



**UNIVERSIDAD
DE GRANADA**



ESCUELA INTERNACIONAL DE POSGRADO
Departamento de Estadística e Investigación Operativa

TRABAJO DE FIN DE MÁSTER

SERIES TEMPORALES APLICADAS

APPLIED TIME SERIES

Autor: Andrés Alejo Podadera Molero
Titulación: Máster en Estadística Aplicada
Tutor: Francisco Javier Alonso Morales
Departamento: Estadística e Investigación Operativa
Rama de conocimiento: Ciencias

ZÚRICH, junio de 2022

I. Declaración de originalidad del Trabajo de Fin de Máster:

Certifico que los documentos presentados son originales e inéditos, no habiéndose utilizado para la evaluación de ninguna otra asignatura del plan de estudios cursado. Así como, haber debidamente citado, las fuentes empleadas para la elaboración del mismo.

Esta declaración se realiza bajo la responsabilidad de quien la suscribe, a los efectos de participar en la Defensa del Trabajo de Fin de Máster.



Andrés Alejo Podadera Molero, Zúrich, junio de 2022.

II. Resumen

Se describe la estructura de los modelos de series temporales ARMA, ARIMA y SARIMA. Así como, las funciones de autocorrelación, autocorrelación parcial y autocorrelación extendida (acf, pacf y eacf).

Además, se presentan varios modelos construidos con datos reales de aplicación práctica en el campo de los riesgos financieros, esto es; los tipos de interés para los distintos bancos centrales en función de sus variables macroeconómicas.

Se realiza un análisis de los modelos, y se comparan. Pasando por todas las fases correspondientes de la modelización de series temporales. Identificación del modelo, estimación de los parámetros, significación, correlación, validación y predicción de futuros valores.

Se complementa el análisis, con el ajuste del problema a través de un modelo de Redes Neuronales. Y también, con la utilización de modelos conjuntos SARIMA y de Redes Neuronales.

El modelo con menor error es aquel que combina la red neuronal con el modelo SARIMA. Si bien es cierto, que el modelo SARIMA de por sí, ofrece una precisión bastante alta. Que, de forma individual, supera a la red neuronal.

Como conclusión podemos destacar que el ajuste de un modelo económico real, en concreto, la tasa de interés de la Reserva Federal americana; que se selecciona de forma manual por un comité de expertos, es difícil de modelar. Las mayores dificultades están a la hora de normalizar el modelo y estabilizar su varianza.

No obstante, con un buen ajuste de los parámetros y coeficientes, y seleccionando el modelo adecuado; se pueden conseguir resultados positivos. Como se ha demostrado a lo largo de este documento. En este caso de estudio, se comprueba que los métodos compuestos o ensamblados, pueden ofrecer mejores resultados que los modelos de forma individual. Si bien es cierto que, estos resultados no deben extrapolarse a otros problemas.

En futuros estudios, se pueden tener en cuenta otras variables macroeconómicas como la inflación, el crecimiento del empleo o la cotización de los mercados financieros respecto al producto interior bruto.

III. Palabras Clave

Modelización, Series Temporales, AR, MA, ARMA, ARIMA, SARIMA, Autocorrelación, ACF, PACF, EACF, Redes Neuronales, Perceptrón Multicapa, Híbrido, Riesgos Financieros, Tipos de Interés.

IV. Abstract

The structure of the ARMA, ARIMA and SARIMA time series models is described. As well as the autocorrelation, partial autocorrelation and extended autocorrelation functions (acf, pacf and eacf).

In addition, several models built with real data of practical application in the field of financial risks, i.e. interest rates for different central banks as a function of their macroeconomic variables, are presented.

The models are analysed and compared. Going through all the corresponding phases of time series modelling. Model identification, parameter estimation, significance, correlation, validation and prediction of future values.

The analysis is complemented with the adjustment of the problem through a Neural Network model. And also, with the use of joint SARIMA and Neural Network models.

The model with the lowest error is the one that combines the neural network with the SARIMA model. Although it is true that the SARIMA model by itself offers a fairly high accuracy. That, individually, surpasses the neural network.

As a conclusion we can highlight that the adjustment of a real economic model, specifically the interest rate of the American Federal Reserve, which is selected manually by a committee of experts, is difficult to model. The greatest difficulties are in normalising the model and stabilising its variance.

However, with a good fit of the parameters and coefficients, and by selecting the right model, positive results can be achieved. As has been demonstrated throughout this paper. In this case study, it is shown that composite or ensemble methods can provide better results than individual models. However, these results should not be extrapolated to other problems.

In future studies, other macroeconomic variables such as inflation, employment growth or the price of financial markets relative to gross domestic product can be taken into account.

V. Keywords

Modelling, Time Series, AR, MA, ARMA, ARIMA, SARIMA, Autocorrelation, ACF, PACF, EACF, Artificial Neural Network, Multilayer Perceptron, Hybrid, Financial Risks, Interest Rates.

VI. Agradecimientos

Al profesorado de la UGR por su paciencia y trato.

A mis padres por su cariño y apoyo incondicional.

ÍNDICE

TRABAJO DE FIN DE MÁSTER	1
I. <i>Declaración de originalidad del Trabajo de Fin de Máster:</i>	3
II. <i>Resumen</i>	4
III. <i>Palabras Clave</i>	4
IV. <i>Abstract</i>	5
V. <i>Keywords</i>	5
VI. <i>Agradecimientos</i>	6
1. Introducción a los Modelos SARIMA para series temporales	9
1.1. Modelos autorregresivos AR.....	16
1.2. Modelos de medias móviles MA.....	16
1.3. Modelos autorregresivos de medias móviles ARMA	17
1.4. ARIMA.....	17
1.5. SARIMA	18
1.6. Funciones de autocorrelación.....	19
1.7. Funciones de autocorrelación parcial	20
1.8. Funciones de autocorrelación extendida	22
1.9. Elección del modelo SARIMA	24
1.10. Validación del modelo SARIMA	25
1.11. Método híbrido de Redes Neuronales y SARIMA	27
2. Ajuste del modelo SARIMA usando R	31
2.1. Presentación de los datos.....	31
2.2. Identificación del modelo: funciones muestrales y comportamiento.....	32
2.3. Estimación del modelo: parámetros, significación y correlación	42
2.4. Validación del modelo: estudio de los residuos	48
2.5. Predicción del modelo SARIMA	50
2.6. Predicción del modelo híbrido SARIMA+NN.....	52
3. RESULTADOS	60
4. CONCLUSIONES	63
5. BIBLIOGRAFIA	64
5.1. Libros	64
5.2. Publicaciones.....	64
5.3. Enlaces web.....	65
6. ANEXOS	67
6.1. Código en R.....	67
6.2. Otros anexos.....	77

6.3. Datos utilizados:.....77

1. Introducción a los Modelos SARIMA para series temporales

Una serie temporal es un conjunto de observaciones de una variable ordenadas en el tiempo. Así, tenemos una variable que toma distintos valores en cada instante de tiempo. El análisis de las series temporales puede resultar útil para obtener información relevante de la variable estudiada o predecir valores futuros.

Los modelos clásicos consideran la serie temporal como la función de cuatro componentes: Tendencia (T_t), Ciclo (C_t), Estacionalidad (S_t) y Ruido (E_t). El conjunto, de los mismos generan la serie temporal final.

$$X_t = f(T_t, C_t, S_t, E_t) \quad (1)$$

Estos, se pueden combinar bien de forma aditiva:

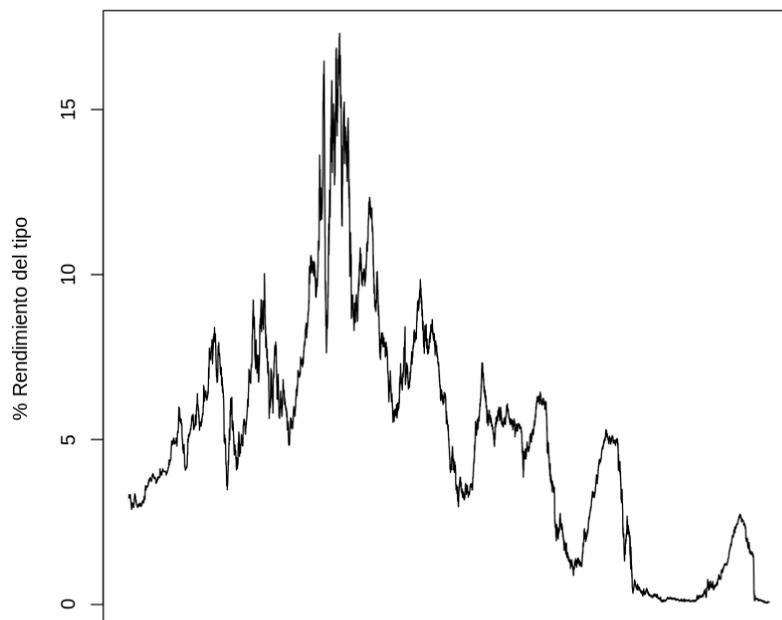
$$X_t = T_t + C_t + S_t + E_t \quad (2)$$

O bien de forma multiplicativa:

$$X_t = T_t * C_t * S_t * E_t \quad (3)$$

Consideramos como ejemplo para ilustrar el concepto anterior la serie temporal que describe el rendimiento porcentual de las letras del tesoro americanas a un año vista.

Rendimiento del tipo del tesoro americano a 1 año



Enero de 1962 a julio 2021

Ilustración 1 Evolución del rendimiento del interés de los bonos del tesoro americanos de 1962 a 2021.

Para elaborar los gráficos se utiliza el código en R incluido en los anexos. A continuación, se descompondrá de forma aditiva definiendo los componentes de la serie temporal.

La Tendencia (T_t) es el movimiento o dirección regular de la serie a largo plazo. Puede ser creciente, decreciente o estable. En la siguiente imagen, podemos observar la componente de tendencia del rendimiento mencionado anterior:

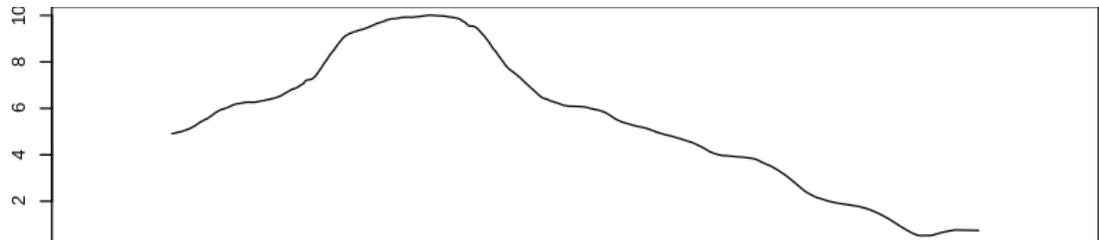


Ilustración 2 Componente tendencial de la descomposición de la serie temporal del rendimiento descrito para el mismo periodo

El componente Cíclico (C_t) es un movimiento oscilante en torno a la tendencia que provoca variaciones no necesariamente periódicas de la variable. En el diagrama anterior, se incluye el movimiento descrito. Siendo realmente, la componente tendencia-cíclica. Ya que, suelen representarse de forma conjunta.

El componente Estacional (S_t) es un movimiento oscilante periódico que se repite en espacios temporales de mayor escala, por ejemplo, de forma trimestral o anual. En el ejemplo utilizado, se utilizan décadas, por el comportamiento del producto financiero. También, podemos representar esta componente en un diagrama que se repite como podemos observar debajo:

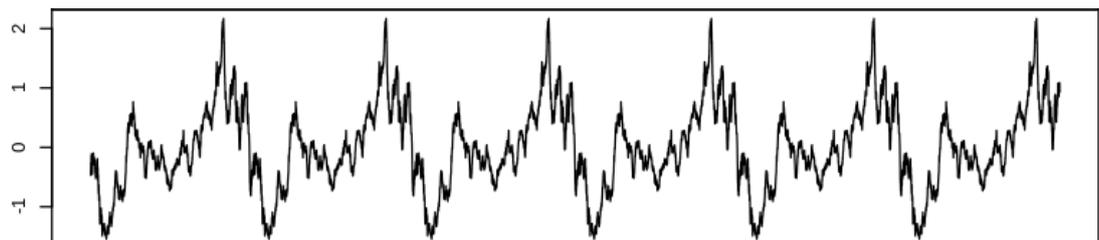


Ilustración 3 Componente estacional de la serie temporal definida anteriormente

El Ruido o Error (E_t) se considera el movimiento provocado por anomalías temporales o variaciones desconocidas. Por su naturaleza se considera arbitrario y no predecible. El ruido tiene una representación visual poco ordenada, para nuestra serie temporal es:

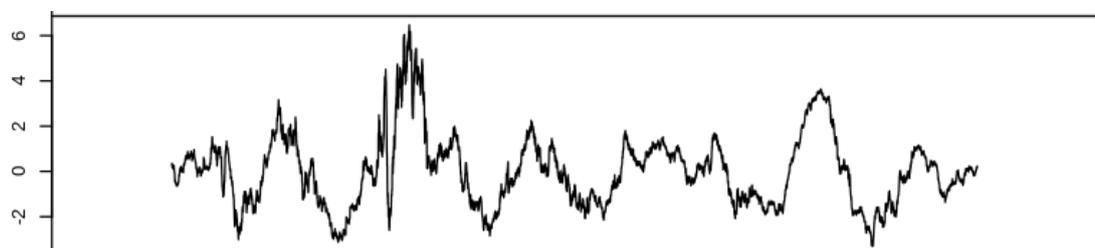
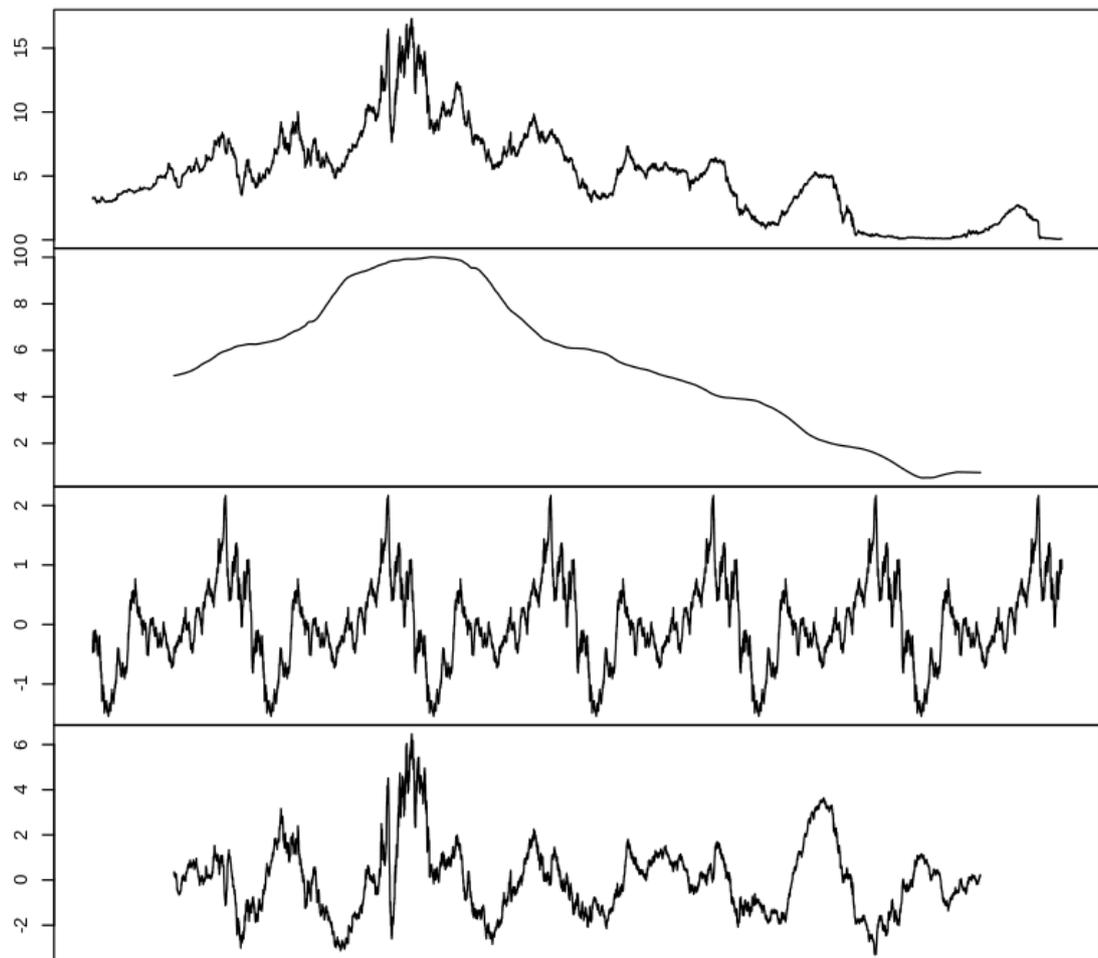


Ilustración 4 Ruido o error del ejemplo de serie temporal.

Podemos representar de forma conjunta, la serie temporal y sus componentes. Apreciando los rangos de valor que representan cada una de las componentes, del total de la serie.



Rendimiento del tipo del tesoro americano a 1 año

Ilustración 5 Serie temporal descompuesta de forma aditiva con sus componentes. Serie temporal completa, tendencia-cíclica, estacional y error de forma respectiva.

Una serie temporal puede ser estacionaria o no. Una serie temporal estacionaria es aquella cuyas propiedades no dependen del momento en que se observa la serie. Por lo tanto, las series temporales con tendencias, no son estacionarias, ya que afectará al valor de la serie temporal en diferentes momentos.

Una serie temporal y_t será estrictamente estacionaria si la distribución conjunta de $y_{t_1}, y_{t_2}, \dots, y_{t_n}$ es la misma que la de $y_{t_1-k}, y_{t_2-k}, \dots, y_{t_n-k}$ para todos los valores de tiempo 't' y retardos 'k'.

Para 'n=1' tenemos que las distribuciones unidimensionales son iguales, por tanto:

$$E(y_t) = E(y_{t-k}) \quad (4)$$

$$\text{Var}(y_t) = \text{Var}(y_{t-k}) \quad (5)$$

De forma similar para 'n=2' tenemos que la distribución bivariada de ' y_t ' e ' y_s ' deben ser iguales a las de ' y_{t-k} ' e ' y_{s-k} '. Por tanto, se deduce que para todo 't', 's' y 'k':

$$\text{Cov}(y_t, y_s) = \text{Cov}(y_{t-k}, y_{s-k}) \quad (6)$$

Para 'k=s' y después 'k=t' tenemos que:

$$\gamma_{t,s} = \text{Cov}(y_{t-s}, y_0) = \text{Cov}(y_0, y_{s-t}) = \text{Cov}(y_0, y_{|t-s|}) = \gamma_{0,|t-s|} \quad (7)$$

Esto significa que la covarianza entre ' y_t ' e ' y_s ' solo depende de $|t-s|$ y no de los instantes 't' y 's'. De esta forma el proceso estacionario queda definido como:

$$\gamma_k = \text{Cov}(y_t, y_{t-k}) \quad (8)$$

$$\rho_k = \text{Corr}(y_t, y_{t-k}) \quad (9)$$

Además:

$$\rho_k = \frac{\gamma_k}{\gamma_0} \quad (10)$$

Algunas propiedades de un proceso estacionario serán por tanto:

$$\left\{ \begin{array}{l} \gamma_0 = \text{var}(y_t) \\ \gamma_k = \gamma_{-k} \\ |\gamma_k| \leq \gamma_0 \\ \rho_0 = 1 \\ \rho_k = \rho_{-k} \\ |\rho_k| \leq 1 \end{array} \right. \quad (11)$$

Si es estrictamente estacionaria y su varianza no es infinita, la función de covarianza depende solo del retardo.

Una serie temporal y_t será débilmente estacionaria o de segundo grado, si la función media es constante en el tiempo y se cumple que $\gamma_{t,t-k} = \gamma_{0,k}$ para cualquier instante t y retardo k.

Así, podemos identificar tres tipos de estacionariedad en las series temporales:

- Estrictamente estacionario: como se ha mencionado anteriormente un proceso estocástico es estrictamente estacionario si la distribución conjunta de $y_{t_1}, y_{t_2}, \dots, y_{t_n}$ es la misma que la de $y_{t_1-k}, y_{t_2-k}, \dots, y_{t_n-k}$ para todos los valores de tiempo 't' y retardos 'k'.
- Estacionario en media o débilmente estacionario de primer orden: un proceso estocástico será estacionario de primer orden si y solo si $E[y_t]$ es

constante, es decir, no depende de los valores de tiempo 't'.

- Estacionario en autocovarianza o débilmente estacionario de segundo orden: un proceso estocástico será estacionario de segundo orden si y solo si $E[y_t]$, $Var[y_t]$, $Cov[y_t, y_{t+k}]$ son constantes, es decir, no dependen de los valores de tiempo 't'. Pudiendo depender de los retardos 'k'.

A lo largo del documento, cuando se habla de estacionariedad se asume esta como débilmente estacionaria de segundo orden.

Por otro lado, una serie con ruido blanco es estacionaria, no importa el momento en que se observe. Al ser este, una secuencia de variables aleatorias independientes e idénticamente distribuidas.

Es importante definir la estacionariedad, como se ha hecho anteriormente. Pues, para hacer inferencias estadísticas sobre la estructura de un proceso estocástico o sus observaciones, debemos hacer algunas suposiciones. El supuesto más importante es el de estacionariedad.

