237758  
L\_GG.l1 -0.2778916 3.36639420 -1.31418394 -0.2134218 -0.8948099  
L\_GC.l1 -0.1847404 -1.88785850 -0.48758497 1.3354645 -1.3605263  
L\_GK.l1 -0.2480989 -1.37287595 0.01989781 -1.0902060 0.4782761  
L\_IF.l1 -0.1274489 0.06782783 0.80838209 -0.3852260 -1.6148731  
L\_IVA.l1 0.5791197 1.20062298 1.23863346 1.4955178 1.5141218  
L\_ICE.l1 0.7588437 -1.09644442 0.41950093 -1.0926320 2.1507986  
L\_IR.l1 1.3174608 -0.32179846 -0.11028251 0.4897667 0.6610095  
constant -21.3593180 7.20167336 -9.16030864 -4.6071778 -14.4253446  
 L\_IF.l1 L\_IVA.l1 L\_ICE.l1 L\_IR.l1 constant  
L\_PP.l1 1.00000000 1.0000000 1.0000000 1.0000000 1.000000000  
L\_PIB.l1 2.31180164 2.1965497 0.8382612 -0.4410457 0.441582292  
L\_GG.l1 -0.55387332 0.1661096 -0.1437685 0.1185211 -0.563672905  
L\_GC.l1 -0.54347756 -0.4730255 1.6080807 -0.8404113 -0.240905965  
L\_GK.l1 -0.11323347 1.0887838 -1.1336760 0.9718469 -1.549307168  
L\_IF.l1 -0.96615237 1.3870214 -0.8986143 -1.3342308 -0.491574728  
L\_IVA.l1 0.97208201 -4.9683112 -0.3383076 1.9828351 -0.690720871  
L\_ICE.l1 0.53517813 1.8773423 1.0909467 -0.1795450 0.840946996  
L\_IR.l1 -0.01142896 -0.8260108 -3.7907235 0.2523096 0.009541541  
constant -18.09336666 -5.5963588 18.1598105 -9.0688320 12.989079589  
  
Weights W:  
(This is the loading matrix)  
  
 L\_PP.l1 L\_PIB.l1 L\_GG.l1 L\_GC.l1 L\_GK.l1  
L\_PP.d -0.170837587 -0.003662898 0.0575157278 -0.131035864 0.013545373  
L\_PIB.d -0.005989066 -0.017212711 0.0395195792 0.093970593 -0.008844093  
L\_GG.d -0.014904228 -0.131448226 0.0984319174 0.031597461 0.065662423  
L\_GC.d -0.202032138 0.037061936 0.0239556762 -0.053822660 0.023277256  
L\_GK.d 0.107879149 0.040323523 0.0399504908 0.097749889 0.035514573  
L\_IF.d -0.130986197 0.022476659 -0.0809102756 -0.027650974 0.078040213  
L\_IVA.d 0.034755025 -0.021334929 -0.1195008720 -0.006213521 0.037895083  
L\_ICE.d -0.086456592 -0.026877165 -0.0523795313 0.110418160 -0.018931774  
L\_IR.d -0.244164374 -0.033087565 -0.0007176711 0.055140201 -0.006378733  
 L\_IF.l1 L\_IVA.l1 L\_ICE.l1 L\_IR.l1 constant  
L\_PP.d -0.065563423 -0.012443347 -0.015033901 -0.0218652619 7.964410e-16  
L\_PIB.d -0.057023595 0.001508587 -0.029896084 0.0119994823 -2.777417e-17  
L\_GG.d 0.024103879 -0.007603056 -0.010517381 0.0042684620 2.576200e-15  
L\_GC.d 0.025564965 0.013161977 -0.008089357 0.0011980029 5.434942e-16  
L\_GK.d -0.033141892 0.009359561 0.011561300 -0.0161955978 5.953449e-16  
L\_IF.d -0.050478050 -0.015256264 -0.014706610 0.0046688332 -2.897984e-16  
L\_IVA.d -0.023889710 0.023095483 -0.008423701 -0.0001829324 -2.094494e-16  
L\_ICE.d 0.069818273 -0.004381045 -0.008119242 -0.0154794202 -1.182713e-15  
L\_IR.d -0.007205277 0.003809481 0.018052746 0.0066262475 -6.074391e-16

choose\_r\_trace <- function(jobj, alpha = 0.05) {  
 ts <- jobj@teststat  
 cv <- jobj@cval[,"5pct"]  
 sum(ts > cv)  
}  
r <- max(1, choose\_r\_trace(j\_tr, 0.05)) # asegura r≥1 para poder estimar VECM  
cat("\nRango de cointegración estimado (trace, 5%): r =", r, "\n")

Rango de cointegración estimado (trace, 5%): r = 1

La salida proviene de la selección del número óptimo de rezagos en un modelo VAR.

Los criterios de información (AIC, HQ, SC, FPE) comparan la bondad de ajuste y la penalización por complejidad del modelo (número de rezagos).

El objetivo es elegir el valor de que minimiza el criterio seleccionado.

-Normalmente **AIC** (Akaike Information Criterion) o **BIC/SC** (Schwarz Criterion).

vecm\_fit <- cajorls(j\_tr, r = r)  
cat("\n--- Beta (vectores cointegrados estimados) ---\n")

--- Beta (vectores cointegrados estimados) ---

print(vecm\_fit$beta)

ect1  
L\_PP.l1 1.0000000  
L\_PIB.l1 0.7324992  
L\_GG.l1 -0.2778916  
L\_GC.l1 -0.1847404  
L\_GK.l1 -0.2480989  
L\_IF.l1 -0.1274489  
L\_IVA.l1 0.5791197  
L\_ICE.l1 0.7588437  
L\_IR.l1 1.3174608  
constant -21.3593180

var\_from\_vecm <- vec2var(j\_tr, r = r)

Función para elejir r 5%

cat("\n--- Portmanteau ---\n"); print(serial.test(var\_from\_vecm, lags.pt = 12, type = "PT.asymptotic"))

--- Portmanteau ---

Portmanteau Test (asymptotic)  
  
data: Residuals of VAR object var\_from\_vecm  
Chi-squared = 854.21, df = 819, p-value = 0.191

