Análisis de datos económicos
Trabajo Fin de Grado

Ramón Trinidad Acedo

Grado en Matemáticas

Sevilla, Junio de 2025

Índice general

Resumen

Este Trabajo de Fin de Grado aborda el análisis y la predicción de indicadores económicos, centrándose de manera particular en la evolución de la inflación dentro del contexto de los países de la Unión Europea. El estudio se fundamenta en la aplicación y comparación de un amplio espectro de metodologías estadísticas y de aprendizaje automático.

En primer lugar, se exploran los modelos ARIMA, detallando sus componentes autorregresiva, de integración y de medias móviles. Posteriormente, se introducen los modelos lineales aplicados a series temporales, incluyendo una discusión sobre técnicas de regularización cruciales para manejar la multicolinealidad y la alta dimensionalidad, tales como la Regresión Ridge, Lasso y Elastic Net. Se profundiza también en los modelos lineales dinámicos, que permiten incorporar la autocorrelación en los errores.

Finalmente, el trabajo investiga la aplicabilidad de métodos de aprendizaje automático más avanzados, como los Árboles de Regresión, Bosques Aleatorios (Random Forest), Gradient Boosting y Support Vector Regression (SVR). El objetivo principal es realizar una evaluación comparativa exhaustiva de la capacidad predictiva de todos estos modelos al ser aplicados a series temporales de datos económicos reales, con el fin de identificar las herramientas más efectivas para la predicción de la inflación y otros indicadores relevantes.

Abstract

This Bachelor's Thesis addresses the analysis and prediction of economic indicators, with a particular focus on the evolution of inflation within the context of European Union countries. The study is based on the application and comparison of a wide spectrum of statistical and machine learning methodologies.

Firstly, ARIMA models are explored, detailing their autoregressive, integrated, and moving average components. Subsequently, linear models applied to time series are introduced, including a discussion of crucial regularization techniques for handling multicollinearity and high dimensionality, such as Ridge Regression, Lasso, and Elastic Net. Dynamic linear models, which allow for the incorporation of autocorrelation in errors, are also examined in depth.

Finally, the work investigates the applicability of more advanced machine learning methods, including Regression Trees, Random Forests, Gradient Boosting, and Support Vector Regression (SVR). The main objective is to conduct a comprehensive comparative evaluation of the predictive capability of all these models when applied to real economic time series data, in order to identify the most effective tools for forecasting inflation and other relevant indicators.

Índice de figuras

Introducción

El análisis y predicción fiable de indicadores económicos constituye uno de los principales retos a los que se someten los gobiernos, para mantener una economía responsable de los bienes del estado, o los economistas y agentes de bolsa a la hora de tomar decisiones de inversión.

El objetivo de este trabajo es describir y aplicar los métodos estadísticos de regresión de variables con el fin de realizar predicciones sobre indicadores económicos de interés. Para ello, será necesario trabajar y manipular las variables económicas mediante el uso de series temporales.

Vamos a presentar los diferentes conceptos económicos que intervendrán en el trabajo y los indicadores que nos permiten medirlos. Todas estas definiciones han sido extraídas y sintetizadas a partir de autores como Tamames, Ramón & Gallego, Santiago (2006), el diccionario de la Real Academia Española, y el metadata de la base de datos europea, Eurostat.

Economía

La palabra **economía** tiene su origen etimológico en el griego "oikos", casa, y "nomos" administración, pues se entendía como la administración recta y prudente de los bienes. Más formalmente, la economía es la ciencia social que estudia cómo los individuos, las empresas, los gobiernos y las sociedades asignan recursos escasos para satisfacer sus necesidades ilimitadas. Se divide en dos grandes ramas: la **microeconomía**, que analiza el comportamiento de agentes individuales (como consumidores y empresas), y la **macroeconomía**, que estudia los fenómenos agregados, como el crecimiento económico, la inflación o el desempleo.

Si bien no existe un único indicador para el estudio de la economía en su conjunto, se pueden utilizar indicadores clave como el Producto Interior Bruto (PIB), la tasa de desempleo y la inflación para evaluar su estado general.

Producto Interior Bruto (PIB)

El PIB es el valor monetario de todos los bienes y servicios finales producidos en un país durante un período de tiempo determinado (generalmente un año o un trimestre). Es el indicador más utilizado para medir el tamaño y el crecimiento de una economía.

Índice de Precios al Consumo (IPC)

El IPC es un indicador estadístico que mide la evolución del nivel de precios de una cesta de bienes y servicios adquiridos por los hogares. Se utiliza para evaluar cambios en el costo de vida y para ajustar salarios, pensiones y otros pagos.

Inflación

La **inflación** es el aumento generalizado y sostenido de los precios de bienes y servicios en una economía durante un período de tiempo. Se habla de **inflación de costes** cuando este aumento se debe al alza de los factores de producción (salarios, tipos de interés, coste de la vivienda,...); y de **inflación de demanda** cuando es imputable al aumento de las intenciones de la demanda, que junto a la rigidez de la oferta, hacen que suban los precios.

La inflación suele cuantificarse como la tasa de cambio, anual o mensual, del *Índice de Precios al Consumo*. En nuestro caso, como vamos a analizar la inflación en los países de la Unión Europea, existe un índice denominado *Índice de Precios al Consumo Armonizado* que se trata de una medida común de la inflación que permite realizar comparaciones internacionales. Se obtiene como resultado de homogeneizar los aspectos metodológicos más importantes de cada uno de los Índices de Precios de Consumo de cada uno de los estados miembros de la Unión Europea (UE) para hacerlos comparables.

Desempleo

El desempleo se refiere a la situación en la que una persona que forma parte de la población activa (en edad de trabajar y dispuesta a hacerlo) no tiene un empleo remunerado. Es un indicador clave de la salud económica de un país, ya que refleja la capacidad de la economía para generar empleo.

Tipos de interés

Los tipos de interés son el costo del dinero, es decir, el precio que se paga por pedir prestado o el rendimiento que se obtiene por ahorrar. Son fijadas por los bancos centrales y tienen un impacto directo en la inversión, el consumo y la inflación.

Además, los tipos de interés son una herramienta clave que utilizan los bancos centrales para influir en la economía y controlar la inflación. Si la inflación es alta, los bancos centrales pueden subir los tipos de interés para frenar el gasto excesivo y controlar la inflación. Si la inflación es baja, se reducen para estimular la inversión y el consumo.

Los siguientes datos recogidos de la base de la Unión Europea (Eurostat) y la página web *Datosmacro* reflejan el control de la inflación que ejerce el Banco Central Europeo mediante las tasas de interés:

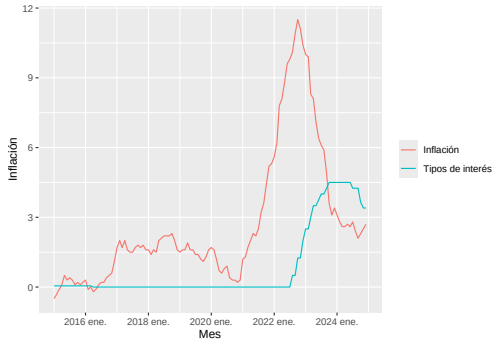

Figura 1: Valores de la inflación y tasas de interés en los países de la Unión Europea desde el año 2015.

Capítulo 1

Metodología. Análisis de series temporales

El análisis de series temporales constituye una herramienta fundamental en la modelización estadística de datos secuenciales. En este capítulo se abordan los modelos ARIMA a través de sus tres componentes esenciales, para luego expandirlos hacia modelos lineales con series temporales y técnicas de regularización como Ridge, Lasso y Elastic Net.

Además, se describe el modelo lineal dinámico, que combina los anteriores y que funciona especialmente bien en el ajuste de series de variables económicas.

Por último, se da una idea básica de modelos más complejos de aprendizaje automático, proporcionando así un marco metodológico completo para analizar, modelar y predecir comportamientos temporales en diversos contextos aplicados.

1.1. El modelo ARIMA

ARIMA(AutoRegressive Integrated Moving Average) es un algoritmo estadístico que consiste en construir una serie de modelos que expliquen la relación existente entre valores actuales y valores pasados de una determinada serie temporal. De esta manera, podemos usar los modelos ARIMA con el fin de predecir valores futuros en la serie.

El modelo ARIMA consta de tres parámetros fundamentales p, q, d. Estos caracterizan el modelo, pues nos brindan la información necesaria para construir cada una de sus componentes.

A continuación, entraremos en detalle sobre cada una de dichas partes. La información recogida sobre este modelo, así como el código de R para la generación de modelos y gráficas proviene de Dayal, V. (2020).

1.1.1. Componente Autorregresiva (Componente AR)

El parámetro p tiene que ver con la componente AR (autorregresiva) del modelo. Esta se basa en la idea de que, fijado un cierto tiempo t, el valor actual X_t de la serie está relacionado con valores pasados $X_{t-1}, X_{t-2}, \ldots$, luego p indica cuántos valores pasados

(en inglés conocido como lagged values) se utilizan en el modelo. Más formalmente, el modelo AR consiste en,

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + ... + \phi_p X_{t-p} + \epsilon_t$$

Aquí los ϕ_k son los coeficientes autorregresivos y ϵ_t es el error aleatorio, conocido como *ruido blanco*. Un ejemplo básico de modelo autorregresivo sería:

$$Hoy = 0.5Ayer + ruido\ blanco$$

Una serie temporal como la anterior se ve de la siguiente manera,

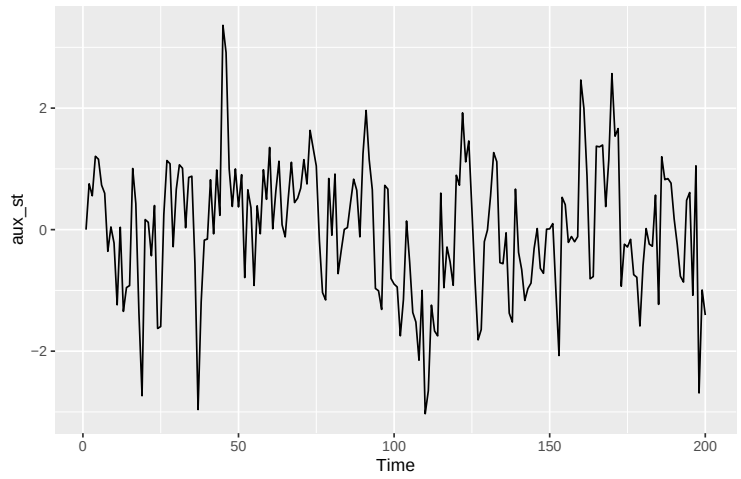

Figura 1.1: Ejemplo de serie temporal AR (1) simulada.

De hecho, es interesante calcular el modelo ARIMA(1,0,0) si la serie temporal con la queremos trabajar fuera la anterior.

	Model	Parameter	stimation	std.error
ar1	ARIMA(1,0,0)	ar1	0.505199	0.06098
intercept	ARIMA(1,0,0)	intercept	-0.007517	0.13310

Nótese que el modelo generado es,

$$X_t = -0.0075172 + 0.5051989 X_{t-1} + \epsilon_t$$

es decir, el coeficiente ϕ_1 se acerca a 0.5, pues hemos generado forzosamente los datos.

1.1.2. Componente Integración

El segundo de los parámetro modifica nuestro modelo aportándole una componente de integración. Para entender la integración, en primer lugar, debemos definir la estacionariedad en una serie temporal.

Una serie estacionaria es aquella en la que la media, la varianza, o la covarianza no cambian con el tiempo (Chan, K.S. & Cryer, J.D. (2008)). Formalmente, considerada una serie temporal $\{X_t\}_{t=1}^{T}$ se dirá que es estacionaria si:

- $E[X_t] = \mu$
- $E[(X_t - \mu)(X_{t-k} - \mu)] = \gamma_k \quad \forall t = 1, ..., T$ y $k = 1, ..., t$

El proceso de integración consiste en calcular las diferencias (diferenciación) entre valores consecutivos de la serie hasta que se logre estacionariedad, haciendo este proceso tantas veces como indique el parámetro d.

$$X'_t = X_t - X_{t-1}$$

Nota: Si $d = 0$, entonces la serie ya es estacionaria.

Un ejemplo de serie no estacionaria es la conocida como Random Walk. Se correponde con el modelo:

$$Hoy = Ayer + error$$

Este tipo de series no sigue ningún patrón determinado, pues su varianza ya no es constante, además de que sigue una tendencia aleatoria, de ahí el nombre que recibe

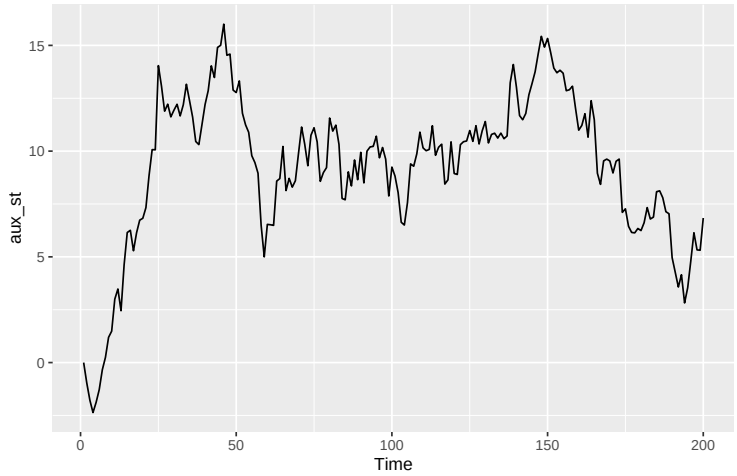

Figura 1.2: Ejemplo de serie temporal Random Walk.

Se observa la tendencia y la heterocedasticidad en el tiempo, por tanto, la serie no es estacionaria. Tal y como dijimos anteriormente, el proceso de integración consistirá en hacer una transformación lineal de las variables de manera que consigamos una serie temporal estacionaria. Entonces, aplicando a la anterior serie el proceso de integración, obtenemos la siguiente serie estacionaria:

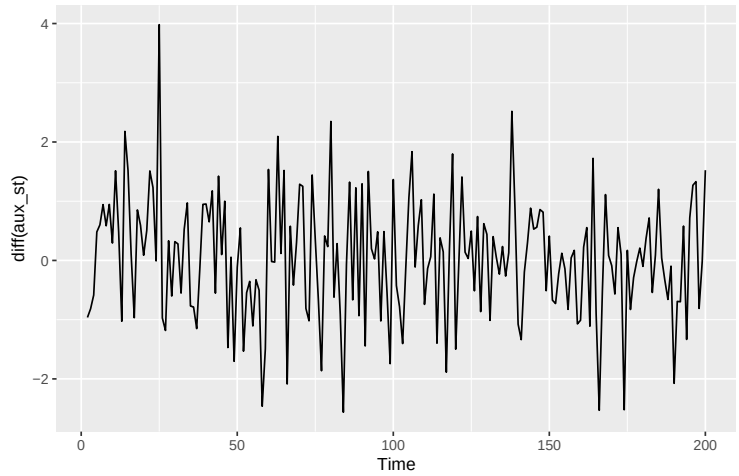

Figura 1.3: Serie Random Walk diferenciada (primer orden).

Como su propio nombre indica los Random Walk (caminos aleatorios), no dejan apreciar ningún patrón común en su forma. A continuación, se generan tres realizaciones de Random Walk diferentes en el mismo marco temporal:

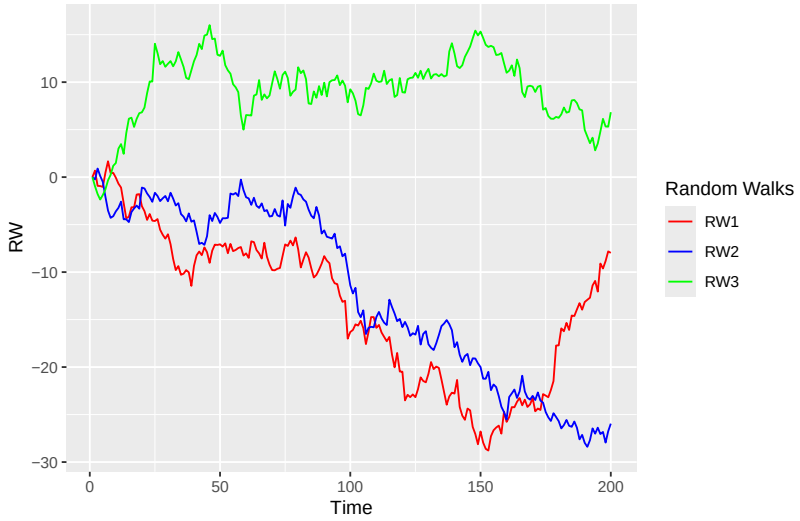

Figura 1.4: Comparación de tres realizaciones de Random Walk.

1.1.3. Componente de Medias Móviles

Por último, el parámetro q representa las medias móviles (MA), que trata de modelar el valor actual mediante las variables errores aleatorios de tiempos pasados. Es decir,

$$X_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q}$$

Como ejemplo básico, podemos considerar el siguiente modelo.

$$Hoy = error + 0.3 error_{ayer}$$

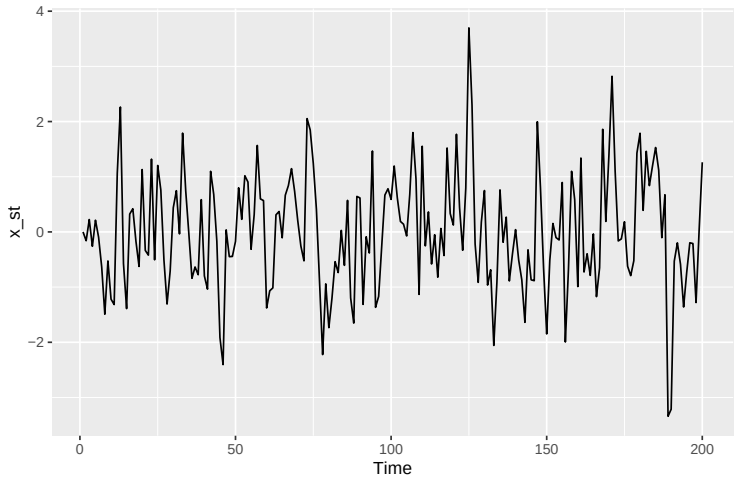

Figura 1.5: Ejemplo de serie temporal MA(1) simulada.

En definitiva, un ejemplo de modelo ARIMA con argumentos $p = 1, d = 1, q = 1$ sería,

$$X_t = \phi_1 X'_{t-1} + \epsilon_t + \theta_1 \epsilon_{t-1}$$

siendo $X'_{t-1} = X_t - X_{t-1}$.

1.1.3.1. Ejemplo: Modelo ARIMA(1,0,1)

Vamos a generar automáticamente una serie temporal que sigue un modelo ARIMA(1,0,1). Además, se presentarán y comprobarán las hipótesis del modelo.

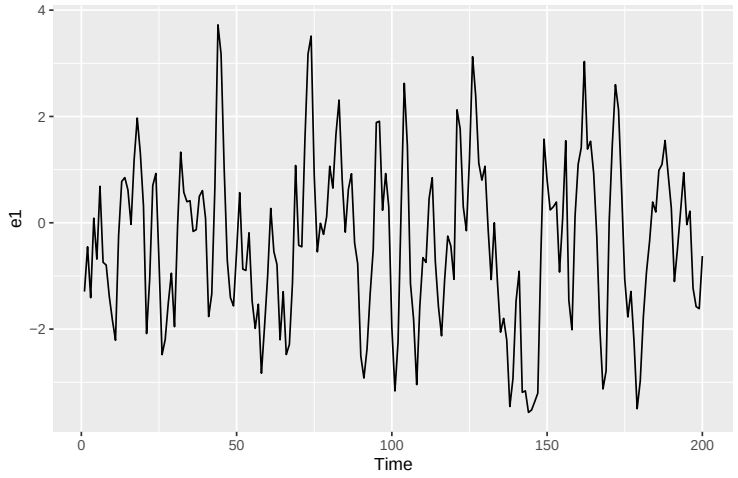

Figura 1.6: Simulación de una serie temporal ARIMA(1,0,1).

La gráfica de la función de autocorrelación(ACF) mide los valores de las correlaciones de la serie temporal con respecto los tiempos pasados.