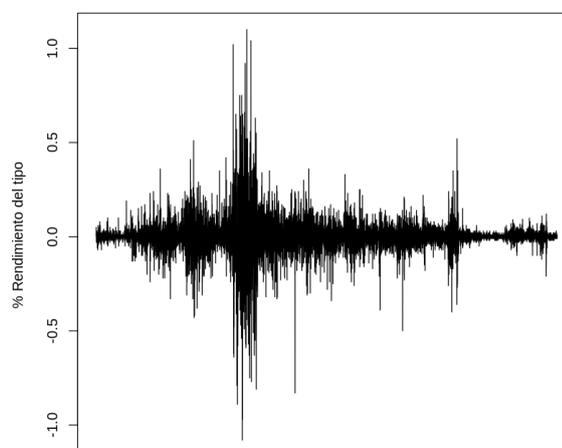
La idea básica de estacionariedad es que las leyes de probabilidad que gobiernan el comportamiento del proceso no cambian con el tiempo. En cierto sentido, el proceso está en equilibrio estadístico.

De esta forma es común transformar las series temporales no estacionarias para poder extraer mejores conclusiones de su estudio. Una de las técnicas utilizadas es la diferenciación.

La diferenciación se realiza restando la observación anterior a la actual. Puede utilizarse para eliminar la tendencia de las series en el tiempo y se estabiliza la media.

Podemos ver un ejemplo en la imagen inferior:

S. T. Diferenciada, rendimiento del tipo del tesoro americano a 1 año



Enero de 1962 a julio 2021

Ilustración 6 Serie temporal del ejemplo diferenciada y estacionaria

También, se puede utilizar la transformación logarítmica para la variable analizada, si esta tiene un comportamiento exponencial. Esto es típico en algunos fenómenos económicos por consecuencia del interés compuesto, entre otros.

Ésta, se utiliza para los casos en los que el aumento de la dispersión parece estar asociado a los niveles más altos de la serie. Es decir, cuanto más alto es el nivel de la serie, más variación hay y viceversa.

Otro método más complejo, pero que por lo general proporciona mejores resultados en el caso de datos sesgados, es la transformación de Box-Cox. Que realiza la transformación de variables dependientes no normales en una forma normal. Mediante, el uso de la siguiente función:

$$y(\lambda) = \begin{cases} \frac{(y^\lambda - 1)}{\lambda}, & \text{si } \lambda \neq 0; \\ \log y, & \text{si } \lambda = 0. \end{cases} \quad (12)$$

Esto se consigue mediante el cálculo de un exponente, lambda (λ). Éste, varía de -5 a 5. Se consideran todos los valores de λ y se selecciona el valor óptimo para los datos presentados. Que será, el que da como resultado la mejor aproximación a una curva de distribución normal.

‘R’ nos permite considerar este rango de valores de lambda y calcular un valor de log-verosimilitud para cada valor de lambda basado en una función de probabilidad normal. Eligiendo así, el valor óptimo.

La transformación de la potencia sólo se aplica a los valores positivos de los datos. Si algunos de los valores son negativos o cero, se puede añadir una constante positiva a todos los valores para hacerlos positivos antes de realizarla.

Cabe mencionar que estas técnicas de transformación no siempre conducen a obtener series estacionarias. Pero en la mayoría de los casos, mejoran el “equilibrio estadístico” de las mismas, facilitando el ajuste de los modelos predictivos.

Una vez presentados los principales métodos de transformación de una serie temporal para hacerla estacionaria, debemos evaluar, si dicha transformación ha sido satisfactoria. Para ello, utilizamos el test de Dickey-Fuller.

El test de Dickey-Fuller es una prueba de validación de hipótesis. Así considerando un modelo AR(1) (definido en 1.1.) de la forma:

$$y_t = \rho y_{t-1} + u_t \quad (13)$$

En el ‘ y_t ’ es la variable estudiada, ‘ ρ ’ un coeficiente y ‘ u_t ’ es el error o ruido blanco. El modelo sería no estacionario con ‘ $\rho = 1$ ’. El modelo de regresión se define como:

$$\nabla y_t = (\rho - 1)y_{t-1} + u_t = \delta y_{t-1} + u_t \quad (14)$$

Siendo ‘ ∇ ’ el operador de primera diferenciación y ‘ $\delta \equiv \rho - 1$ ’. El estadístico ‘ t ’ tiene una distribución específica que se recoge en la tabla Dickey-Fuller al realizarse el test sobre los residuos y no sobre los datos en bruto.

El test se puede definir de distintas formas, siendo una de ellas la que incluye la raíz unitaria con constante y tendencia determinística:

$$\nabla y_t = a_0 + a_1 t + \delta y_{t-1} + u_t \quad (15)$$

La hipótesis nula es en todos los casos que exista una raíz unitaria. Esto es, ‘ $\delta = 0$ ’. Por tanto, la hipótesis alternativa es que ‘ y_t ’ es estacionario.

En el trabajo se utiliza una versión mejorada de la prueba conocida como Dickey Fuller aumentado (ADF), que elimina todos los efectos estructurales (autocorrelación) en la serie temporal y luego realiza la prueba utilizando el mismo procedimiento.

En el ejemplo anterior (Ilustración 6), se puede apreciar a simple vista que la serie no es estacionaria. Sin embargo, esto puede ser comprobado con la prueba de Dickey-Fuller:

```
Dickey-Fuller = -2.3785, Lag order = 1, p-value = 0.418
alternative hypothesis: stationary
```

Como podemos observar tiene un p-valor mayor del 5%. No se considera que haya suficientes indicios para descartar la hipótesis nula. Por tanto, se comprueba que las series no son estacionarias. Sin embargo, si diferenciamos la serie temporal. Y repetimos la prueba, obtenemos un resultado distinto:

```
Dickey-Fuller = -78.942, Lag order = 1, p-value = 0.01
alternative hypothesis: stationary
```

Siendo el p-valor menor del 5% podemos descartar la hipótesis de nulidad, es decir, es estacionaria después de ser diferenciada.

Tras esta breve introducción, podemos definir el objeto de estudio de este documento. Que serán los modelos de series temporales SARIMA. No obstante, debemos introducir previamente los modelos clásicos anteriores.

Estos son; autorregresivos AR, de medias móviles MA y autorregresivos de medias móviles ARIMA. Y se describen en las próximas secciones (1.1. – 1.4.).

Las siglas AR de ARIMA significan "autorregresivo", la I significa "integrado" y MA es "media móvil". Por tanto especificamos el orden de los modelos ARIMA como p,d,q.

Así, por ejemplo, un modelo AR(2) es equivalente a un ARIMA(2,0,0). También, un modelo MA(1) es equivalente a un ARIMA(0,0,1). Si tuviéramos un modelo ARMA(3,1) que aplicáramos a datos que se han diferenciado dos veces, entonces tendríamos un modelo ARIMA(3,2,1).

Habiendo definido la estacionariedad, cabe destacar que utilizaremos los modelos AR, MA y ARMA para series temporales estacionarias. Por otro lado, los modelos ARIMA y SARIMA serán utilizados para series temporales no estacionarias.

Se procede a introducir dichos modelos.

1.1. Modelos autorregresivos AR

Los modelos autorregresivos se utilizan bajo el pretexto de que los valores pasados tienen un efecto sobre los valores actuales, lo que hace que esta técnica estadística sea popular para analizar la naturaleza, la economía y otros procesos que varían con el tiempo, esto es, series temporales.

En un modelo de autorregresión o autorregresivo (AR), predecimos la variable de interés utilizando una combinación lineal de valores históricos de la variable. El término autorregresión indica que se trata de una regresión de la variable contra sí misma.

Así, un modelo autorregresivo de orden p puede escribirse como:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t \quad (16)$$

La letra 'c' representa una constante numérica. ϵ_t es el ruido blanco o residuo que representa la diferencia entre nuestra predicción para el periodo t y el valor correcto.

El coeficiente ϕ_1 es una constante numérica por la que multiplicamos la variable retardada y_{t-1} . Este coeficiente, siempre toma valores entre -1 y 1.

El orden de una autorregresión es el número de valores inmediatamente anteriores de la serie que se utilizan para predecir el valor en el momento actual. La complejidad del modelo crece con el orden del modelo. Ya que aumenta el número de coeficientes a determinar, haciendo más probable que algunos de ellos no sean significativos.

1.2. Modelos de medias móviles MA

En las series temporales, se observan similitudes entre los errores del pasado y los valores del presente. Esto se debe a que se producen ciertos acontecimientos imprevisibles que deben tenerse en cuenta.

Es decir, sabiendo lo lejos que estuvo la estimación anterior, en comparación con el valor real, se puede ajustar un modelo, para que responda en consecuencia. El modelo de medias móviles (MA) tiene en cuenta los residuos del pasado para hacer el mencionado ajuste.

$$y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_q \epsilon_{t-q} \quad (17)$$

Siendo ϵ_t ruido blanco o residuo que representa la diferencia entre nuestra predicción para el periodo t y el valor correcto.

La letra 'c' representa una constante numérica. La variable θ es el equivalente de ϕ en el modelo AR. Se cambia la letra griega, para diferenciarlas por notación.

Así, cada valor de y_t puede considerarse como una media móvil ponderada de los últimos errores de previsión.

Un modelo de media móvil simple es equivalente a un modelo autorregresivo de retardo infinito con ciertas restricciones que no son objeto de estudio de este documento.

Los modelos de media móvil son siempre débilmente estacionarios porque son combinaciones lineales finitas de una secuencia de ruido blanco para la que los dos primeros momentos son invariantes en el tiempo.

1.3. Modelos autorregresivos de medias móviles ARMA

Los modelos autorregresivos de medias móviles, ARMA(p,q), son la suma de un proceso autorregresivo de orden p y uno de medias móviles de orden q. Las funciones de autocorrelación simple y parcial de un proceso ARMA también son la suma de las correspondientes a un proceso MA y a otro AR.

En consecuencia, utiliza los datos del pasado como base para las estimaciones, pero también pueden ajustarse rápidamente a las perturbaciones imprevisibles. Un modelo autorregresivo de medias móviles (ARMA) se puede expresar como:

$$y_t = c + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \dots + \theta_q\epsilon_{t-q} + \phi_1y_{t-1} + \phi_2y_{t-2} + \dots + \phi_p y_{t-p} \quad (18)$$

Siendo los parámetros los mismos de las ecuaciones descritas anteriormente para los modelos AR y MA.

Por lo tanto, este modelo puede explicar la relación de una serie temporal tanto con el ruido aleatorio (parte de la media móvil) como con ella misma en el pasado (parte autorregresiva).

ACF y PACF son herramientas importantes cuando se trata de identificar el orden apropiado de p y q, además de otros usos. Estas, se describirán en las secciones 1.6-1.8.

1.4. ARIMA

ARIMA, es un acrónimo de media móvil integrada autorregresiva. Los modelos ARIMA, surgen de inducir estacionariedad en procesos no estacionarios. Así, combinamos la diferenciación con la autorregresión y un modelo de media móvil, para obtener un modelo ARIMA.

Cualquier serie temporal, no estacionaria, que presente patrones y no sea un ruido blanco aleatorio puede ajustarse y representarse con modelos ARIMA. El modelo puede escribirse como:

$$y'_t = c + \phi_1 y'_{t-1} + \dots + \phi_p y'_{t-p} + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q} + \epsilon_t \quad (19)$$

Siendo los parámetros los mismos de las ecuaciones descritas anteriormente para los modelos AR, MA y ARMA. Siendo y'_t la serie diferenciada.

Los "predictores" del lado derecho incluyen los valores retardados de y_t y los errores retardados. Tomando la forma ARIMA(p,d,q). Esto es, orden de la parte autorregresiva, grado de la primera diferenciación y orden de la parte de la media móvil; respectivamente.

Algunos casos particulares del modelo ARIMA utilizando esta nomenclatura son:

Caso	ARIMA(p,d,q)
Ruido blanco	ARIMA(0,0,0)
Paseo aleatorio	ARIMA(0,1,0) sin constante
Paseo aleatorio con deriva	ARIMA(0,1,0) con una constante
Autorregresión	ARIMA(p,0,0)
Media móvil	ARIMA(0,0,q)

Tabla 1 Casos particulares del modelo ARIMA(p,d,q)

No obstante, la clase SARIMA es más apropiada para las series temporales que presentan estacionalidad. Esto es, el ARIMA estacional, una subclase que se presenta en el siguiente apartado (1.5).

1.5. SARIMA

Los modelos SARIMA son una forma relativamente sencilla de modelar series temporales de tiempo cuyas tendencias estacionales no son tan regulares como las que tendríamos con un modelo de tendencia estacional determinista.

Un modelo ARIMA estacional (SARIMA) se forma incluyendo términos estacionales adicionales al modelo ARIMA presentado anteriormente. La parte estacional del modelo se compone de términos similares a los componentes no estacionales, pero que implican retrocesos del periodo estacional.

El modelo añade tres nuevos parámetros para especificar la autorregresión (AR), la diferenciación (I) y la media móvil (MA) para el componente estacional de la serie, así como un parámetro adicional para el periodo de la estacionalidad.

Respecto a su notación, se utilizan letras mayúsculas para representar las partes estacionales del modelo. Esto es orden autorregresivo estacional (P), orden de diferencia estacional (D), orden de media móvil estacional (Q), número de pasos de tiempo para un solo período estacional (s).

El modelo SARIMA se puede representar de la siguiente forma:

$$SARIMA(p, d, q)x(P, D, Q)s \quad (20)$$

SARIMA sólo incorpora términos y diferenciales de una estacionalidad. Esto significa que, si los datos presentan varios ciclos distintos, se deben diferenciar con anterioridad.

De forma alternativa, construimos el modelo con la dependencia regular y estacional por separado, y luego incorporamos ambas de forma multiplicativa. Que representamos como:

$$\Phi_P(B^S)\phi_p(B)\nabla_s^D\nabla^d y_t = \theta_q(B)\Theta_Q(B^S)\epsilon_t \quad (21)$$

Siendo ‘ $\Phi_P(B^S)$ ’ el operador estacional AR de orden P, ‘ ϕ_p ’ el operador AR de orden p, ‘ ∇_s^D ’ las diferencias estacionales, ‘ ∇^d ’ las diferencias regulares, ‘ $\Theta_Q(B^S)$ ’ el operador media móvil estacional de orden Q, ‘ $\theta_q(B)$ ’ el operador media móvil de orden q y ϵ_t el ruido blanco.

El modelo SARIMA será la base de este documento, y se presenta un caso práctico en la sección 1.9. para mostrar la construcción del modelo, su análisis y resultados. No obstante, antes se deben presentar las funciones de autocorrelación 1.6.-1.8. que serán esenciales en el análisis del modelo.

1.6. Funciones de autocorrelación

La función de autocorrelación, o por sus siglas en inglés ACF, representa los valores de autocorrelación de cualquier serie con sus valores retardados. Trazamos estos valores junto con la banda de confianza para obtener el gráfico ACF.

Describe como se relaciona el valor actual de la serie con sus valores pasados. Una serie temporal puede tener componentes como la tendencia, la estacionalidad, la cíclica o el residuo.

Sea la serie temporal Y_t , entonces la función de autocorrelación entre el instante t y el instante s se puede expresar como:

$$\rho(s, t) = \frac{\gamma(s, t)}{\sqrt{(\gamma(s, s)\gamma(t, t))}} \quad (22)$$

Entre sus propiedades se encuentra primero, el hecho de ser simétrica y como segunda propiedad cabe destacar el hecho que se encuentra acotada entre -1 (correlación negativa) y 1 (correlación positiva).

Un ACF mide la correlación media entre los puntos de datos de una serie temporal y los valores anteriores de la serie medidos para diferentes longitudes de retardo. Por ejemplo, la correlación en el primer retardo se mide como la correlación entre los valores de la serie temporal medidos en el tiempo t con todos los valores de la serie medidos en el tiempo (t - 1).

La correlación en el segundo retraso mide la correlación entre los valores de la serie temporal medidos en el momento t con todos los valores de la serie medidos en el momento $(t - 2)$. Esto continúa durante tantos retrasos como sea necesario, pero es normal que se incluyan entre $T/3$ y $T/4$ siendo T el número de datos.

En el siguiente apartado, se define también la función de autocorrelación parcial. Que, junto con la función de autocorrelación, permite explicar el funcionamiento de los modelos AR y MA (final sección 1.7.)

1.7. Funciones de autocorrelación parcial

La función de autocorrelación parcial, o por sus siglas en inglés PACF, en lugar de encontrar correlaciones del presente con los retardos como el ACF, encuentra la correlación de los residuos de una regresión sobre las variables intermedias.

La función de autocorrelación parcial de una serie temporal Y_t , ϕ_{kk} se puede expresar como:

$$\phi_{kk} = \text{Corr}(Y_t, Y_{t-k} | Y_{t-1}, Y_{t-2}, \dots, Y_{t-k+1}) \quad (23)$$

Es decir, ϕ_{kk} es la correlación en la distribución bivariada de Y_t e Y_{t-k} condicionada a $Y_{t-1}, Y_{t-2}, \dots, Y_{t-k+1}$.

En el segundo retraso, el PACF mide la correlación entre los puntos de datos en el momento t con los puntos de datos en el momento $t - 2$ después de controlar la correlación entre los puntos de datos en el momento t con los del momento $t - 1$.

Por lo tanto, si hay alguna información oculta en el residuo que pueda ser modelada por el siguiente retardo, podríamos obtener una buena correlación y mantendríamos ese siguiente retardo como una característica durante el modelado.

Es conveniente evitar demasiadas características que estén correlacionadas ya que eso puede crear problemas de multicolinealidad. Por lo tanto, es necesario mantener sólo las características pertinentes.

Habiendo definido la función de autocorrelación y la función de autocorrelación parcial, podemos ver como se utilizan para explicar el comportamiento de los modelos AR y MA.

Normalmente no es posible saber, simplemente a partir de un gráfico temporal, qué valores de p y q son apropiados para el ajuste. No obstante, podemos utilizar el gráfico ACF, y el gráfico PACF de forma conjunta, para determinar los valores que más se adaptan de p y q .

Será posible únicamente en los casos que los modelos sean AR o MA o en su respectiva notación ARIMA($p,0,0$) y ARIMA($0,0,q$). Si el modelo ARIMA toma valores tanto de p como de q , las funciones de autocorrelación no proporcionarán una ayuda suficiente. Aunque existen alternativas como la función de autocorrelación extendida (1.8.).

En general, el análisis de los diagramas se puede resumir en la siguiente tabla:

Función a interpretar	AR / ARIMA(p,0,0)	MA / ARIMA(0,0,q)
ACF	Decae exponencialmente / sinusoidal	Es nula a partir del retardo 'q'
PACF	Es nula a partir del retardo 'p'	Decae exponencialmente / sinusoidal

Tabla 2 Interpretación de las funciones de autocorrelación ACF y PACF para los modelos AR y MA

Esto es, los datos pueden seguir un ajuste ARIMA(p,0,0) si los gráficos ACF y PACF de los datos diferenciados muestran los siguientes comportamientos:

- El ACF decae exponencialmente o bien es sinusoidal.
- La función variará hasta el retardo 'p' en la PACF, pero será nula más allá del mismo.

Podemos generar un ejemplo teórico de un proceso AR(2) y sus respectivos diagramas ACF y PACF con la función de R 'ARMAacf':

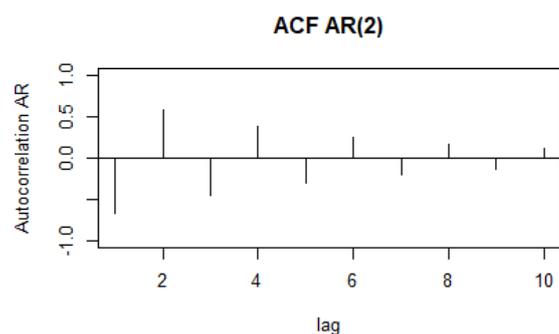


Ilustración 7 Función de autocorrelación de un proceso AR(2)

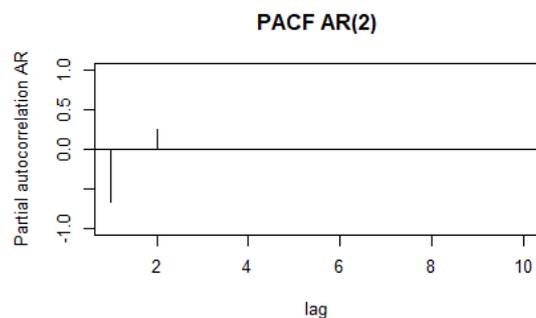


Ilustración 8 Función de autocorrelación parcial de un proceso AR(2)

Como podemos observar en la ilustración 7 el gráfico ACF decae sinusoidalmente. Además, en la ilustración 8 podemos ver como en la función PACF se produce un último pico en el retardo '2' que coincide con el valor de 'p' como se ha mencionado anteriormente.

De forma análoga, los datos pueden ajustarse con un modelo ARIMA(0,0,q) si los gráficos ACF y PACF de los datos diferenciados muestran el siguiente comportamiento:

- El PACF decae exponencialmente o es sinusoidal.
- La función variará hasta el retardo 'q' en el ACF, pero será nula más allá del mismo.