$serial  
  
 Portmanteau Test (asymptotic)  
  
data: Residuals of VAR object var\_from\_vecm  
Chi-squared = 854.21, df = 819, p-value = 0.191

cat("\n--- ARCH ---\n"); print(arch.test(var\_from\_vecm, lags.multi = 5))

--- ARCH ---

ARCH (multivariate)  
  
data: Residuals of VAR object var\_from\_vecm  
Chi-squared = 2925, df = 10125, p-value = 1

$arch.mul  
  
 ARCH (multivariate)  
  
data: Residuals of VAR object var\_from\_vecm  
Chi-squared = 2925, df = 10125, p-value = 1

cat("\n--- Normalidad ---\n"); print(normality.test(var\_from\_vecm))

--- Normalidad ---

$JB  
  
 JB-Test (multivariate)  
  
data: Residuals of VAR object var\_from\_vecm  
Chi-squared = 12.03, df = 18, p-value = 0.8457  
  
  
$Skewness  
  
 Skewness only (multivariate)  
  
data: Residuals of VAR object var\_from\_vecm  
Chi-squared = 7.3634, df = 9, p-value = 0.5993  
  
  
$Kurtosis  
  
 Kurtosis only (multivariate)  
  
data: Residuals of VAR object var\_from\_vecm  
Chi-squared = 4.6671, df = 9, p-value = 0.8623

$jb.mul  
$jb.mul$JB  
  
 JB-Test (multivariate)  
  
data: Residuals of VAR object var\_from\_vecm  
Chi-squared = 12.03, df = 18, p-value = 0.8457  
  
  
$jb.mul$Skewness  
  
 Skewness only (multivariate)  
  
data: Residuals of VAR object var\_from\_vecm  
Chi-squared = 7.3634, df = 9, p-value = 0.5993  
  
  
$jb.mul$Kurtosis  
  
 Kurtosis only (multivariate)  
  
data: Residuals of VAR object var\_from\_vecm  
Chi-squared = 4.6671, df = 9, p-value = 0.8623

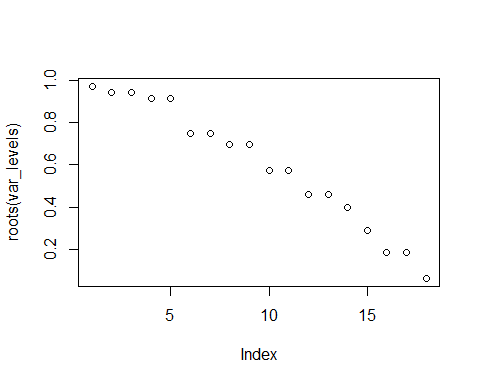
**Estimación Estabilidad**

var\_levels <- VAR(Y\_ts, p = Kj, type = "const")  
mod\_roots <- roots(var\_levels, modulus = TRUE)  
cat("\nMódulos de autovalores (VAR en niveles, p=Kj):\n"); print(round(sort(mod\_roots, decreasing = TRUE), 4))

Módulos de autovalores (VAR en niveles, p=Kj):

[1] 0.9700 0.9390 0.9390 0.9112 0.9112 0.7489 0.7489 0.6939 0.6939 0.5709  
[11] 0.5709 0.4576 0.4576 0.3993 0.2880 0.1840 0.1840 0.0639

plot(roots(var\_levels)) # puntos dentro del círculo => estable



## VAR equivalente del VECM (para IRF/FEVD/forecast)

dY\_ts <- na.omit(diff(Y\_ts))  
sel\_d <- VARselect(dY\_ts, lag.max = 4, type = "const")  
p\_d <- ifelse(is.na(as.integer(sel\_d$selection["AIC(n)"])), 2L, as.integer(sel\_d$selection["AIC(n)"]))  
cat("\nVAR(Δ) seleccionado (AIC): p\_d =", p\_d, "\n")

VAR(Δ) seleccionado (AIC): p\_d = 1

var\_d <- VAR(dY\_ts, p = p\_d, type = "const")  
stb\_diff <- stability(var\_d, type = "OLS-CUSUM")

png("stability\_VAR\_diff\_CUSUM.png", width = 1400, height = 900, res = 150)  
par(mar = c(4, 4, 2, 1))  
plot(stb\_diff)  
dev.off()

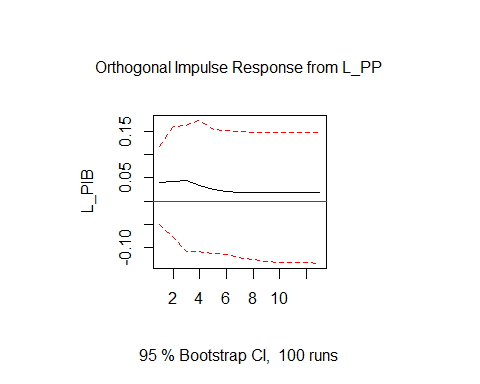
png   
 2

cat("Gráfico guardado en: stability\_VAR\_diff\_CUSUM.png\n")

Gráfico guardado en: stability\_VAR\_diff\_CUSUM.png

**IRF**

irf\_pp\_pib <- irf(var\_from\_vecm, impulse = "L\_PP", response = "L\_PIB",  
 n.ahead = 12, boot = TRUE, ci = 0.95)  
plot(irf\_pp\_pib)



fevd\_v <- fevd(var\_from\_vecm, n.ahead = 12)  
  
  
# Cierra cualquier gráfico abierto  
while (!is.null(dev.list())) dev.off()  
  
# Exporta el FEVD a un PNG  
png("FEVD\_VECM.png", width = 1400, height = 900, res = 150)  
par(mar = c(4,4,2,1)) # Márgenes reducidas  
plot(fevd\_v)  
dev.off()

pdf   
 2

cat("✅ Gráfico FEVD guardado como 'FEVD\_VECM.png' en tu directorio de trabajo.\n")

✅ Gráfico FEVD guardado como 'FEVD\_VECM.png' en tu directorio de trabajo.