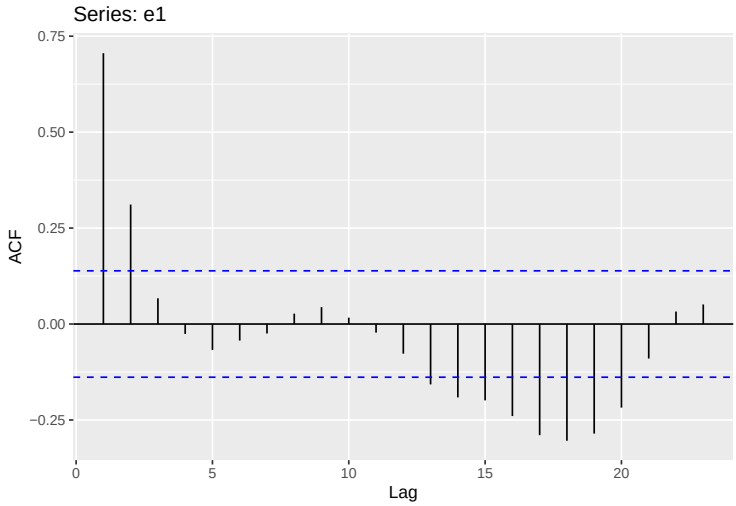

Figura 1.7: Función de Autocorrelación (ACF) de la serie ARIMA(1,0,1) simulada.

Concretamente, el modelo ARIMA generado automáticamente para esta serie es:

	Model	Parameter	stimation	std.error
ar1	ARIMA(1,0,1)	ar1	0.5295	0.07108
ma1	ARIMA(1,0,1)	ma1	0.4291	0.06686
intercept	ARIMA(1,0,1)	intercept	-0.3622	0.21961

Según se menciona en el capítulo 14 de Dayal, V. (2020), la manera de verificar las hipótesis del modelo se realiza automáticamente en R mediante la función `checkresiduals()`, que aplicada a un modelo comprueba que los residuos presenten normalidad, media igual a 0 y homocedasticidad:

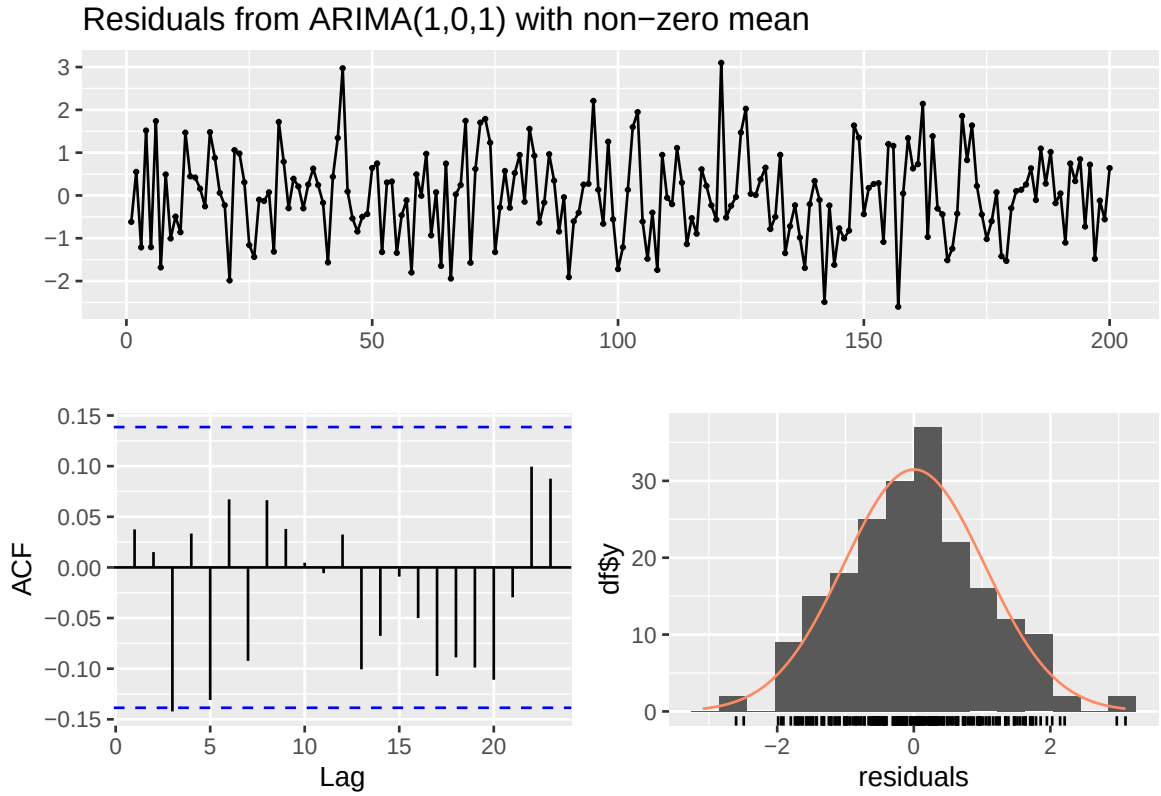

Figura 1.8: Diagnóstico de residuos del modelo ARIMA(1,0,1) ajustado a la serie simulada.

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1) with non-zero mean
## Q* = 12.219, df = 8, p-value = 0.1417
##
## Model df: 2.   Total lags used: 10
```

En las gráficas se observa que los residuos del modelo parecen seguir una distribución normal con media cero aunque todavía hay que hacer el contraste. Además, los residuos no siguen ningún patrón específico sino que muestran cierta aleatoriedad.

El test de Ljung-Box realiza un contraste sobre la autocorrelación de los residuos, al no poder rechazar la hipótesis nula, estaríamos asumiendo que no existe correlación significativa.

Para el contraste de normalidad se realiza el test de Shapiro-Wilk

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(mode1)
## W = 0.99482, p-value = 0.7228
```

Se asume la normalidad de los residuos.

Backshift notation

En ciertas ocasiones, usaremos una *notación de retroceso*, pues es más cómoda a la hora de trabajar con varios valores pasados. Esta notación ha sido extraída de Hyndman, R.J. & Athanasopoulos, G. (2021).

Se considera un operador de retroceso (*backshift operator*) B, tal que,

$$B^k X_t = X_{t-k}$$

Entonces se escribe el proceso de diferenciación como:

$$X'_t = X_t - X_{t-1} = (1 - B)X_t$$

Aún más, la diferenciación de orden d vendrá dada por

$$y'^{(d)} = (1 - B)^d y_t$$

De esta manera, podemos escribir un modelo $ARIMA(p, d, q)$ como:

$$(1 - \phi_1 B - ... - \phi_p B^p)z_t = (1 + \theta_1 B + ... + \theta_q B^q)\epsilon_t$$

siendo $z_t = (1 - B)^d X_t$.

1.2. Modelos lineales con series temporales

Los modelos lineales constituyen una pieza fundamental en la modelización estadística explicable. En ellos, buscamos construir una combinación lineal de una o varias variables explicativas con el objetivo de predecir valores de una serie temporal respuesta y.

Esta sección se ha elaborado en base a lo aprendido en la asignatura Modelos Lineales y Diseños de Experimentos y a información detallada en Hyndman, R.J. & Athanasopoulos, G. (2021), además de Hastie, T., Tibshirani, R. & Friedman, J. (2017).

En el documento que nos ocupa se pretende predecir la inflación, luego usaremos series temporales de distintos indicadores económicos como el desempleo, el PIB, el IPC, etc. para construir los modelos

Dadas r series temporales tomadas en t observaciones de tiempo $t = 1, ..., T$. Buscamos modelar la variable respuesta como:

$$y_t = \beta_0 + \beta_1 x_{1,t} + ... + \beta_r x_{r,t} + \epsilon_t \quad \forall t = 1, ..., T$$

Donde y es la variable respuesta , $x_1, ..., x_r$ las variables explicativas, los coeficientes $\beta_1, ..., \beta_r$ indican el efecto de cada variable explicativa en el modelo y ϵ es un cierto factor aleatorio, que ya denominamos en los modelos ARIMA como ruido blanco.

Hipotesis del modelo

Los factores aleatorios de error $\epsilon_1, ..., \epsilon_T$ deben verificar las siguientes hipótesis:

- Tienen media cero. Es decir, $E[\epsilon_j] = 0 \ \forall j$.
- Están incorrelados, $Cov[\epsilon_i, \epsilon_j] = 0 \ \forall i \neq j$
- Son independientes de las variables explicativas.

Además, Hyndman, R.J. & Athanasopoulos, G. (2021) añaden también las hipótesis de normalidad y homocedasticidad sobre los errores a la hora de construir modelos eficientes y con intervalos de predicción fiables.

Formulación matricial

A partir de las series temporales dadas, consideramos la siguiente matriz de diseño,

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & ... & x_{1r} \\ . & & & \\ . & & & \\ . & & & \\ 1 & x_{T1} & ... & x_{Tr} \end{pmatrix}$$

De manera, que el modelo con el que trabajaremos será,

$$\begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_T \end{pmatrix} = \mathbf{X} \begin{pmatrix} \beta_0 \\ \beta_1 \\ . \\ . \\ . \\ \beta_r \end{pmatrix} + \underline{\epsilon}$$

Donde $\underline{\epsilon}$ es un vector aleatorio tal que $E[\underline{\epsilon}] = \underline{0}$ y $Cov[\underline{\epsilon}] = \sigma^2 I_T$

Estimación de parámetros

Buscamos minimizar la suma de los errores cuadrados, esto es

$$\min_{\underline{\beta}} \quad SCE(\underline{\beta}) = \min_{\underline{\beta}} \quad (\underline{y} - \mathbf{X}\underline{\beta})^t (\underline{y} - \mathbf{X}\underline{\beta})$$

Desarrollando y aplicando las condiciones necesarias de optimalidad, obtenemos los estimadores:

$$\underline{\widehat{\beta}} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\underline{y}$$

De esta manera obtenemos el ajuste de la variable respuesta:

$$\underline{\widehat{y}} = \mathbf{X}\underline{\widehat{\beta}}$$

Es necesario para la estimación que la matriz $\mathbf{X}^t\mathbf{X}$ sea no singular. De lo contrario, si es singular o está próxima a serlo, la varianza de los coeficientes es grande, lo cual se traduce en un escaso poder predictivo en nuevas observaciones.

Como solución a este problema usaremos métodos de regularización, que aplicaremos con el fin de obtener nuevos estimadores de los coeficientes del modelo de regresión.

1.2.1. Técnicas de regularización

Como hemos indicado, en ocasiones necesitamos estimadores de los coeficientes que produzcan un modelo estable en las predicciones aunque se sacrifiquen ciertas propiedades del estimador $\widehat{\underline{\beta}}$. En esta sección se presentan diferentes técnicas de regularización para la estimación de los coeficientes del modelo lineal.

El estimador $\widehat{\underline{\beta}}$ puede ser numéricamente inestable cuando:

- Hay multicolinealidad (exacta o casi exacta)
- El número p de regresores es grande

Este problema se traduce en que el estimador adquiere una gran varianza con lo que las predicciones se vuelven imprecisas. Resulta por tanto de interés buscar estimadores de $\underline{\beta}$ que no sean insesgados, pero que a cambio tengan una menor varianza y permitan realizar mejores predicciones.

Las técnicas de regularización se aplican para obtener estimaciones de los coeficientes de regresión imponiendo una serie de restricciones al conjunto de soluciones admisibles, regularizándolas de alguna manera. Estas restricciones comprimen al vector $(\widehat{\beta}_1, \ldots, \widehat{\beta}_r)$ acotando su norma, permitiendo así una menor varianza. Las restricciones no acotarán al estimador de β_0, pues dependiendo de la escala que se use puede ser necesario que sea grande.

1.2.1.1. Regresión Ridge

Este método de regularización fue introducido en el artículo de Hoerl, A. E., & Kennard, R. W. (1970).

$$\min_{\underline{\beta}\in\mathbb{R}^{p+1}} \left\{ SCE(\underline{\beta}) + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}.$$

El vector de coeficientes se restringe de las covariables mediante la norma $\|\cdot\|_2$ (también llamada ℓ_2), sin embargo, la restricción no se aplica al término independiente.

1.2.1.2. Regresión Lasso

La regresión Lasso (Least Absolute Shrinkage and Selection Operator) fue introducida en Tibshirani, R. (1996). En este caso se utiliza la norma $\|\cdot\|_1$ o ℓ_1 para, dado $\lambda \geq 0$, plantear el problema de optimización

$$\min_{\underline{\beta}\in\mathbb{R}^{p+1}} \left\{ SCE(\underline{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

1.2.1.3. Elastic Net

Este es un método de regularización más sofisticado que usa como penalización una combinación lineal de las funciones de penalización de la regresión ridge y la regresión lasso. Fue publicado en Zou, H., & Hastie, T. (2005).

Para $\alpha \in [0,1]$ y para $\lambda \geq 0$, el método plantea el problema

$$\min_{\underline{\beta} \in \mathbb{R}^{p+1}} \left\{ \frac{SCE(\underline{\beta})}{2n} + \lambda \left[\frac{1-\alpha}{2} \sum_{j=1}^{p} \beta_j^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right] \right\}.$$

Si $\alpha = 1$ se tiene la regresión lasso y si $\alpha = 0$ se tiene la regresión ridge, o más bien, formulaciones equivalentes salvo constante multiplicativa que quedan cubiertas gracias al parámetro λ.

En todos los métodos, el parámetro λ controla la fuerza de la penalización y se suele seleccionar mediante validación cruzada para encontrar el modelo que mejor generalice a nuevos datos.

1.3. Modelos lineales dinámicos.

A la hora de trabajar con series temporales no estacionarias, los modelos lineales no resultan eficientes y pueden producir predicciones poco fiables, esto es debido a que se suele violar la hipótesis de no autocorrelación en los errores. La notación y el planteamiento del modelo se han extraído del capítulo 10 de Hyndman, R.J. & Athanasopoulos, G. (2021).

Recordemos que la forma de plantear el modelo lineal era,

$$y_t = \beta_0 + \beta_1 x_{1,t} + ... + \beta_r x_{r,t} + \epsilon_t \quad \forall t = 1, ..., T$$

con ϵ_t el error ruido blanco.

Sin embargo, si permitimos que los errores presenten autocorrelación, podemos construir un modelo que extienda a los modelos ARIMA y lineales, de manera que ahora el error ϵ_t de los modelos lineales se sustituya por un factor η_t que siga un modelo ARIMA.

Así, sirviéndonos de la notación *backshift notation* de los modelos ARIMA, suponiendo que η es un modelo $ARIMA(p,q,d)$, definimos el modelo lineal dinámico como:

$$y_t = \beta_0 + \beta_1 x_{1,t} + ... + \beta_r x_{r,t} + \eta_t$$

donde para cada t, el error η_t se modela a su vez como:

$$(1 - \phi_1 B - ... - \phi_p B^p)(1-B)^d \eta_t = (1 + \theta_1 B + ... + \theta_q B^q)\epsilon_t$$

El error η_t es propiamente un modelo $ARIMA(p,d,q)$, luego el error aleatorio ϵ_t sigue siendo una serie ruido blanco.

Estimación de parámetros

Análogamente al modelo lineal, se busca minimizar la suma de errores cuadrados, sin embargo, ya no del error η_t, pues encontraríamos problemas en las estimaciones de los coeficientes del modelo, en la construcción de sus intervalos de confianza y en el contraste fundamental del modelo. En este caso, se trata de minimizar la suma de cuadrados de ϵ_t, que evita los problemas mencionados.

Otro aspecto a tener en cuenta es que para que la estimación de paramétros sea consistente, necesitaremos que todas las series temporales que intervienen en el modelo sean estacionarias. No obstante, si tuviéramos inicialmente variables no estacionarias podríamos usar el proceso de integración vistos en los modelos ARIMA, dando lugar a un modelo equivalente al modelo dinámico original, denominado como *modelo en diferencias*.

$$y'_t = \beta_0 + \beta_1 x'_{1,t} + ... + \beta_p x'_{p,t} + \eta'_t$$

siendo,

$$(1 - \phi_1 B - ... - \phi_p B^p)(1 - B)^d \eta'_t = (1 + \theta_1 B + ... + \theta_q B^q)\epsilon_t$$

Aunque no todas las variables necesiten diferenciación, si hacemos esta transformación, será necesario hacerla en todas para no alterar la relación entre ellas.

1.4. Introducción a métodos de Machine learning

En esta sección se describen métodos de aprendizaje automático como Árboles de Regresión, Bosques Aleatorios, Gradient Boosting y Support Vector Regression. Estos representan herramientas fundamentales para el análisis económico moderno, permitiendo modelar relaciones no lineales complejas entre variables económicas, lo que resulta particularmente valioso en entornos económicos dinámicos donde las correlaciones entre variables pueden cambiar rápidamente.

Estos modelos están en continua evolución y presentan muchos matices, es por ello que en este documento se presentan las ideas básicas de cada uno de ellos. La información ha sido recopilada a partir de la asignatura Análisis de Datos Multivariantes, de Hastie, T., Tibshirani, R. & Friedman, J. (2017) y de Pérez, C. (2024).

1.4.1. Árboles de regresión

Los árboles de decisión constituyen una técnica predictiva consistente en dividir de manera secuencial la muestra, según una variable de interés, para obtener una clasificación fidedigna en grupos homógeneos.

La asignación de un elemento poblacional a un nodo se realiza de acuerdo a los valores de las diferentes variables en él. El método trata entonces de seleccionar las variables explicativas que son más discriminantes para la variable respuesta y construir una regla de decisión que establezca la predicción en cada nodo.

Nuestro objetivo, entonces, será modelar Y mediante un conjunto de variables explicativas $X_1, ..., X_r$, mediante un proceso de bifurcación en las variables.

El proceso consistirá en buscar la variable X_j para $j = 1, ..., r$ que mejor explique la variable Y, esta será la primera que defina una división de la muestra en dos subconjuntos L y R, que llamaremos nodos. Después, se reitera el procedimiento en el interior de cada uno de estos conjuntos buscando la segunda mejor variable y así sucesivamente. Una vez establecidos los nodos terminales, deberemos establecer la predicción en esos nodos, de manera que si tenemos M^* nodos terminales y dada una realización muestral $(x_1, ..., x_r)$ entonces el modelo será

$$\widehat{f}(X) = \sum_{m^*=1}^{M} c_{m^*} I_{(x_1,...x_r)\in R_{m^*}}$$

siendo R_{m^*} el nodo terminal m^* y c_{m^*} la constante que dicta el valor de $\widehat{y}$ en el nodo.

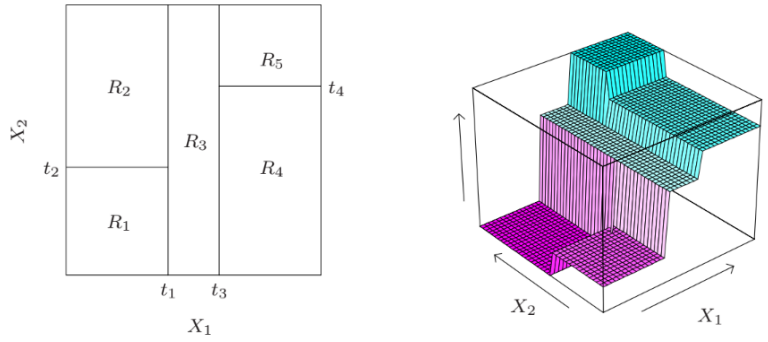

Figura 1.9: Ejemplo de Árbol de regresión con dos variables explicativas, 5 nodos terminales y constantes c1 = -5, c2 = -7, c3 = 0, c4 = 2 c5 = 4. (Hastie, T., Tibshirani, R. , Friedman, J. (2017))

En este punto, existen varios factores a tener en cuenta.

- **Estrategias de predicción** Debemos determinar el valor de la constante en cada nodo. Lo habitual será usar como constante c_{m^*} la media de las y pertenecientes al nodo m^*. Es decir,

$$c_{m^*} = \frac{1}{|R_{m^*}|} \sum_{i\in R_{m^*}} y_i$$

- **Estrategias de ramificación del árbol.** Supongamos que realizamos una cierta partición de nuestro conjunto de datos total en dos conjuntos L y R. Se define la impureza de la partición anterior como:

$$i(L, R) = \frac{|L|}{|L \cup R|}(\frac{1}{|L|}\sum_{i\in L}(y_i - c_L)^2) + \frac{|R|}{|L \cup R|}(\frac{1}{|R|}\sum_{i\in R}(y_i - c_R)^2)$$

  Donde c_L y c_R serán las medias muestrales de los conjuntos L y R, respectivamente.