Podemos también en este caso, generar un ejemplo teórico de un proceso MA y sus respectivos diagramas ACF y PACF con la función de R ‘ARMAacf’:

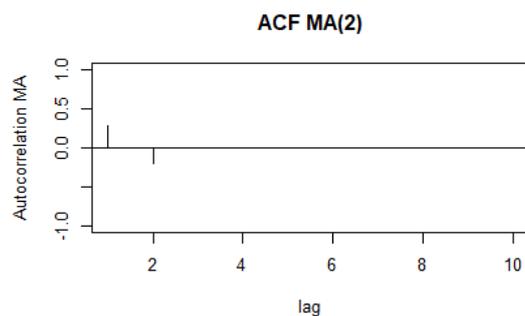


Ilustración 9 Función de autocorrelación de un proceso MA(2)

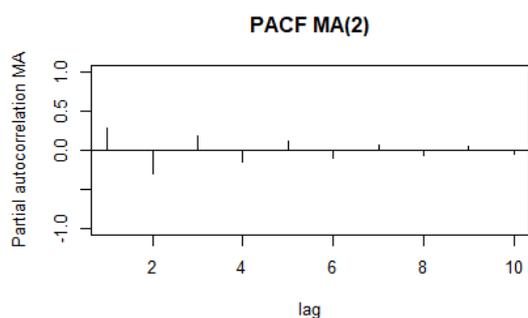


Ilustración 10 Función de autocorrelación parcial de un proceso MA(2)

Se aprecia en la ilustración 9, que el gráfico ACF deja de tomar valores a partir del retardo ‘2’ que coincide con ‘q’, de forma similar al PACF del caso anterior. Además, en la ilustración 10, podemos ver como la función PACF decae sinusoidalmente; de forma similar, al ACF del proceso AR(2).

1.8. Funciones de autocorrelación extendida

El ACF y el PACF proporcionan herramientas eficaces para identificar modelos AR(p) o MA(q). Sin embargo, para un modelo ARMA mixto, su ACF y PACF teóricos tienen infinitos valores distintos de cero, lo que dificulta la identificación de modelos mixtos a partir de la ACF y PACF de la muestra. La principal herramienta gráfica para facilitar la identificación de los órdenes ARMA, es el método de autocorrelación extendida (EACF) propuesto por Tsay y Tiao en 1984 [7].

El método EACF utiliza el hecho de que si se conoce la parte AR de un modelo ARMA mixto, al “filtrar” la serie temporal con el ajuste previo se obtiene un proceso MA, cuya función se anula en el retardo ‘q’. Así, los coeficientes AR pueden estimarse mediante una secuencia finita de regresiones.

De forma general, en el caso ARMA(p,q) se realizan tantas regresiones como sean necesarias hasta obtener:

$$X_t = \sum_{i=1}^p \phi_i^{(q)} X_{t-1} + e_t^{(q)} + \sum_{i=1}^q \theta_i e_{t-i}^{(q)} \quad (24)$$

Donde $e_t^{(q)}$ son los errores de la regresión número 'q' en el instante 't'.

Se realiza la regresión sobre 'p' valores de X. Estimados los parámetros ϕ_i se puede construir el proceso:

$$W_t = X_t - \sum_{i=1}^p \hat{\phi}_i^{(q)} X_{t-i} \quad (25)$$

Y estudiar la correlación de este. Para determinar el orden de los polinomios se sigue el procedimiento, siendo:

$$W_{t,k,j} = X_t - \sum_{i=1}^k \phi_i^{(j)} X_{t-i} \quad (26)$$

$W_{t,k,j}$ los residuos de la regresión suponiendo un ARMA(k,j). Las funciones de autocorrelación muestrales de los mismos serán las autocorrelaciones muestrales extendidas. Por tanto, la función de autocorrelación extendida será:

$$\hat{\rho}(k, l) = \hat{\rho}_l(\hat{W}_{t,k,l}) \quad (27)$$

Siendo 'l' la correlación del proceso ARMA(k,l). Cuando ' $k > p$ ' y ' $|c| < 1$ ' se cumple que:

$$\hat{\rho}(k, l) \xrightarrow{p} \begin{cases} c, & l = q + k - p \\ 0, & l > q + k - p \end{cases} \quad (28)$$

Así, podemos disponer los valores en una tabla que contendrá los valores $\hat{\rho}(k, l)$ para las filas AR y las columnas MA. Como se muestra a continuación:

AR / MA	0	1	2	...
0	$\hat{\rho}(0,1)$	$\hat{\rho}(0,2)$	$\hat{\rho}(0,3)$...
1	$\hat{\rho}(1,1)$	$\hat{\rho}(1,2)$	$\hat{\rho}(1,3)$...
2	$\hat{\rho}(2,1)$	$\hat{\rho}(2,2)$	$\hat{\rho}(2,3)$...
...

Tabla 3 EACF genérica

Se compara el valor obtenido con ± 1.96 veces la desviación típica con la aproximación de Bartlett o $(T - j - k)^{-1}$ bajo la hipótesis de independencia. Los p-valores no significativos con este criterio se representan con círculos y con cruces los que sí lo son.

Se muestra un ejemplo de una matriz EACF o función de autocorrelación extendida obtenida con 'R', que nos ayuda a determinar las posibilidades del modelo ARMA.

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1	x	x	x	o	o	o	o	o	x	x	o	o	o	o
2	x	o	x	o	o	o	o	o	o	x	o	o	o	o
3	x	x	x	o	o	o	o	o	o	x	o	o	o	o
4	x	x	o	o	o	o	o	o	o	x	o	o	o	o
5	x	x	x	o	o	o	o	o	o	o	o	o	o	o
6	x	o	x	x	o	o	o	o	o	x	o	o	o	o
7	x	o	o	x	x	o	x	o	o	o	o	o	o	o

Ilustración 11 Ejemplo de matriz EACF

Si el proceso del ejemplo fuera ARMA(p,q), los valores con círculos deberían formar un triángulo cuyo vértice será los valores ‘p’ y ‘q’ del mismo. Siendo ‘p’ la fila y ‘q’ la columna. No obstante, en el ejemplo de la matriz EACF expuesto, no se aprecia este patrón de forma clara; y por tanto, no podemos determinar que un modelo ARMA se ajuste correctamente.

1.9. Elección del modelo SARIMA

Para el elegir un modelo el primer paso es inspeccionar de forma detallada la serie temporal a estudiar. Visualmente y de forma genérica se puede reconocer si una gráfica presenta tendencia, estacionalidad, etc. Además, podemos identificar cualquier observación inusual que se presente en el gráfico.

Si es necesario, recurrimos a la transformación de Box-Cox para estabilizar la varianza. Esta se ha explicado anteriormente, y es especialmente relevante en aquellos casos que no presenten una distribución normal o tiene datos sesgados.

Es importante distinguir entre series temporales estacionarias y no estacionarias. Pues esto determina si utilizaremos modelos más simples como AR, MA o ARMA; o en su defecto, otros modelos como ARIMA y SARIMA. Además, la presencia de patrones estacionales también determinará la distinción entre estos.

Es interesante probar con los respectivos test, las hipótesis que se plantean desde la visualización del gráfico. Así, comprobamos matemáticamente la estacionariedad con la prueba de Dickey-Fuller presentada anteriormente. Lo que nos permite elegir entre los modelos anteriores.

A continuación, debemos representar el diagrama ACF de la serie temporal, para estudiar la correlación entre los retardos. La existencia de una persistencia en la ACF, indica la necesidad de realizar un ajuste de diferenciación.

Por otro lado, si en el diagrama ACF se presenta un patrón en los retardos, repetido en el tiempo (por ejemplo, en $t = 6, 12, 18, \dots$), indica la necesidad de realizar un ajuste de diferenciación estacional.

De forma análoga ocurre con el diagrama PACF tal y como se ha presentado al final de la sección 1.7.

De esta forma, podemos realizar sucesivas iteraciones o tentativas hasta llegar a un resultado que ajuste los valores de '(p,d,q)' de forma adecuada. En cualquier caso, los ajustes deben ser validados con posterioridad (1.10.)

Otra forma de realizar el ajuste es con el criterio de información de Akaike (AIC). Que normalmente, se utiliza para seleccionar los predictores en la regresión, también es útil para determinar el orden de un modelo ARIMA. Se puede escribir como:

$$AIC = -2 \log(L) + 2(p + q + k + 1) \quad (29)$$

Siendo 'L' la verosimilitud de los datos, y '(p + q + k + 1)' el número de parámetros del modelo incluyendo la varianza de los residuos; 'k' tendrá valor unidad si los valores de 'c' son distintos de cero; y valor nulo, de lo contrario.

Su versión para ARIMA será:

$$AIC_c = AIC + \frac{2(p + q + k + 1)(p + q + k + 2)}{T - p - q - k - 2} \quad (30)$$

Este criterio permite elegir de forma cuantitativa los valores de p y q del modelo, mediante la minimización de AIC_c . Teniendo el orden del modelo, podemos determinar los parámetros o coeficientes (ϕ_p, θ_q) .

Para ello, se utiliza de forma genérica la estimación de máxima verosimilitud (MLE). Esta técnica encuentra los valores de los parámetros que maximizan la probabilidad de obtener los datos que hemos observado.

Es importante señalar que estos criterios de información no suelen ser una buena guía para seleccionar el orden de diferenciación adecuado 'd' de un modelo, sino sólo para seleccionar los valores de 'p' y 'q'.

Esto se debe a que la diferenciación cambia los datos sobre los que se calcula la verosimilitud, lo que hace que los valores del AIC entre modelos con diferentes órdenes de diferenciación no sean comparables. Así que resulta más apropiado utilizar algún otro enfoque, como el presentado anteriormente, para elegir 'd' y luego podemos utilizar el AIC_c para seleccionar 'p' y 'q'.

1.10. Validación del modelo SARIMA

Una vez seleccionado el modelo (S)ARIMA debemos validar que se ajusta correctamente a nuestros datos. De lo contrario, no será correcto para su propósito. Para ello, se analizan los residuos del modelo, la correlación de los estimadores, si existe sobreajuste, y la predicción de este; entre otros.

En primer lugar, debemos comprobar que los residuos, se comporten como ruido blanco. Esto significa, que toda la señal de la serie temporal se ha ajustado en nuestro

modelo. Se puede comprobar, mediante el test de Ljung-Box sobre los residuos para validar la hipótesis nula de independencia. Podemos definirlo de la siguiente manera:

H_0 : Se cumple que: ' $\rho_1 = \rho_2 = \rho_3 = \dots = \rho_h = 0$ '.

H_1 : Se cumple que alguna: ' $\rho_h \neq 0$ ' ó bien que: ' $i \leq h - p - q$ '.

$Q(h)$, definido a continuación, sigue aproximadamente una distribución chi-cuadrado bajo la hipótesis nula. Es decir, se cumple que las 'h' primeras autocorrelaciones son nulas.

Siendo el estadístico:

$$Q(h) = n(n+2) \sum_{k=1}^h \frac{\hat{p}_k^2}{n-k} \quad (31)$$

El tamaño de la muestra es 'n', ' \hat{p}_k ' es la autocorrelación de la muestra en el retardo 'k', y 'h' el número de retardos considerados. Para el nivel de significación α , la región crítica para el rechazo de la hipótesis de aleatoriedad es:

$$Q > X_{1-\alpha, h-p-q}^2 \quad (32)$$

Siendo $X_{1-\alpha, h-p-q}^2$ el cuantil de la distribución chi-cuadrado con 'h-p-q' grados de libertad.

Si obtenemos un p-valor inferior a 0.05, se considera que sigue existiendo correlación que no es fruto del azar. Esto es, que los residuos no se comportan como ruido blanco. Por tanto, que el modelo no presenta un ajuste suficientemente correcto. Esto es, debemos seguir iterando con otras alternativas.

Otro componente que considerar, son los valores de significancia de los parámetros modelo. Esto es, como de apropiados son los parámetros del ajuste. O qué impacto tiene cada uno de ellos sobre los datos.

En la siguiente imagen podemos observar un test de significancia de los parámetros de un ajuste SARIMA incorrecto. Se genera con la función de R 'coefstest()'. Como se puede ver, de acuerdo con la leyenda que se presenta debajo, uno de los parámetros en concreto el 'drift', no es significativo para el modelo.

z test of coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
ar1	0.42522760	0.00074782	568.6256	< 2.2e-16	***
sar1	-0.01855690	0.00334636	-5.5454	2.933e-08	***
sar2	-0.03356698	0.00221799	-15.1339	< 2.2e-16	***
drift	-0.00029696	0.00823191	-0.0361	0.9712	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					

Ilustración 12 Significación de los parámetros de un modelo SARIMA mal ajustado

Para el resto de los parámetros el ajuste si parece estar justificado. No obstante, no podemos considerar que este modelo sea válido. Y en este ejemplo, debemos seguir probando con otras posibilidades, como eliminar el parámetro drift.

También podemos observar visualmente, como se comportan los residuos. Para ello, observamos su normalidad y los gráficos de autocorrelación, aplicando los criterios de comportamiento de la sección 1.7.

A continuación, se presentan dichos diagramas generados en R con la función ‘checkresiduals()’ para el modelo del ejemplo anterior (Ilustración 13):

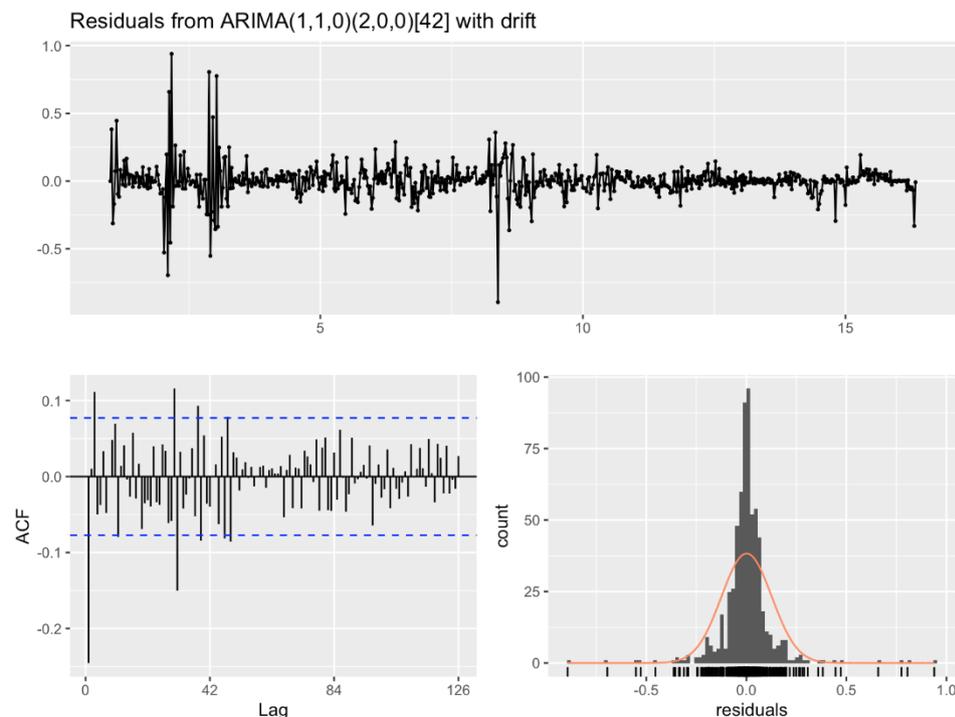


Ilustración 13 Residuos del modelo SARIMA mal ajustado

Se aprecia en el gráfico ACF que existen picos fuera de los intervalos de confianza que no se detienen después de ‘p’. Además, es destacable que los residuos siguen una distribución no normal. Por tanto, por este criterio tampoco podemos validar el modelo del ejemplo y debemos seguir iterando.

1.11. Método híbrido de Redes Neuronales y SARIMA

Los métodos conjuntos son una técnica de aprendizaje automático que combina varios modelos base para producir un modelo predictivo óptimo. En el caso de los métodos conjuntos de Redes Neuronales (NN) se refieren a la práctica de utilizar un sistema que incluye tanto redes neuronales como otros métodos y funcionan de forma conjunta.

En lugar de crear varios modelos y elegir el más preciso, los métodos de conjunto tienen en cuenta varios y los promedian para producir un modelo final. O bien, cada modelo se ocupa de una parte de la tarea.

En el caso práctico de este trabajo, se propone la creación y evaluación de un modelo SARIMA de forma completa e individual, además de un método híbrido con Redes Neuronales. Se compararán sus resultados para determinar qué estrategia es más adecuada en el caso tratado.

El método híbrido estará basado en la integración de los datos obtenidos del modelo estacional de medias móviles integradas autorregresivas (SARIMA) con el modelo de redes neuronales (NN) para predecir la variable objetivo.

Es decir, la previsión en varios pasos se realiza de forma recursiva. El modelo se alimenta de las predicciones del modelo SARIMA o de las predicciones de sus residuos, para generar su predicción.

El modelo de red neuronal (NN) se inspira en la forma en que el cerebro humano procesa la información. Se ha utilizado ampliamente en diferentes campos debido a su gran capacidad de generalización, como la identificación de imágenes, el reconocimiento de enfermedades, la previsión del precio y la demanda de energía y el área de mercado.

La red utilizada de perceptrón multicapa, está compuesto por una capa de entrada, una o más capas ocultas y una capa de salida. Cada capa está compuesta por unidades llamadas neuronas, y cada neurona está conectada a todas las unidades de la capa anterior.

Las entradas se multiplican por los pesos y se suman al sesgo, y los resultados pasan por una función como la siguiente:

$$v_k = \sum_{j=1}^m w_{kj} a_j + b_k \quad (33)$$

$$z_k = \varphi(v_k) \quad (34)$$

Se puede representar de forma visual con el siguiente esquema:

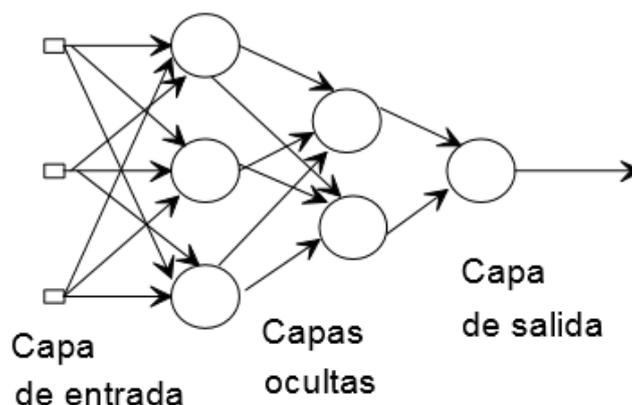


Ilustración 14 Esquema de una red neuronal de perceptrón multicapa. Fuente: "Desarrollo de un sistema inteligente para la clasificación de documentos ya digitalizados aplicando redes neuronales supervisadas" Henry Paz et al. (2015)

O bien el modelo híbrido de forma conjunta:

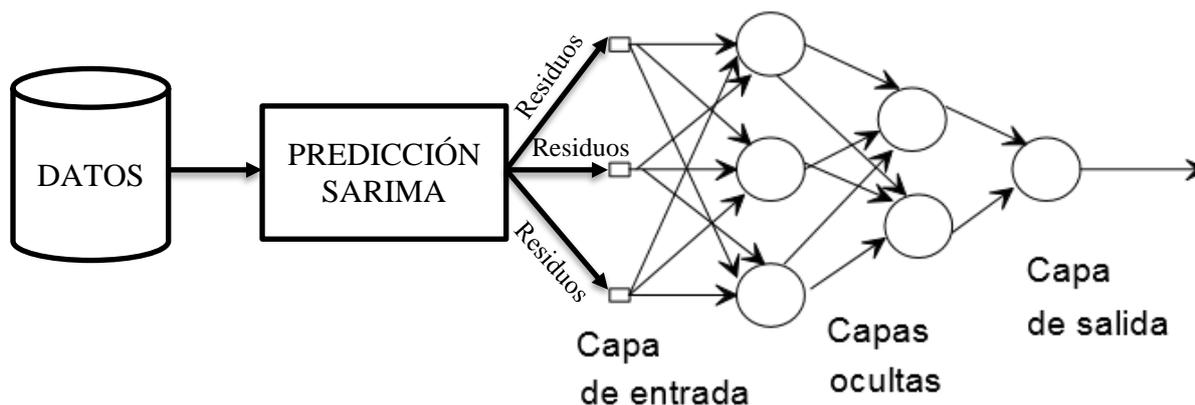


Ilustración 15 Esquema del modelo híbrido SARIMA+NN

El ajuste de la red neuronal se consigue teniendo en cuenta varios elementos. En primer lugar, la función de activación que determinará para cada una de las neuronas si es tenida en cuenta en la función lineal:

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (35)$$

También, para el caso de la regresión se debe considerar la función de descenso error cuadrático esto es:

$$\text{Descenso}(\hat{y}, y, W) = \frac{1}{2} \|\hat{y} - y\|_2^2 + \frac{\alpha}{2} \|W\|_2^2 \quad (36)$$

Donde $\frac{\alpha}{2} \|W\|_2^2$ es un término de regularización L2 que penaliza los modelos complejos. Siendo α un hiperparámetro positivo que controla dicha penalización.

Partiendo de unos pesos iniciales aleatorios, el perceptrón multicapa (MLP) minimiza la función de pérdida actualizando los pesos de forma iterativa. Después de calcular la pérdida, un paso hacia atrás la propaga desde la capa de salida a las capas anteriores, proporcionando a cada parámetro de peso un valor de actualización destinado a disminuir la pérdida.