# IRF: L\_PP → L\_PIB  
png("IRF\_L\_PP\_to\_L\_PIB.png", width = 1400, height = 900, res = 150)  
par(mar = c(4,4,2,1))  
plot(irf\_pp\_pib)  
dev.off()

png   
 2

cat("✅ Gráfico IRF guardado como 'IRF\_L\_PP\_to\_L\_PIB.png'\n")

✅ Gráfico IRF guardado como 'IRF\_L\_PP\_to\_L\_PIB.png'

## **Modelo SVAR**

# =========================================================  
# 0) Paquetes y setup  
# =========================================================  
# install.packages(c("zoo","ggplot2","tseries","vars","urca","forecast","dplyr","tidyr"))  
library(zoo)  
library(ggplot2)  
library(tseries)  
library(vars)  
library(urca)  
library(forecast)  
library(dplyr)  
library(tidyr)  
  
set.seed(42)  
  
# =========================================================  
# 1) Calendario y variables (72 trimestres 2007Q1–2024Q4)  
# =========================================================  
T <- 72  
fq <- 4  
fechas <- as.yearqtr(seq(from = as.Date("2007-01-01"),  
 by = "quarter", length.out = T))  
vars <- c("L\_PP","L\_PIB","L\_GG","L\_GC","L\_GK","L\_IF","L\_IVA","L\_ICE","L\_IR")  
k <- length(vars)  
  
# =========================================================  
# 2) DGP: I(1) con 2 cointegraciones plausibles (para simular)  
# =========================================================  
beta <- matrix(0, nrow = k, ncol = 2, dimnames = list(vars, c("CI\_gasto","CI\_tributos")))  
beta["L\_GG","CI\_gasto"] <- 1; beta["L\_GC","CI\_gasto"] <- -0.7; beta["L\_GK","CI\_gasto"] <- -0.3  
beta["L\_IF","CI\_tributos"] <- 1; beta["L\_IVA","CI\_tributos"] <- -0.5; beta["L\_IR","CI\_tributos"] <- -0.3; beta["L\_ICE","CI\_tributos"] <- -0.2  
  
alpha <- matrix(0, nrow = k, ncol = 2, dimnames = list(vars, c("CI\_gasto","CI\_tributos")))  
alpha["L\_GG","CI\_gasto"] <- -0.25  
alpha["L\_GC","CI\_gasto"] <- -0.10  
alpha["L\_GK","CI\_gasto"] <- -0.05  
alpha["L\_IF","CI\_tributos"] <- -0.20  
alpha["L\_IVA","CI\_tributos"] <- -0.10  
alpha["L\_IR","CI\_tributos"] <- -0.08  
alpha["L\_ICE","CI\_tributos"] <- -0.05  
alpha["L\_PIB","CI\_gasto"] <- -0.02  
alpha["L\_PIB","CI\_tributos"] <- -0.01  
alpha["L\_PP",] <- c(0,0) # petróleo no corrige directamente  
  
Gamma1 <- matrix(0, nrow = k, ncol = k, dimnames = list(vars, vars))  
Gamma1["L\_PIB","L\_PP"] <- 0.10  
Gamma1["L\_IF","L\_PP"] <- 0.08  
Gamma1["L\_IF","L\_IVA"] <- 0.10; Gamma1["L\_IF","L\_IR"] <- 0.07; Gamma1["L\_IF","L\_ICE"] <- 0.05  
Gamma1["L\_GG","L\_PIB"] <- 0.06  
Gamma1["L\_GC","L\_GG"] <- 0.10  
Gamma1["L\_GK","L\_GG"] <- 0.06  
  
Sigma <- diag(c(0.20, 0.18, 0.15, 0.12, 0.12, 0.18, 0.15, 0.12, 0.12))  
dimnames(Sigma) <- list(vars, vars)  
Sigma["L\_PIB","L\_PP"] <- Sigma["L\_PP","L\_PIB"] <- 0.05  
Sigma["L\_IF","L\_PP"] <- Sigma["L\_PP","L\_IF"] <- 0.04  
Sigma["L\_IF","L\_IVA"] <- Sigma["L\_IVA","L\_IF"] <- 0.06  
Sigma["L\_GG","L\_PIB"] <- Sigma["L\_PIB","L\_GG"] <- 0.04  
Sigma <- (Sigma + t(Sigma))/2  
C <- t(chol(Sigma))  
  
# =========================================================  
# 3) Simulación VECM (ΔY\_t = Γ1 ΔY\_{t-1} + α β' Y\_{t-1} + ε\_t)  
# =========================================================  
Y <- matrix(0, nrow = k, ncol = T, dimnames = list(vars, NULL))  
dY <- matrix(0, nrow = k, ncol = T, dimnames = list(vars, NULL))  
Y[,1] <- c(4.6, 8.5, 9.2, 8.9, 7.8, 8.6, 7.9, 6.5, 7.6)  
  
for (t in 2:T) {  
 eps\_t <- C %\*% rnorm(k)  
 EC\_lag <- t(beta) %\*% Y[, t-1] # (2x9)\*(9x1) = (2x1)  
 dY[,t] <- Gamma1 %\*% dY[,t-1] + alpha %\*% EC\_lag + eps\_t  
 Y[,t] <- Y[,t-1] + dY[,t]  
}  
  
Y\_ts <- ts(t(Y), start = c(2007,1), frequency = fq); colnames(Y\_ts) <- vars

dY\_ts <- na.omit(diff(Y\_ts))  
sel\_d <- VARselect(dY\_ts, lag.max = 4, type = "const")  
p\_d <- ifelse(is.na(as.integer(sel\_d$selection["AIC(n)"])), 2L, as.integer(sel\_d$selection["AIC(n)"]))  
cat("Rezagos VAR(Δ) (AIC): p\_d =", p\_d, "\n")

Rezagos VAR(Δ) (AIC): p\_d = 1

var\_d <- VAR(dY\_ts, p = p\_d, type = "const") # <- 'varest'  
cat("\nResumen VAR(Δ):\n"); print(summary(var\_d))

Resumen VAR(Δ):

VAR Estimation Results:  
=========================   
Endogenous variables: L\_PP, L\_PIB, L\_GG, L\_GC, L\_GK, L\_IF, L\_IVA, L\_ICE, L\_IR   
Deterministic variables: const   
Sample size: 70   
Log Likelihood: -196.861   
Roots of the characteristic polynomial:  
0.3844 0.3699 0.3699 0.2873 0.2873 0.2268 0.1286 0.1286 0.1173  
Call:  
VAR(y = dY\_ts, p = p\_d, type = "const")  
  