Algoritmo de construcción

1. Construir un árbol, formado inicialmente con un nodo raíz y asociar a este el conjunto total de datos.

2. Mientras queden en el árbol nodos con más de κ_0 registros, para cada uno de estos nodos $\tilde{n}$:

    1. Seleccionar la variable y el corte que produzcan mínima impureza
    2. Ramificar $\tilde{n}$ en los nodos L y R y asociar a ellos los resgistros de $\tilde{n}$ correspondientes

3. Una vez haya menos de κ_0 registros en cada nodo, llamamos M^* al número final de nodos y a cada nodo final R_{m^*}. Asociar a cada nodo una predicción c_{m^*}

Uno de los parámetros a elegir para el modelo de árboles de regresión sería el número mínimo de registros en los nodos κ_0. Podríamos entrenar el modelo con varios valores de κ_0 y escoger el modelo que minimice el error respecto a una muestra de prueba. No obstante, existe otro algoritmo que nos permite ajustar el número de nodos terminales, de manera que la elección de κ_0 no será tan trascendente, únicamente deberemos escogerlo suficientemente pequeño para obtener un árbol completo al que aplicarle el siguiente algoritmo.

Poda del árbol

Una vez construido nuestro árbol, será conveniente comprobar que no está sobre ajustado. Para esto realizaremos el siguiente proceso. Secuencialmente, deberemos ir eliminando los nodos terminales mientras se mejore la siguiente medida de complejidad

$$C_\alpha(T) = \sum_{t \in M^*} |R_{m^*}| i(R_{m^*}) + \alpha |M^*|$$

siendo $\alpha \geq 0$, M^* el conjunto de nodos terminales y $i(R_{m^*})$ la impureza en el nodo terminal R_{m^*}.

Una de las mayores ventajas de los árboles es su gran explicabilidad, ya que su forma y construcción es bastante intuitiva y sencilla de representar.

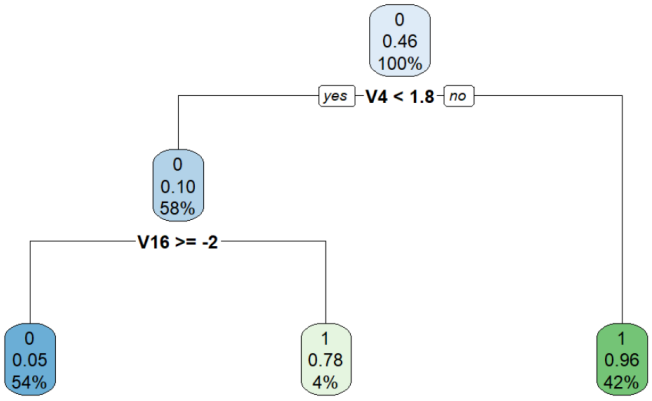

Figura 1.10: Ejemplo de Árbol construido con la función rpart de R

Por si solos los árboles suelen tener menos poder predictivo que otros métodos de regresión vistos hasta ahora. Sin embargo, usando técnicas que involucran varios árboles

de decisión como los métodos de *Bosques aleatorios* o *Boosting* mejoran sustancialmente la bondad de las predicciones.

1.4.2. Bosques Aleatorios

Nace de la idea de trabajar con varios árboles de clasificación y regresión simultáneamente. Actualmente es un método muy usado en el aprendizaje automático e inteligencia artificial explicable. Se basa en la generación de un gran número de árboles en una muestra *bootstrap*.

Consiste en aplicar secuencialmente el siguiente algoritmo,

1. Tomar $B = 500, 1000, \ldots$ Desde $b = 1$ hasta B
    1. Generar una muestra con reposición del mismo tamaño que la muestra de aprendizaje (muestra bootstrap)
    2. Construir un árbol T_b en la muestra anterior repitiendo recursivamente los siguientes pasos hasta que cada nodo terminal del árbol alcance un número mínimo κ_0 de registros:
        1. Generar una muestra aleatoria simple sin reposición de $\tilde{m}$ variables de las r totales.
        2. Seleccionar la mejor variable entre las $\tilde{m}$ escogidas y el corte que minimice la impureza.
        3. Ramificar el nodo en sus dos nodos hijos
2. Para hacer predicciones sobre nuevos puntos x, consideramos la media de las predicciones de los distintos árboles.

$$\widehat{f}_{rf}^{B} = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

Se observa que la muestra generada para la construcción de los árboles de regresión de los Bosques Aleatorios se considera con reposición, esto quiere decir que podemos encontramos el mismo registro varias veces y a su vez, registros de la muestra de aprendizaje que no aparezcan en la muestra usada por el árbol. Estos últimos se conocen como registros *fuera de la bolsa* y nos pueden ser útiles como muestra independiente para estimar el error de predicción del bosque $\frac{1}{B}\sum_b e(T_b)$, siendo

$$e(T_b) = \frac{1}{T}\sum_{t=1}^{T}(y_t - T_b(x_t))^2$$

Luego gracias a los registros fuera de bolsa se construye el estimador de error de predicción del bosque $\frac{1}{B}\sum_b \widehat{e(T_b)}$ mediante el estimador del error de predicción de cada árbol,

$$\widehat{e(T_b)} = \frac{1}{U}\sum_{i=1}^{U}(y_i - T_b(x_i))^2$$

con U el número de elementos en la muestra fuera de bolsa.

Esta estimación nos será útil para medir la importancia de cada variable en la regresión, lo cuál se hará mediante un proceso de permutación en los registros fuera de la bolsa por cada variable.

Es decir, supongamos que queremos medir la importancia de la variable j. Realizamos la permutación en la columna j de los registros fuera de bolsa. Una vez hecha la permutación, revaluamos el error de predicción, si este es muy diferente al error antes de la permutación, diremos que la variable tiene mucha importancia, de lo contrario la importancia de la variable será escasa.

1.4.3. Gradient Boosting

Su fundamento reside en la idea de construir un modelo predictivo robusto a partir de la combinación secuencial de múltiples modelos más simples, comúnmente denominados aprendices débiles (weak learners). La principal diferencia con el modelo Random Forests reside en su proceso de construcción secuencial y adaptativo. En lugar de entrenar modelos de forma independiente y promediar sus predicciones, el Gradient Boosting construye el modelo de forma aditiva: cada nuevo aprendiz débil se entrena para corregir los errores o las deficiencias del modelo construido hasta ese momento. El Gradient Boosting puede entenderse como un procedimiento para ajustar modelos aditivos mediante una estrategia progresiva (stagewise).

1. Modelos Aditivos: Un modelo aditivo toma la forma general:

$$F_m(x) = \sum_m \beta_m h_m(x)$$

   Donde $h_m(x)$ son las funciones base (los aprendices débiles, por ejemplo, árboles de decisión), y β_m son los coeficientes (o pesos) de expansión. $F_0(x)$ suele ser una estimación inicial simple, como la media de la variable objetivo.

2. Función de pérdida: El "gradient" en Gradient Boosting proviene de la idea de que cada nuevo aprendiz débil $h_m(x)$ debe ajustarse para apuntar en la dirección del gradiente negativo de la función de pérdida con respecto a las predicciones del modelo $F_{m-1}(x_i)$ en cada observación x_i. Estos gradientes negativos se conocen como pseudo-residuales:

$$r_{im} = -[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)}$$

   El aprendiz débil $h_m(x)$ se entrena para predecir estos pseudo-residuales r_{im}. Una vez que $h_m(x)$ está entrenado, se determina su contribución óptima β_m para minimizar la pérdida general.

Algoritmo de Gradient Boosting

1. Inicialización del modelo: Para una función de pérdida $L(y, F)$, el modelo inicial $F_0(x)$ se establece como:

$$F_0(x) = arg\,\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma)$$

2. Iterar para construir los aprendices débiles ($m = 1$ hasta M):

    1. Calcular los pseudo-residuales: Para cada observación $i = 1, ..., N$, calcular los pseudo-residuales r_{im} basados en el gradiente negativo de la función de pérdida con respecto a las predicciones del modelo de la iteración anterior $F_{m-1}(x)$:

$$r_{im} = -[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x_i)}$$

    2. Ajustar un aprendiz débil: Entrenar un aprendiz débil $h_m(x)$ (por ejemplo, un árbol de regresión) utilizando los pseudo-residuales r_{im} como valores objetivo. Es decir, se ajusta $h_m(x)$ a los datos $(x_i, r_{im})_{i=1,..,N}$.
    3. Calcular el multiplicador óptimo (o peso de la hoja para árboles): Determinar el valor óptimo del multiplicador γ_m para el aprendiz débil $h_m(x)$. Resolver:

$$\gamma_m = arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

    4. Actualizar el modelo aditivo:

$$F_m(x) = F_{m-1}(x) + \nu\gamma_m h_m(x)$$

    Donde ν es la tasa de aprendizaje (o "shrinkage"), un hiperparámetro entre 0 y 1 que reduce la contribución de cada nuevo árbol. Esto ayuda a prevenir el sobreajuste y mejora la generalización.

3. Salida del modelo final: El modelo final es $F_M(x)$.

1.4.4. Support Vector Regression

La Regresión de Vectores de Soporte, conocida por sus siglas en inglés SVR (Support Vector Regression), es una técnica de aprendizaje automático supervisado que extiende los principios de las Máquinas de Vectores de Soporte (SVM), originalmente diseñadas para problemas de clasificación, al ámbito de la regresión.

El método resulta especialmente útil en el campo de los datos económicos, donde las relaciones entre variables a menudo son complejas y no lineales, SVR ofrece una alternativa potente a los modelos de regresión lineal tradicionales. Se tratará de dar una idea básica acerca del método. Además de los autores mencionados al inicio de la sección, para este método se aportan referencias de Sethi, A. (2025).

A diferencia de otros métodos que buscan minimizar el error cuadrático medio sobre todos los puntos de datos, SVR se enfoca en encontrar una función que se ajuste a los datos dentro de un margen de error específico, ignorando los puntos que ya están bien predichos, lo que puede llevar a modelos más generalizables y menos sensibles a valores atípicos.

A grandes rasgos, el objetivo será construir un **hiperplano** en el que el modelo SVR intenta ajustar a los datos. El objetivo no es que el hiperplano pase por la mayor cantidad de puntos posible, sino que represente la tendencia central de los datos. A la hora de construir el modelo deberemos tener en cuenta:

- Los **márgenes de tolerancia**, también conocidos como el *tubo épsilon-insensible*, son cruciales en SVR. Se definen dos límites paralelos al hiperplano, a una distancia ϵ por encima y por debajo de él. La característica distintiva de SVR es que los errores de predicción para los puntos de datos que caen dentro de este tubo no se penalizan.
- La **elección de** ϵ es fundamental: un valor pequeño de ϵ resultará en un tubo más estrecho, lo que puede llevar a un modelo más complejo que intente ajustarse más de cerca a los datos de entrenamiento (potencialmente causando sobreajuste), mientras que un valor grande de ϵ permitirá un tubo más ancho, resultando en un modelo más simple y posiblemente más generalizable, pero que podría ignorar variaciones importantes en los datos.
- Los **vectores de soporte** son aquellos puntos de datos que se encuentran exactamente en los límites del margen de tolerancia o fuera de él. Estos son los puntos críticos que "soportan" o definen la estructura del hiperplano y los márgenes.
- **Escalado de Características**: SVR es sensible a la escala de las variables de entrada. Luego, es habitual hacer una estandarización de las variables.
- **Selección del Kernel:** Seleccionaremos la función kernel radial o gaussiano por su capacidad de captar las relaciones no lineales en las variables.

  $$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2)$$

  Con γ un parámetro a determinar.

El modelo tratará de encontrar un hiperplano $f(x) = wx + b$, mediante la resolución de un problema de optimización cuadrática para encontrar los valores óptimos de w y b.

1.5. Comparación de modelos. Medidas de bondad de ajuste

Coeficiente de determinación: En los modelos lineales, consideramos el coeficiente de determinación R^2 como medio para evaluar la bondad de ajuste a los datos observados.

$$R^2 = \frac{SCR}{SCT} = \frac{\sum(\widehat{y}_t - \overline{y})^2}{\sum(y_t - \overline{y})^2} = 1 - \frac{\sum(\widehat{y}_t - y_t)^2}{\sum(y_t - \overline{y})^2} = 1 - \frac{SCE}{SCT}$$

verificándose que $0 \leq R^2 \leq 1$. Se interpreta como la proporción de variabilidad total que es explicada por el modelo, luego cuanto más proxima esté de 1 , o equivalentemente cuando más cercano a 0 sea SCE, mejor será el ajuste.

Además tambien tenemos medidas de comparación de bondad de ajuste entre varios modelos.

Criterio de información de Akaike.

La idea de este coeficiente es penalizar la suma de errores cuadrados, con el número de parámetros a estimar (grados de libertad). Generalmente, su expresión es,

$$AIC = Tlog(\frac{SCE}{T}) + 2(p + 2)$$

Criterio de información de Akaike con corrección. Para valores pequeños de T,

$$AIC_c = AIC + \frac{2(p+2)(p+3)}{T-p-3}$$

Criterio de información bayesiano

$$BIC = Tlog(\frac{SCE}{T}) + log(T)(p+2)$$

Nos interesará quedarnos con el modelo que menor valor presente en estos criterios.

Capítulo 2

Análisis de datos de inflación

El objetivo de este capítulo es abordar el análisis y predicción de la inflación mediante los modelos propuestos en el Capítulo 1. La metodología se aplica a datos mensuales de inflación de países de la Unión Europea e incorpora variables explicativas influyentes con el fin de realizar un estudio de las relaciones entre los indicadores y producir estimaciones fiables.

Este análisis no solo busca identificar los patrones históricos de la inflación, sino también evaluar la capacidad predictiva de los diferentes modelos en el contexto económico.

2.1. Modelos ARIMA

Para el desarrollo de los modelos ARIMA, se usarán datos de la inflación desde 1997 en los países de la Unión Europea, extraídos de la base de datos Eurostat. Se representan en la siguiente serie temporal:

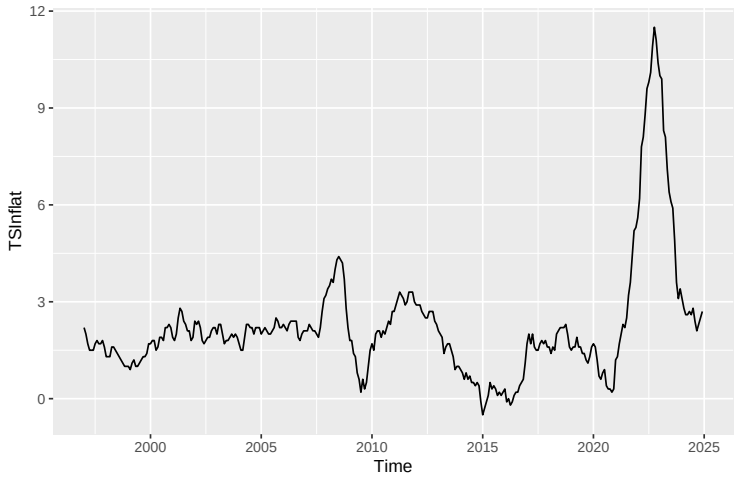

Figura 2.1: Serie temporal de inflación desde 1997 en la UE.

En la serie se puede observar ciertos periodos de inestabilidad provocados por grandes crisis económicas cíclicas, entre las que destaca la explosión de la burbuja inmobiliaria de 2008, aunque también se observa la repercusión que tuvo en la economía europea crisis exógenas como la sanitaria provocada por el Covid-19 y la guerra de Ucrania.

De este contexto tan volátil se deduce que la inflación no tendrá un comportamiento estacionario. Esto se observa en el gráfico de autocorrelaciones, pues la correlación con tiempos pasados disminuye muy lentamente (Dayal, V. (2020)).

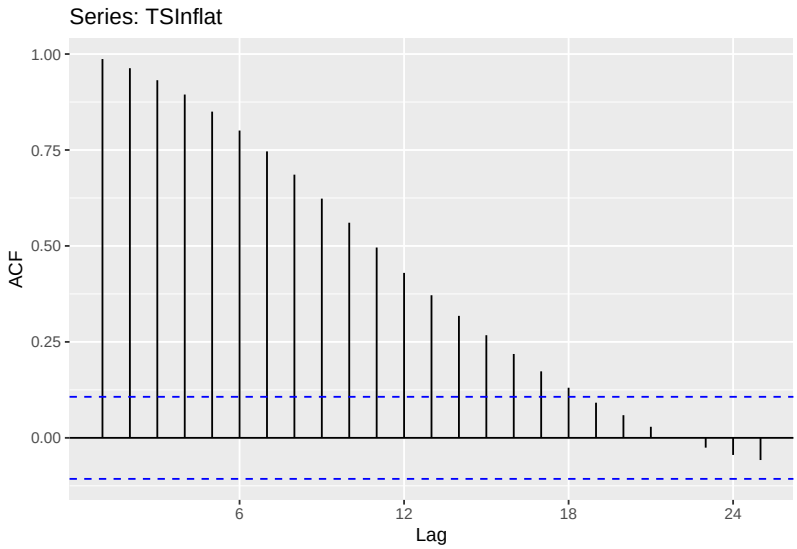

Figura 2.2: ACF de la serie temporal de inflación.

No obstante, es necesario realizar un contraste de estacionariedad para llegar a una conlusión.

Se realiza el test ADF (Fuller, W. A. (1976) y Dickey, D. A. (1984)) a los datos de inflación:

statistic	p.value	parameter	method	alternative
-4.2	0.01	6	Augmented Dickey-Fuller Test	stationary

A pesar de lo que sugiere la gráfica, el test indica que la serie es estacionaria. Ante esta situación contradictoria, es lógico preguntarse si el test está siendo efectivo.

Nota: Realmente, los test que verifican estacionariedad tienen como hipótesis nula que el polinomio característico de la parte autorregresiva del modelo ARIMA tenga al uno como raíz. El motivo es el siguiente:

Recordemos que un modelo autorregresivo $AR(p)$ es de la forma:

$$(1 - \phi_1 B - ... - \phi_p B^p) X_t = \epsilon_t$$

siendo B el operador de retroceso. El polinomio característico de este proceso estocástico es $P(z) = (1 - \phi_1 z - ... - \phi_p z^p)$. Supongamos que este polinomio tiene a 1 como raíz, entonces

$$P(1) = 0 \Leftrightarrow (1 - \phi_1 - ... - \phi_p) = 0 \Leftrightarrow \sum_{i=1}^{p} \phi_i = 1$$

El caso más simple donde ya pasaba esto es en las series Random Walk, descritas en el Capítulo 1:

$$X_t = X_{t-1} + \epsilon_t = (X_{t-2} + \epsilon_{t-1}) + \epsilon_t = \ldots = X_0 + \sum_{i=1}^{t} \epsilon_i$$

Fijado $X_0 = x_0$ y teniendo en cuenta que el modelo toma como hipótesis que los residuos son incorrelados y tienen varianza constante, entonces

$$Var(X_t) = Var(x_0 + \sum_{i=1}^{t} \epsilon_i) = \sum_{i=1}^{t} Var(\epsilon_t) = t\sigma^2$$

Es decir, la varianza de este tipo de series aumenta con el tiempo y por tanto son no estacionarias.

A grandes rasgos, los test de estacionariedad contrastan que la parte autorregresiva tenga de raíces de módulo cercano a uno, pues esto indica que la serie es no estacionaria.

Con el fin de esclarecer lo ocurrido con el test ADF, vamos a plantear el modelo y calcular las raíces de la parte autorregresiva.

	Model	Parameter	estimate	std.error
ar1	ARIMA(1,1,1)	ar1	0.82	0.065
ma1	ARIMA(1,1,1)	ma1	-0.52	0.098

La raíz de la parte autoregresiva es:

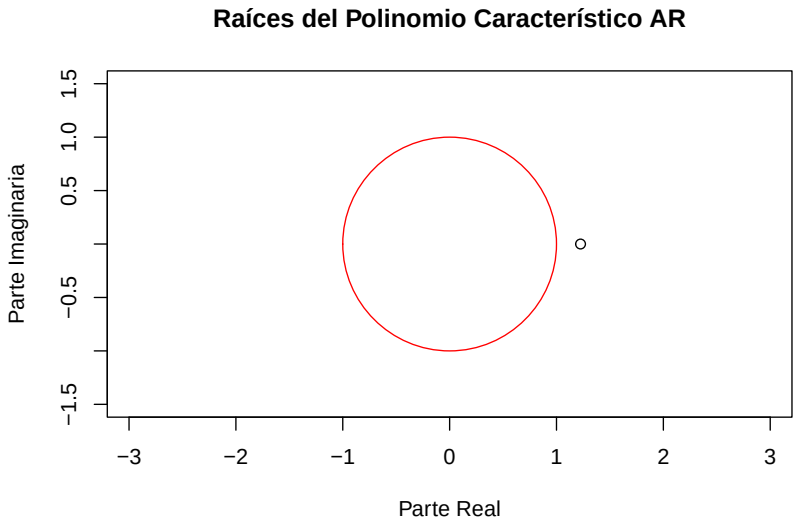

Figura 2.3: Representación de la ráiz del modelo ARIMA para la inflación

La raíz es de módulo relativamente próximo a uno, lo que indica no estacionariedad, luego todo parece indicar que el test ADF no es efectivo para la serie temporal que nos ocupa.