Esta disminución se realiza con el gradiente descenso $\nabla \text{Descenso}_W$. Esto es, el gradiente de la pérdida con respecto a los pesos, se calcula y se deduce de W . Se puede expresar como:

$$W^{i+1} = W^i - \epsilon \nabla \text{Descenso}_W^i \quad (37)$$

donde 'i' es el paso de iteración, y 'ε' es la tasa de aprendizaje con un valor positivo.

El algoritmo se detiene cuando alcanza un número máximo preestablecido de iteraciones; o cuando la mejora de la pérdida es inferior a un número determinado y pequeño.

La integración con el modelo SARIMA consiste en alimentar la red neuronal de los valores de la serie temporal o bien de los valores de los residuos para intentar predecir como se comportarán en el futuro. Después, estos resultados, se integran con el modelo SARIMA para mejor el ajuste utilizando bien los mismos datos o bien el ruido blanco que resta del mismo.

2. Ajuste del modelo SARIMA usando R

Para ajustar los modelos mencionados en la sección anterior se divide el proceso en los pasos que se enumeran a continuación (2.1.-2.6.): presentación de los datos, identificación del modelo (funciones muestrales y comportamiento), estimación del modelo (parámetros, significación y correlación), validación del modelo (estudio de los residuos) y, por último; predicción del modelo simple e híbrido.

2.1. Presentación de los datos

La serie temporal elegida son los tipos de interés de la Reserva Federal de Estados Unidos (FED). Se elige esta institución y no el Banco Central Europeo, porque es la referencia internacional principal del resto de bancos centrales, por su peso en la economía global y el uso del dólar como divisa del comercio internacional. Además, su recorrido histórico es mayor, facilitando una mayor cantidad de datos disponibles.

El tipo de interés de los fondos federales es el tipo de interés al que las instituciones de depósito intercambian fondos federales. Esto es, los saldos mantenidos en los bancos de la Reserva Federal entre en un día.

Cuando una institución de depósito tiene excedentes de saldos en su cuenta de reserva, presta a otros bancos que necesitan mayores saldos. En términos más sencillos, un banco con exceso de efectivo, lo que suele denominarse liquidez, prestará a otro banco que necesite obtener liquidez rápidamente.

El Comité Federal de Mercado Abierto (FOMC) determina el tipo objetivo de los fondos federales. Este tipo influye en el tipo efectivo de los fondos federales mediante operaciones de mercado abierto o mediante la compra y venta de deuda pública.

Más concretamente, la Reserva Federal puede disminuir la liquidez vendiendo bonos del Estado, con lo que aumenta el tipo de los fondos federales porque los bancos tienen menos liquidez para comerciar con otros bancos.

Del mismo modo, la Reserva Federal puede aumentar la liquidez comprando bonos del Estado, disminuyendo el tipo de los fondos federales porque los bancos tienen un exceso de liquidez para comerciar.

Que la Reserva Federal quiera comprar o vender bonos depende del estado de la economía. Si el FOMC considera que la economía está creciendo demasiado rápido y que las presiones inflacionistas son muy altas, se puede establecer un objetivo de tipo de interés de los fondos federales más alto para moderar la actividad económica.

De lo contrario, el FOMC puede fijar un objetivo de tipo de interés de los fondos federales más bajo para estimular una mayor actividad económica. Por lo tanto, el FOMC debe observar el estado actual de la economía para determinar el mejor curso de la política monetaria que maximice el crecimiento.

Para tomar sus decisiones, la Reserva Federal utiliza datos como la evolución de los precios y los salarios, el empleo, el gasto y los ingresos de los consumidores, las inversiones de las empresas y los mercados de divisas.

Las decisiones de la Reserva Federal sobre los tipos de interés tienen una importancia muy grande en la economía global, y puede cambiar la dirección de la cotización múltiples mercados financieros acciones, divisas, bonos, materias primas, etc.

Tras introducir el fenómeno representado, presentamos su serie temporal a continuación:

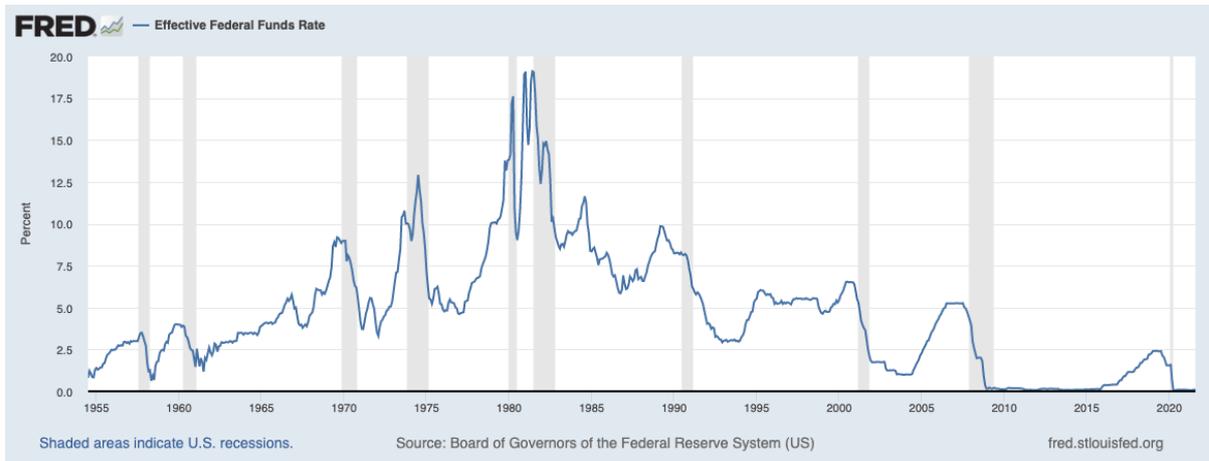


Ilustración 16 Tipo de interés de la FED desde 1954 hasta la actualidad. Oscurecidas las épocas de recesión de la economía americana.

Se puede ver en gris las épocas de recesión de la economía americana y como coinciden con el conocido como ‘tampering’ o rebaja de los tipos de interés. Por otro lado, se puede ver como la tasa de interés fluctúa a lo largo del tiempo, además; se mantienen mucho más bajos en las dos últimas décadas. Analizaremos el comportamiento de la serie temporal en las siguientes secciones de forma cuantitativa.

2.2. Identificación del modelo: funciones muestrales y comportamiento

El código en R que se ha utilizado para todo el documento está disponible en la sección de Anexos (6.). Se recomienda su visualización para aclarar los pasos seguidos durante el proceso.

En primer lugar, se descargan los datos de la Ilustración 16 presentada anteriormente, y se grafican. De esta forma podemos, comprobar que los datos se han cargado correctamente. En la Ilustración 17 se observa que los datos son los mismos confirmándolo.

También podemos observar, que no se presentan valores anormales y la gráfica es continua en el tiempo. A priori, no necesita transformaciones en este sentido. Además, se realiza una exploración básica de la estacionalidad con las ilustraciones 17 y 18 donde se puede ver los años y meses ordenados por sus tasas, y agrupados en un código de colores

que muestra los datos agrupados por fechas. No se ve estacionalidad de forma clara, al no haber patrones marcados.

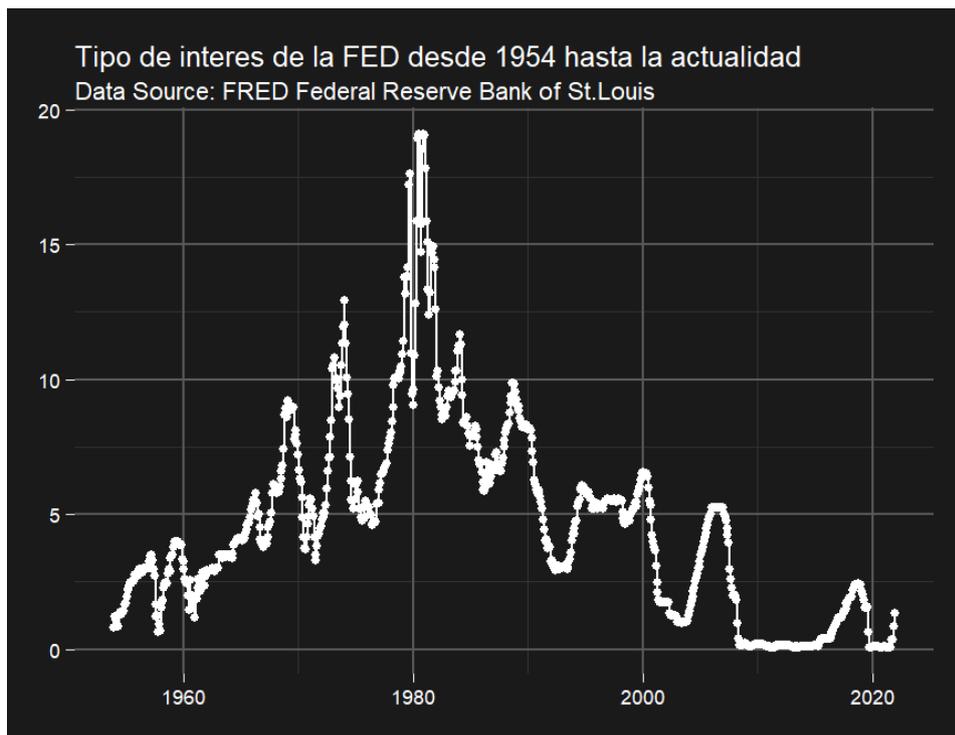


Ilustración 17 Tipo de interés de la FED desde 1954 hasta la actualidad generado con R

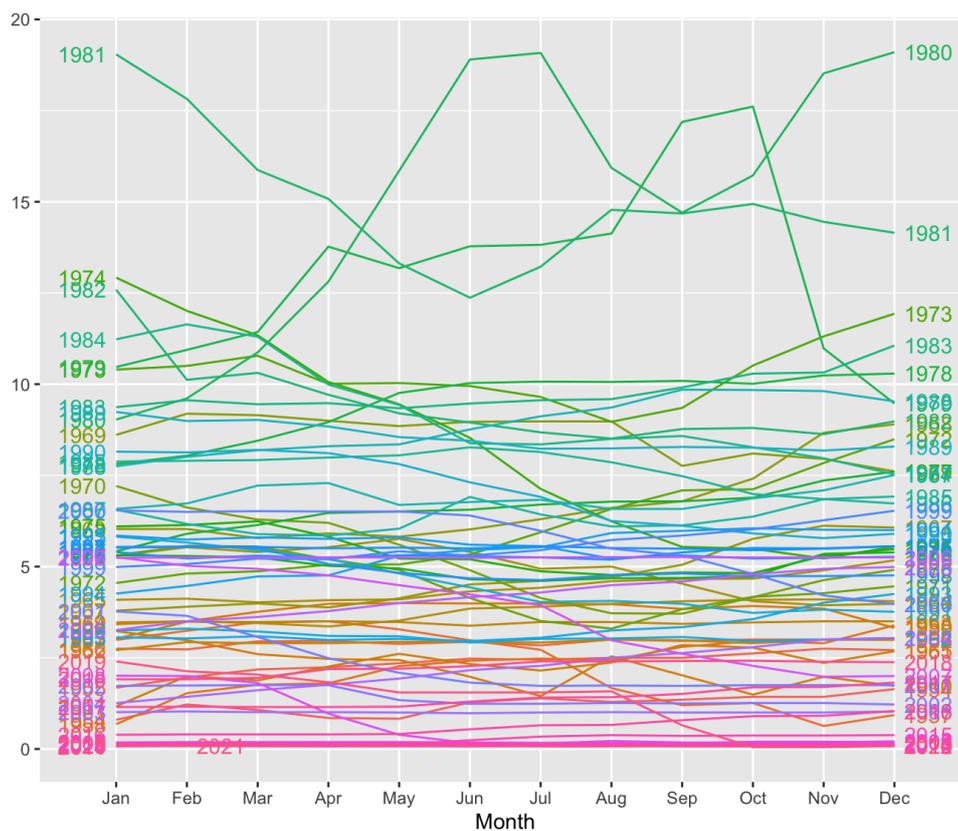


Ilustración 18 Valor de la tasa de interés en cada año

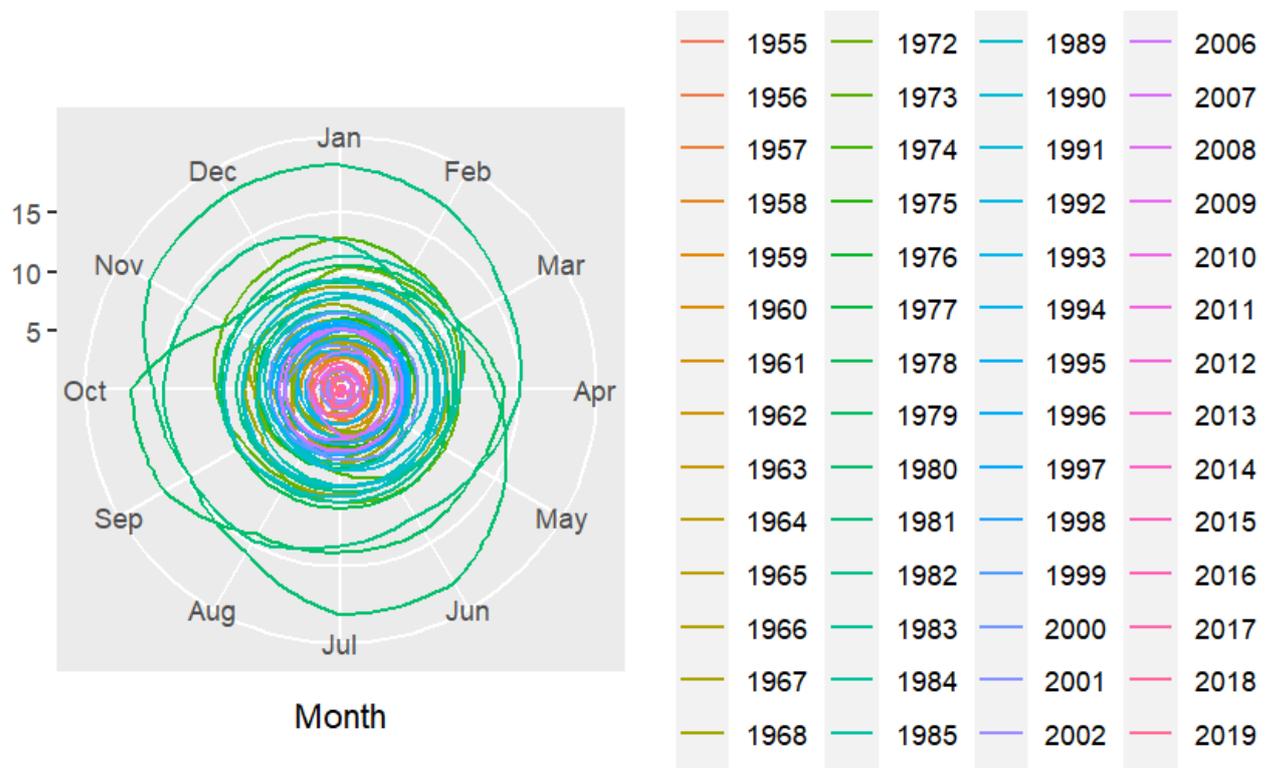


Ilustración 19 Tasas de interés representadas por meses y años de forma polar

En segundo lugar, se analiza la estacionariedad de los datos. Se realiza, mediante la prueba de Dickey-Fuller:

```
> adf.test(yt)
```

Augmented Dickey-Fuller Test

```
data: yt
```

```
Dickey-Fuller = -3.1209, Lag order = 9, p-value = 0.1038
```

```
alternative hypothesis: stationary
```

Podemos observar que el p-valor es de $0.1038 > 0.05$ por tanto se considera no estacionario. Esto, además, indica la necesidad de utilizar un modelo ARIMA o SARIMA en vez de un modelo mas simple como AR, MA o ARMA.

A continuación, podemos estudiar la normalidad del modelo. Para ello, realizamos un grafico Q-Q como el que se muestra a continuación:

```
> qqnorm(yt, pch = 19, col = "gray50")
```

```
> qqline(yt)
```

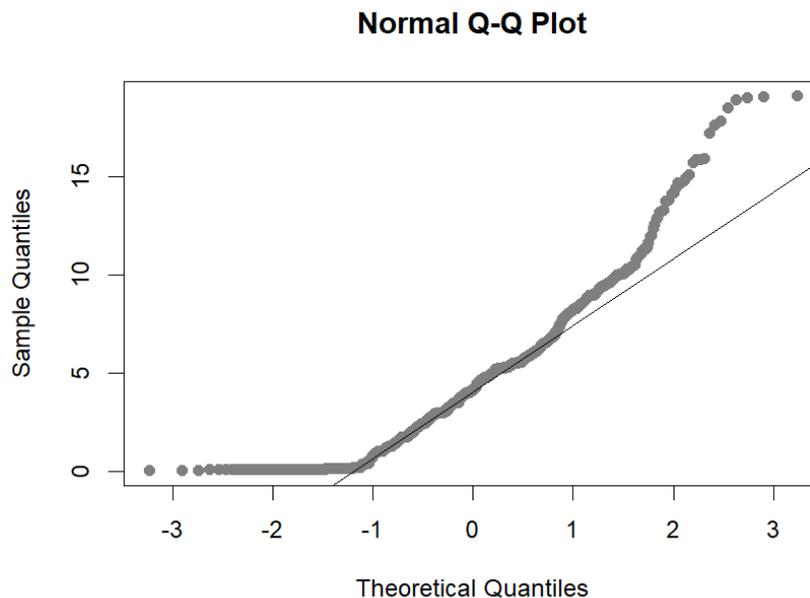


Ilustración 20 Gráfico Q-Q antes de la transformación Box-Cox

En la ilustración 20, se observa a simple vista que la serie temporal no es normal. En cualquier caso, se pueden realizar pruebas con validación de hipótesis para confirmarlo. Un ejemplo, es el test Lilliefors que es una modificación del Kolmogorov-Smirnov.

```
> lillie.test(x = yt)
```

Lilliefors (Kolmogorov-Smirnov) normality test

```
data: yt
```

```
D = 0.10444, p-value < 2.2e-16
```

Como podemos observar el p-valor es $2.2e-16 < 0.05$. Por tanto, no es posible aceptar la hipótesis nula de normalidad. Entonces, debemos intentar transformar la serie para obtener un mejor ajuste del modelo.

Para ello, se analiza la necesidad de realizar una diferenciación mediante el test de raíces unitarias de Dickey-Fuller.

```
> dfuller<-ur.df(yt,lags=0,type="trend")
```

```
> summary(dfuller)
```

```
Value of test-statistic is: -2.4327, 2.231, 3.3458
```

Critical values for test statistics:

```
1pct 5pct 10pct
```

```
tau3 -3.96 -3.41 -3.12
```

```
phi2 6.09 4.68 4.03
```

```
phi3 8.27 6.25 5.34
```

No podemos rechazar la hipótesis de raíz unitaria al ser $-2.43 > -3.41$ para el 5% de confianza.

También podemos determinar mediante la misma prueba la necesidad de una segunda diferenciación. Para ello repetimos el proceso:

```
> dfuller<-ur.df(diff(yt),lags=0,type="trend")
> summary(dfuller)
```

Value of test-statistic is: -19.0336, 120.7639, 181.1459

Critical values for test statistics:

	1pct	5pct	10pct
<i>tau3</i>	-3.96	-3.41	-3.12
<i>phi2</i>	6.09	4.68	4.03
<i>phi3</i>	8.27	6.25	5.34

En este caso, podemos rechazar la hipótesis de raíz unitaria al ser $-19.03 < -3.41$ para el 5% de confianza.

Podemos representar en un grafico la serie representada:

```
> dyt <-diff(yt)
> plot(dyt)
```

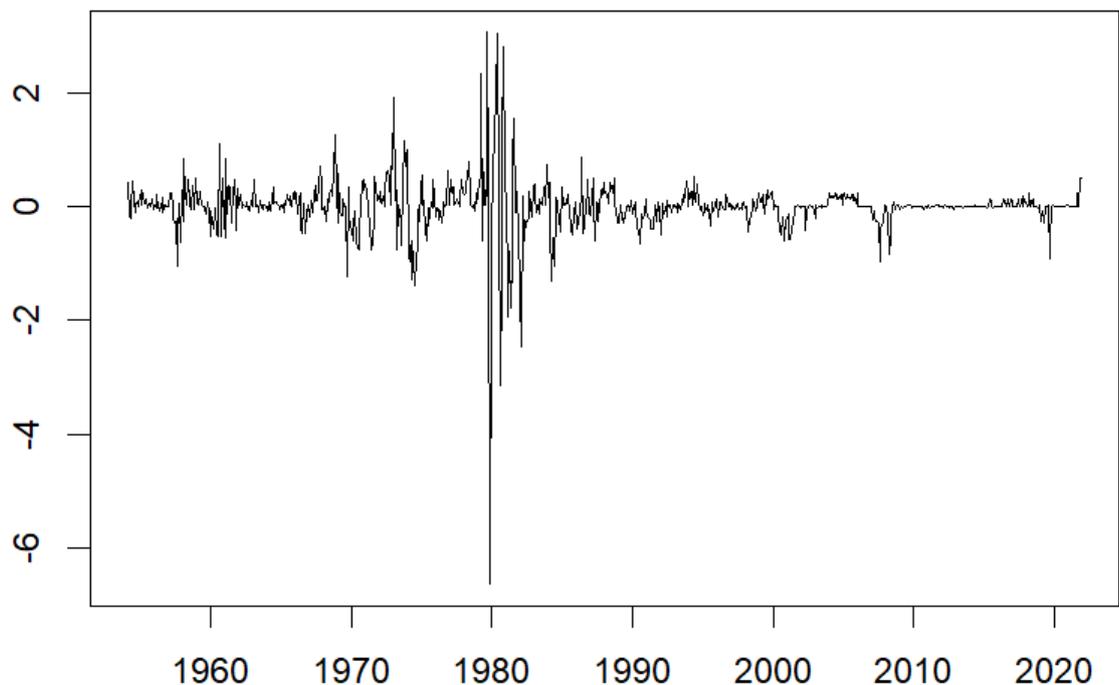


Ilustración 21 Primera diferenciada de la serie temporal

Volvemos a estudiar su estacionariedad con el test de Dickey-Fuller.

```
> adf.test(dyt)
```

Augmented Dickey-Fuller Test

```
data: dyt
Dickey-Fuller = -8.4992, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary
```

Al ser el p-valor de $0.01 < 0.05$ la serie diferenciada se considera estacionaria.