  
Estimation results for equation L\_PP:   
=====================================   
L\_PP = L\_PP.l1 + L\_PIB.l1 + L\_GG.l1 + L\_GC.l1 + L\_GK.l1 + L\_IF.l1 + L\_IVA.l1 + L\_ICE.l1 + L\_IR.l1 + const   
  
 Estimate Std. Error t value Pr(>|t|)   
L\_PP.l1 0.006144 0.139235 0.044 0.9649   
L\_PIB.l1 -0.270701 0.160059 -1.691 0.0960 .  
L\_GG.l1 0.050977 0.124205 0.410 0.6830   
L\_GC.l1 -0.262953 0.220490 -1.193 0.2377   
L\_GK.l1 -0.160439 0.177812 -0.902 0.3705   
L\_IF.l1 0.190770 0.162370 1.175 0.2447   
L\_IVA.l1 -0.229206 0.182196 -1.258 0.2133   
L\_ICE.l1 -0.145232 0.156537 -0.928 0.3572   
L\_IR.l1 0.318083 0.179151 1.775 0.0809 .  
const -0.021665 0.060722 -0.357 0.7225   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
  
Residual standard error: 0.4629 on 60 degrees of freedom  
Multiple R-Squared: 0.1471, Adjusted R-squared: 0.01921   
F-statistic: 1.15 on 9 and 60 DF, p-value: 0.3431   
  
  
Estimation results for equation L\_PIB:   
======================================   
L\_PIB = L\_PP.l1 + L\_PIB.l1 + L\_GG.l1 + L\_GC.l1 + L\_GK.l1 + L\_IF.l1 + L\_IVA.l1 + L\_ICE.l1 + L\_IR.l1 + const   
  
 Estimate Std. Error t value Pr(>|t|)   
L\_PP.l1 0.013107 0.117795 0.111 0.9118   
L\_PIB.l1 -0.108425 0.135413 -0.801 0.4265   
L\_GG.l1 0.139025 0.105079 1.323 0.1908   
L\_GC.l1 -0.115211 0.186538 -0.618 0.5392   
L\_GK.l1 0.016799 0.150431 0.112 0.9115   
L\_IF.l1 0.081897 0.137368 0.596 0.5533   
L\_IVA.l1 0.174036 0.154141 1.129 0.2634   
L\_ICE.l1 -0.233048 0.132433 -1.760 0.0835 .  
L\_IR.l1 0.246361 0.151565 1.625 0.1093   
const -0.009392 0.051372 -0.183 0.8556   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
  
Residual standard error: 0.3916 on 60 degrees of freedom  
Multiple R-Squared: 0.1263, Adjusted R-squared: -0.004795   
F-statistic: 0.9634 on 9 and 60 DF, p-value: 0.4789   
  
  
Estimation results for equation L\_GG:   
=====================================   
L\_GG = L\_PP.l1 + L\_PIB.l1 + L\_GG.l1 + L\_GC.l1 + L\_GK.l1 + L\_IF.l1 + L\_IVA.l1 + L\_ICE.l1 + L\_IR.l1 + const   
  
 Estimate Std. Error t value Pr(>|t|)   
L\_PP.l1 0.129842 0.147409 0.881 0.3819   
L\_PIB.l1 0.037703 0.169457 0.222 0.8247   
L\_GG.l1 0.147612 0.131497 1.123 0.2661   
L\_GC.l1 0.106570 0.233435 0.457 0.6497   
L\_GK.l1 0.107221 0.188251 0.570 0.5711   
L\_IF.l1 -0.006656 0.171903 -0.039 0.9692   
L\_IVA.l1 0.329128 0.192893 1.706 0.0931 .  
L\_ICE.l1 0.083153 0.165728 0.502 0.6177   
L\_IR.l1 0.304865 0.189670 1.607 0.1132   
const -0.023708 0.064287 -0.369 0.7136   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
  
Residual standard error: 0.4901 on 60 degrees of freedom  
Multiple R-Squared: 0.1326, Adjusted R-squared: 0.002536   
F-statistic: 1.019 on 9 and 60 DF, p-value: 0.4352   
  
  
Estimation results for equation L\_GC:   
=====================================   
L\_GC = L\_PP.l1 + L\_PIB.l1 + L\_GG.l1 + L\_GC.l1 + L\_GK.l1 + L\_IF.l1 + L\_IVA.l1 + L\_ICE.l1 + L\_IR.l1 + const   
  
 Estimate Std. Error t value Pr(>|t|)   
L\_PP.l1 -0.027326 0.086987 -0.314 0.7545   
L\_PIB.l1 0.027759 0.099997 0.278 0.7823   
L\_GG.l1 0.000186 0.077597 0.002 0.9981   
L\_GC.l1 -0.116962 0.137752 -0.849 0.3992   
L\_GK.l1 0.030642 0.111088 0.276 0.7836   
L\_IF.l1 -0.157829 0.101441 -1.556 0.1250   
L\_IVA.l1 0.025977 0.113827 0.228 0.8203   
L\_ICE.l1 0.198581 0.097797 2.031 0.0467 \*  
L\_IR.l1 0.039181 0.111925 0.350 0.7275   
const -0.069440 0.037936 -1.830 0.0722 .  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
  
Residual standard error: 0.2892 on 60 degrees of freedom  
Multiple R-Squared: 0.1607, Adjusted R-squared: 0.03475   
F-statistic: 1.276 on 9 and 60 DF, p-value: 0.2687   
  
  
Estimation results for equation L\_GK:   
=====================================   
L\_GK = L\_PP.l1 + L\_PIB.l1 + L\_GG.l1 + L\_GC.l1 + L\_GK.l1 + L\_IF.l1 + L\_IVA.l1 + L\_ICE.l1 + L\_IR.l1 + const   
  
 Estimate Std. Error t value Pr(>|t|)  
L\_PP.l1 -0.01966 0.10528 -0.187 0.852  
L\_PIB.l1 0.06611 0.12103 0.546 0.587  
L\_GG.l1 -0.04100 0.09392 -0.437 0.664  
L\_GC.l1 0.10285 0.16673 0.617 0.540  
L\_GK.l1 -0.11631 0.13445 -0.865 0.390  
L\_IF.l1 0.19534 0.12278 1.591 0.117  
L\_IVA.l1 -0.13374 0.13777 -0.971 0.336  
L\_ICE.l1 -0.16472 0.11837 -1.392 0.169  
L\_IR.l1 0.12174 0.13547 0.899 0.372  
const -0.04998 0.04592 -1.088 0.281  
  