Como alternativa, se realiza otro test que verifica estacionariedad, en este caso vamos a usar el test PP (Phillips, P. C. B. & Perron, P. (1988)):

statistic	p.value	parameter	method	alternative
-12.36	0.4178	5	Phillips-Perron Unit Root Test	stationary

En este caso se obtiene el resultado esperado, la serie no es estacionaria. Este desacuerdo entre test es una situación bastante común en el análisis de series temporales y se debe a las diferencias en su construcción.

El test ADF realiza una corrección paramétrica de la autocorrelación en los residuos añadiendo términos rezagados de la variable dependiente diferenciada a la regresión, por lo que su validez depende de que los residuos finales de esta regresión sean ruido blanco.

En cambio, el test PP utiliza una corrección no paramétrica para la autocorrelación y la heterocedasticidad en los residuos, lo que lo hace más robusto ante series cuyos modelos no verifican las hipótesis fundamentales.

En conclusión, en modelos donde previsiblemente no se verifiquen las hipótesis fundamentales de los residuos, el test ADF no siempre es fiable, mientras que el test PP constituye una alternativa más robusta.

La información acerca de ambos test y las diferencias entre ellos han sido recopilados de Cryer, J.D. & Chan, K.S. (2008) y de Shumway, R.H. & Stoffer, D.S. (2017)

La poca eficiencia del test ADF hace ver que los residuos del modelo ARIMA no se comportan según las hipótesis fundamentales. No obstante, esto se debe comprobar con rigor estadístico:

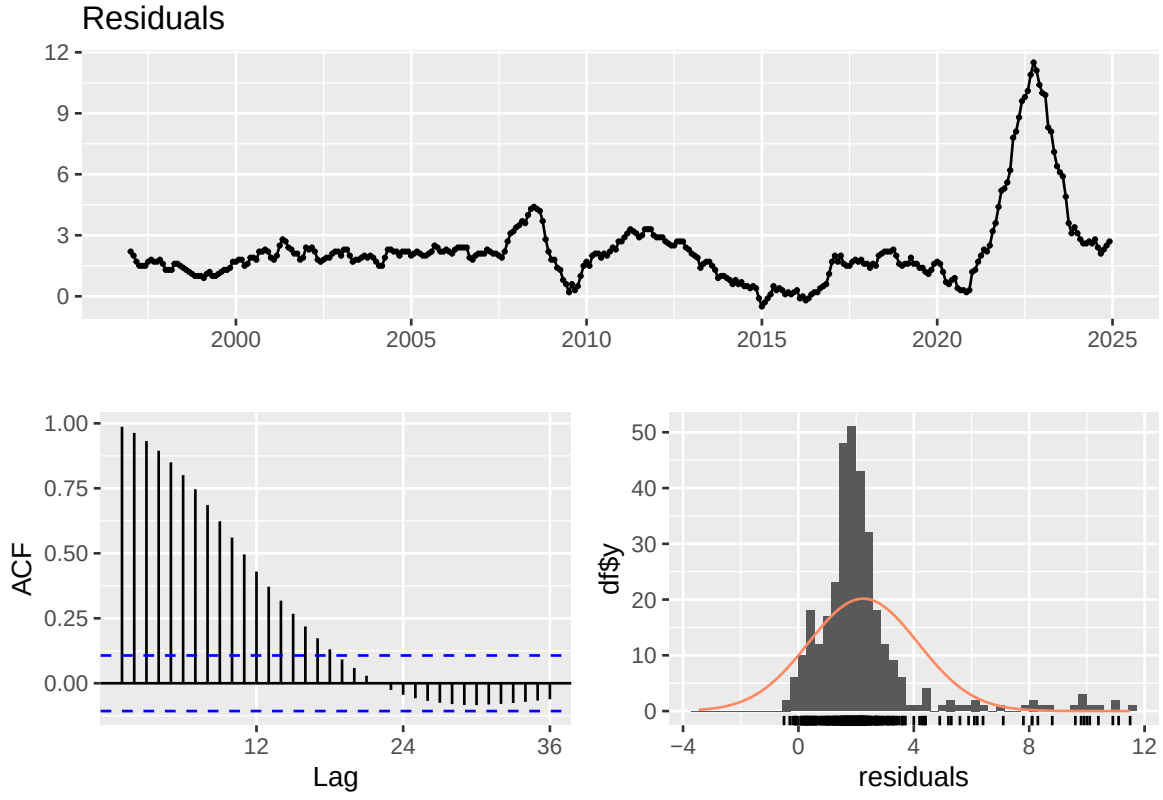

Figura 2.4: Diagnóstico de residuos para TSInflat.

```
##
##  Ljung-Box test
##
## data:  Residuals
## Q* = 2582.6, df = 24, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 24
```

Como se venía previendo, según el test de Ljung-Box existe evidencia estadística de que el modelo no verifica las hipótesis de homocedasticidad y no autocorrelación en los residuos. Esto indica que no es un modelo adecuado para los datos que nos ocupan.

A pesar de ello, veamos las predicciones que genera.

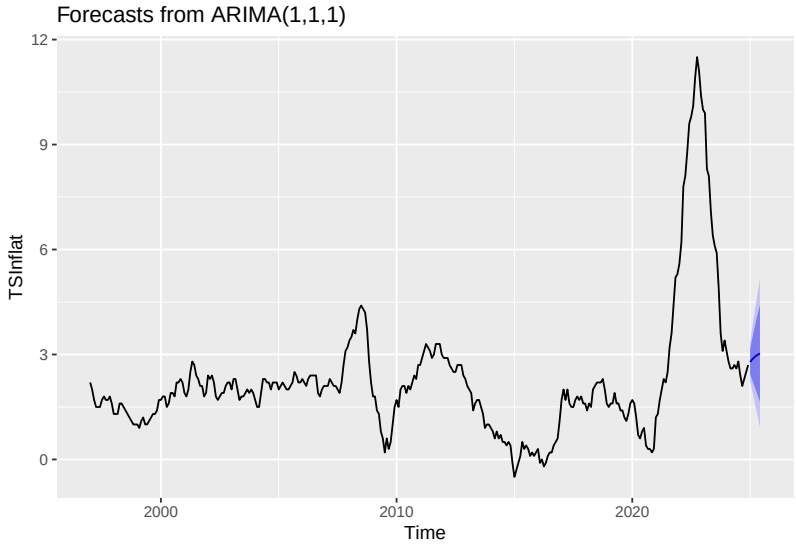

Figura 2.5: Predicción de inflación con modelo inicial (modInfl).

Vamos a tratar de eliminar un poco de ruido en los datos, ignorando la etapa post Covid, donde la inflación ha seguido un comportamiento difícil de explicar mediante los datos de tiempos pasados.

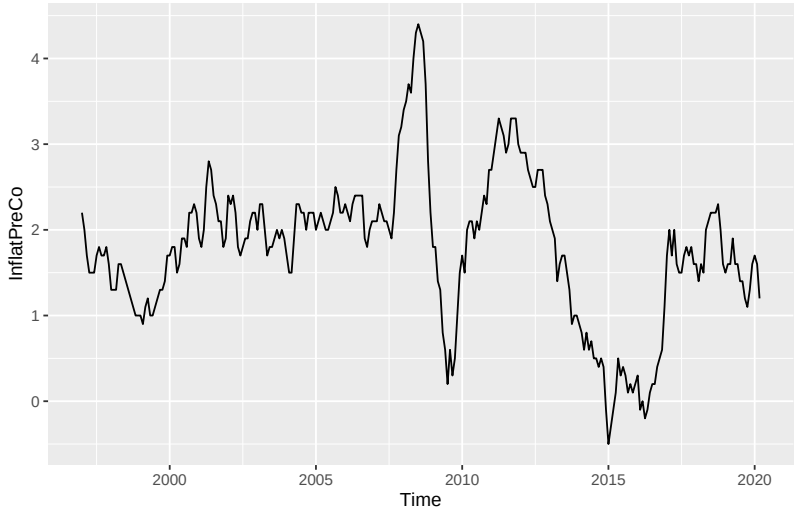

Figura 2.6: Serie temporal y ACF de inflación pre-COVID.

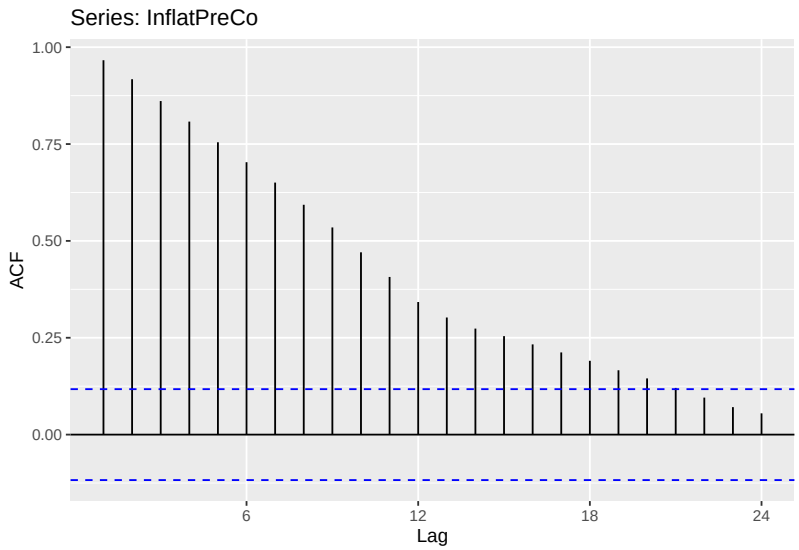

Figura 2.7: Serie temporal y ACF de inflación pre-COVID.

Se deduce no estacionariedad en la serie, pues las correlaciones con tiempos pasados va disminuyendo paulatinamente. No obstante, se realizan los test ADF y PP de contraste de estacionariedad:

statistic	p.value	parameter	method	alternative
-3	0.16	6	Augmented Dickey-Fuller Test	stationary

statistic	p.value	parameter	method	alternative
-14.97	0.2704	5	Phillips-Perron Unit Root Test	stationary

Ahora sí, ambos test coinciden y concuerdan con la gráfica, la serie es no estacionaria. Aplicando la integración:

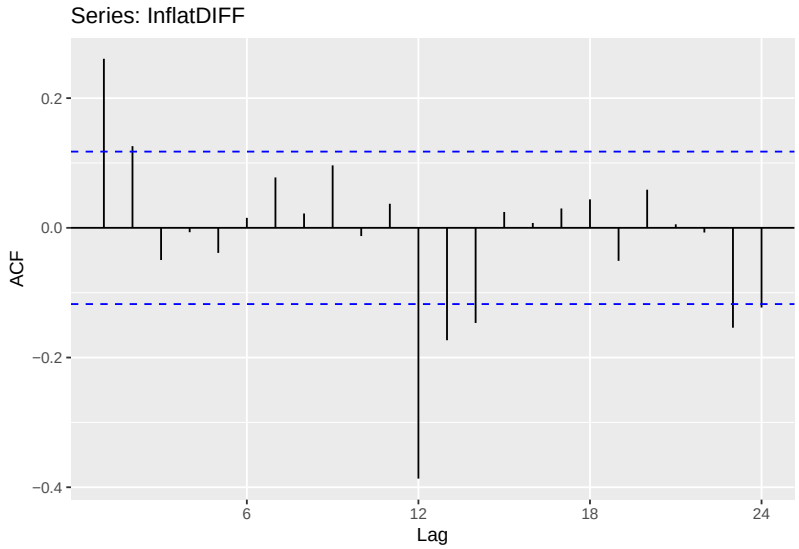

Figura 2.8: ACF de la serie diferenciada (InflatDIFF).

statistic	p.value	parameter	method	alternative
-202	0.01	5	Phillips-Perron Unit Root Test	stationary

Se asume la estacionariedad en la serie tras el proceso de integración de orden uno. Seguidamente, se formula el modelo y se comprueban sus hipótesis:

	Model	Parameter	stimation	std.error
ar1	ARIMA(1,1,0)	ar1	0.26	0.058

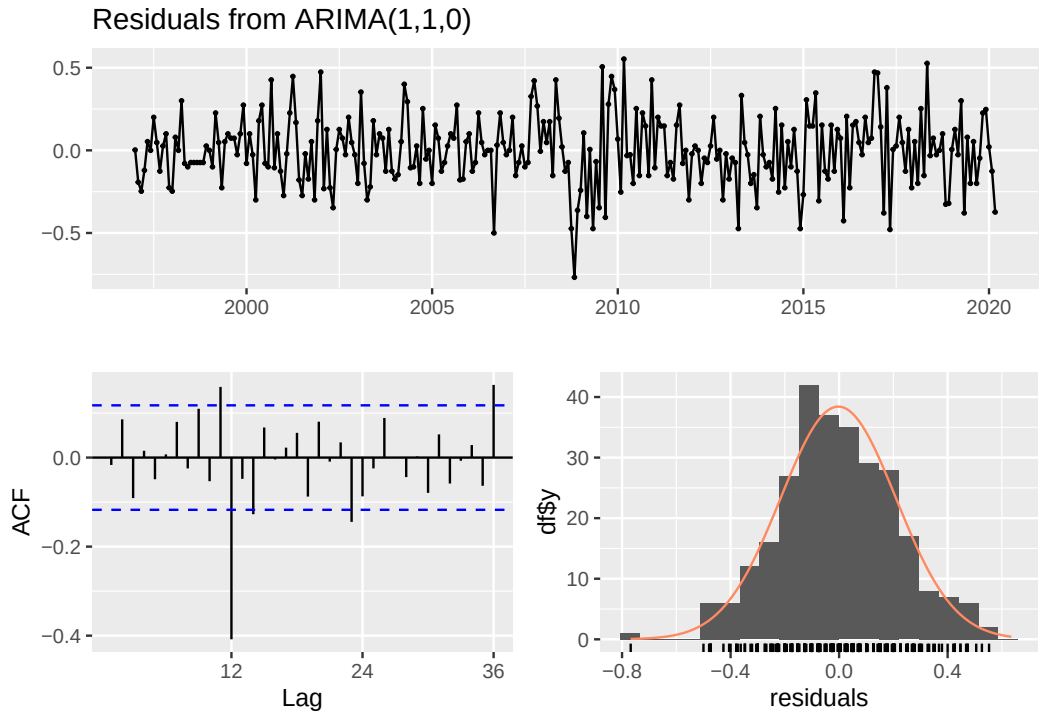

Figura 2.9: Diagnóstico de residuos (modelo modInfl).