Volvemos a estudiar la normalidad, para averiguar si necesitamos realizar ajustes adicionales a la diferenciación. Para ello usamos la prueba de Lilliefors:

```
> lillie.test(x = dyt)

Lilliefors (Kolmogorov-Smirnov) normality test
```

```
data: dyt
D = 0.19791, p-value < 2.2e-16
```

Podemos observar que el p-valor es $2.2e-16 < 0.05$, por tanto, la serie temporal sigue sin ser normal. La mejor posibilidad para la normalización es la realización de la transformación Box Cox sobre la serie diferenciada. Primero, hacemos los valores positivos:

```
> ndyt <- dyt - min(dyt) + 1
> plot(ndyt)
```

Después, calculamos lambda.

```
> lambda <- BoxCox.lambda(ndyt)
> lambda
[1] -0.9999242 ≈ -1
```

Obtenemos un valor de λ de aproximadamente -1 para la serie temporal estudiada. Y se estabiliza la varianza realizando una transformación Box Cox:

```
> BCyt <- BoxCox(ndyt, -1)
> plot(BCyt)
```

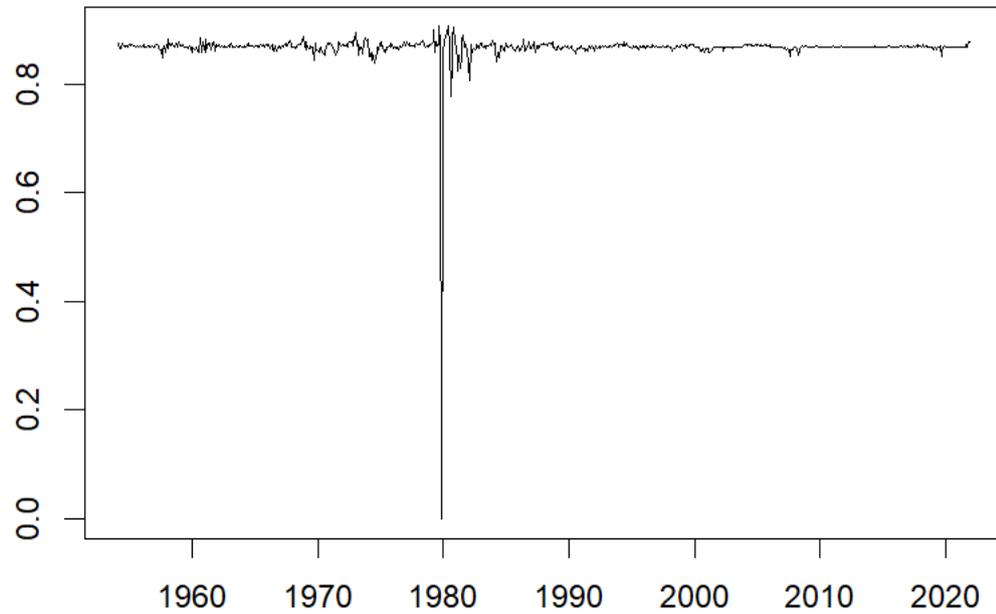


Ilustración 22 Transformada de Box-Cox de la serie diferenciada

Podemos observar la normalidad de la nueva serie en un gráfico Q-Q después de realizar la transformación.

```
> qqnorm(BCyt, pch = 19, col = "gray50")  
> qqline(BCyt)
```

Normal Q-Q Plot

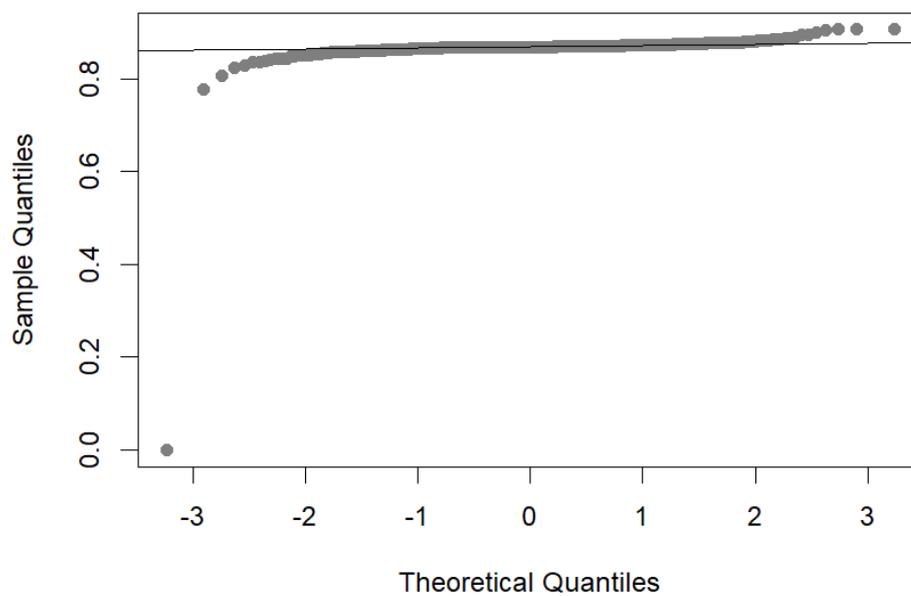


Ilustración 23 Gráfico Q-Q tras la normalización Box-Cox.

También podemos comprobarlo mediante el test de Lilliefors de forma cuantitativa.

```
> lillie.test(x = BCyt)
```

```
Lilliefors (Kolmogorov-Smirnov) normality test
```

```
data: BCyt
```

```
D = 0.3384, p-value < 2.2e-16
```

Al ser $2.2e-16 < 0.05$ se comprueba que la serie sigue sin ser normal. Pese a que se observa que los puntos pasan de no estar alineados con la línea diagonal, a estarlo mucho más tras realizar la respectiva transformación.

A simple vista, se intuye que se ha reducido la varianza. No obstante, el hecho de que la serie siga sin ser normal tras las respectivas transformaciones dificultará el ajuste del modelo y añadirá ruido al resultado final.

Se separan los datos en dos segmentos. El primero será el conjunto de entrenamiento para los modelos y el segundo será el conjunto de datos de prueba o validación.

De esta manera, nos podemos asegurar que los modelos no están sobreajustados. Ya que, una vez, realizados los ajustes los resultados se validan fuera del conjunto de entrenamiento.

En nuestro caso, utilizamos un conjunto reducido de datos para la prueba. Tres meses, pues la predicción de un modelo SARIMA pierde precisión en un horizonte temporal amplio.

```
> p80 <- round(length(yt)*0.996)
> train_bc <- BCyt[0:p80] %>% ts(frequency=frec)
> test_bc <- BCyt[(p80+1):(length(yt))] %>% ts(frequency=frec)
```

Seguiremos con la generación de las funciones de autocorrelación parcial y completa. ACF y PACF. Estas, son de utilidad para realizar el ajuste del modelo y estudiar las transformaciones necesarias. A continuación, mostramos las funciones de autocorrelación tras la transformación Box-Cox realizada:

```
> acf(BCyt,lag = 200)
```

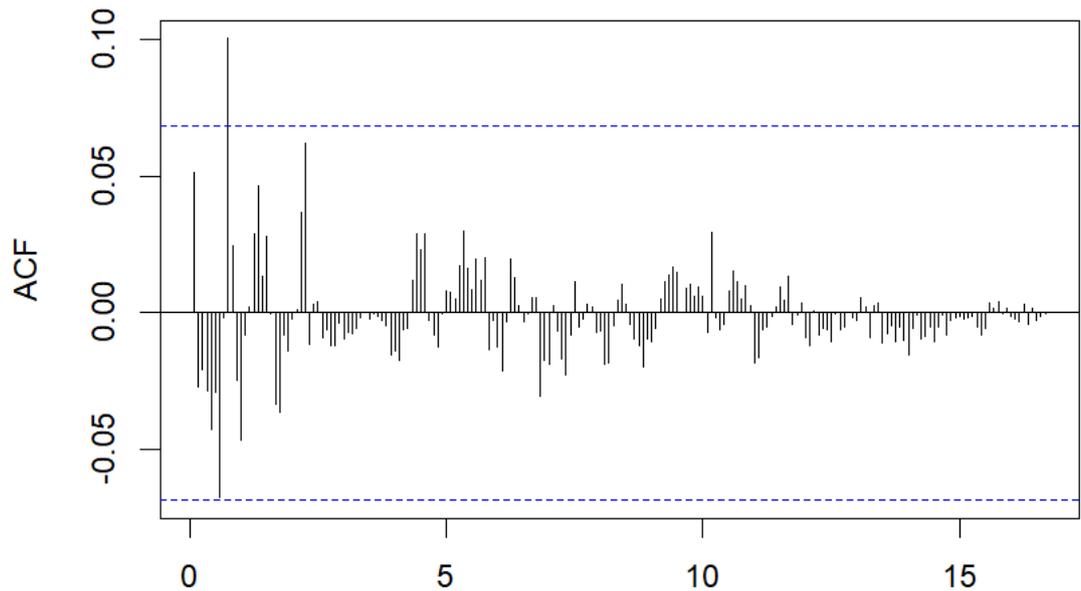


Ilustración 24 Diagrama ACF de la transformación Box-Cox.

También se puede observar el mismo gráfico ACF con menos retardos:

```
> acf(BCyt,lag = 48)
```

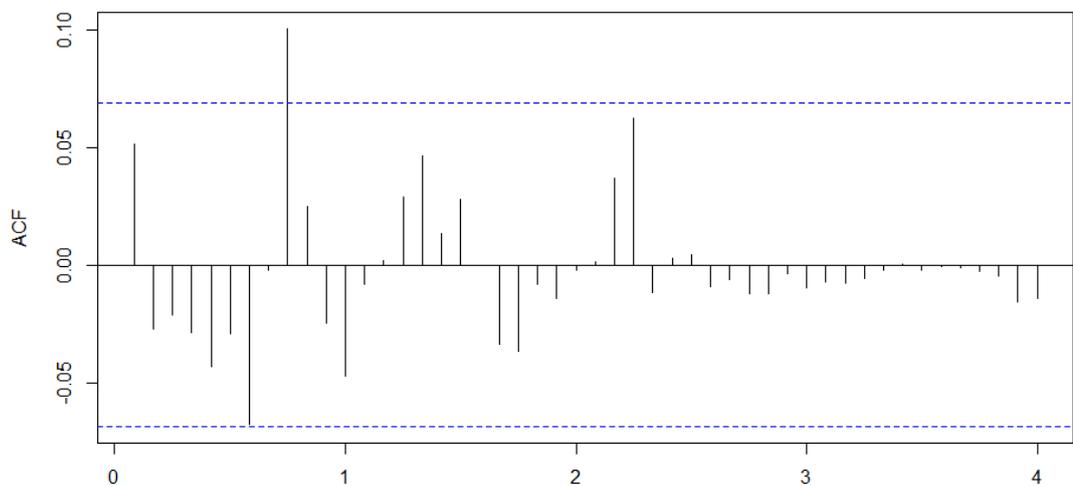


Ilustración 25 Diagrama de la transformación Box-Cox con 'lag'=48

Se puede observar que solo un valor se sitúa fuera de los márgenes de confianza en los primeros instantes en el diagrama ACF. Los retrasos se estabilizan y se sitúan dentro de los mismos en general. Por tanto, las transformaciones aplicadas parecen adecuadas.

Igualmente, para el diagrama PACF:

```
> pacf(BCyt, lag = 200)
```

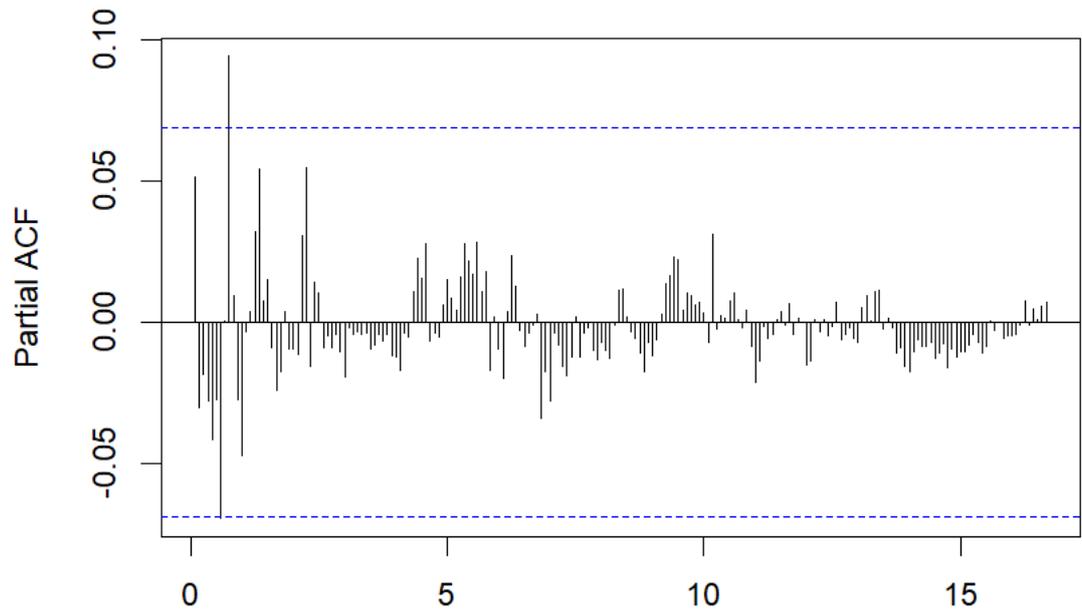


Ilustración 26 Diagrama PACF de la transformación Box-Cox.

Si utilizamos menos retardos (48):

```
> pacf(BCyt, lag = 48)
```

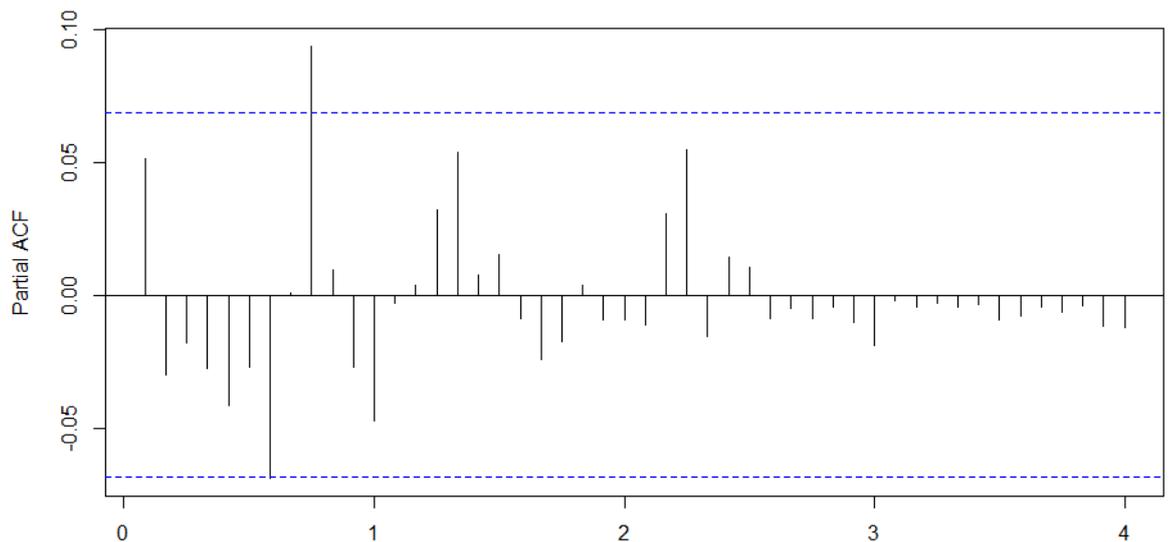


Ilustración 27 Diagrama PACF de la transformación Box-Cox con 'lag'=48

Se puede observar solo un valor se sitúa fuera de los márgenes de confianza en los primeros instantes en el diagrama PACF. Después los retrasos se estabilizan y se sitúan dentro de los márgenes de confianza, por norma general. Por tanto, las transformaciones aplicadas parecen adecuadas también en este diagrama.

Es importante recordar que el diagrama PACF ayudará a estimar la parte MA del modelo y el ACF la parte AR del mismo.

2.3. Estimación del modelo: parámetros, significación y correlación

Se eligen los parámetros del modelo SARIMA teniendo en cuenta las pruebas y transformaciones presentadas en la sección anterior y los resultados de los diagramas ACF, PACF. No obstante, son necesarias varias iteraciones siguiendo un proceso de prueba y error.

En primer lugar, determinamos el ciclo medio de la serie temporal con la función de R:

```
> findfrequency(yt)
[1] 16
```

Con esta función la frecuencia dominante se determina a partir de un análisis espectral de la serie temporal. En primer lugar, se elimina la tendencia lineal y, a continuación, se estima la función de densidad espectral a partir del modelo autorregresivo que mejor se ajuste (según el AIC).

A priori, una frecuencia de 16, resulta un valor poco razonable. Puesto que representa un año y un cuatrimestre. Aunque ésta, será comparada con otras. Puesto que el criterio AIC, no siempre es acertado, como se discute en la sección 1.9.

Se realiza una primera iteración, del que llamaremos modelo A, con la solución propuesta por la función en R `auto.arima()`. En la que se incluye el parámetro 'seasonal=TRUE' para que utilice un modelo estacional. El mejor resultado propuesto por la función es:

```
> fit <- auto.arima(ts(train_bc,frequency=16),trace=TRUE, d=1, D=0)
ARIMA(5,1,0)(0,0,1)[16] : -3166.152
```

Es decir, el modelo A es:

$$ARIMA(5,1,0) \times ARIMA(0,0,1)[9]$$

En primer lugar, observamos el ajuste:

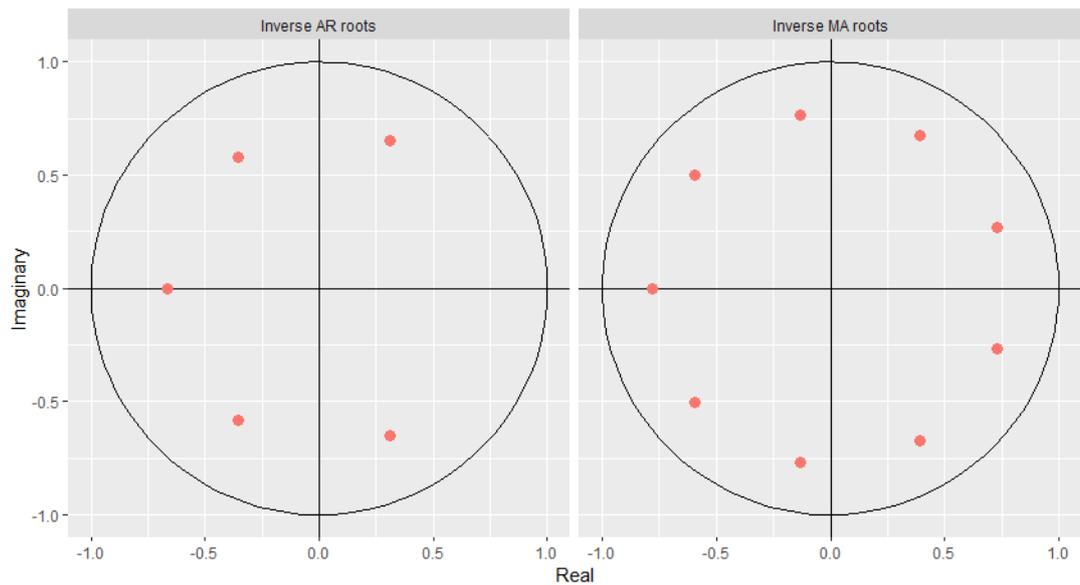


Ilustración 28 Ajuste del modelo generado por `auto.arima()`

Parece correcto porque los puntos se sitúan dentro del área del círculo. Seguimos con el estudio del modelo.

Se puede analizar el siguiente resumen de sus valores:

```
> summary(fit)
```

```
Series: ts(train_bc, frequency = 16)
ARIMA(5,1,0)(0,0,1)[16]
```

Coefficients:

```
   ar1   ar2   ar3   ar4   ar5   sma1
-0.7585 -0.6150 -0.4556 -0.3011 -0.1634 0.0507
s.e. 0.0347 0.0425 0.0448 0.0423 0.0346 0.0352
```

```
sigma^2 estimated as 0.001184: log likelihood=1586.48
AIC=-3158.95 AICc=-3158.81 BIC=-3126.06
```

Training set error measures:

ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
- 9.0112e-06	0.03425491	0.006001837	Inf	Inf	0.7442305	-0.0179

Tabla 4 Errores del ajuste SARIMA del modelo A

Podemos ver que los errores MPE y MAPE tienen valor infinito. Esto es un claro indicativo de que el modelo está sobreajustado. Y es un motivo suficiente para descartar el modelo.