  
Residual standard error: 0.35 on 60 degrees of freedom  
Multiple R-Squared: 0.1415, Adjusted R-squared: 0.01277   
F-statistic: 1.099 on 9 and 60 DF, p-value: 0.3773   
  
  
Estimation results for equation L\_IF:   
=====================================   
L\_IF = L\_PP.l1 + L\_PIB.l1 + L\_GG.l1 + L\_GC.l1 + L\_GK.l1 + L\_IF.l1 + L\_IVA.l1 + L\_ICE.l1 + L\_IR.l1 + const   
  
 Estimate Std. Error t value Pr(>|t|)  
L\_PP.l1 -0.09814 0.12374 -0.793 0.431  
L\_PIB.l1 -0.08250 0.14224 -0.580 0.564  
L\_GG.l1 -0.11260 0.11038 -1.020 0.312  
L\_GC.l1 -0.14983 0.19595 -0.765 0.447  
L\_GK.l1 -0.25196 0.15802 -1.594 0.116  
L\_IF.l1 0.22234 0.14430 1.541 0.129  
L\_IVA.l1 0.24038 0.16192 1.485 0.143  
L\_ICE.l1 0.08121 0.13911 0.584 0.562  
L\_IR.l1 -0.04844 0.15921 -0.304 0.762  
const -0.04600 0.05396 -0.852 0.397  
  
  
Residual standard error: 0.4114 on 60 degrees of freedom  
Multiple R-Squared: 0.2204, Adjusted R-squared: 0.1035   
F-statistic: 1.885 on 9 and 60 DF, p-value: 0.07158   
  
  
Estimation results for equation L\_IVA:   
======================================   
L\_IVA = L\_PP.l1 + L\_PIB.l1 + L\_GG.l1 + L\_GC.l1 + L\_GK.l1 + L\_IF.l1 + L\_IVA.l1 + L\_ICE.l1 + L\_IR.l1 + const   
  
 Estimate Std. Error t value Pr(>|t|)  
L\_PP.l1 -0.12070 0.11075 -1.090 0.280  
L\_PIB.l1 0.19782 0.12732 1.554 0.126  
L\_GG.l1 -0.02788 0.09880 -0.282 0.779  
L\_GC.l1 0.15912 0.17539 0.907 0.368  
L\_GK.l1 -0.11119 0.14144 -0.786 0.435  
L\_IF.l1 0.03385 0.12916 0.262 0.794  
L\_IVA.l1 0.18860 0.14493 1.301 0.198  
L\_ICE.l1 0.03268 0.12452 0.262 0.794  
L\_IR.l1 -0.14104 0.14251 -0.990 0.326  
const -0.03429 0.04830 -0.710 0.481  
  
  
Residual standard error: 0.3682 on 60 degrees of freedom  
Multiple R-Squared: 0.1624, Adjusted R-squared: 0.03672   
F-statistic: 1.292 on 9 and 60 DF, p-value: 0.2601   
  
  
Estimation results for equation L\_ICE:   
======================================   
L\_ICE = L\_PP.l1 + L\_PIB.l1 + L\_GG.l1 + L\_GC.l1 + L\_GK.l1 + L\_IF.l1 + L\_IVA.l1 + L\_ICE.l1 + L\_IR.l1 + const   
  
 Estimate Std. Error t value Pr(>|t|)   
L\_PP.l1 0.11530 0.11148 1.034 0.3052   
L\_PIB.l1 -0.07117 0.12816 -0.555 0.5807   
L\_GG.l1 -0.08798 0.09945 -0.885 0.3798   
L\_GC.l1 -0.07259 0.17654 -0.411 0.6824   
L\_GK.l1 -0.20319 0.14237 -1.427 0.1587   
L\_IF.l1 -0.11829 0.13001 -0.910 0.3665   
L\_IVA.l1 0.31812 0.14588 2.181 0.0331 \*  
L\_ICE.l1 -0.13038 0.12534 -1.040 0.3024   
L\_IR.l1 0.06344 0.14344 0.442 0.6599   
const -0.06346 0.04862 -1.305 0.1968   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
  
Residual standard error: 0.3706 on 60 degrees of freedom  
Multiple R-Squared: 0.1263, Adjusted R-squared: -0.004702   
F-statistic: 0.9641 on 9 and 60 DF, p-value: 0.4784   
  
  
Estimation results for equation L\_IR:   
=====================================   
L\_IR = L\_PP.l1 + L\_PIB.l1 + L\_GG.l1 + L\_GC.l1 + L\_GK.l1 + L\_IF.l1 + L\_IVA.l1 + L\_ICE.l1 + L\_IR.l1 + const   
  
 Estimate Std. Error t value Pr(>|t|)   
L\_PP.l1 0.019853 0.097462 0.204 0.8393   
L\_PIB.l1 -0.083797 0.112039 -0.748 0.4574   
L\_GG.l1 -0.006524 0.086942 -0.075 0.9404   
L\_GC.l1 -0.052402 0.154340 -0.340 0.7354   
L\_GK.l1 0.012229 0.124466 0.098 0.9221   
L\_IF.l1 0.127941 0.113657 1.126 0.2648   
L\_IVA.l1 -0.260281 0.127535 -2.041 0.0457 \*  
L\_ICE.l1 -0.006577 0.109574 -0.060 0.9523   
L\_IR.l1 -0.022772 0.125403 -0.182 0.8565   
const -0.034965 0.042504 -0.823 0.4140   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
  
Residual standard error: 0.324 on 60 degrees of freedom  
Multiple R-Squared: 0.08923, Adjusted R-squared: -0.04738   
F-statistic: 0.6532 on 9 and 60 DF, p-value: 0.7471   
  
  
  