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)
## Q* = 89.175, df = 23, p-value = 9.774e-10
##
## Model df: 1.   Total lags used: 24
```

Aunque se observa una mejora sustancial del comportamiento de los residuos, el contraste de no autocorrelación y homocedasticidad se rechaza de nuevo (aunque con un p.valor bastante mayor).

Es decir, el modelo ARIMA en ningún caso ha resultado ser un modelo adecuado para nuestros datos, ya sea incluyendo o no el periodo de Covid.

Debido a la gran inestabilidad económica, las series reflejan un alto grado de variabilidad, esto hace que no se verifiquen las hipótesis fundamentales del modelo y por tanto las predicciones tengan un gran nivel de incertidumbre.

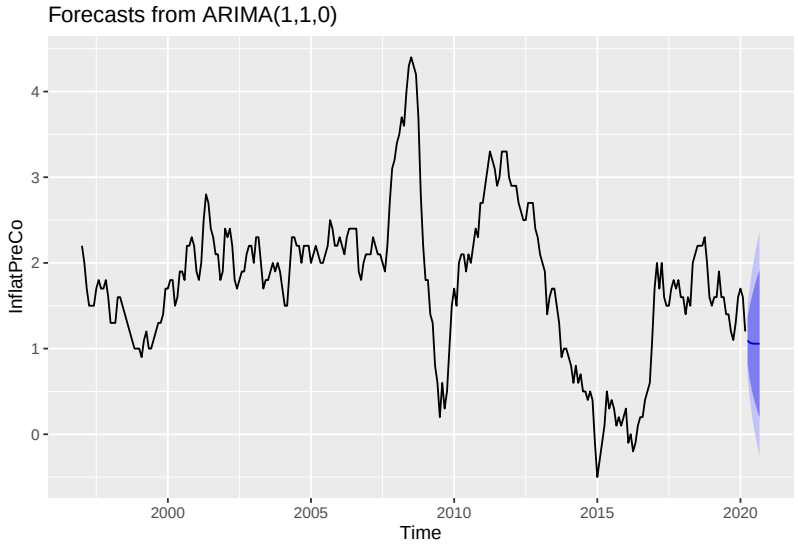

Figura 2.10: Predicción de inflación con modelo pre-COVID.

2.2. Planteamiento de los modelos.

En esta sección, se necesitará recoger un conjunto de variables explicativas con el fin de predecir la inflación mediante las técnicas de regresión estudiadas. Nuestro objetivo será plantear los modelos, verificar sus hipótesis y analizar su comportamiento.

Los códigos de R usados para la generación de los modelos y gráficas han sido elaborados a partir de Hyndman, R.J. & Athanasopoulos, G. (2021), de Pérez, C. (2024) y de las respectivas librerías de R.

Variables explicativas. Composición de la base de datos

Recopilaremos las variables explicativas, se hará un análisis exploratorio y se llegarán a conclusiones sobre la importancia estadística de las mismas.

Consideramos una base de datos elaborada a partir de la base de datos europea Eurostat. Los datos recogidos están comprendidos en el periodo de enero de 2015 hasta diciembre de 2024. Algunas de las variables incluidas en la base datos vienen definidas en la introducción del presente documento, otras serán presentadas a continuación:

- Food price monitoring tool: Nos referimos a ella como *FPMT(rez.)*. Según la sección de metadata de Eurostat este índice pretende analizar los datos disponibles sobre la evolución de los precios a lo largo de la cadena de suministro, para ello, compara los índices de 4 indicadores:
  - HIPC: Harmonised Indices of Consumer Prices.
  - Índice de precios al productor para la producción nacional (PPId): mide la variación de los precios de los bienes y servicios que se producen en un país y que están destinados tanto al mercado interno como a la exportación.
  - Índice de precios a la importación de productos: rastrea la evolución mensual de los precios de los productos que un país importa de otras naciones.
  - Índice de precios a los productos agrícolas (ACP): mide los cambios en los precios que reciben los agricultores por los productos que venden.

Se pueden consultar las bases de datos históricas de estos indicadores en el apartado "other price statistics" de la página web de Eurostat.

- HIPC: Los datos se recogen mensualmente en el apartado "prices" de Eurostat.
- Desempleo: También tiene periodicidad mensual. Extraídos del apartado "Labour market" de Eurostat.
- Inflación: Se mide en este caso como la tasa de cambio anual del HIPC. Los datos históricos se encuentran en el apartado "prices" de Eurostat.
- Índice porcentual de variación de la energía: En la base de datos se incluye como *EI(rez.)*. Mide de la variación interanual de los precios de los productos energéticos que consumen los hogares. Incluye electricidad, gas, combustibles líquidos (como gasolina, diésel) y combustibles sólidos. Se puede extraer de la carpeta "prices" de Eurostat

- Índice de Precios de la Vivienda: Se incluye como *HP* en nuestra base de datos. Mide los cambios en los precios de transacción de todo tipo de viviendas residenciales compradas por los hogares. Esto incluye tanto viviendas nuevas como existentes. Sirve como un indicador de tendencias de precios específicas en el mercado inmobiliario.

FPMT(rez.)	HIPC(rez.)	Desempleo(rez.)	Inflacion	EI(rez.)	HP(rez.)	Mes
99.00	100.15	10.8	-0.5	-2.0	1.0	2015 ene.
99.17	100.14	10.8	-0.3	-1.6	1.2	2015 feb.
99.16	99.93	10.9	-0.1	-2.1	1.2	2015 mar.
99.20	99.84	10.7	0.1	-5.8	1.2	2015 abr.
99.70	98.57	10.6	0.5	-8.6	1.4	2015 may.
100.13	99.07	10.6	0.3	-7.6	1.4	2015 jun.

Nota: Un aspecto a tener en cuenta en la base de datos es que las variables explicativas toman valores rezagados cuatro meses, por ejemplo, el valor de HIPC que tomamos en enero de 2015 es realmente su valor en septiembre de 2014. El motivo de hacerlo así es que nuestro estudio tiene un importante factor temporal, para predecir la inflación en un tiempo t futuro, no tiene sentido considerar las variables explicativas en dicho tiempo t, pues necesitaríamos también predicciones sobre esas variables. Esto se indica en la base de datos mediante el sufijo -(rez.) en el nombre.

Análisis exploratorio de los datos

Se puede observar en la evolución de estos indicadores a lo largo del periodo como todos presentan tendencias en sus comportamientos, en el caso de los indicadores relacionados con el IPC, ascendente, en cambio, en el caso del desempleo esta tendencia es descendente. Esto presenta cierta lógica, pues cuanto menor sea el número de personas desempleadas, más dinero estará en movimiento, lo cual se traduce en el aumento de precios. A su vez, se aprecia como los indicadores relacionados con la energía y el mercado inmobiliario tienen un comportamiento similar al de la inflación, aunque en diferentes escalas.

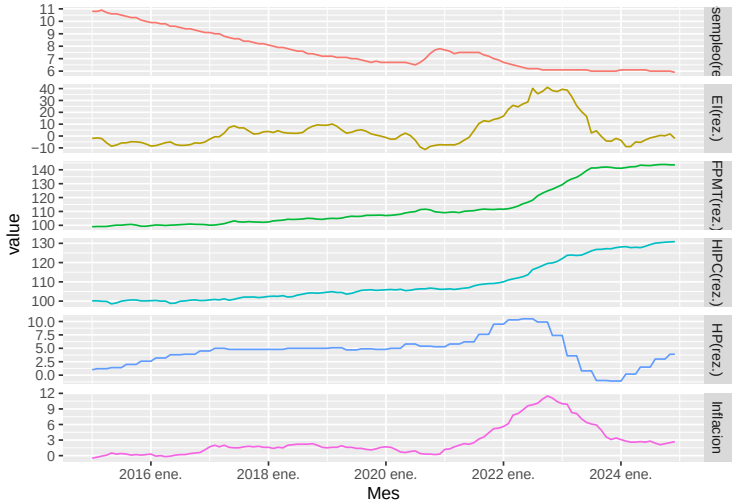

Figura 2.11: Evolución de indicadores económicos.

En cuanto al estudio estadístico de estas series temporales se deduce no estacionariedad, pues la tendencia y la variabilidad es no uniforme a lo largo del tiempo.

Veamos cómo se relacionan entre si las variables mediante sus gráficos de dispersión y la matriz de correlaciones.

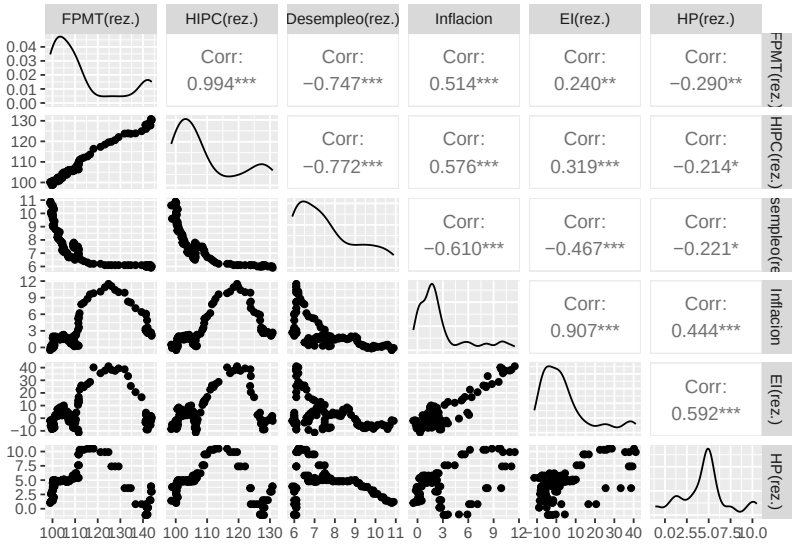

Figura 2.12: Matriz de dispersión de indicadores económicos.

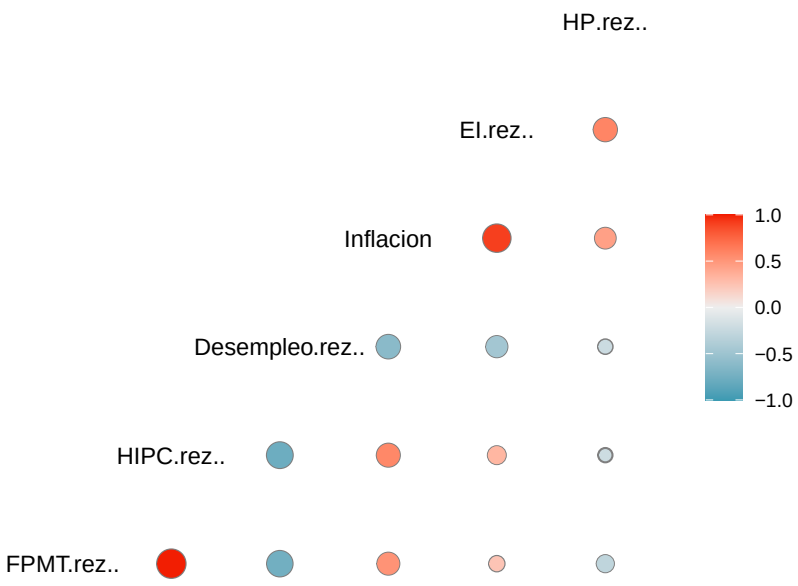

Figura 2.13: Matriz de correlaciones de indicadores económicos.

Se observa en los gráficos de dispersión patrones no lineales en la mayoría de relaciones entre variables, lo cual es usual en el contexto económico. Debido a esta falta de linealidad parece difícil que los modelos lineales puedan ser eficientes en estos casos. No obstante, se debe analizar su comportamiento.

2.2.1. Modelo lineal múltiple

Antes de nada, haremos una selección de variables, eligiendo el modelo con menor AIC en este caso. El método que usaremos será un procedimiento paso a paso, que funciona añadiendo o quitando variables hasta que nuestras métricas de bondad de ajuste no mejore más. La aplicación de esta técnica en R ha sido extraída de la asignatura Modelos Lineales y Diseño de Experimentos.

```
## lm(formula = Inflacion ~ `EI(rez.)` + `FPMT(rez.)` + `HP(rez.)` +
##     `Desempleo(rez.)` + `HIPC(rez.)`, data = datosEcon)
```

En este caso, se ha aplicado el método paso a paso hacia delante y ha tenido como resultado el siguiente modelo:

term	estimate	std.error	statistic	p.value
(Intercept)	-5.4833	3.773134	-1.453	1.489e-01
`EI(rez.)`	0.1833	0.009607	19.075	2.848e-37
`FPMT(rez.)`	0.2556	0.072515	3.524	6.122e-04
`HP(rez.)`	0.2327	0.053614	4.341	3.089e-05
`Desempleo(rez.)`	0.3389	0.105852	3.201	1.773e-03
`HIPC(rez.)`	-0.2296	0.104562	-2.195	3.015e-02

El modelo resultante es el que incluye todas las variables. Esta es la evolución del AIC según se añaden variables a partir del modelo nulo:

Step	AIC
	260.42
+ `EI(rez.)`	54.41
+ `FPMT(rez.)`	-33.22
+ `HP(rez.)`	-37.26
+ `Desempleo(rez.)`	-43.63
+ `HIPC(rez.)`	-46.59

Se observa que el modelo final tiene $AIC = -46.59$ y que la variable de mayor relevancia para modelar la inflación es EI(rez.).

Veamos un gráfico que compara los valores reales de inflación y los valores ajustados por el modelo:

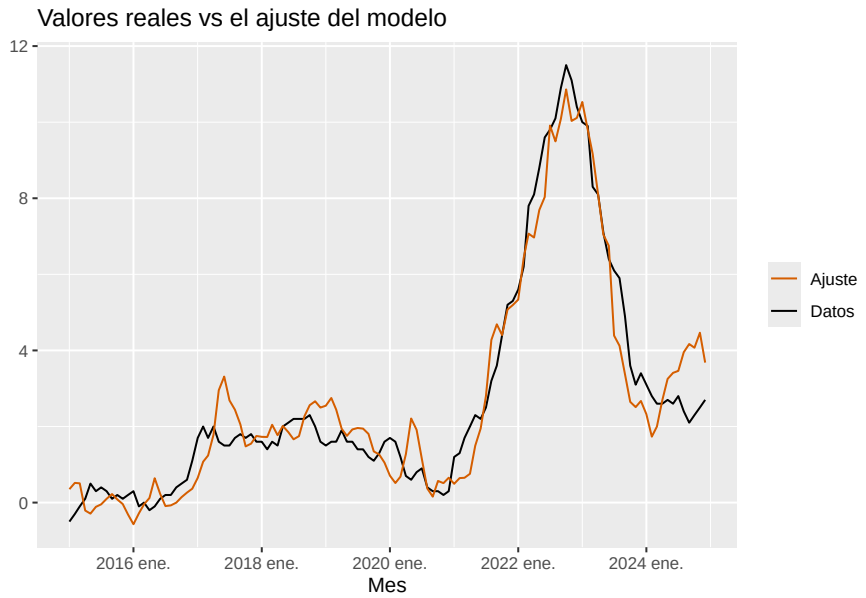

Figura 2.14: Ajuste del modelo lineal múltiple (TSLM) vs. datos reales.

El modelo ajusta con un $R^2 = 0.928765$ sobre los datos muestrales. A continuación, se comprueban las hipótesis del modelo:

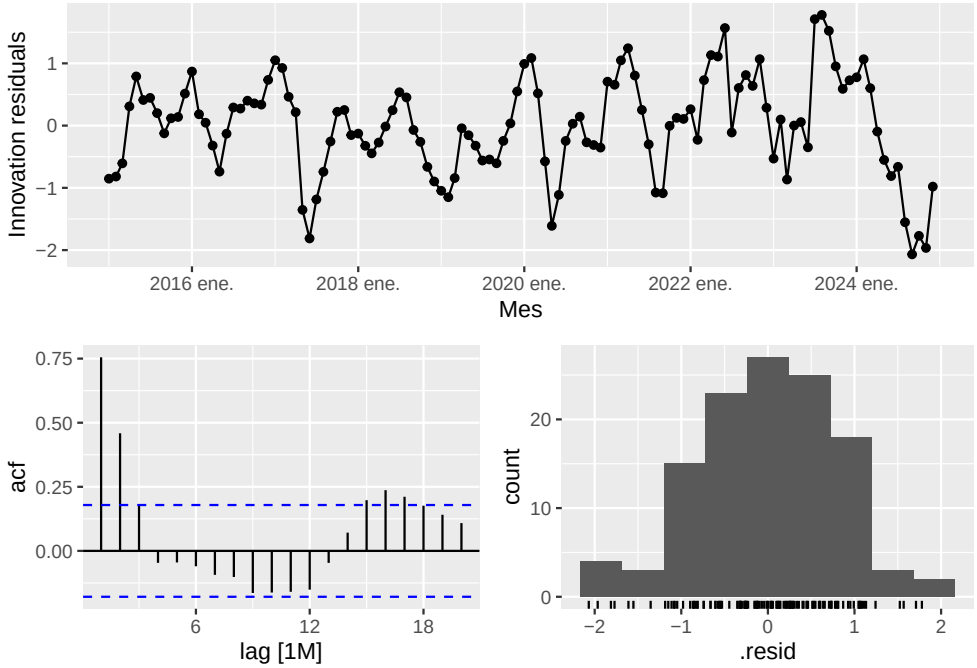

Figura 2.15: Diagnóstico de residuos del modelo TSLM.

```
##
##  Shapiro-Wilk normality test
##
## data:  pull(augment(m_ts), .innov)
## W = 0.9899, p-value = 0.5258
```

.model	lb_stat	lb_pvalue
TSLM(Inflacion ~ `FPMT(rez.)` + `Desempleo(rez.)` + `EI(rez.)` + `HIPC(rez.)` + `HP(rez.)`)	111	0

Por un lado, se acepta la normalidad y el ajuste R^2 del modelo es alto. No obstante, se puede apreciar que se incumplen las hipótesis de no autocorrelación en los residuos y homocedasticidad. Recordemos que las variables tienen un comportamiento no lineal entre ellas, luego, aunque el modelo lineal se ajuste bien a los datos que ya conoce, es posible que las predicciones ante nuevos datos presenten un alto grado de incertidumbre.

2.2.2. Modelos lineales dinámicos

Primero de todo, debemos ver si las series son estacionarias. Aplicaremos a los datos el contraste de estacionariedad.

	HIPC(rez.)	FPMT(rez.)	Desempleo(rez.)	Inflacion	EI(rez.)	HP(rez.)
p.value	0.97	0.98	0.88	0.87	0.73	0.78

Se confirma lo que dedujimos en las representaciones gráficas, ninguna de las series es estacionaria, veamos si con un proceso de diferenciación de primer orden en cada variable basta para que sean no estacionarias.

	HIPC(rez.)	FPMT(rez.)	Desempleo(rez.)	Inflacion	EI(rez.)	HP(rez.)
p.value	0.01	0.01	0.01	0.01	0.01	0.01

El contraste verifica que tras el proceso de integración de orden uno todas las series son estacionarias. Es importante tener estos resultados en cuenta a la hora de ajustar los parámetros del modelo dinámico.

Visualización de las series después de aplicar diferenciación:

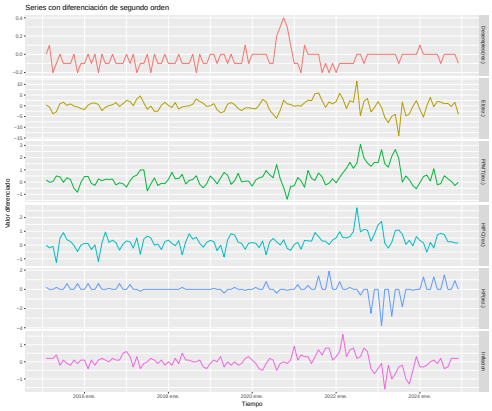

Figura 2.16: Series tras el proceso de integración

A continuación, se plantea el modelo dinámico con todas las variables explicativas. Por defecto R escoge los parámetros p, d, q según los criterios de información. No obstante, hemos observado que las series no son estacionarias, luego podemos forzar que d tome valor igual a uno.

Nota: El criterio de selección de R para los parámetros p, d, q es similar a los procesos paso a paso usados en los modelos lineales. Se plantean los modelos con diferentes parámetros y se escogen los modelos con menor AIC sobre la muestra. No obstante, en ocasiones el modelo que mejor ajuste tenga sobre la muestra no es el que mejor funciona ante nuevos datos, es importante también que se verifiquen las hipótesis del modelo.

Por ejemplo, en el caso que nos ocupa, R toma en el proceso de selección de parámetros $d = 0$. Sin embargo, las series son no estacionarias por lo cual esta elección no es adecuada si queremos tener mejores predicciones.

Es por esto, que es importante analizar la estacionariedad de las series que intervienen en el modelo y ajustar manualmente el parámetro d según la necesidad de tus datos.

Se plantea el modelo imponiendo $d = 1$:

.model	term	estimate	std.error	statistic	p.value
Dinámico	ar1	0.844	0.091	9.23	1.2e-15
Dinámico	ma1	-0.579	0.138	-4.20	5.2e-05
Dinámico	`FPMT(rez.)`	-0.071	0.057	-1.25	2.1e-01
Dinámico	`Desempleo(rez.)`	-0.996	0.355	-2.80	5.9e-03
Dinámico	`EI(rez.)`	0.019	0.013	1.49	1.4e-01
Dinámico	`HIPC(rez.)`	0.038	0.067	0.58	5.7e-01
Dinámico	`HP(rez.)`	0.013	0.051	0.26	8.0e-01

Comprobamos las hipótesis:

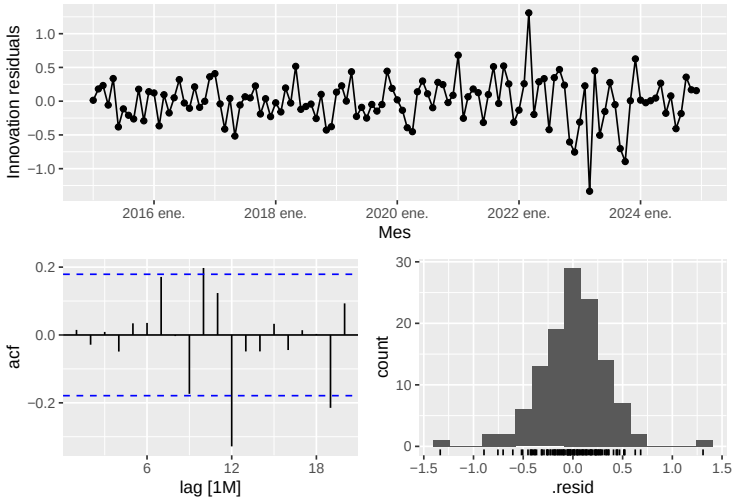

Figura 2.17: Diagnóstico de residuos del modelo lineal dinámico.