No obstante, procedemos con el análisis de significancia de los parámetros.

```
> coeftest(fit)
```

z test of coefficients:

```

      Estimate Std. Error z value Pr(>|z|)
ar1 -0.758549  0.034683 -21.8708 < 2.2e-16 ***
ar2 -0.615043  0.042483 -14.4774 < 2.2e-16 ***
ar3 -0.455554  0.044811 -10.1662 < 2.2e-16 ***
ar4 -0.301142  0.042281  -7.1224 1.061e-12 ***
ar5 -0.163357  0.034554  -4.7276 2.272e-06 ***
sma1 0.050681  0.035226  1.4388  0.1502
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Podemos ver que la 'sma1' no es relevante para el modelo. Por tanto, debemos descartar este ajuste. Por otro lado, esto puede indicar, que la frecuencia seleccionada por la función descrita anteriormente no sea correcta.

Como mencionado anteriormente, los resultados del ajuste del modelo A con esta función resultan poco prometedores. Dado que el ajuste no es bueno, analizando en conjunto los parámetros, significación y correlación.

La función de R '*auto.arima()*' tiene limitaciones, debido a la insuficiencia del criterio de elección de Akaike, que se ha discutido en la sección 1.9. Además, tiende a seleccionar modelos sobreajustados.

Seguimos el estudio con otro modelo, al que llamaremos modelo B. Este consiste, en un modelo simple que también utiliza las transformaciones realizadas anteriormente, pero con una frecuencia mayor (40). Siendo equivalente a 3 años y un cuatrimestre.

Se utiliza esta frecuencia por motivos teóricos. Ya, que se supone, que es la duración media de un ciclo económico corto. Bajo este pretexto, el modelo B sobre los datos transformados es $ARIMA(0,0,0)(0,0,1)[40]$:

$$ARIMA(0,0,0) \times ARIMA(0,0,1)[40]$$

En primer lugar, observamos el ajuste:

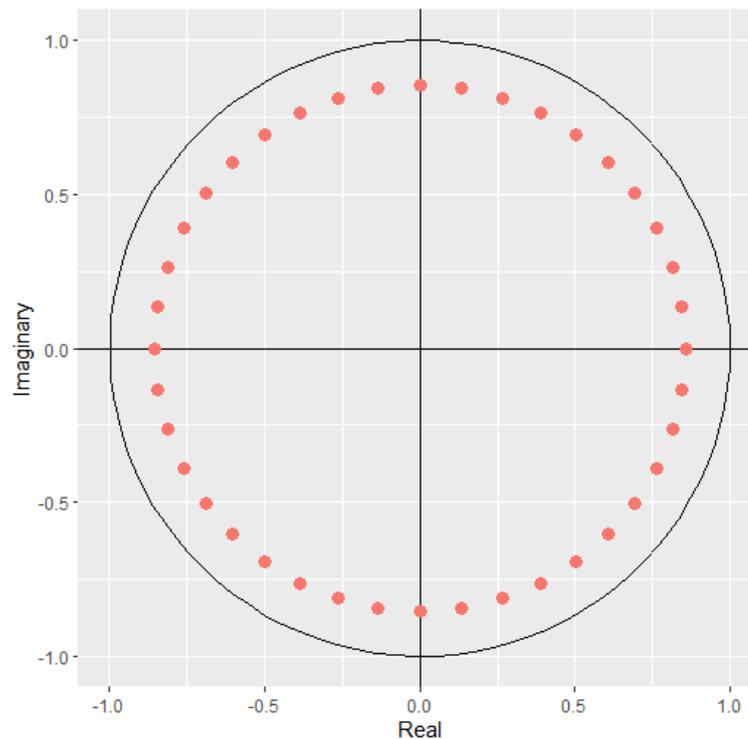


Ilustración 29 Ajuste del modelo B

Como parece correcto porque los puntos se sitúan dentro de la circunferencia, seguimos con el estudio del modelo.

Se puede analizar el siguiente resumen de sus valores:

```
> summary(fit)
```

```
Series: ts(train_bc, frequency = 40)
ARIMA(0,0,0)(0,0,1)[40] with non-zero mean
```

Coefficients:

```
      sma1  mean
-0.0019  0.8676
s.e.  0.0341  0.0011
```

```
sigma^2 estimated as 0.0009904: log likelihood=1659.31
AIC=-3312.62  AICc=-3312.59  BIC=-3298.52
```

Training set error measures:

ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
2.4507e-08	0.0314326	0.005493184	Inf	Inf	0.6282684	0.05132196

Tabla 5 Errores del ajuste SARIMA del modelo A

Podemos ver que los errores MPE y MAPE tienen valor infinito. Esto es un claro indicativo de que el modelo está sobreajustado. Y es un motivo suficiente para descartar el modelo.

No obstante, comprobamos si los parámetros son significativos:

```
> coeftest(fit)
```

z test of coefficients:

```
Estimate Std. Error z value Pr(>|z|)
sma1     -0.0019487 0.0340877 -0.0572 0.9544
intercept 0.8676187 0.0011009 788.0879 <2e-16 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Se observa que el parámetro 'sma1' no es significativo de acuerdo a la leyenda. Por tanto, el modelo es definitivamente descartado.

Tras realizar diversas iteraciones del modelo, con diferentes frecuencias, se obtiene el mejor resultado con ciclos más cortos; de tres trimestres o nueve meses. Así describimos a continuación un modelo C que conlleva el siguiente ajuste:

```
> fit<- arima(train_bc,order=c(0,0,0),seasonal = list(order=c(0,0,1),period=9))
```

ARIMA(0,0,0)(0,0,1)[9].

Así, el modelo propuesto es el:

$$ARIMA(0,0,0) \times ARIMA(0,0,1)[9] \quad (19)$$

Podemos ver como se ajusta dicho modelo en la imagen inferior:

```
> autoplot(fit)
```

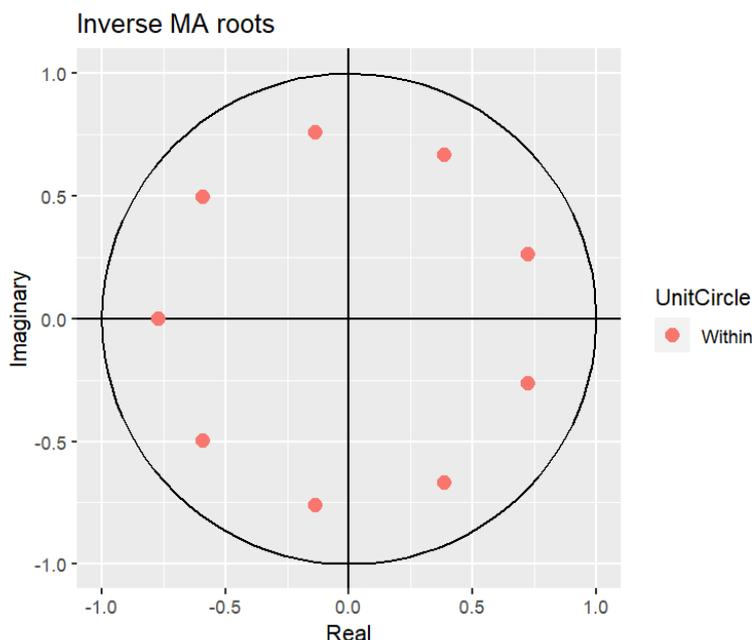


Ilustración 30 Ajuste del modelo SARIMA

Al estar los valores dentro de las fronteras de la circunferencia es un indicativo de que el modelo se ajusta correctamente. Tanto en su parte AR, como en su parte MA.

Se puede analizar el siguiente resumen de sus valores:

```
> summary(fit)
```

Call:

```
arima(x = train_bc, order = c(0, 0, 0), seasonal = list(order = c(0, 0, 1),
  period = 9))
```

Coefficients:

```
      sma1 intercept
      0.0968   0.8676
s.e. 0.0344   0.0012
```

sigma^2 estimated as 0.0009784: log likelihood = 1663.26, aic = -3322.51

Training set error measures:

ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
0.001462805	0.004985828	0.003980084	0.2231935	0.3210954	0.00398008	-0.10872

Tabla 6 Errores del ajuste SARIMA

Donde cabe destacar, los coeficientes del modelo y sus valores de AIC, AICc y BIC. Que han servido, para comparar el ajuste de los parámetros con otras opciones. Estos factores, deben ser minimizados. Y se han tenido en cuenta de cara a la elección del ajuste.

Por otro lado, se presenta la significación de los parámetros:

```
> coeftest(fit)
```

z test of coefficients:

```
      Estimate Std. Error   z value Pr(>|z|)
sma1    0.0967870 0.0344066   2.813 0.004908 **
intercept 0.8676206 0.0012025 721.510 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Se aprecia que todos los parámetros usados son significativos de acuerdo con la leyenda inferior. Esto es importante, para evaluar la salud del modelo. Pues aquellos parámetros no significativos son susceptibles de ser eliminados. Este, es otro de los factores que se han tenido en cuenta en la elección del modelo.

En la siguiente sección (2.4) se validará el modelo C mediante el estudio de sus residuos. Si el análisis de los mismos, descarta el modelo propuesto, deberemos seguir iterando con otros ajustes, o plantear el problema de una manera distinta.

De lo contrario, si el modelo C es válido, continuaremos con las predicciones del mismo, y el análisis de los resultados.

2.4. Validación del modelo: estudio de los residuos

Una vez estimado el modelo, debemos validarlo con el estudio de sus residuos. Para ello, generamos un diagrama con los mismos:

```
> checkresiduals(fit, lag = 20)
```

Ljung-Box test

data: Residuals from ARIMA(0,0,0)(0,0,1)[9] with non-zero mean
Q* = 16.2, df = 18, p-value = 0.5786

Model df: 2. Total lags used: 20

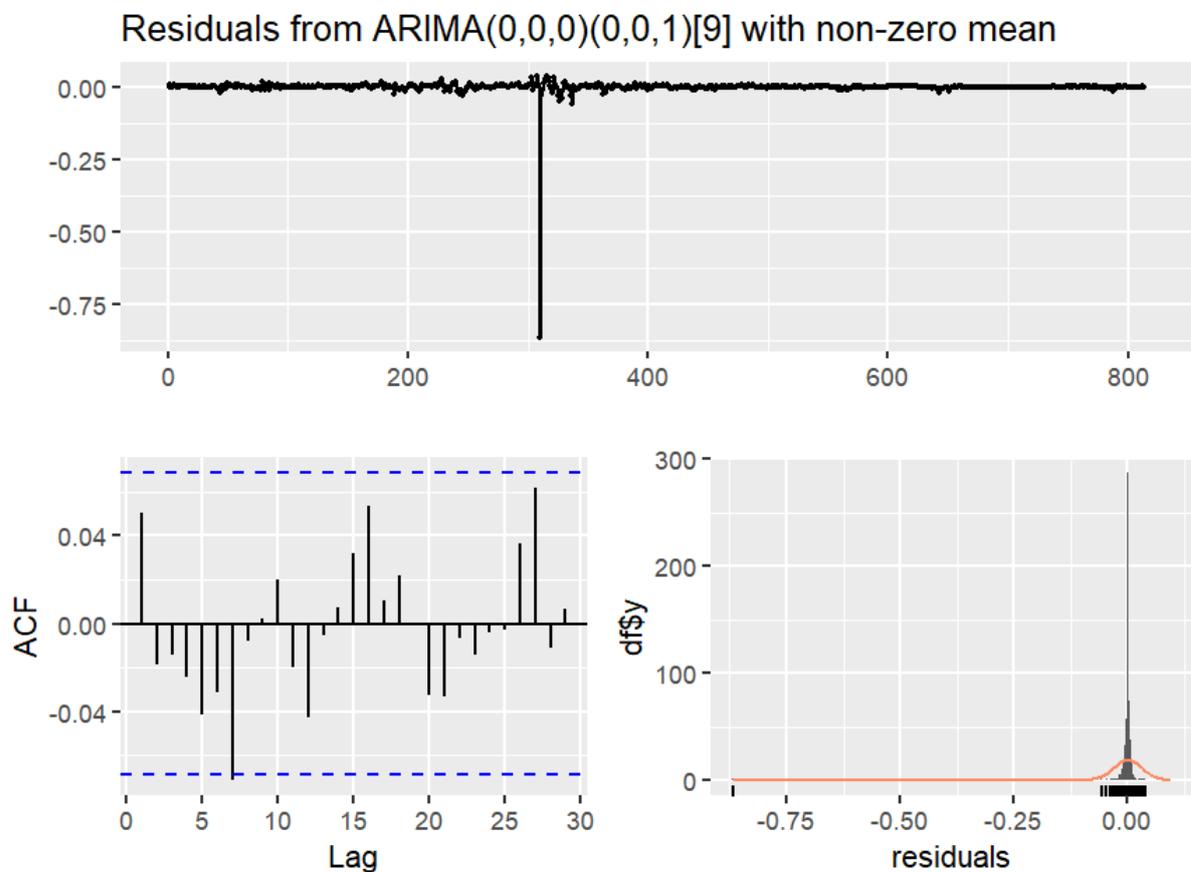


Ilustración 31 Residuos del ajuste SARIMA realizado

Podemos ver como los residuos se encuentran dentro de los intervalos de confianza en el diagrama ACF, salvo por un instante. La distribución parece sesgada, pero

es una consecuencia de la asimetría de los ciclos debido a la naturaleza económica de los mismos y la dificultad de normalizar la distribución existente.

De hecho, cuando se realiza el test de Ljung-Box sobre los residuos para validar la hipótesis nula de independencia obtenemos un p-valor de 0.5786 superior a 0.05, es decir, se considera que no sigue existiendo correlación que no sea fruto del azar.

Por tanto, el ajuste del modelo parece suficientemente positivo para seguir el análisis y se evaluarán sus resultados en las siguientes secciones. Vamos a mostrar también los diagramas ACF y PACF de los residuos del modelo ajustado:

```
> acf(fit$residuals, lag=48)
```

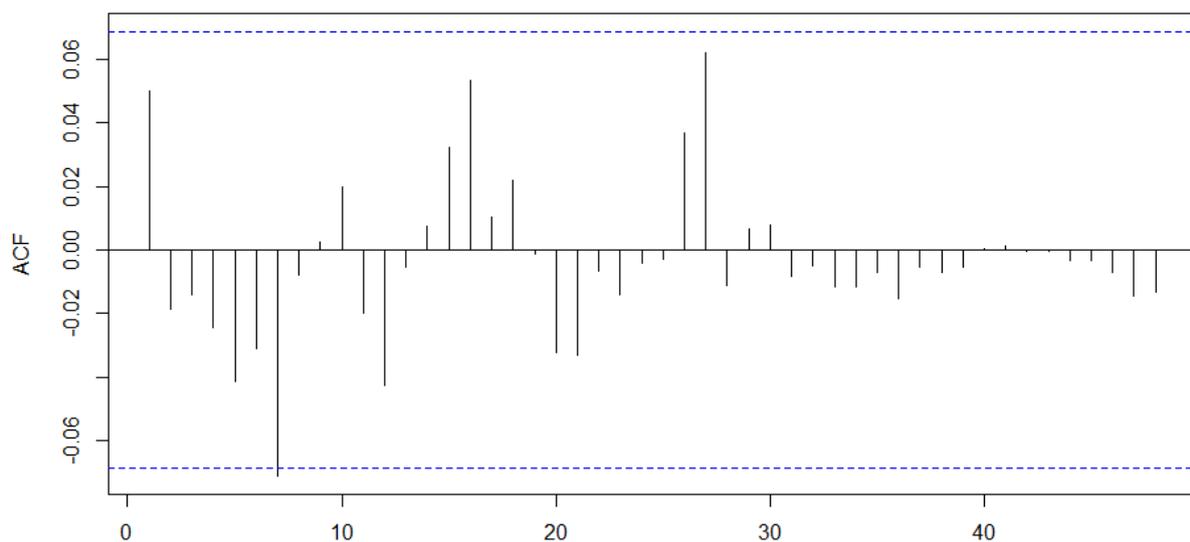


Ilustración 32 Diagrama ACF de los residuos del modelo SARIMA

```
> pacf(fit$residuals, lag=48)
```

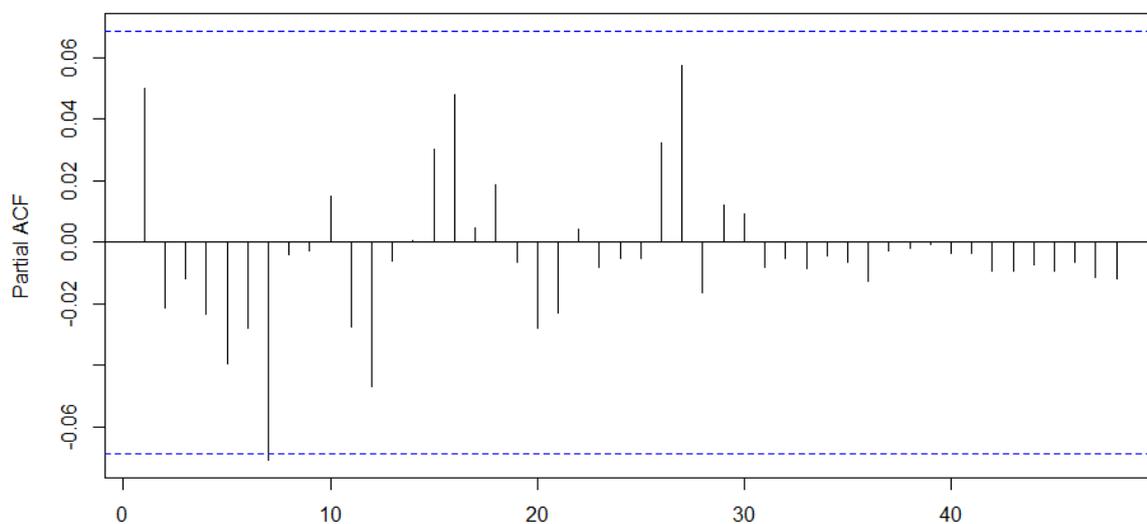


Ilustración 33 Diagrama PACF de los residuos del modelo SARIMA

Se puede ver como en el caso del PACF también se encuentran dentro de los intervalos propuestos salvo en un caso. Se considera suficientemente válido ya que los datos son de un caso práctico real, que por la naturaleza de sus datos es difícil de normalizar completamente.

Se han realizado multitud de iteraciones, más allá de los modelos 'A', 'B' y 'C'. Siendo el mejor resultado el propuesto por el último modelo.

2.5. Predicción del modelo SARIMA

Para ver los resultados del modelo, debemos deshacer la transformación realizada con anterioridad. El proceso se lleva a cabo en varios pasos. Primero deshacemos la transformación de Box-Cox:

```
> invBoxCox <- function(x, lambda){
+   if (lambda == 0) exp(x) else (lambda*x + 1)^(1/lambda)
+ }
```

Después se deshace el cambio de signo y diferencia:

```
> predBCp <- predBC + min(dyt) - 1
```

Por último, se agrupan los datos observados con las predicciones mediante:

```
> a <- 0
> horizonte <- length(test)
> ll <- length(yt)
> for (i in 1:ll){ a[i] <- yt[i] }
> for (j in 1:horizonte){ a[ll+j] <- predBCp[j] + a[ll+j-1] }
> a <- a[1:horizonte] %>% ts(frequency=1)
```

Una vez desecha la transformación se muestran los resultados de la predicción del modelo SARIMA. Los datos completos, se recogen en la siguiente tabla de resultados:

```
> df_sarima <- data_frame(Actual = test %>% as.vector(), Predicted = a %>%
round(2), Error = Predicted - Actual, Error_Percent = round(Error / Actual, 2))
> df_sarima %>% knitr::kable()
```

Actual	Predicted	Error
0.33	0.80	0.47
0.83	1.22	0.39
1.33	1.07	-0.26

Tabla 7 Errores del ajuste SARIMA

El error es relativamente grande, pero no lo es en términos absolutos. El modelo parece estimar de forma razonable los siguientes tres meses.

Podemos observarlo en las siguientes ilustraciones. Los valores pronosticados se encuentran entre los máximos y mínimos del valor real. No obstante, la predicción es mejorable.