Covariance matrix of residuals:  
 L\_PP L\_PIB L\_GG L\_GC L\_GK L\_IF L\_IVA  
L\_PP 0.2142737 0.019537 0.024792 0.0385602 0.0003262 0.066916 -0.0272871  
L\_PIB 0.0195370 0.153364 0.055971 -0.0060646 -0.0080788 0.035731 0.0096208  
L\_GG 0.0247924 0.055971 0.240173 -0.0075004 -0.0231070 -0.003968 -0.0263040  
L\_GC 0.0385602 -0.006065 -0.007500 0.0836341 -0.0236269 0.023771 0.0008537  
L\_GK 0.0003262 -0.008079 -0.023107 -0.0236269 0.1225169 -0.022639 -0.0023202  
L\_IF 0.0669159 0.035731 -0.003968 0.0237714 -0.0226385 0.169227 0.0631867  
L\_IVA -0.0272871 0.009621 -0.026304 0.0008537 -0.0023202 0.063187 0.1355797  
L\_ICE -0.0335923 -0.016986 0.016064 -0.0051152 -0.0121047 -0.024500 0.0048957  
L\_IR -0.0173999 -0.011026 0.019756 0.0168764 -0.0151485 -0.003703 -0.0114097  
 L\_ICE L\_IR  
L\_PP -0.033592 -0.017400  
L\_PIB -0.016986 -0.011026  
L\_GG 0.016064 0.019756  
L\_GC -0.005115 0.016876  
L\_GK -0.012105 -0.015149  
L\_IF -0.024500 -0.003703  
L\_IVA 0.004896 -0.011410  
L\_ICE 0.137370 0.012749  
L\_IR 0.012749 0.104990  
  
Correlation matrix of residuals:  
 L\_PP L\_PIB L\_GG L\_GC L\_GK L\_IF L\_IVA  
L\_PP 1.000000 0.10777 0.10929 0.288047 0.002013 0.35141 -0.160094  
L\_PIB 0.107773 1.00000 0.29164 -0.053548 -0.058937 0.22180 0.066719  
L\_GG 0.109288 0.29164 1.00000 -0.052921 -0.134705 -0.01968 -0.145768  
L\_GC 0.288047 -0.05355 -0.05292 1.000000 -0.233409 0.19982 0.008017  
L\_GK 0.002013 -0.05894 -0.13470 -0.233409 1.000000 -0.15722 -0.018003  
L\_IF 0.351407 0.22180 -0.01968 0.199815 -0.157223 1.00000 0.417151  
L\_IVA -0.160094 0.06672 -0.14577 0.008017 -0.018003 0.41715 1.000000  
L\_ICE -0.195798 -0.11703 0.08844 -0.047723 -0.093306 -0.16069 0.035873  
L\_IR -0.116008 -0.08690 0.12441 0.180101 -0.133567 -0.02778 -0.095632  
 L\_ICE L\_IR  
L\_PP -0.19580 -0.11601  
L\_PIB -0.11703 -0.08690  
L\_GG 0.08844 0.12441  
L\_GC -0.04772 0.18010  
L\_GK -0.09331 -0.13357  
L\_IF -0.16069 -0.02778  
L\_IVA 0.03587 -0.09563  
L\_ICE 1.00000 0.10616  
L\_IR 0.10616 1.00000

# Diagnósticos rápidos  
cat("\nPortmanteau (autocorrelación):\n"); print(serial.test(var\_d, lags.pt = 12, type = "PT.asymptotic"))

Portmanteau (autocorrelación):

Portmanteau Test (asymptotic)  
  
data: Residuals of VAR object var\_d  
Chi-squared = 824.07, df = 891, p-value = 0.9464

$serial  
  
 Portmanteau Test (asymptotic)  
  
data: Residuals of VAR object var\_d  
Chi-squared = 824.07, df = 891, p-value = 0.9464

cat("\nARCH (heterocedasticidad):\n"); print(arch.test(var\_d, lags.multi = 5))

ARCH (heterocedasticidad):

ARCH (multivariate)  
  
data: Residuals of VAR object var\_d  
Chi-squared = 2925, df = 10125, p-value = 1

$arch.mul  
  
 ARCH (multivariate)  
  
data: Residuals of VAR object var\_d  
Chi-squared = 2925, df = 10125, p-value = 1

cat("\nNormalidad (Jarque-Bera multivariante):\n"); print(normality.test(var\_d))

Normalidad (Jarque-Bera multivariante):

$JB  
  
 JB-Test (multivariate)  
  
data: Residuals of VAR object var\_d  
Chi-squared = 15.505, df = 18, p-value = 0.6271  
  
  
$Skewness  
  
 Skewness only (multivariate)  
  
data: Residuals of VAR object var\_d  
Chi-squared = 6.9774, df = 9, p-value = 0.6395  
  
  
$Kurtosis  
  
 Kurtosis only (multivariate)  
  
data: Residuals of VAR object var\_d  
Chi-squared = 8.5275, df = 9, p-value = 0.482

$jb.mul  
$jb.mul$JB  
  
 JB-Test (multivariate)  
  
data: Residuals of VAR object var\_d  
Chi-squared = 15.505, df = 18, p-value = 0.6271  
  
  
$jb.mul$Skewness  
  
 Skewness only (multivariate)  
  
data: Residuals of VAR object var\_d  
Chi-squared = 6.9774, df = 9, p-value = 0.6395  
  
  
$jb.mul$Kurtosis  
  
 Kurtosis only (multivariate)  
  
data: Residuals of VAR object var\_d  
Chi-squared = 8.5275, df = 9, p-value = 0.482

# Estabilidad (raíces del VAR(Δ))  
mod\_roots <- roots(var\_d, modulus = TRUE)  
cat("\nMódulos de autovalores (VAR(Δ)):\n"); print(round(sort(mod\_roots, decreasing = TRUE), 4))

Módulos de autovalores (VAR(Δ)):

[1] 0.3844 0.3699 0.3699 0.2873 0.2873 0.2268 0.1286 0.1286 0.1173

vnames\_d <- colnames(var\_d$y)  
K <- length(vnames\_d)  
  