```
##
##  Shapiro-Wilk normality test
##
## data:  pull(augment(fit7), .innov)
## W = 0.96358, p-value = 0.00248
```

.model	lb_stat	lb_pvalue
ARIMA(Inflacion ~ `FPMT(rez.)` + `Desempleo(rez.)` + `EI(rez.)` + `HIPC(rez.)` + `HP(rez.)` + pdq(d = 1) + PDQ(P = 0, D = 0, Q = 0))	13.71	0.1866

Se observa como este modelo sí que verifica las hipótesis que se imponen sobre los residuos. Por otra parte, los residuos no presentan normalidad, esto se traduce en unos intervalos de predicción poco fiables.

Veamos en este caso, el ajuste del modelo dinámico en contraposición con los valores reales de inflación.

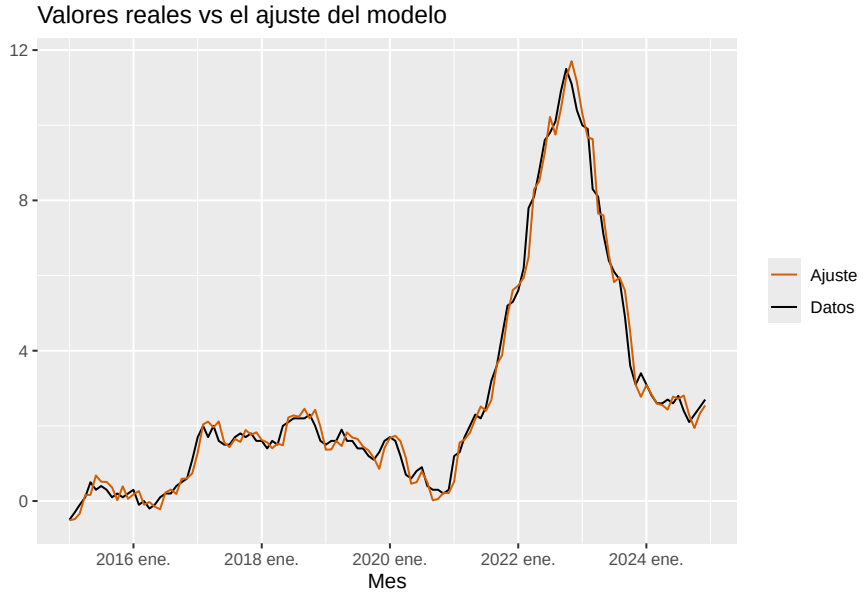

Figura 2.18: Ajuste del modelo lineal dinámico vs. datos reales.

2.2.3. Regularización, Bosques aleatorios, Gradient Boosting y SVR

En esta sección, se plantean los diferentes modelos de aprendizaje automático vistos en el Capítulo 1.

Para la regularización de los coeficientes del modelo lineal múltiple, se aplican las técnicas Elastic Net, con $\alpha = 0$ (Regularización Ridge), $\alpha = 1$ (Regularización Lasso), y con $alpha = 0.5$, eligiendo los respectivos λ mediante validación cruzada.

Además, implementaremos Bosques aleatorios y Gradient Boosting con 100 árboles de decisión y SVR con parámetros $cost = 2$ y $\epsilon = 0,1$.

El objetivo será ver qué modelos leen mejor las relaciones complejas entre los datos y se ajustan mejor a los datos reales de inflación.

Técnicas de regularización

Cada técnica de regularización se representa mediante dos gráficos complementarios: uno que contrasta las predicciones con los valores reales y otro que muestra la distribución de los residuos en función de los valores predichos.

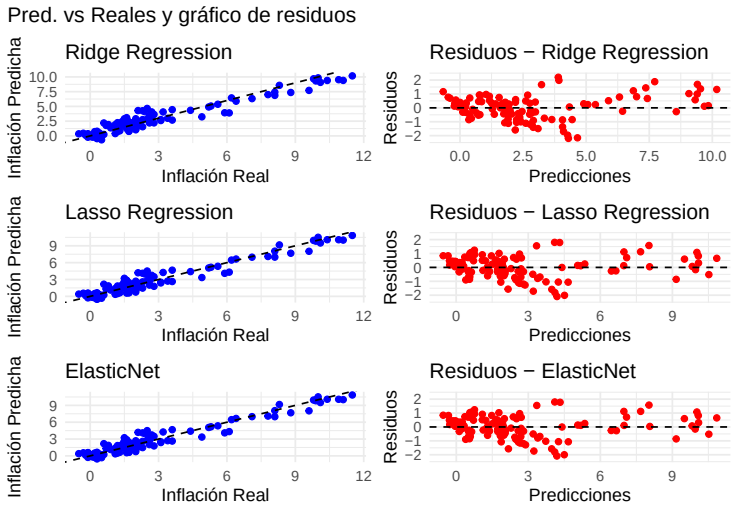

Figura 2.19: Gráfico de valores predichos versus reales y residuos de los modelos con técnicas de regularización.

En los gráficos de predicción versus valores reales, la diagonal representa la predicción perfecta. Se observa que, aunque los tres modelos siguen la tendencia general de los datos, existe una dispersión considerable alrededor de esta línea ideal.

En definitiva, a pesar de las ventajas de la regularización para controlar la complejidad y evitar el sobreajuste, no conseguimos tener buen ajuste del modelo, pues se sigue heredando el problema de los modelos lineales debido a las relaciones no lineales entre variables.

A continuación, otra perspectiva temporal del ajuste de los modelos lineales en comparación con los datos reales de inflación durante el período analizado:

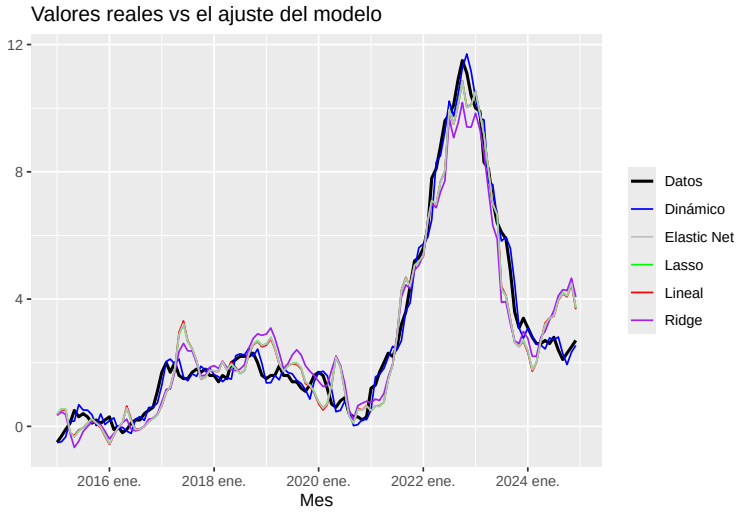

Figura 2.20: Ajuste de los modelos lineales versus los datos reales.

Se han incluido, además de las técnicas de regularización, el modelo lineal y el modelo lineal dinámico.

Apreciemos el buen comportamiento del modelo lineal dinámico frente a los demás. Esta diferencia en el rendimiento se explica por la incorporación de componentes autorregresivos en el modelo Dinámico, que le permiten capturar la dependencia temporal.

Bosques aleatorios y Gradient Boosting

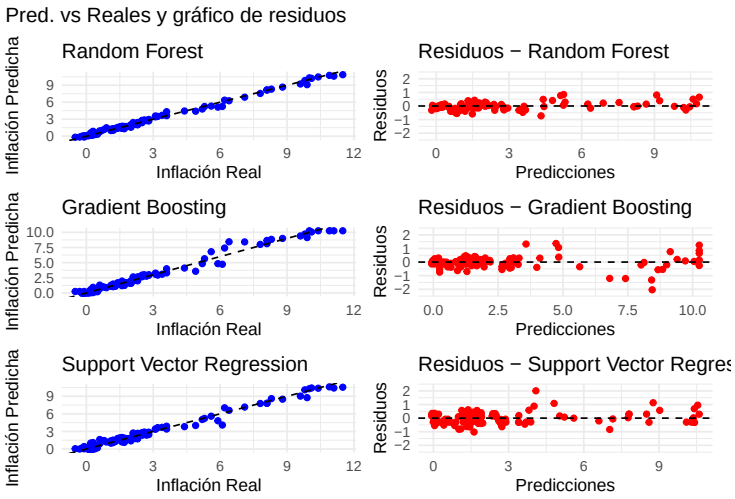

Figura 2.21: Gráfico de valores predichos versus reales y residuos de los modelos con técnicas avanzadas.

En este caso, los modelos muestran una notable mejora en la capacidad predictiva respecto a los modelos lineales regularizados analizados anteriormente. La dispersión de puntos alrededor de la línea diagonal es significativamente menor, de hecho, la alineación de los puntos con la diagonal es notablemente precisa a lo largo de todo el rango de valores de inflación.

En general, el gráfico de residuos muestra una distribución más compacta y centrada en el cero y con menor variabilidad, sobre todo en Bosques aleatorios.

La superioridad de estos modelos avanzados frente a las técnicas de regularización lineal se explica por su capacidad para capturar relaciones no lineales y efectos de interacción complejos entre las variables explicativas.

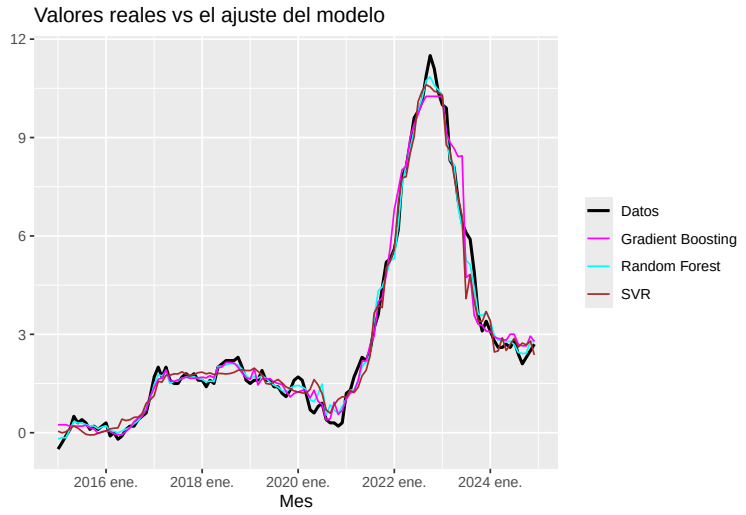

Figura 2.22: Ajuste de los modelos avanzados versus los datos reales.

De nuevo, el gráfico revela un rendimiento notablemente superior de los modelos avanzados en comparación con los modelos lineales. Los tres modelos logran seguir con mayor precisión las fluctuaciones de la inflación, incluyendo el periodo del Covid.

Además de los gráficos, podemos calcular la cantidad de información explicada por cada modelo mediante la comparación de sus R^2:

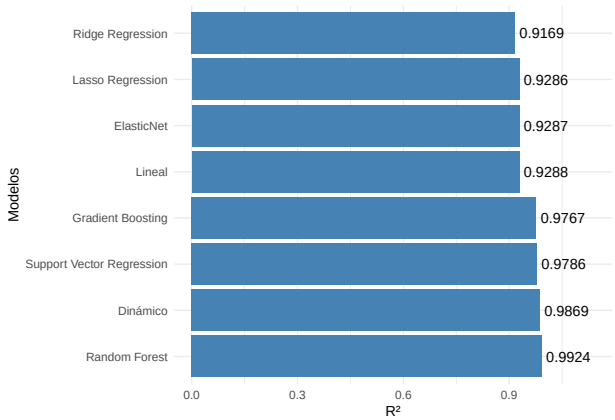

Figura 2.23: Comparativa de la variabilidad explicada por cada uno de los modelos

En síntesis, los modelos avanzados de aprendizaje automático, junto con los modelos dinámicos emergen como herramientas eficaces para la predicción de la inflación debido a que pueden adaptarse a la complejidad de las relaciones entre las variables económicas.

Importancia de las variables

El método de Bosques aleatorios obtuvo uno de los mejores resultados de R^2 entre los modelos planteados, alcanzando alrededor del 99 % de variabilidad explicada. Una ventaja de este método es su gran interpretabilidad, al fin y al cabo su algoritmo se basa en construir un número finito de árboles y calcular su predicción como la media de las predicciones de cada árbol. A continuación, con el fin de entender cómo se constituye el método de Bosques aleatorios, visualizaremos el árbol de regresión construido con los datos que nos ocupan.

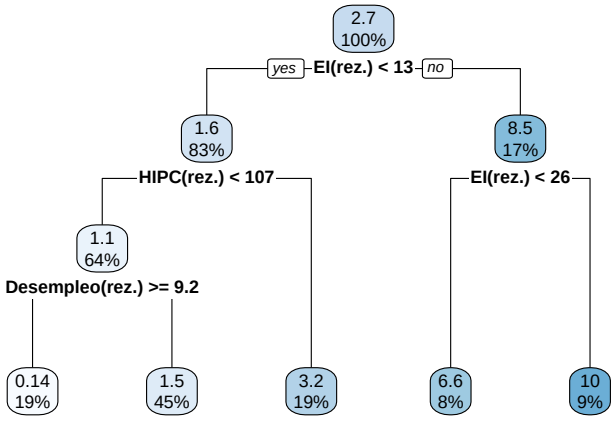

Figura 2.24: Árbol de regresión como ejemplo para entender Bosques aleatorios

Se observa como el árbol actúa como una clasificación de los registros de nuestra base de datos, de manera que al introducir un nuevo registro se posiciona en uno de los nodos, atribuyéndole así el valor de inflación más acorde al valor de sus variables explicativas. Es por esto que la técnica de Bosques aleatorios no solo constituye un eficiente método de regresión sino que también desvela la relación y el contexto de las variables explicativas con respecto la inflación. Como se mencionó en el Capítulo 1, esta relación se puede cuantificar gracias a que durante el algoritmo de construcción del modelo se produce un conjunto de elementos fuera de la bolsa, los cuáles son útiles para producir estimaciones de la importancia de las variables. Esto son los resultados que se obtienen:

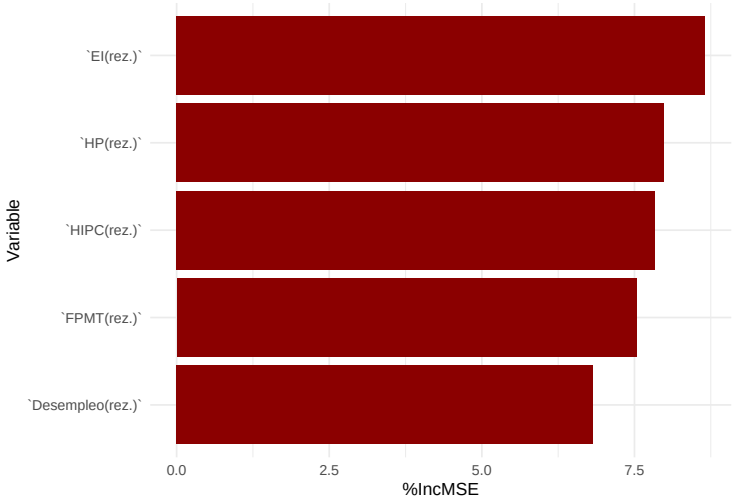

Figura 2.25: Importancia de las variables

Estadísticamente, la variable más importante es el índice de variación porcentual de la energía y esto concuerda perfectamente con la realidad económica, pues tal y como recoge Rodríguez, D. (2022), los precios energéticos impactan directamente en la cesta de consumo y en la estructura de costes empresariales, siendo el factor principal del reciente proceso inflacionario post-pandemia y crisis de Ucrania.

En definitiva, el método de Bosques aleatorios, además de producir un gran ajuste de la inflación, establece con rigor estadístico la importancia las variables, resultando ser estimaciones fiables de la influencia de dichas variables en la inflación desde el punto de vista económico.

2.3. Validación y comparación de modelos.

El objetivo fundamental de esta sección es determinar qué enfoque metodológico proporciona estimaciones más precisas y robustas de la inflación futura.

Anteriormente, hemos visualizado el ajuste de cada uno de los modelo con respecto los mismos datos que hemos usado para definirlos. Esto significa que no hemos comprobado todavía cómo funcionan las predicciones ante nuevos conjuntos de datos. La manera habitual de resolver este problema es dividir nuestra muestra en dos conjuntos de datos diferentes. El primero de ellos el conjunto de "entrenamiento" donde se definen los modelos, y el segundo, el conjunto de "prueba" sobre el cual se hacen las predicciones.

Dividiremos los datos en una proporción aproximada de 5/6 para el "entrenamiento", 1/6 para la "prueba". Al tratarse de series temporales, no podemos hacer una división aleatoria de los datos pues romperíamos la dependencia temporal y puede llevar a una evaluación demasiado optimista del rendimiento del modelo, ya que el modelo podría "ver" datos futuros durante el entrenamiento. Por este motivo, nuestro conjunto de entrenamiento contendrá los tiempos $t = 1, ..., 108$ y el de prueba los 12 restantes. Es decir, haremos la predicción de la inflación del año 2024.

En la siguiente gráfica se pondrán en contraposición las predicciones versus la inflación real a partir del año 2024, es decir en los datos de test. Recordemos que en este caso el modelo realiza las predicciones sobre datos nuevos, luego los resultados serán clave para determinar el rendimiento de los modelos.

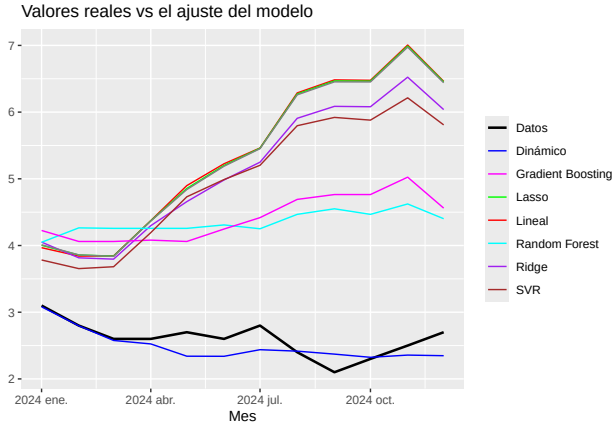

Figura 2.26: Predicciones de los modelos lineales versus los datos reales.

Destaca por su rendimiento el modelo dinámico, manteniendo sus predicciones cercanas a los valores reales observados. Por otra parte, los modelos basados en árboles no alcanzan la capacidad predictiva que se podía esperar de ellos teniendo en cuenta el buen ajuste que daban. Por último, los modelos lineales muestran un escaso poder predictivo, cosa que ya habíamos previsto en apartados anteriores.

El criterio de evaluación empleado para la comparación de los modelos será el error cuadrático medio (MSE):

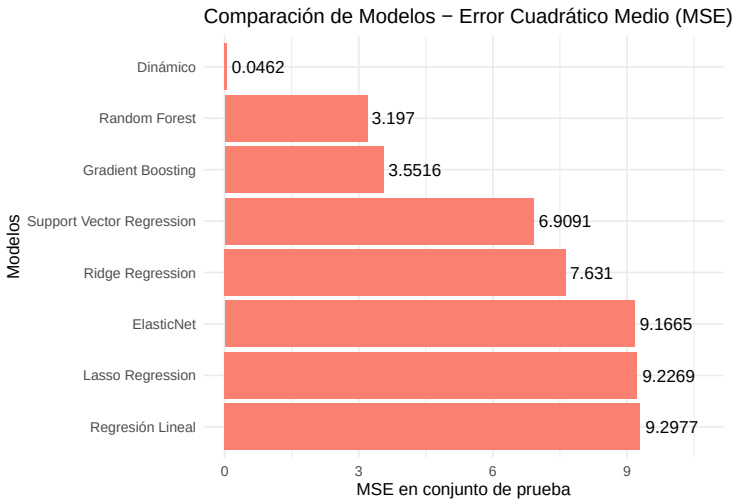

Figura 2.27: Comparación de MSE.

El MSE refuerza las ideas extraídas del gráfico, el modelo dinámico exhibe un rendimiento excepcional, con valores bajos de MSE. Les siguen los modelos Bosques aleatorios, Gradient Boosting y SVR y por último los modelos lineales.

2.4. Predicción de la inflación

En los apartados anteriores, se ha realizado un análisis del contexto económico en el que nos encontramos, se han planteado los modelos, comprobado sus hipótesis y visualizado su ajuste sobre los datos reales. También, se han establecido, mediante procesos estadísticos, las variables más influyentes sobre el comportamiento de la inflación y se ha sometido a los distintos modelos a un proceso de validación con el fin de obtener las técnicas más adecuadas para su predicción.

El objetivo de esta sección es realizar un pronóstico real de la inflación. Se usará el modelo dinámico, por ser el que mejor resultados obtuvo en la validación. Por la naturaleza de nuestros datos se realizará el estudio con un horizonte temporal de 4 meses. Como se pretende prever la inflación en el primer cuatrimestre de 2025, necesitamos los datos de las variables explicativas del último cuatrimestre de 2024. A continuación se exponen dichos datos:

FPMT(rez.)	HIPC(rez.)	Desempleo(rez.)	EI(rez.)	HP(rez.)
143.9	130.8	5.9	-4.7	3.9
144.9	131.2	5.8	-3.3	4.9
145.3	131.0	5.8	-1.2	4.9
145.2	131.4	5.8	0.7	4.9

Se procede ahora al planteamiento del modelo dinámico. Este proceso ya se realizó en su correspondiente apartado (2.2.2). Finalmente, el modelo realiza el siguiente pronóstico:

Mes	Predicción
2025 ene.	2.674
2025 feb.	2.801
2025 mar.	2.842
2025 abr.	2.935

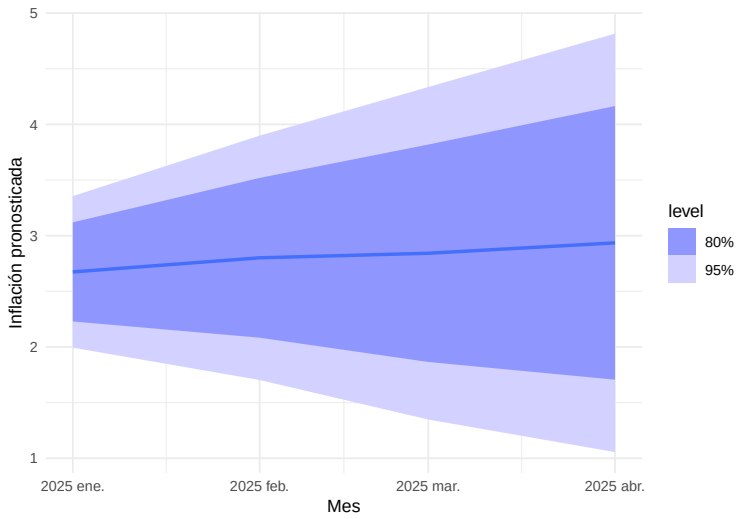

Figura 2.28: Pronóstico de la inflación para el primer cuatrimestre de 2025

La situación económica de este estudio es volátil, por lo que es conveniente aclarar la incertidumbre de las predicciones. Las regiones sombreadas de azul representan el intervalo de predicción del modelo y la línea los valores de inflación pronosticados. A partir del primer mes de predicción, el modelo usa su propio pronóstico sobre la inflación como dato para el siguiente mes. Por esto, se observa que cuanto mayor sea nuestro horizonte temporal, más amplios son los intervalos de predicción y mayor es la incertidumbre del pronóstico.

Conclusión

En este trabajo se ha realizado un estudio de la situación económica actual, analizando datos de indicadores claves en la economía y poniendo en contexto las variables más influyentes en su desarrollo. El objetivo principal ha sido describir y plantear las técnicas estadísticas más adecuadas para el análisis de los datos económicos.

Es fundamental reconocer que cada enfoque metodológico puede resultar útil dependiendo del contexto específico de aplicación. Los modelos lineales ofrecen ventajas en términos de interpretabilidad y explicabilidad. Por otro lado, los modelos avanzados como Random Forest o SVR destacan en la captura de relaciones no lineales complejas, resultando más adecuados para periodos de inestabilidad económica. El modelo lineal dinámico representa un equilibrio entre complejidad e interpretabilidad, con excelente rendimiento en series temporales.

También, hay que tener en cuenta el horizonte temporal del pronóstico, que dependerá del contexto económico en el que nos situemos. En un periodo de estabilidad económica, es viable realizar predicciones a largo plazo, mientras que en situaciones económicamente inestables, como la de este estudio, las condiciones se vuelven imprevisibles, por lo que es necesario reducir el horizonte temporal de la predicción con el objetivo de obtener un análisis fiable.

En definitiva, la práctica óptima del análisis de datos económicos conlleva realizar estudios exhaustivos del contexto económico, comparar y validar el rendimiento de los diferentes modelos, establecer un marco de predicción conveniente y declarar la incertidumbre del pronóstico.

Apéndice A

Código R del Capítulo 1

Bloque: tabla_pander

Bloque: cap1-fig1

```
library(forecast)
#Generamos los vectores auxiliares
aux <- numeric(200) #generamos un vector
#de 0 con 500 componentes
w <- rnorm(200) .
for (t in 2:200) {
  aux[t]<-0.5*aux[t-1]+w[t]
}
#Construimos la serie temporal y la visualizamos
aux_st <- ts(aux,start=1)
autoplot(aux_st)
```

Bloque: cap1-calc1

```
library(kableExtra)
modInfl<-arima(aux_st,order = c(1,0,0))
# Extraer coeficientes y estadísticas del modelo
coefs <- data.frame(
  Model=c("ARIMA(1,0,0)"),
  Parameter = names(modInfl$coef),
  stimation = modInfl$coef,
  std.error = sqrt(diag(modInfl$var.coef))
)
AR<-modInfl
tabla_pander(coefs,
             decimales = 4)
phi_1 <- (AR$coef)[1]
ter.ind<-AR$coef[2]
```

Bloque: cap1-fig2

```r
library(forecast)
aux <- numeric(200) #generamos un vector de 0 con 500 componentes
w <- rnorm(200)
#Forzamos que en la generación de datos la
#relación con el tiempo inmediatamente anterior.
for (t in 2:200) {
  aux[t]<-aux[t-1]+w[t]
}
#Construimos la serie temporal y la visualizamos
aux_st <- ts(aux,start=1)
autoplot(aux_st)
```

Bloque: cap1-fig3

```r
autoplot(diff(aux_st))
```

Bloque: cap1-fig4

```r
library(ggplot2)
aux2 <- numeric(200) #generamos un vector de 0 con 500 componentes
w <- rnorm(200)
#Forzamos que en la generación de datos la
#relación con el tiempo inmediatamente anterior.
for (t in 2:200) {
  aux2[t]<-aux2[t-1]+w[t]
}

aux3 <- numeric(200) #generamos un vector de 0 con 500 componentes
w <- rnorm(200)
for (t in 2:200) {
  aux3[t]<-aux3[t-1]+w[t]
}
RW <- ts(cbind(aux3,aux2,aux))
autoplot(RW) +scale_color_manual(
    name = "Random Walks",  # Título de la leyenda
    values = c("red", "blue", "green"),  # Colores de las series
    labels = c("RW1", "RW2", "RW3")  # Nuevos nombres de la leyenda
  )
```

Bloque: cap1-fig5

```r
x<-numeric(200)
w<-rnorm(200)
for (t in 2:200){
  x[t]<-w[t]+0.3*w[t-1]
}
x_st=ts(x)
autoplot(x_st)
```

Bloque: cap1-fig6

```
e1<-arima.sim(n=200,list(order=c(1,0,1),ar=c(0.6),ma=c(0.4)))
autoplot(e1)
```

Bloque: cap1-fig7

```
library(forecast)
ggAcf(e1)
```

Bloque: cap1-calc2

```
mode1 <- Arima(e1,order = c(1,0,1))
modInfl<-mode1

coefs <- data.frame(
  Model=c("ARIMA(1,0,1)"),
  Parameter = names(modInfl$coef),
  stimation = modInfl$coef,
  std.error = sqrt(diag(modInfl$var.coef))
)
mode1<-modInfl

tabla_pander(coefs,
             decimales = 4)
```

Bloque: cap1-fig8

```
checkresiduals(mode1)
```

Bloque: cap1-calc3

```
shapiro.test(residuals(mode1))
```

Bloque: fotos1

```
knitr::include_graphics(c("C:/Users/rtrin/OneDrive/Imágenes/
Capturas de pantalla/Captura de pantalla 2025-05-29 184951.png",
"C:/Users/rtrin/OneDrive/Imágenes/
Capturas de pantalla/
Captura de pantalla 2025-05-13 184230.png"))
```

Bloque: fotos2

```
knitr::include_graphics("C:/Users/rtrin/OneDrive/Imágenes/

Capturas de pantalla/Captura de pantalla 2025-05-29 191338.png")
```

Apéndice B

Código R del Capítulo 2

Bloque: setup-ch2

```
source("cabecera_chunk_inicio.R")
```

Bloque: load-libs-ch2

```
library(forecast)
library(tseries)
library(kableExtra)
library(fpp3)
library(GGally)
library(purrr)
library(broom)
library(readxl)       # Para leer archivos Excel
library(dplyr)        # Para manipulación de datos
library(caret)        # Para división de datos y evaluación de modelos
library(randomForest) # Para modelo Random Forest
library(gbm)          # Para Gradient Boosting
library(e1071)        # Para Support Vector Regression
library(glmnet)       # Para Ridge, Lasso y ElasticNet
library(ggplot2)      # Para visualización
library(patchwork)
library(h2o)
library(rpart)
library(rpart.plot)
```

Bloque: tabla_pander2

```
#Función auxiliar para las tablas
tabla_pander <- function(datos, titulo = NULL, decimales = 4) {
  library(pander)

  # Configurar opciones de pander
  panderOptions('table.split.table', Inf)
  panderOptions('table.alignment.default', 'center')
```

```
  panderOptions('table.alignment.rownames', 'left')
  panderOptions('keep.trailing.zeros', TRUE)
  panderOptions('digits', decimales)

  # Crear la tabla con pander
  if (!is.null(titulo)) {
    cat(paste0("**", titulo, "**\n\n"))
  }

  # Convertir a formato markdown
  pander(datos)
}
```

Bloque: load-data-plot-ch2

```
ROC<- read.csv("C:/Users/rtrin/Downloads/
estat_prc_hicp_manr_filtered_en (1).csv.gz")
Inflacion <- ROC[,"OBS_VALUE"]
TSInflat<-ts(data = Inflacion,start = 1997,frequency = 12)
autoplot(TSInflat)
```

Bloque: acf-plot-ch2

```
ggAcf(TSInflat)
```

Bloque: adf-test-ch2

```
test_adf<-adf.test(TSInflat)
tidy_test_adf<-tidy(adf.test(TSInflat))
tabla_pander(tidy_test_adf,
             decimales = 2)
```

Bloque: autoarima-ch2

```
modInfl<-auto.arima(TSInflat,D=0,max.P = 0,max.Q = 0)
# Extraer coeficientes y estadísticas del modelo
coefs <- data.frame(
  Model=c("ARIMA(1,1,1)","ARIMA(1,1,1)"),
  Parameter = names(modInfl$coef),
  estimate = modInfl$coef,
  std.error = sqrt(diag(modInfl$var.coef))
)
info_modelo<- data.frame(Loglik=modInfl$loglik,
  AIC=modInfl$aic,
  BIC=modInfl$bic,
  Sigma2=modInfl$sigma2)
```

```r
# Mostrar tabla
tabla_pander(coefs,
             decimales = 2)
```

Bloque: roots

```r
# Acceder a las raíces (si deseas hacerlo manualmente)
# Los coeficientes AR se encuentran en model_arima$coef
ar_coeffs_model <- modInfl$coef[grepl("ar", names(modInfl$coef))]

# Crea el polinomio para las raíces AR
# Necesitas invertir el signo y añadir el 1 para el término constante
ar_poly_coeffs <- c(1, -ar_coeffs_model)
ar_roots <- polyroot(ar_poly_coeffs)
plot(ar_roots, xlim = c(-1.5, 1.5), ylim = c(-1.5, 1.5),
     xlab = "Parte Real", ylab = "Parte Imaginaria",
     main = "Raíces del Polinomio Característico AR",
     asp=1)
# Dibuja el círculo unitario
symbols(0, 0, circles = 1, inches = FALSE, add = TRUE, fg = "red")
```

Bloque: perron_test

```r
tidy_test_pp<-tidy(pp.test(TSInflat))
tabla_pander(tidy_test_pp,
             decimales = 4)
```

Bloque: checkres-tsinflat-ch2

```r
checkresiduals(TSInflat)
```

Bloque: forecast-plot-ch2

```r
modInfl %>% forecast(h = 6) %>%
autoplot()
```

Bloque: precovid-plots-ch2

```r
InflatPreCo <- window(TSInflat,end=c(2020,3))
autoplot(InflatPreCo)
ggAcf(InflatPreCo)
```

Bloque: adf-test-precovid-ch2

```r
test_adf_PreCovid<-adf.test(InflatPreCo)
tidy_test_adf_PreCovid<-tidy(test_adf_PreCovid)
tabla_pander(tidy_test_adf_PreCovid,
             decimales = 2)
tidy_test_pp_preC<-tidy(pp.test(InflatPreCo))
tabla_pander(tidy_test_pp_preC,
             decimales = 4)
```

Bloque: diff-plots-ch2

```
InflatDIFF<-diff(InflatPreCo, differences = 1)
ggAcf(InflatDIFF)

tabla_pander(tidy(pp.test(InflatDIFF)),
             decimales = 2)
```

Bloque: autoarima-precovid-ch2

```
modInfl2<-auto.arima(InflatPreCo,D=0,max.P = 0,max.Q = 0)
# Extraer coeficientes y estadísticas del modelo
coefs <- data.frame(
  Model=c("ARIMA(1,1,0)"),
  Parameter = names(modInfl2$coef),
  stimation = modInfl2$coef,
  std.error = sqrt(diag(modInfl2$var.coef))
)
info_modelo<- data.frame(Loglik=modInfl2$loglik,
  AIC=modInfl2$aic,
  BIC=modInfl2$bic,
  Sigma2=modInfl2$sigma2)


tabla_pander(coefs,
             decimales = 2)
```

Bloque: checkres-modinfl-ch2

```
checkresiduals(modInfl)
```

Bloque: forecast-plot-precovid-ch2

```
modInfl2 %>% forecast(h = 6) %>%
autoplot()
```

Bloque: load-excel-ch2

```
datosEcon <- read_excel("C:/Users/rtrin/OneDrive/
                        Escritorio/TFG/datosEcon.xlsx",
    col_names = FALSE,n_max = 120)
colnames(datosEcon)<-c("FPMT(rez.)","HIPC(rez.)","Desempleo",
                       "Inflacion","EI(rez.)","HP")
```

Bloque: create-tsibble-ch2

```r
año_inicio <- 2015
mes_inicio <- 1

# Crear una secuencia de fechas mensuales
fechas_mensuales <- seq(from = as.Date(paste(año_inicio, mes_inicio,
                                             "01", sep = "-")),
                        by = "month",
                        length.out = 120)

# Extraer el año y el mes y
#combinarlos en un formato adecuado
indice_temporal <- format(fechas_mensuales, "%Y-%m")

#Convertimos los datos en
#una tabla de datos temporales

datosEcon1<-datosEcon
datosEcon1$Mes<-indice_temporal

datosEcon1 |>
  mutate(Mes=yearmonth(Mes))|>
  as_tsibble(index = Mes) ->datosEcon1
tabla_pander(head(datosEcon1),
             decimales = 4)
```

Bloque: indicadores-plot-ch2

```r
datosEcon1 |>
 pivot_longer(-Mes) |>
  ggplot(aes(Mes,value,colour = name))+
  geom_line()+
  facet_grid(name ~ ., scales = "free_y") +
  guides(colour = "none")
```

Bloque: pairs-corr-plots-ch2

```r
library(GGally)
datosEcon1 |>
  GGally::ggpairs(columns = 1:6)
datosEcon1 |>
  GGally::ggcorr(geom = "circle")
```

Bloque: tslm-fitting-ch2

```r
rmult<-lm(Inflacion~.,datosEcon)
nulo<-lm(Inflacion~1,datosEcon)

s<-step(nulo,direction = "forward",
     scope=list(lower=nulo, upper=rmult),trace=0)
s$call
```

Bloque: tidy_s

```
tabla_pander(tidy(s),
             decimales = 4)
```

Bloque: AIC_tidy

```
tabla_pander(s$anova[,c(1,6)],
             decimales = 4)
```

Bloque: tslm-fit-plot-ch2

```
datosEcon1|>
  model(TSLM(Inflacion~`FPMT(rez.)`+Desempleo+`EI(rez.)`
+`HIPC(rez.)`+`HP`))->m_ts
augment(m_ts)|>
  pull(.fitted)->prediccionLineal

r_2<-glance(m_ts)|>
  pull(r_squared)

augment(m_ts) |> #augment crea la columna .fitted
  ggplot(aes(x = Mes)) +
  geom_line(aes(y = Inflacion, colour = "Datos")) +
  geom_line(aes(y = .fitted, colour = "Ajuste")) +
  labs(y = NULL,
    title = "Valores reales vs el ajuste del modelo"
  ) +
  scale_colour_manual(values=c(Datos="black",Ajuste="#D55E00")) +
  guides(colour = guide_legend(title = NULL))
```

Bloque: tslm-residuals-ch2

```
gg_tsresiduals(m_ts)
augment(m_ts)|>
  pull(.innov)|>
shapiro.test()
augment(m_ts) |>
  features(.innov,ljung_box,lag=10)|>
  tabla_pander(decimales = 4)
```

Bloque: dynlm-ljungbox1-ch2

```
x<-datosEcon1[1:6]
z<-apply(x,2,pp.test)
f1<-z$`FPMT(rez.)`$p.value
f2<-z$`HIPC(rez.)`$p.value
f3<-z$Desempleo$p.value
f4<-z$Inflacion$p.value
```

```
f5<-z$`EI(rez.)`$p.value
f7<-z$HP$p.value
z<-c(f1,f2,f3,f4,f5,f7)
T1<-matrix(z,1,6,byrow = TRUE)
rownames(T1)<-"p.value"
colnames(T1)<-(c("HIPC(rez.)","FPMT(rez.)","Desempleo",
"Inflacion","EI(rez.)","HP"))

tabla_pander(T1,
             decimales = 2)
```

Bloque: dynlm-adf-ch2

```
x_diff<-apply(x,2,function(x) diff(x,differences=1))
z<-apply(x_diff,2,pp.test)
f1<-z$`FPMT(rez.)`$p.value
f2<-z$`HIPC(rez.)`$p.value
f3<-z$Desempleo$p.value
f4<-z$Inflacion$p.value
f5<-z$`EI(rez.)`$p.value
f7<-z$HP$p.value
z<-c(f1,f2,f3,f4,f5,f7)
T1<-matrix(z,1,6,byrow = TRUE)
rownames(T1)<-"p.value"
colnames(T1)<-(c("HIPC(rez.)","FPMT(rez.)","Desempleo",
"Inflacion","EI(rez.)","HP"))

tabla_pander(T1,
             decimales = 2)
```

**Bloque:

```
# Crear un dataframe con las series diferenciadas de segundo orden
diff_df <- as.data.frame(x_diff)
diff_df$Mes <- yearmonth(datosEcon1$Mes[-c(1)])

# Convertir a tsibble
diff_tsibble <- as_tsibble(diff_df,index = Mes)

# Visualizar series diferenciadas de segundo orden
diff_tsibble |>
  pivot_longer(-Mes) |>
  ggplot(aes(Mes, value, colour = name)) +
  geom_line() +
  facet_grid(name ~ ., scales = "free_y") +
  guides(colour = "none") +
  labs(title = "Series con diferenciación de segundo orden",
       x = "Tiempo",
       y = "Valor diferenciado")
```

Bloque: dynlm-fitting-ch2

```
fit7 <- datosEcon1 |>
  model(ARIMA(Inflacion ~ `FPMT(rez.)`+Desempleo+`EI(rez.)`+`HIPC(rez.)`
+`HP`+pdq(d=1)+PDQ(P = 0, D = 0, Q = 0)))
```

Bloque: dynlm-report-ch2

```
modelo_tidy_fit <- tidy(fit7,conf.int=TRUE)
modelo_tidy_fit$.model<-c("Modelo Dinámico","Modelo Dinámico",
"Modelo Dinámico","Modelo Dinámico",
"Modelo Dinámico","Modelo Dinámico","Modelo Dinámico")
tabla_pander(modelo_tidy_fit,
             decimales = 2)
```

Bloque: dynlm-residuals-ch2

```
fit7 |> gg_tsresiduals()
augment(fit7)|>
  pull(.innov)|>
shapiro.test()
augment(fit7) |>
  features(.innov,ljung_box,lag=10)|>
  tabla_pander(decimales = 4)
```

Bloque: dynlm-fit-plot-ch2

```
augment(fit7)|>
  pull(.fitted)->prediccionDin
augment(fit7) |> #augment crea la columna .fitted
  ggplot(aes(x = Mes)) +
  geom_line(aes(y = Inflacion, colour = "Datos")) +
  geom_line(aes(y = .fitted, colour = "Ajuste")) +
  labs(y = NULL,
    title = "Valores reales vs el ajuste del modelo"
  ) +
  scale_colour_manual(values=c(Datos="black",Ajuste="#D55E00")) +
  guides(colour = guide_legend(title = NULL))
```

Bloque: prep-adv-models-ch2

```
X<-model.matrix(Inflacion~.,datos)[,-1]
y<-datos$Inflacion
```

Bloque: define-eval-func-ch2

```r
evaluar_modelo <- function(nombre, modelo, predicciones) {
  # Calcular métricas en conjunto de entrenamiento
  mse <- mean((y - predicciones)^2)
  mae <- mean(abs(y - predicciones))
  r2 <- 1 - sum((y - predicciones)^2) / sum((y - mean(y))^2)


  # Crear gráfico de predicciones vs valores reales
  df_plot <- data.frame(Real = y, Prediccion = predicciones)
  p1 <- ggplot(df_plot, aes(x = Real, y = Prediccion)) +
    geom_point(color = "blue") +
    geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
    labs(title = paste0( nombre),
         x = "Inflación Real",
         y = "Inflación Predicha") +
    theme_minimal()

  p3<-ggplot(df_plot,aes(x = 1:120)) +
    geom_line(aes(y = Real, colour = "Datos")) +
    geom_line(aes(y = Prediccion, colour = "Ajuste")) +
    labs(y = NULL,
         title = "Valores reales vs el ajuste del modelo"
    ) +
    scale_colour_manual(values=c(Datos="black",Ajuste="#D55E00")) +
    guides(colour = guide_legend(title = NULL))


  # Crear gráfico de residuos
  residuos <- y - predicciones
  df_residuos <- data.frame(Prediccion = predicciones, Residuo = residuos)
  p2 <- ggplot(df_residuos, aes(x = Prediccion, y = Residuo)) +
    geom_point(color = "red") +
    geom_hline(yintercept = 0, linetype = "dashed") +
    labs(title = paste0("Residuos - ", nombre),
         x = "Predicciones",
         y = "Residuos") +
    theme_minimal()


  # Devolver resultados
  return(list(
    nombre = nombre,
    modelo = modelo,
    mse = mse,
    mae = mae,
    r2 = r2,
    predicciones = predicciones,
    p1=p1,
```

```
    p3=p3,
    p2=p2
  ))
}
```

Bloque: run-adv-models-ch2

```
resultados <- list()
x_train_matrix <- as.matrix(X)

modelo_ridge <- cv.glmnet(x_train_matrix, y, alpha = 0)

pred_ridge <- predict(modelo_ridge, newx = x_train_matrix,
s = "lambda.min")[,1]
resultados[[2]] <- evaluar_modelo("Ridge Regression",
modelo_ridge, pred_ridge)

# 3. Lasso Regression

modelo_lasso <- cv.glmnet(x_train_matrix, y, alpha = 1)
pred_lasso <- predict(modelo_lasso, newx = x_train_matrix,
s = "lambda.min")[,1]
resultados[[3]] <- evaluar_modelo("Lasso Regression",
modelo_lasso, pred_lasso)