# A: 1 en diagonal; 0 = cero fijo; NA = libre  
Amat\_d <- diag(1, K); dimnames(Amat\_d) <- list(vnames\_d, vnames\_d)  
  
# ΔL\_PP (shock de petróleo) NO recibe contemporáneos del resto:  
Amat\_d["L\_PP", setdiff(vnames\_d, "L\_PP")] <- 0  
  
# Canales contemporáneos plausibles (libres)  
Amat\_d["L\_PIB","L\_PP"] <- NA # petróleo → PIB  
Amat\_d["L\_GG","L\_PIB"] <- NA # PIB → gasto total  
Amat\_d["L\_GC","L\_GG"] <- NA # gasto total → gasto corriente  
Amat\_d["L\_GK","L\_GG"] <- NA # gasto total → gasto capital  
Amat\_d["L\_IF","L\_PP"] <- NA # petróleo → ingresos fiscales  
Amat\_d["L\_IF","L\_IVA"] <- NA # IVA → ingresos fiscales  
Amat\_d["L\_IF","L\_IR"] <- NA # IR → ingresos fiscales  
Amat\_d["L\_IF","L\_ICE"] <- NA # ICE → ingresos fiscales  
# (todo lo no especificado queda en 0; ajusta NA/0 según tu narrativa)  
  
# B: diagonal (NA para las sd de los shocks), 0 fuera  
Bmat\_d <- diag(NA, K); dimnames(Bmat\_d) <- list(vnames\_d, vnames\_d)  
Bmat\_d[row(Bmat\_d) != col(Bmat\_d)] <- 0  
  
# Estimar SVAR(Δ)  
svar\_d <- SVAR(var\_d, Amat = Amat\_d, Bmat = Bmat\_d, estmethod = "direct")  
cat("\nSVAR(Δ) estimado con éxito.\n")

SVAR(Δ) estimado con éxito.

while (!is.null(dev.list())) dev.off()  
  
# IRF: shock petróleo → PIB  
irf\_pp\_pib\_d <- irf(svar\_d, impulse = "L\_PP", response = "L\_PIB",  
 n.ahead = 12, boot = TRUE, ci = 0.95)  
png("IRF\_SVAR\_DIFF\_LPP\_to\_LPIB.png", width = 1400, height = 900, res = 150)  
par(mar = c(4,4,2,1)); plot(irf\_pp\_pib\_d); dev.off()

pdf   
 2

cat("✅ Guardado: IRF\_SVAR\_DIFF\_LPP\_to\_LPIB.png\n")

✅ Guardado: IRF\_SVAR\_DIFF\_LPP\_to\_LPIB.png

# IRF: shock petróleo → ingresos fiscales  
irf\_pp\_if\_d <- irf(svar\_d, impulse = "L\_PP", response = "L\_IF",  
 n.ahead = 12, boot = TRUE, ci = 0.95)  
png("IRF\_SVAR\_DIFF\_LPP\_to\_LIF.png", width = 1400, height = 900, res = 150)  
par(mar = c(4,4,2,1)); plot(irf\_pp\_if\_d); dev.off()

pdf   
 2

cat("✅ Guardado: IRF\_SVAR\_DIFF\_LPP\_to\_LIF.png\n")

✅ Guardado: IRF\_SVAR\_DIFF\_LPP\_to\_LIF.png

# FEVD (12 pasos)  
fevd\_d <- fevd(svar\_d, n.ahead = 12)  
png("FEVD\_SVAR\_DIFF\_12.png", width = 1400, height = 900, res = 150)  
par(mar = c(4,4,2,1)); plot(fevd\_d); dev.off()

pdf   
 2

cat("✅ Guardado: FEVD\_SVAR\_DIFF\_12.png\n")

✅ Guardado: FEVD\_SVAR\_DIFF\_12.png

# IRFs de petróleo hacia todas las respuestas (lote)  
responses <- c("L\_PIB","L\_GG","L\_GC","L\_GK","L\_IF","L\_IVA","L\_ICE","L\_IR")  
for (resp in responses) {  
 f <- irf(svar\_d, impulse = "L\_PP", response = resp, n.ahead = 12, boot = TRUE, ci = 0.95)  
 fn <- paste0("IRF\_SVAR\_DIFF\_LPP\_to\_", resp, ".png")  
 png(fn, width = 1400, height = 900, res = 150)  
 par(mar = c(4,4,2,1)); plot(f); dev.off()  
 cat("✅ Guardado: ", fn, "\n", sep = "")  
}

✅ Guardado: IRF\_SVAR\_DIFF\_LPP\_to\_L\_PIB.png

✅ Guardado: IRF\_SVAR\_DIFF\_LPP\_to\_L\_GG.png

✅ Guardado: IRF\_SVAR\_DIFF\_LPP\_to\_L\_GC.png

✅ Guardado: IRF\_SVAR\_DIFF\_LPP\_to\_L\_GK.png

✅ Guardado: IRF\_SVAR\_DIFF\_LPP\_to\_L\_IF.png

✅ Guardado: IRF\_SVAR\_DIFF\_LPP\_to\_L\_IVA.png

✅ Guardado: IRF\_SVAR\_DIFF\_LPP\_to\_L\_ICE.png

✅ Guardado: IRF\_SVAR\_DIFF\_LPP\_to\_L\_IR.png

cat("\n=== FIN: SVAR en diferencias (estacionario) con gráficos exportados ===\n")

=== FIN: SVAR en diferencias (estacionario) con gráficos exportados ===

Forecast

# ===============================  
# FORECAST con VAR en diferencias  
# ===============================  
  
# Horizonte de pronóstico (trimestres)  
h <- 8 # 2 años  
  
# 1) Pronóstico de diferencias (Δlog) con el VAR estacionario  
fc\_d <- predict(var\_d, n.ahead = h, ci = 0.95)  
  
# Extraer medias, límites inferior/superior por variable  
vnames\_d <- colnames(var\_d$y)  
dhat\_mean <- sapply(vnames\_d, function(v) fc\_d$fcst[[v]][, "fcst"])  
dhat\_low <- sapply(vnames\_d, function(v) fc\_d$fcst[[v]][, "lower"])  
dhat\_high <- sapply(vnames\_d, function(v) fc\_d$fcst[[v]][, "upper"])  
  
# 2) Reconstruir niveles (logs) desde el último dato observado  
last\_level <- as.numeric(tail(Y\_ts, 1)) # vector de 9 niveles (logs) en 2024Q4  
names(last\_level) <- colnames(Y\_ts)  
  
# Función auxiliar: acumula difs para pasar a niveles  
rebuild\_levels <- function(last\_level, diffs\_mat) {  
 # diffs\_mat: h x k (Δlog pronosticadas)  
 lev <- matrix(NA\_real\_, nrow = nrow(diffs\_mat), ncol = ncol(diffs\_mat))  
 colnames(lev) <- colnames(diffs\_mat)  
 prev <- last\_level  
 for (t in 1:nrow(diffs\_mat)) {  
 lev[t, ] <- prev + diffs\_mat[t, ]  
 prev <- lev[t, ]  
 }  
 lev  
}  
  
# Niveles pronosticados (punto), y bandas aprox. por acumulación (didáctico)  
lev\_hat\_mean <- rebuild\_levels(last\_level, dhat\_mean)  
lev\_hat\_low <- rebuild\_levels(last\_level, dhat\_low)  
lev\_hat\_high <- rebuild\_levels(last\_level, dhat\_high)  
  
# 3) Construir fechas futuras (trimestres posteriores a 2024Q4)  
# Reusamos la secuencia "fechas" que ya tienes (2007Q1..2024Q4) y la extendemos  
fechas\_all <- as.yearqtr(seq(from = as.Date("2007-01-01"), by = "quarter", length.out = T + h))  
fechas\_fc <- tail(fechas\_all, h) # fechas pronóstico  
  
# 4) Armar data.frames para exportar a CSV  
# a) Pronóstico en diferencias  
fc\_diff\_df <- data.frame(  
 fecha\_q = fechas\_fc,  
 as.data.frame(dhat\_mean),  
 check.names = FALSE  
)  
  
# b) Pronóstico en niveles (logs)  
fc\_level\_df <- data.frame(  
 fecha\_q = fechas\_fc,  
 as.data.frame(lev\_hat\_mean),  
 check.names = FALSE  
)  
  
# 5) Exportar a CSV (carpeta de trabajo actual)  
write.csv(fc\_diff\_df, "forecast\_VARdiff\_deltas.csv", row.names = FALSE)  
write.csv(fc\_level\_df, "forecast\_VARdiff\_levels.csv", row.names = FALSE)  
cat("✅ CSV guardados: forecast\_VARdiff\_deltas.csv, forecast\_VARdiff\_levels.csv\n")

✅ CSV guardados: forecast\_VARdiff\_deltas.csv, forecast\_VARdiff\_levels.csv

# 6) Gráficos exportados (histórico + pronóstico) para 4 variables clave  
plot\_series\_fc <- function(varname, file\_png) {  
 # Serie histórica (niveles log)  
 hist\_ts <- Y\_ts[, varname]  
 # Serie pronosticada (niveles log)  
 fc\_ts <- ts(lev\_hat\_mean[, varname],  
 start = c(2007 + (T)/4, (T %% 4) + 1), # arranque inmediatamente después  
 frequency = 4)  
  
 # Cierra y abre dispositivo PNG grande para evitar márgenes  
 while (!is.null(dev.list())) dev.off()  
 png(file\_png, width = 1400, height = 900, res = 150)  
 par(mar = c(4,4,2,1))  
 # Gráfico base del histórico  
 plot(hist\_ts, type = "l", lwd = 2, xlab = "Trimestre", ylab = "Log(nivel)",  
 main = paste("Histórico y Forecast (niveles log) -", varname))  
 # Añadir pronóstico  
 lines(window(fc\_ts, start = tsp(fc\_ts)[1]), lwd = 2, lty = 2)  
 # Límites (aprox.) de confianza en niveles  
 lines(ts(lev\_hat\_low[, varname],  
 start = start(fc\_ts), frequency = 4), lty = 3)  
 lines(ts(lev\_hat\_high[, varname],  
 start = start(fc\_ts), frequency = 4), lty = 3)  
 legend("topleft", bty = "n",  
 legend = c("Histórico", "Pronóstico (media)", "Banda baja", "Banda alta"),  
 lwd = c(2,2,1,1), lty = c(1,2,3,3))  
 dev.off()  
 cat("✅ Gráfico guardado:", file\_png, "\n")  
}  
  
# Elige algunas variables representativas para la clase:  
for (vn in c("L\_PP","L\_PIB","L\_GG","L\_IF")) {  
 plot\_series\_fc(vn, paste0("FC\_LEVELS\_", vn, ".png"))  
}

✅ Gráfico guardado: FC\_LEVELS\_L\_PP.png

✅ Gráfico guardado: FC\_LEVELS\_L\_PIB.png

✅ Gráfico guardado: FC\_LEVELS\_L\_GG.png

✅ Gráfico guardado: FC\_LEVELS\_L\_IF.png

# 7) (Opcional) Si quieres pronóstico en diferencias (Δlog) también en PNG:  
plot\_series\_fc\_diff <- function(varname, file\_png) {  
 while (!is.null(dev.list())) dev.off()  
 png(file\_png, width = 1400, height = 900, res = 150)  
 par(mar = c(4,4,2,1))  
 plot(dY\_ts[, varname], type = "l", lwd = 2, xlab = "Trimestre", ylab = "Δ log",  
 main = paste("Histórico Δlog y Forecast -", varname))  
 lines(ts(dhat\_mean[, varname],  
 start = c(2007 + (T)/4, (T %% 4) + 1), frequency = 4), lwd = 2, lty = 2)  
 lines(ts(dhat\_low[, varname],  
 start = c(2007 + (T)/4, (T %% 4) + 1), frequency = 4), lty = 3)  
 lines(ts(dhat\_high[, varname],  
 start = c(2007 + (T)/4, (T %% 4) + 1), frequency = 4), lty = 3)  
 legend("topleft", bty = "n",  
 legend = c("Δlog histórico", "Δlog pronóstico (media)", "Banda baja", "Banda alta"),  
 lwd = c(2,2,1,1), lty = c(1,2,3,3))  
 dev.off()  
 cat("✅ Gráfico guardado:", file\_png, "\n")  
}  
  
for (vn in c("L\_PP","L\_PIB","L\_GG","L\_IF")) {  
 plot\_series\_fc\_diff(vn, paste0("FC\_DIFF\_", vn, ".png"))  
}

✅ Gráfico guardado: FC\_DIFF\_L\_PP.png

✅ Gráfico guardado: FC\_DIFF\_L\_PIB.png

✅ Gráfico guardado: FC\_DIFF\_L\_GG.png

✅ Gráfico guardado: FC\_DIFF\_L\_IF.png

cat("\n=== FORECAST listo: CSV + PNG exportados ===\n")

=== FORECAST listo: CSV + PNG exportados ===