# 4. ElasticNet

modelo_elastic <- cv.glmnet(x_train_matrix, y, alpha = 0.5)
pred_elastic <- predict(modelo_elastic, newx = x_train_matrix,
s = "lambda.min")[,1]

resultados[[4]] <- evaluar_modelo("ElasticNet",
modelo_elastic, pred_elastic)

# 5. Random Forest

modelo_rf <- randomForest(x = X, y = y, ntree = 100,importance = TRUE)
pred_train_rf <- predict(modelo_rf, newdata = X)

resultados[[5]] <- evaluar_modelo("Random Forest", modelo_rf,
pred_train_rf)

# 6. Gradient Boosting

modelo_gbm <- gbm(Inflacion ~ `FPMT(rez.)`+Desempleo+`EI(rez.)`+
`HIPC(rez.)`+`HP`,
                  data = datosEcon,
                  distribution = "gaussian",
                  n.trees = 100,
```

```
                  interaction.depth = 3,
                  shrinkage = 0.1)
pred_train_gbm <- predict(modelo_gbm, newdata = datosEcon, n.trees = 100)
resultados[[6]] <- evaluar_modelo("Gradient Boosting", modelo_gbm,
pred_train_gbm)

# 7. Support Vector Regression

# Escalar datos para SVR
preproc <- preProcess(X, method = c("center", "scale"))
X_train_scaled <- predict(preproc, X)


modelo_svr <- svm(x = X_train_scaled, y = y, kernel = "radial", cost = 2,
epsilon = 0.1)
pred_train_svr <- predict(modelo_svr, newdata = X_train_scaled)

resultados[[7]] <- evaluar_modelo("Support Vector Regression",
modelo_svr, pred_train_svr)
```

Bloque: regularization-plots-ch2

```
p1 <- resultados[[2]]$p1
p2 <- resultados[[2]]$p2
p3 <- resultados[[3]]$p1
p4 <- resultados[[3]]$p2
p5 <- resultados[[4]]$p1
p6 <- resultados[[4]]$p2

combined_plot <- (p1 + p2 )/(p3+p4) /  ( p5 + p6) +
  plot_annotation(
    title = "Pred. vs Reales y gráfico de residuos",
  )
print(combined_plot)
```

Bloque: plot_mod_lin

```
resultados[[1]]<-evaluar_modelo("Lineal", m_ts,

predicciones = prediccionLineal)
resultados[[8]]<-evaluar_modelo("Dinámico", fit7,
predicciones = prediccionDin)
df_plotTodas <- data.frame(Inflacion_Real = y,
PrediccionLineal= resultados[[1]]$predicciones,
PrediccionL2 = resultados[[2]]$predicciones,
PrediccionL1 =resultados[[3]]$predicciones,
PrediccionElasNet =resultados[[4]]$predicciones,
PrediccionRF =resultados[[5]]$predicciones,
```

```
PrediccionGBM =resultados[[6]]$predicciones,
PrediccionSVR =resultados[[7]]$predicciones,
PrediccionDin= resultados[[8]]$predicciones
)

df_plotTodas$Mes<-indice_temporal

df_plotTodas |>
  mutate(Mes=yearmonth(Mes))|>
  as_tsibble(index = Mes) ->df_plotTodas


ggplot(df_plotTodas,aes(x = Mes)) +
    geom_line(aes(y = Inflacion_Real, colour = "Datos"),linewidth=0.9) +
    geom_line(aes(y = PrediccionLineal, colour = "Lineal")) +
    geom_line(aes(y = PrediccionDin, colour = "Dinámico"))+
    geom_line(aes(y = PrediccionL1, colour = "Lasso")) +
    geom_line(aes(y = PrediccionL2, colour = "Ridge")) +
    geom_line(aes(y = PrediccionElasNet, colour = "Elastic Net")) +

    labs(y = NULL,
         title = "Valores reales vs el ajuste del modelo"
    ) +
  scale_colour_manual(values = c("Datos" = "black",
                                 "Lineal" = "red",
                                 "Dinámico" = "blue",
                                 "Lasso" = "green",
                                 "Ridge" = "purple",
                                  "Elastic Net"="grey"))+

  guides(colour = guide_legend(title = NULL))
```

Bloque: rf-plots-ch2

```
p7<-resultados[[5]]$p1
p8<-resultados[[5]]$p2
p9<-resultados[[6]]$p1
p10<-resultados[[6]]$p2
p11<-resultados[[7]]$p1
p12<-resultados[[7]]$p2
combined_plot1 <- (p7 + p8)/(p9+p10)/(p11+p12) +
  plot_annotation(
    title = "Pred. vs Reales y gráfico de residuos",
  )
print(combined_plot1)
```

Bloque: plot_mod_adv

```r
ggplot(df_plotTodas,aes(x = Mes)) +
    geom_line(aes(y = Inflacion_Real, colour = "Datos"),linewidth=0.9) +
    geom_line(aes(y = PrediccionRF, colour = "Random Forest")) +
    geom_line(aes(y = PrediccionGBM, colour = "Gradient Boosting")) +
    geom_line(aes(y = PrediccionSVR, colour = "SVR")) +
     labs(y = NULL,
         title = "Valores reales vs el ajuste del modelo"
    ) +
  scale_colour_manual(values = c("Datos" = "black",
                                 "Elastic Net" = "orange",
                                "Random Forest" = "cyan",
                                "Gradient Boosting" = "magenta",
                                "SVR" = "brown")) +
  guides(colour = guide_legend(title = NULL))
```

Bloque: comp_mod

```r
nombres <- sapply(resultados, function(x) x$nombre)
r2_scores <- sapply(resultados, function(x) x$r2)
df_mse <- data.frame(Modelo = nombres, R2 = r2_scores)

ggplot(df_mse, aes(x = reorder(Modelo, -R2), y = R2)) +
      geom_bar(stat = "identity", fill = "steelblue") +
      geom_text(aes(label = round(R2, 4)),
                hjust = -0.1,
                position = position_dodge(width = 0.9)) +
      labs(
           x = "Modelos",
           y = "R²") +

      scale_y_continuous(expand = expansion(mult = c(0.05, 0.20))) +
      coord_flip() +
      theme_minimal()
```

Bloque: arbol

```r
datos.rpart <- rpart(Inflacion ~ ., data=datosEcon
                     , method="anova")
rpart.plot(datos.rpart)
```

Bloque: var_imp

```r
var_importance <- importance(modelo_rf)
importance_accuracy <- as.data.frame(var_importance) %>%
  tibble::rownames_to_column("Variable") %>%
  arrange(desc(`%IncMSE`))

ggplot(importance_accuracy, aes(x = reorder(Variable, `%IncMSE`),
```

```
  y = `%IncMSE`)) +
  geom_bar(stat = "identity", fill = "darkred") +
  coord_flip() + # Para que las barras sean horizontales
  labs(
    title = "Importancia de Variables",
    x = "Variable"
  ) +
  theme_minimal()
```

Bloque: r mods_pred

```
n_total <- nrow(datosEcon)
n_train <- 108 # Número de observaciones para entrenamiento


train_indices <- 1:n_train
test_indices <- (n_train + 1):n_total

train_df <- datosEcon[train_indices, ]
test_df <- datosEcon[test_indices, ]

# Preparar variables predictoras y objetivo

y<-datosEcon$Inflacion
X_train <- train_df[,-4]
y_train <- train_df$Inflacion
X_test <- test_df[,-4]
y_test <- test_df$Inflacion


evaluar_modelo <- function(nombre, modelo, predicciones_train,
predicciones_test) {
  # Calcular métricas en conjunto de entrenamiento
  mse_train <- mean((y_train - predicciones_train)^2)
  mae_train <- mean(abs(y_train - predicciones_train))
  r2_train <- 1 - sum((y_train - predicciones_train)^2) /
    sum((y_train - mean(y_train))^2)

  # Calcular métricas en conjunto de prueba
  mse_test <- mean((y_test - predicciones_test)^2)
  mae_test <- mean(abs(y_test - predicciones_test))
  r2_test<- 1 - sum((y_test - predicciones_test)^2) /
    sum((y_test - mean(y_test))^2)


  # Devolver resultados
  return(list(
    nombre = nombre,
```

```
    modelo = modelo,
    mse_test = mse_test,
    mae_test = mae_test,
    r2_test = r2_test,
    predicciones_test = predicciones_test

  ))
}



# Lista para almacenar resultados
resultados1 <- list()

#  Entrenar y evaluar modelos

# 1. Regresión Lineal (sin cambios en el entrenamiento)

modelo_lr <- lm(Inflacion ~ ., data = train_df)
pred_train_lr <- predict(modelo_lr, newdata = train_df)
pred_test_lr <- predict(modelo_lr, newdata = test_df)
resultados1[[length(resultados1) + 1]] <-

evaluar_modelo("Regresión Lineal", modelo_lr,

pred_train_lr, pred_test_lr)

# 2. Ridge Regression

x_train_matrix <- as.matrix(X_train)
x_test_matrix <- as.matrix(X_test)

modelo_ridge <- cv.glmnet(x_train_matrix, y_train, alpha = 0)
pred_train_ridge <- predict(modelo_ridge, newx = x_train_matrix,
s = "lambda.min")[,1]
pred_test_ridge <- predict(modelo_ridge, newx = x_test_matrix,
s = "lambda.min")[,1]
resultados1[[length(resultados1) + 1]] <-

evaluar_modelo("Ridge Regression", modelo_ridge,

pred_train_ridge,pred_test_ridge)

# 3. Lasso Regression (misma nota que Ridge sobre cv.glmnet)

modelo_lasso <- cv.glmnet(x_train_matrix, y_train, alpha = 1)
pred_train_lasso <- predict(modelo_lasso, newx = x_train_matrix,
s = "lambda.min")[,1]
```

```r
pred_test_lasso <- predict(modelo_lasso, newx = x_test_matrix,
s = "lambda.min")[,1]
resultados1[[length(resultados1) + 1]] <-

evaluar_modelo("Lasso Regression", modelo_lasso,

pred_train_lasso, pred_test_lasso)

# 4. ElasticNet (misma nota que Ridge sobre cv.glmnet)

modelo_elastic <- cv.glmnet(x_train_matrix, y_train, alpha = 0.5)
pred_train_elastic <- predict(modelo_elastic, newx = x_train_matrix,
s = "lambda.min")[,1]
pred_test_elastic <- predict(modelo_elastic, newx = x_test_matrix,

s = "lambda.min")[,1]
resultados1[[length(resultados1) + 1]] <-

evaluar_modelo("ElasticNet", modelo_elastic,

pred_train_elastic, pred_test_elastic)

# 5. Random Forest

train_df_rf <- datosEcon1[train_indices, ]
test_df_rf <- datosEcon1[test_indices, ]

# Preparar variables predictoras y objetivo

X_train_rf <- train_df_rf[,-4]
X_test_rf <- test_df_rf[,-4]

modelo_rf <- randomForest(x = X_train_rf, y = y_train,
ntree = 100,importance = TRUE)
pred_train_rf <- predict(modelo_rf, newdata = X_train_rf)
pred_test_rf <- predict(modelo_rf, newdata = X_test_rf)
resultados1[[length(resultados1) + 1]] <-

  evaluar_modelo("Random Forest",
                                              modelo_rf,

pred_train_rf, pred_test_rf)

# 6. Gradient Boosting

modelo_gbm <- gbm(Inflacion ~ `FPMT(rez.)`+Desempleo+`EI(rez.)`+`

                  HIPC(rez.)`+`HP`,
```

```
                  data = train_df,
                  distribution = "gaussian",
                  n.trees = 100,
                  interaction.depth = 3,
                  shrinkage = 0.1)
pred_train_gbm <- predict(modelo_gbm, newdata = train_df, n.trees = 100)
pred_test_gbm <- predict(modelo_gbm, newdata = test_df, n.trees = 100)
resultados1[[length(resultados1) + 1]] <-

evaluar_modelo("Gradient Boosting", modelo_gbm,

pred_train_gbm, pred_test_gbm)

# 7. Support Vector Regression


preproc <- preProcess(X_train, method = c("center", "scale"))
X_train_scaled <- predict(preproc, X_train)
X_test_scaled <- predict(preproc, X_test)

modelo_svr <- svm(x = X_train_scaled, y = y_train, kernel = "radial",
cost = 2, epsilon = 0.1)
pred_train_svr <- predict(modelo_svr, newdata = X_train_scaled)
pred_test_svr <- predict(modelo_svr, newdata = X_test_scaled)
resultados1[[length(resultados1) + 1]] <-

evaluar_modelo("Support Vector Regression", modelo_svr,

pred_train_svr, pred_test_svr)

#8. Modelo dinámico
datosP<-datosEcon
datosP$Mes<-indice_temporal

datosP |>
  mutate(Mes=yearmonth(Mes))|>
  as_tsibble(index = Mes) ->datosP
datosP|>
  filter(Mes %in% Mes[(n_train+1):120])->datosEcon_test

datosP|>
  filter(Mes %in% Mes[1:n_train])->datosEcon_train


din<-datosEcon_train|>
  model(ARIMA(Inflacion ~ `FPMT(rez.)`+Desempleo+`EI(rez.)`+`HIPC(rez.)`+
`HP`+pdq(d=1)+PDQ(P=0,Q=0,D=0)))
augment(din)|>
```

```
  pull(.fitted)->pred_din_train
forecast(din, new_data = datosEcon_test)|>
  pull(.mean)->pred_din_test

resultados1[[8]]<-

  evaluar_modelo("Dinámico", din,
                                  predicciones_train = pred_din_train,

predicciones_test = pred_din_test)
```

Bloque: r plot_todos_2-code

```
df_plotTodas <- data.frame(Inflacion_Real = y_test,
PrediccionLineal= resultados1[[1]]$predicciones_test,
PrediccionL2 = resultados1[[2]]$predicciones_test,
PrediccionL1 =resultados1[[3]]$predicciones_test,
PrediccionElasNet =resultados1[[4]]$predicciones_test,
PrediccionRF =resultados1[[5]]$predicciones_test,
PrediccionGBM =resultados1[[6]]$predicciones_test,
PrediccionSVR =resultados1[[7]]$predicciones_test,
PrediccionDin= resultados1[[8]]$predicciones_test
)
```

Bloque: plot_pred_mod_lin

```
df_plotTodas$Mes<-indice_temporal[(n_train+1):120]

df_plotTodas |>
  mutate(Mes=yearmonth(Mes))|>
  as_tsibble(index = Mes) ->df_plotTodas


ggplot(df_plotTodas,aes(x = Mes)) +
    geom_line(aes(y = Inflacion_Real, colour = "Datos"),linewidth=0.9) +
    geom_line(aes(y = PrediccionLineal, colour = "Lineal")) +
    geom_line(aes(y = PrediccionDin, colour = "Dinámico"))+
    geom_line(aes(y = PrediccionL1, colour = "Lasso")) +
    geom_line(aes(y = PrediccionL2, colour = "Ridge")) +
    geom_line(aes(y = PrediccionElasNet, colour = "Elastic Net")) +
    geom_line(aes(y = PrediccionRF, colour = "Random Forest")) +
    geom_line(aes(y = PrediccionGBM, colour = "Gradient Boosting")) +
    geom_line(aes(y = PrediccionSVR, colour = "SVR"))+
    labs(y = NULL,
         title = "Valores reales vs el ajuste del modelo"
    ) +
  scale_colour_manual(values = c("Datos" = "black",
                                  "Lineal" = "red",
```

```
                                    "Dinámico" = "blue",
                                    "Lasso" = "green",
                                    "Ridge" = "purple",
                                    "Random Forest" = "cyan",
                                    "Gradient Boosting" = "magenta",
                                    "SVR" = "brown"))+

  guides(colour = guide_legend(title = NULL))
```

Bloque: comp_mod_mse

```
nombres <- sapply(resultados1, function(x) x$nombre)
mse_scores <- sapply(resultados1, function(x) x$mse_test)
df_mse <- data.frame(Modelo = nombres, MSE = mse_scores)

ggplot(df_mse, aes(x = reorder(Modelo, -MSE), y = MSE)) +
      geom_bar(stat = "identity", fill = "salmon") +
      geom_text(aes(label = round(MSE, 4)),
                hjust = -0.1,
                position = position_dodge(width = 0.9)) +
      labs(title = "Comparación de Modelos - Error Cuadrático Medio (MSE)",
           x = "Modelos",
           y = "MSE en conjunto de prueba") +

      scale_y_continuous(expand = expansion(mult = c(0.05, 0.20))) +
      coord_flip() +
      theme_minimal()
```

Bloque: nuevos_datos

```
DatosPred <- read_excel("C:/Users/rtrin/OneDrive/
Escritorio/TFG/DatosPred.xlsx",
    col_names = FALSE, n_max = 4)
colnames(DatosPred)<-c("FPMT(rez.)","HIPC(rez.)",
"Desempleo(rez.)","EI(rez.)","HP(rez.)")
tabla_pander(DatosPred,
             decimales = 4)
```

Bloque: predicc

```
Mod_def<-datosEcon1|>
  model(ARIMA(Inflacion ~ `FPMT(rez.)`+`Desempleo(rez.)`+
`EI(rez.)`+`HIPC(rez.)`+`HP(rez.)`
+pdq(d=1)+PDQ(P=0,Q=0,D=0)))

año_inicio <- 2025
mes_inicio <- 1
```

```r
fechas_mensuales <- seq(

from = as.Date(paste(año_inicio,
                     mes_inicio, "01", sep = "-")),
                        by = "month",
                        length.out = 4)


indice_temporal1 <- format(fechas_mensuales, "%Y-%m")

DatosPred$Mes<-indice_temporal1

DatosPred |>
  mutate(Mes=yearmonth(Mes))|>
  as_tsibble(index = Mes) ->DatosPred

Prediccion_def<-forecast(Mod_def, new_data = DatosPred)
Tabla_Pred<-as.data.frame(Prediccion_def[,c("Mes",".mean")])
colnames(Tabla_Pred)<-c("Mes","Predicción")
tabla_pander(Tabla_Pred,
             decimales = 4)

Prediccion_def|>
  autoplot(linewidth=1)+
  labs(
    ,
     y = "Inflación pronosticada",
     x = "Mes"
  ) +
  theme_minimal()
```

Bibliografía

[1] . «Base de datos de desempleo». https://ec.europa.eu/eurostat/databrowser/view/UNE_RT_M___custom_7680578/bookmark/table?lang=en&bookmarkId=2feeff57-57c9-4278-a50b-7e2279d699c2.

[2] (). «Base de datos de FPMT». https://ec.europa.eu/eurostat/databrowser/view/prc_fsc_idx___custom_16901353/default/table?lang=en.

[3] . «Base de datos de HIPC». https://ec.europa.eu/eurostat/databrowser/view/prc_hicp_midx/default/table?lang=en&category=prc.prc_hicp.

[4] . «Base de datos de HIPC». https://ec.europa.eu/eurostat/databrowser/view/prc_hicp_midx/default/table?lang=en&category=prc.prc_hicp.

[5] (). «Base de datos de las tasas de interés». https://datosmacro.expansion.com/tipo-interes/zona-euro.

[6] (). «Bases de datos de los indicadores económicos europeos». https://ec.europa.eu/eurostat.

[7] Chan, Kung-Sik y Cryer, Jonathan D. (2008). *Time Series Analysis With Applications in R*. Springer, 1.ª edición.

[8] Dayal, Vikram (2020). *Quantitative Economics with R*. Springer, 1.ª edición.

[9] Hastie, Trevor; Tibshirani, Robert y Friedman, Jerome (2017). *The Elements of Statistical Learning*. Springer, 2.ª edición.

[10] Hyndman, Rob J y Athanasopoulos, George (2021). *Forecasting: Principles and Practice*. OText, 3.ª edición.

[11] Perez López, César (2024). *Análisis Multivariante de Datos. Aplicaciones con R*. Ibergaceta, 1.ª edición.

[12] Rodríguez, Diego (2022). «Los precios de la energía y la inflación: las medidas regulatorias y sus efectos». https://revistasice.com/index.php/ICE/article/view/7525/7566.

[13] Sethi, Alakh (2025). «Support Vector Regression Tutorial for Machine Learning». *analytics vidhya*.

[14] Shumway, Robert H. y Stoffer, David S. (2017). *Time Series Analysis and Its Applications: With R Examples*. Springer, 4.ª edición.

[15] Tamames, Ramón y Gallego, Santiago (2006). *Diccionario de Economía y Finanzas*. Alianza, 14.ª edición.

[16] Tibshirani, Robert (1996). «Regression Shrinkage and Selection via the Lasso». http://www.jstor.org/stable/2346178.

[17] Zou, H. and Hastie, T. (2005). «Regularization and Variable Selection via the Elastic Net».
http://www.jstor.org/stable/3647580.