```
> autoplot(predicted_sarima_ts)
```

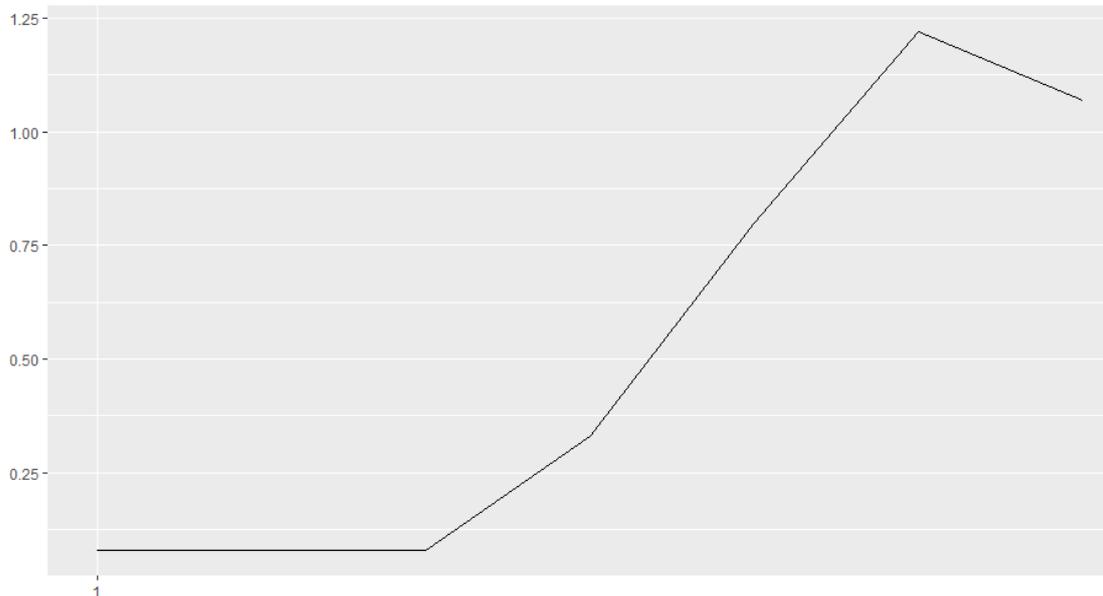


Ilustración 34 Predicción del modelo SARIMA para los valores de prueba

```
> x <- cbind(seq(1,length(yt)-length(train)))
> y <- cbind(df_sarima$Actual, df_sarima$Predicted)
> matplot(x,y,type="l")
> legend("topleft",col = 1:2, lty = 1:2, legend = colnames(df_sarima)[1:2])
> x <- cbind(seq(1,length(yt)-length(train)))
> y <- cbind(df_sarima$Actual, df_sarima$Predicted)
> matplot(x,y,type="l")
> legend("topleft",col = 1:2, lty = 1:2, legend = colnames(df_sarima)[1:2])
```

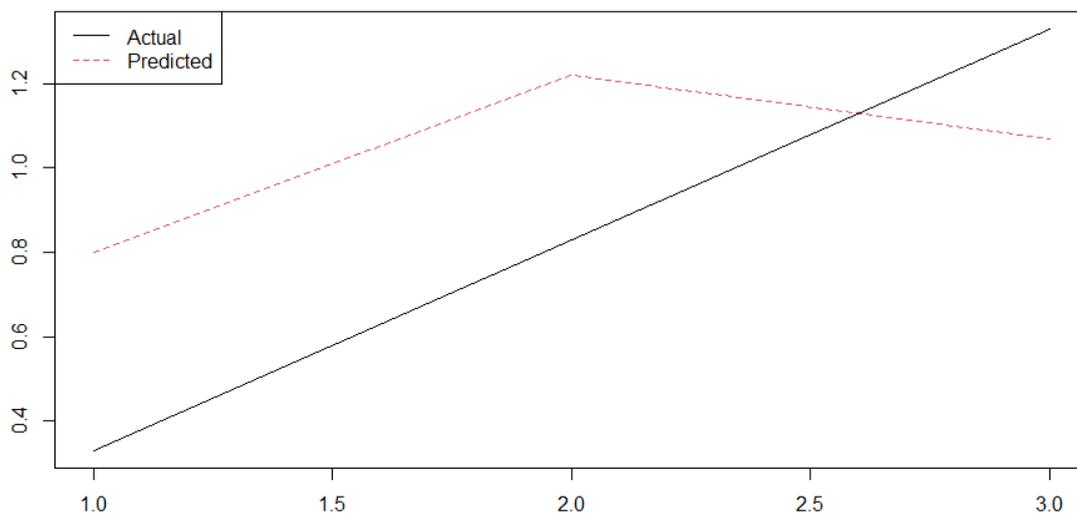


Ilustración 35 Predicción del modelo SARIMA en rojo versus valor real en negro

Como hemos comentado anteriormente se aprecia que la predicción es buena, pero mejorable y por eso se plantea un modelo mixto con redes neuronales en la siguiente sección (2.6).

2.6. Predicción del modelo híbrido SARIMA+NN

En primer lugar, generamos una red neuronal básica de perceptrón multicapa con los datos de entrenamiento. Se realiza de manera automática con la siguiente función de R de la librería ‘nnfor’:

```
> mlp_std <- mlp(train, hd = 5, difforder = 0)
> plot(mlp_std)
```

Siendo ‘train’ el conjunto de datos de entrenamiento sin transformaciones previas. Dado que no hace falta normalizar o reducir la varianza para la red neuronal.

Cuyo diagrama básico podemos observar en la siguiente imagen:

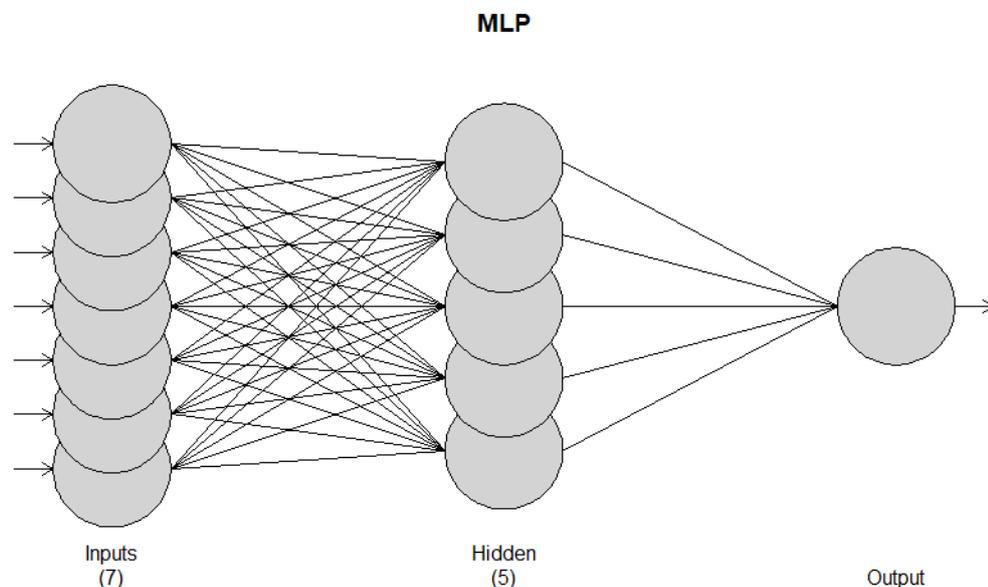


Ilustración 36 Red neuronal de perceptrón multicapa

Los ‘inputs’ en este caso son siete. Reflejan el número de retardos de entrada usados para construir la red neuronal. Esto es, se utiliza la misma serie temporal como valor de entrada con distintos valores de retardo. En este caso se utilizan los siguientes:

```
> mlp_std[["lags"]]
[1] 1 2 5 7 8 10 12
```

Es decir, cada valor de retardo construye una serie temporal, intercalados desde uno hasta llegar a la frecuencia máxima (12). Este vector de vectores es calculado de forma automática por la función ‘mlp()’ de la librería ‘nnfor’.

Estas entradas generarán una combinación distinta para cada una de las neuronas de la capa oculta. Para la que se elige un parámetro 'hd = 5'. No es calculado automáticamente. Tras varias iteraciones, de forma empírica, resulta el valor más apropiado de nodos de la red oculta.

Es una red sencilla, con un bajo número de nodos, que evita el sobreajuste del problema tratado. Tendrá una sola capa intermedia con cinco neuronas como se observa en el diagrama superior.

El output será el vector resultante del ajuste de la red, que se representa a continuación:

```
> predicted_mlp_std <- forecast(mlp_std, h=length(test), model=mlp_std)$mean
%>% as.vector()
> predicted_mlp_std_ts <- ts(c(train[(length(train)-
length(predicted_mlp_std)):length(train)],predicted_mlp_std), frequency = frec)
> autoplot(predicted_mlp_std_ts)
```

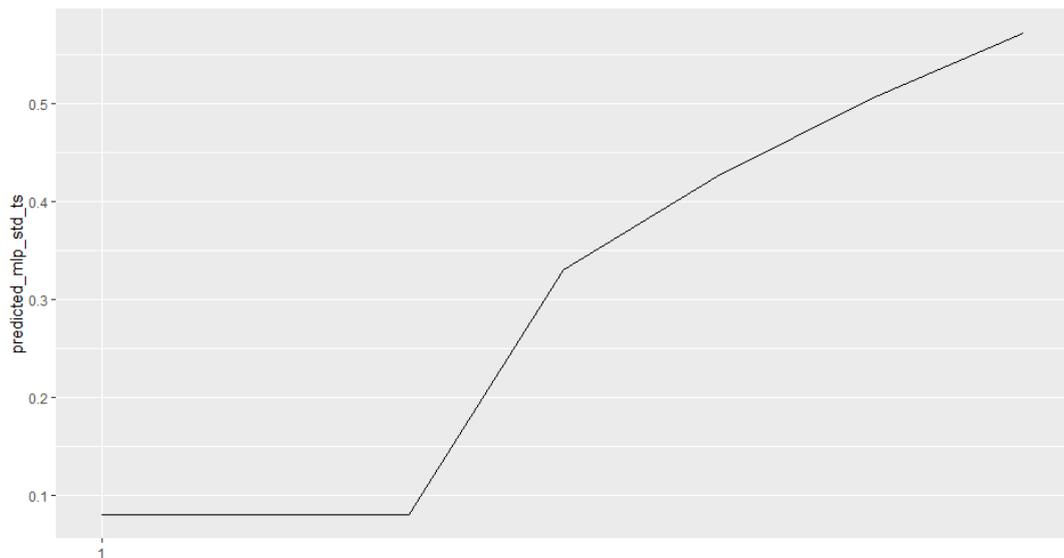


Ilustración 37 Predicción de la red neuronal básica (datos de entrenamiento)

Presentamos estos datos en forma de tabla:

```
> df_mlp_std <- data_frame(Actual = test %>% as.vector(), Predicted =
predicted_mlp_std %>% round(2), Error = Predicted - Actual, Error_Percent =
round(Error / Actual, 2))
> df_mlp_std %>% knitr::kable()
```

Actual	Predicted	Error
0.33	0.43	0.30
0.83	-0.33	-0.40
1.33	-0.76	-0.57

Tabla 8 Errores del ajuste de la red neuronal (datos de entrenamiento)

En segundo lugar, probamos otro modelo de red neuronal pero alimentada con los residuos del modelo SARIMA (C), en vez de los datos de entrenamiento.

```
> mlp_residuals <- mlp(previous_residuals,hd = 5, difforder = 0)
> plot(mlp_residuals)
```

Siendo ‘previous_residuals’ los residuos del modelo SARIMA que hemos definido anteriormente. Tendrá una capa intermedia con cinco neuronas cuyo diagrama básico podemos observar en la siguiente imagen:

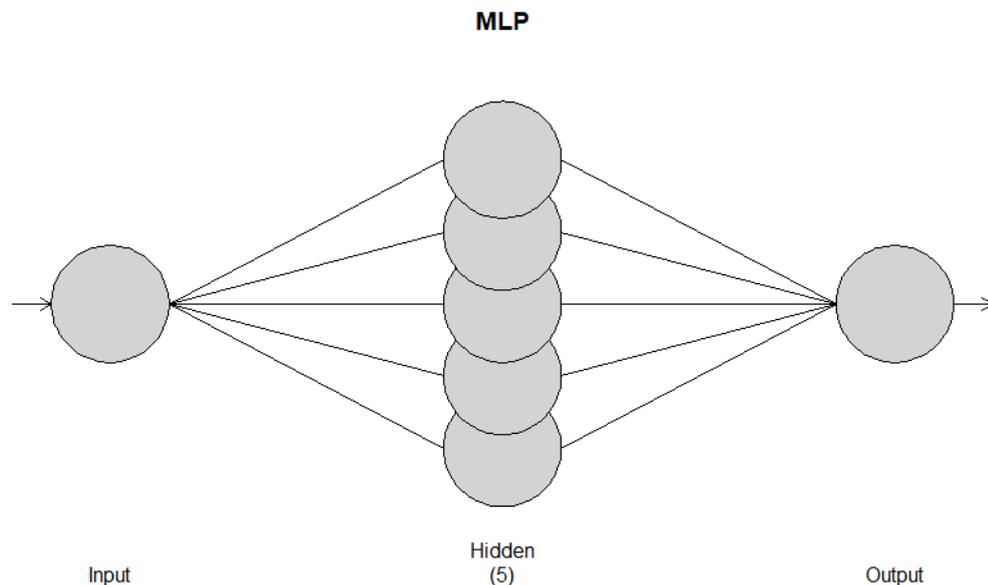


Ilustración 38 Diagrama de la red neuronal básica (datos de los residuos).

El input de esta red neuronal será una única serie temporal con siete retrasos, como se comprueba a continuación:

```
> mlp_residuals[["lags"]]
[1] 7
```

También se elige para esta red un parámetro ‘hd = 5’ que no es calculado automáticamente por la función, porque resulta el valor más apropiado de la red oculta tras varias iteraciones, realizadas de forma empírica.

En este caso, también es una red simple, con un bajo número de nodos, que evita el sobreajuste del problema tratado.

El output será el vector resultante del ajuste de la red, que se representa a continuación:

```
> predicted_residuals <-
forecast(mlp_residuals, h=length(test), model=mlp_residuals)$mean %>% as.vector()
> predicted_residuals_ts <- ts(c(previous_residuals[(length(previous_residuals)-
length(test)):length(previous_residuals)],predicted_residuals), frequency = frec)
> autoplot(predicted_residuals_ts)
```

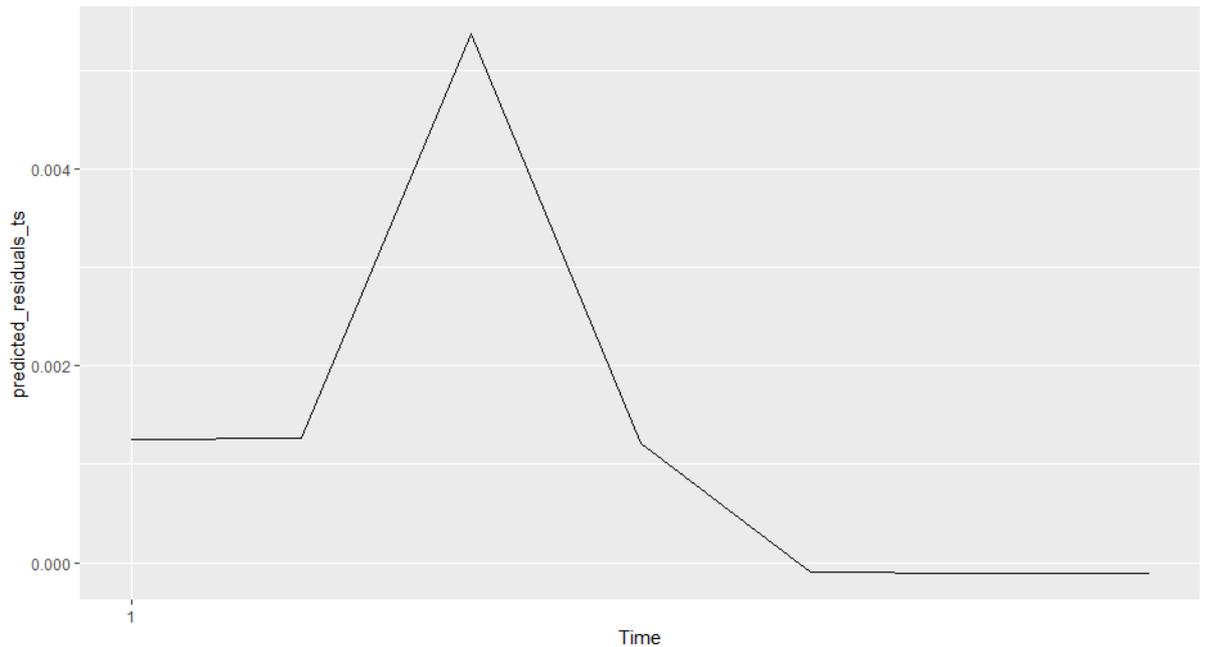


Ilustración 39 Predicción de la red neuronal básica (datos de los residuos).

Podemos ver como la curva es diferente, pero además los valores son muy inferiores al tratarse de una predicción de los residuos, que representan un porcentaje muy pequeño del subtotal.

Se presentan los resultados en forma de tabla:

```
> df_mlp_res <- data_frame(Actual = test %>% as.vector(), Predicted =
predicted_residuals %>% round(2), Error = Predicted - Actual, Error_Percent =
round(Error / Actual, 2))
> df_mlp_res %>% knitr::kable()
```

Actual	Predicted	Error
0.33	≈ 0	-0.33
0.83	≈ 0	-0.83
1.33	≈ 0	-1.33

Tabla 9 Errores del ajuste de la red neuronal (datos de los residuos).

Como se mencionaba anteriormente, este ajuste no tiene poder predictivo, al tratarse de valores muy pequeños en comparación. Pero tendrá sentido, como parte de un modelo conjunto.

A continuación, se muestra el resultado del ajuste del primer modelo combinado. Donde la predicción del modelo SARIMA, se combina con la predicción de la red neuronal (con datos de entrenamiento).

Generamos el modelo conjunto y observamos la predicción:

```
> mlp_ensemble_std <- (predicted_sarima_ts+predicted_mlp_std_ts)/2
> plot(mlp_ensemble_std)
```

```
> mlp_predicted_std <- mlp_ensemble_std[(1+length(mlp_ensemble_std)-
length(test)):length(mlp_ensemble_std)] %>% ts(frequency=1)
```

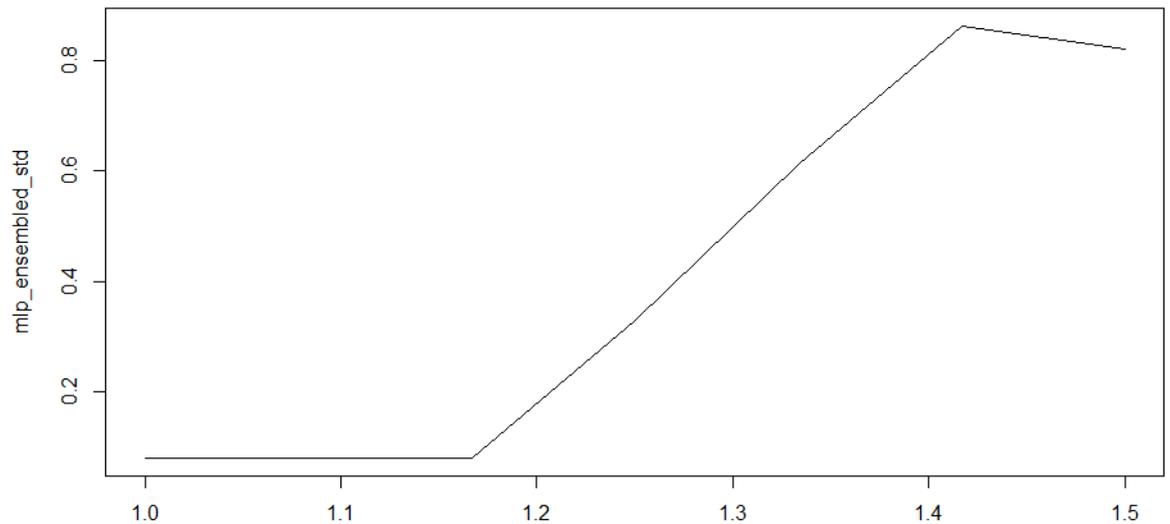


Ilustración 40 Predicción del modelo conjunto SARIMA y red neuronal (con datos de entrenamiento)

Recogemos estos resultados en forma de tabla y medimos el error:

```
> df_ensemble_std <- data_frame(Actual = test %>% as.vector(), Predicted =
mlp_predicted_std %>% round(2), Error = Predicted - Actual, Error_Percent =
round(Error / Actual, 2))
> df_ensemble_std %>% knitr::kable()
```

Actual	Predicted	Error
0.33	0.61	0.28
0.83	0.86	0.03
1.33	0.82	-0.51

Tabla 10 Errores del ajuste del modelo conjunto SARIMA y red neuronal (con datos de entrenamiento)

Para continuar, calculamos el segundo modelo conjunto, donde se combinan las predicciones del modelo SARIMA con las predicciones de la red neuronal (con los datos de los residuos). Con la diferencia, de que los pesos no se reparten equitativamente.

Daremos, una importancia mayor a la predicción de los residuos para compensar que los valores son menores. Estos pesos, se calculan según la importancia de los residuos según la descomposición de la serie temporal:

```
> y2 <- decompose(yt)
> plot(y2,xaxt = "n")
```

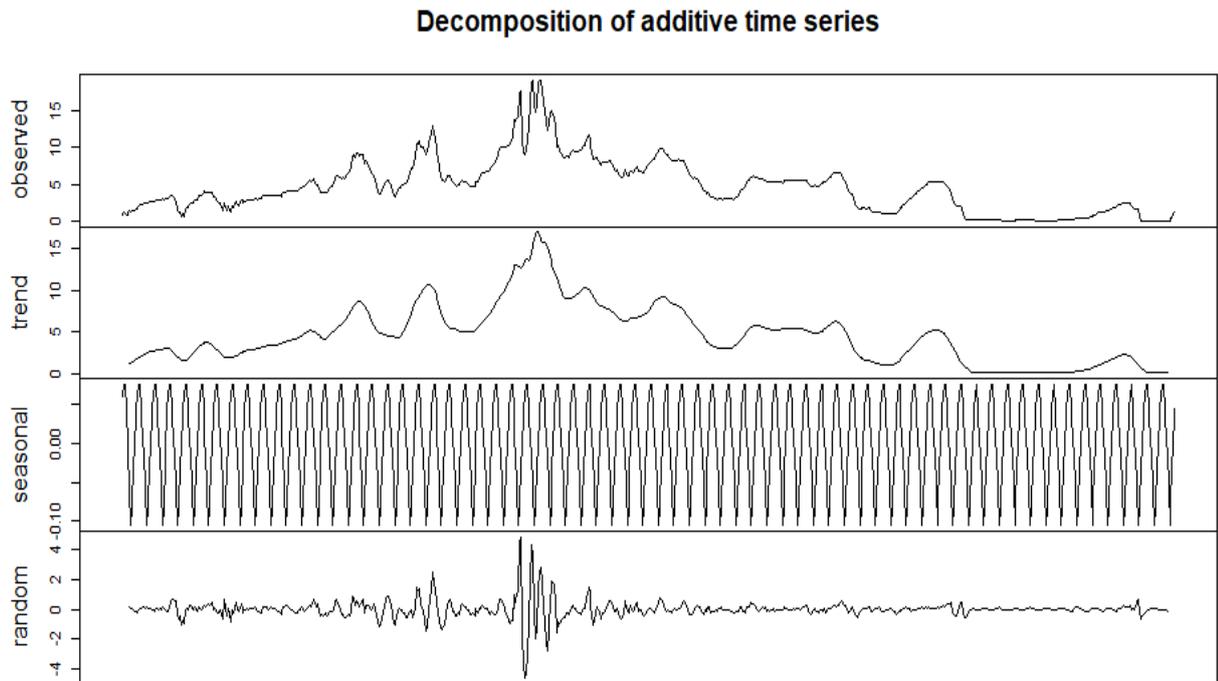


Ilustración 41 Descomposición de la serie temporal estudiada

Se observa que el ruido representa un 21% del total. Por tanto, realizamos el siguiente ajuste para el modelo conjunto:

```
> mlp_res <- (predicted_residuals_ts*0.21+predicted_sarima_ts*0.79)
> plot(mlp_res)
```

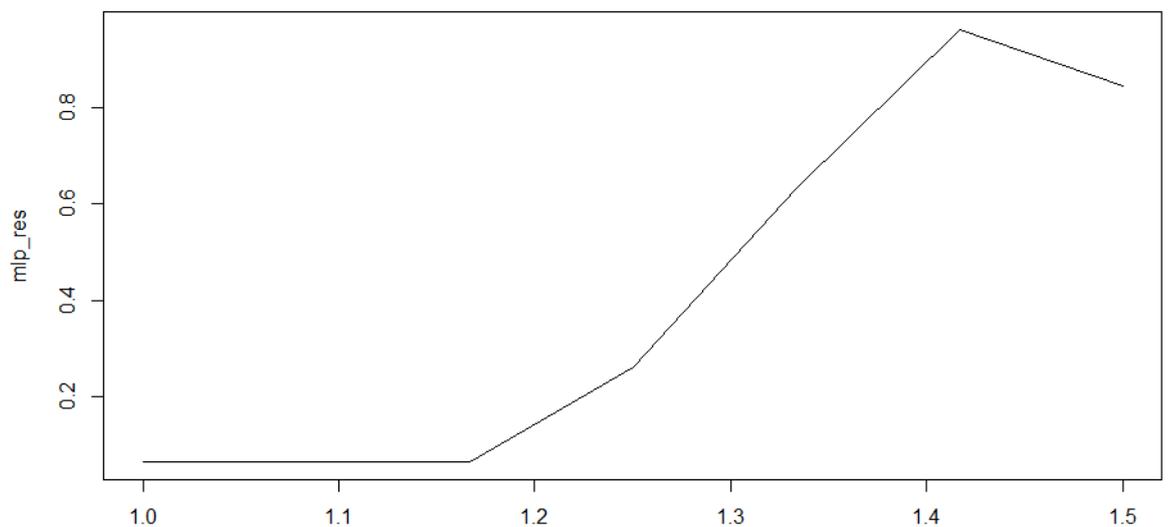


Ilustración 42 Predicción del modelo conjunto SARIMA y red neuronal (con datos de los residuos)

También recogemos sus resultados en una tabla:

```
> mlp_predicted_res <-
mlp_res[(1+length(mlp_res)-length(test)):length(mlp_res)] %>% ts(frequency=1)
```

```
> df_ensemble_res <- data_frame(Actual = test %>% as.vector(), Predicted =
mlp_predicted_res %>% round(2), Error = Predicted - Actual, Error_Percent =
round(Error / Actual, 2))
> df_ensemble_res %>% knitr::kable()
```

Actual	Predicted	Error
0.33	0.63	0.30
0.83	0.96	0.13
1.33	0.85	-0.48

Tabla 11 Errores del ajuste del modelo conjunto SARIMA y red neuronal (con datos de entrenamiento)

Existe la posibilidad de crear un tercer modelo conjunto, que combine los dos modelos conjuntos anteriores. En concreto dos redes neuronales (datos de entrenamiento y residuos del SARIMA) con las predicciones del SARIMA:

```
> mlp_res_train <-
(predicted_residuals_ts*0.21+(predicted_sarima_ts+predicted_mlp_std_ts)/2*0.7)
> plot(mlp_res_train)
```

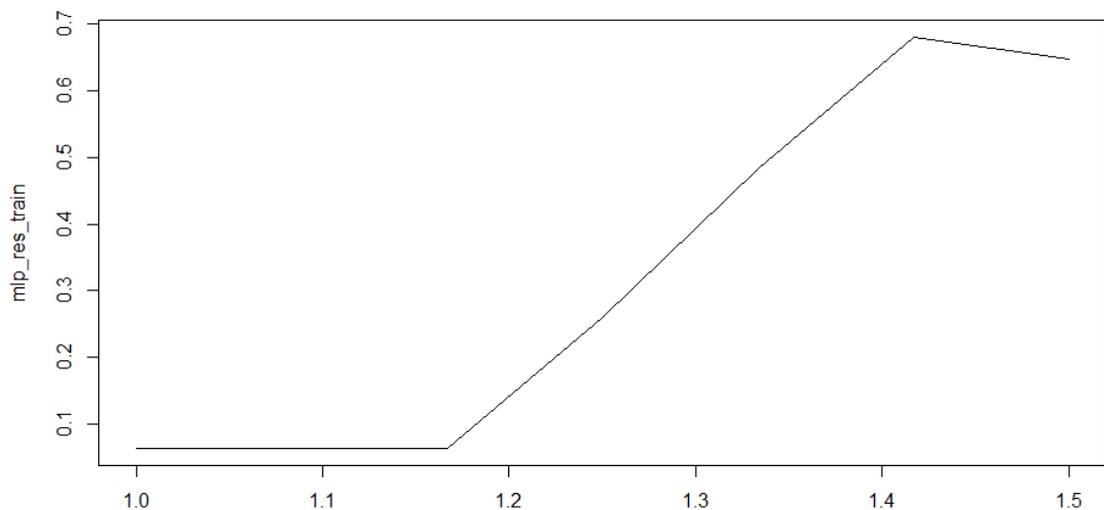


Ilustración 43 Predicción del modelo conjunto SARIMA y ambas redes neuronales

Recogemos sus datos en una tabla:

```
> mlp_predicted_res_train <-
mlp_res_train[(1+length(mlp_res_train)-length(test)):length(mlp_res_train)] %>%
ts(frequency=1)
> df_ensemble_res_train <- data_frame(Actual = test %>% as.vector(), Predicted
= mlp_predicted_res_train %>% round(2), Error = Predicted - Actual, Error_Percent
= round(Error / Actual, 2))
> df_ensemble_res_train %>% knitr::kable()
```

Actual	Predicted	Error
0.33	0.48	0.15
0.83	0.68	-0.15
1.33	0.65	-0.68

Tabla 12 Errores del ajuste del modelo conjunto SARIMA y red neuronal (con datos de entrenamiento)

Así tenemos las predicciones de cuatro nuevos modelos. Que analizaremos junto con los resultados del modelo SARIMA en la siguiente sección de resultados (3).

3. RESULTADOS

Se recogen los resultados de las predicciones de cada uno de los modelos en una tabla resumen. Pero primero debemos introducir los tipos de errores que se mencionan a continuación:

- MAE. Error absoluto medio. Se define con la ecuación:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (20)$$

- MAPE. Error de porcentaje absoluto medio:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{(\hat{y}_i - y_i)}{y_i} \right| \quad (21)$$

- MSE. Error cuadrático medio:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (22)$$

- RMSE. Error cuadrático medio:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (23)$$

Que podemos definir en código R como:

```
> get_accuracy_measures <- function(your_result_df) {
+   act <- your_result_df %>% pull(Actual) # Real
+   pred <- your_result_df %>% pull(Predicted) # Predicado
+   err <- act - pred # Error
+   per_err <- abs(err / act) # % Error
+   mae <- err %>% abs() %>% mean() # Mean Absolute Error (MAE)
+   mape <- 100*mean(per_err) # Mean Absolute Percentage Error (MAPE)
+   mse <- mean(err^2) # Mean Squared Error (MSE)
+   rmse <- sqrt(mse) # Root Mean Squared Error (RMSE)
+   return(data_frame(MAE = mae, MAPE = mape, MSE = mse, RMSE = rmse, N =
length(act))) # Resultados
+ }
```

A continuación, se muestra la tabla de resultados que contiene todos los modelos ajustados en las secciones anteriores:

```
> bind_rows(df_sarima %>% get_accuracy_measures(),
```

```

+   df_mlp_std %>% get_accuracy_measures(),
+   df_ensemble_std %>% get_accuracy_measures(),
+   df_ensemble_res %>% get_accuracy_measures(),
+   df_ensemble_res_train %>% get_accuracy_measures()
+ ) %>%
+ mutate(Approach = c("SARIMA", "MLP-STD", "HIB-STD", "HIB-RES", "HIB-
STD+RES")) %>%
+ select(Approach, everything()) %>%
+ mutate_if(is.numeric, function(x) {round(x, 2)}) %>%
+ knitr::kable()

```

Ajuste	MAE	MAPE	MSE	RMSE
SARIMA	0.37	69.65	0.15	0.38
MLP-STD	0.40	42.40	0.23	0.48
HIB-STD	0.27	42.27	0.11	0.34
HIB-RES	0.30	47.55	0.11	0.34
HIB-STD+RES	0.33	38.22	0.17	0.41

Tabla 13 Tabla de errores de cada ajuste

Como se ve en la tabla, el modelo con menor error MAE, MSE y RMSE es aquel que combina la red neuronal con el modelo SARIMA, combina ambas componentes para hacer un modelo óptimo.

Si bien es cierto, que el modelo SARIMA de por sí, ofrece una precisión bastante alta (RMSE=0.38). Que, de forma individual, supera a la red neuronal con creces.

Por otro lado el modelo conjunto de SARIMA y ambas redes neuronales es el que mejor resultado tiene respecto al error MAPE. No obstante, obtiene resultados MSE y RMSE más débiles.

En conjunto, se puede decir que el mejor modelo para este problema es el híbrido entre red neuronal y SARIMA que no considera los residuos. Puesto, que este ruido es muy difícil de modelar sin sobreajustar el modelo. Obtenemos un RMSE de 0.34 que es equivalente a una precisión teórica del 66%.

Se considera una precisión suficientemente alta, considerando que son los resultados sobre los datos de prueba de un problema real difícilmente normalizable.

Cabe recordar, que estos son resultados particulares de este caso práctico y no tienen por qué ser extrapolables a otros modelos o ajustes SARIMA.

En la siguiente ilustración se aprecian los ajustes que se han realizado en las secciones anteriores. Además, se comparan con el resultado real del segmento de datos de validación:

```

> x <- cbind(seq(1,length(yt)-length(train)))
> y <- cbind(df_sarima$Actual, df_sarima$Predicted, df_mlp_std$Predicted,
df_ensemble_std$Predicted, df_ensemble_res$Predicted,
df_ensemble_res_train$Predicted)
> matplot(x,y,type="l")

```

```
> legend("topleft",col = 1:6, lty = 1:6, legend = c("Valor", "SARIMA", "MLP-STD",  
"HIB-STD", "HIB-RES", "HIB-STD+RES"))
```

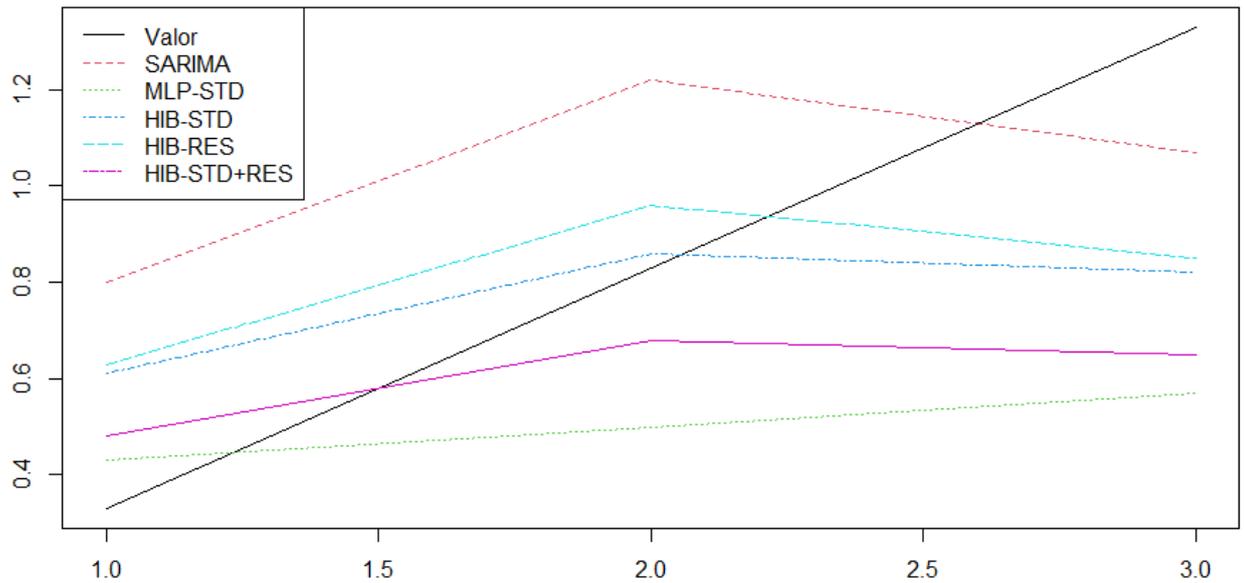


Ilustración 44 Comparación de los diferentes ajustes del modelo

Los modelos MLP-STD y HIB-STD+RES (en verde y rosa) obtienen mejores resultados al principio, pero peores al final.

El modelo SARIMA (en rojo) obtiene malos resultados al inicio, pero es el mejor al final del periodo analizado.

Podemos observar, como los mejores modelos son aquellos que se separan menos de la línea negra a lo largo del tiempo. Los modelos HIB-STD y HIB-RES (en azul claro y oscuro) son más consistentes a lo largo de todo el periodo analizado.

4. CONCLUSIONES

Como conclusión podemos destacar que el ajuste de un modelo económico real, en concreto, la tasa de interés de la Reserva Federal americana; que se selecciona de forma manual por un comité de expertos, es difícil de modelar. Las mayores dificultades están a la hora de normalizar el modelo y estabilizar su varianza.

No obstante, con un buen ajuste de los parámetros y coeficientes, y seleccionando el modelo adecuado; se pueden conseguir resultados positivos. Como se ha demostrado a lo largo de este documento.

De forma particular, el modelo SARIMA resulta apropiado para modelizar series temporales de factores económicos, que presentan además de tendencia, una componente estacional. Como es el caso del ejemplo tratado.

Si bien el resultado de ajustar simplemente con un modelo de esta índole es apropiado; cuando se tienen en cuenta todos los factores y premisas necesarios, se demuestra que se puede mejorar mediante el apoyo en la predicción del modelo de una red neuronal. Que también, se alimenta de la misma serie temporal que el modelo SARIMA.

Es decir, se comprueba que los métodos compuestos o ensamblados, pueden ofrecer mejores resultados que los modelos de forma individual, en el problema tratado. En el ejemplo de este trabajo, la red neuronal en conjunto con el modelo SARIMA es la que ofrece el ajuste con menor error. Esto es en parte, porque el ajuste del modelo SARIMA no es perfecto.

Al presentar residuos independientes ambos modelos, el modelo conjunto puede compensar las ineficiencias de ambos a lo largo del tiempo compensándose entre ellos.

En cuanto a las predicciones realizadas, si bien estiman de forma correcta. Situando en el centro entre el rango máximo y mínimo de la serie temporal, no se ajustan de forma exacta al recorrido de la curva. Por tanto, pueden servir como orientación genérica del entorno al que se moverán los tipos, pero no debe esperarse un valor de alta precisión (66%).

En futuros estudios, se pueden tener en cuenta otras variables macroeconómicas que influyen en las decisiones que toma el comité de expertos sobre la tasa de interés. Estas son la inflación, el crecimiento del empleo o la cotización de los mercados financieros respecto al producto interior bruto.

Además, se pueden considerar otros modelos que podrían ajustar mejor el modelo o ampliar el sistema ensamblado de los mismos aumentando la precisión de la predicción. Como 'Gradient Boosting', 'Prophet' o redes neuronales 'LTSM'.

5. BIBLIOGRAFIA

Dividida en tres categorías: libros, publicaciones y enlaces web.

5.1. Libros

[1] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. oTexts.
<https://www.otexts.org/fpp/8/9>

[2] Box, G., Jenkins, G., Reinsel, G., & Ljung, G. (2016). *Fifth Edition Time Series Analysis Forecasting and Control*, 712.

[3] Cryer, J. D., & Chan, K.-S. (2008). *Springer Texts in Statistics Time Series Analysis With Applications in R Second Edition*.

[4] Navarro, D. (2019). *Learning statistics with R: A tutorial for psychology students and other beginners*.
<https://learningstatisticswithr.com/book/regression.html#r2>

[5] Alberto Mauricio, José. (2013). *Introducción al Análisis de Series Temporales*.
<https://www.ucm.es/data/cont/docs/518-2013-11-11-JAM-IAST-Libro.pdf>

5.2. Publicaciones

[6] Tsay, R. S. (1986). Time Series Model Specification in the Presence of Outliers. *Journal of the American Statistical Association*, 81(393), 132.
<https://doi.org/10.2307/2287980>

[7] Tsay, R. S. and Tiao, G. C. (1984). Consistent Estimates of Autoregressive Parameters and Extended Sample Autocorrelation Function for Stationary and Nonstationary ARMA Models. *J. Amer. Statist. Assoc.* 79

[8] Gea, J., & Uriel, E. (1986). Identificación automática mediante la función de autocorrelación extendida (*), 111.

[9] Bonilla Ruiz, A., & Reche Lorite, F. (2019). *Estudio estadístico de un caso real. Series temporales*. . Universidad de Almería.

[10] SAGE. (2017). Learn About Time Series ACF and PACF in SPSS With Data From the USDA Feed Grains Database Learn About Time Series ACF and PACF in SPSS With Data From the USDA Feed Grains Database (1876-2015) Student Guide.

[11] Bozkurt, Ö. Ö., Biricik, G., & Tayşi, Z. C. (2017). Artificial neural network and SARIMA based models for power load forecasting in Turkish electricity market. *PLOS ONE*, 12(4), e0175915. <https://doi.org/10.1371/JOURNAL.PONE.0175915>

- [12] Ruiz-Aguilar, J. J., Turias, I. J., & Jiménez-Come, M. J. (2014). Hybrid approaches based on SARIMA and artificial neural networks for inspection time series forecasting. *Transportation Research Part E: Logistics and Transportation Review*, 67, 1–13. <https://doi.org/10.1016/J.TRE.2014.03.009>
- [13] Hu, J., Wang, J., & Zeng, G. (2013). A hybrid forecasting approach applied to wind speed time series. *Renewable Energy*, 60, 185–194.
- [14] Noemí Ordóñez Porras. (2005). *LA MODELIZACION DE VARIABLES MACRO ECONOMICAS PERUANAS Los ARCH como Alternativa a los ARIMA*.
- [15] Doris Jimenez, Henry Paz-Arias, & Andres Larco-A. (2015). Desarrollo de un sistema inteligente para la clasificación de documentos ya digitalizados aplicando redes neuronales supervisadas. *Revista Tecnológica ESPOL – RTE*, 28(1), 8–23. Retrieved from https://www.researchgate.net/publication/281380920_Desarrollo_de_un_sistema_inteligente_para_la_clasificacion_de_documentos_ya_digitalizados_aplicando_redes_neuronales_supervisadas
- [16] Dimitriou Asteriou, & Stephen G. Hall. (2011). *Applied Econometrics*, 16.
- [17] Alencar, D. B., Affonso, C. M., Oliveira, R. C. L., & Filho, J. C. R. (2018). Hybrid approach combining SARIMA and neural networks for multi-step ahead wind speed forecasting in Brazil. *IEEE Access*, 6, 55986–55994. <https://doi.org/10.1109/ACCESS.2018.2872720>
- [18] Catalán, C. E. (n.d.). *SERIES TEMPORALES. CSIC Laboratorio de Estadística*.
- [19] Gilli, M., Maringer, D., & Schumann, E. (2019). A gentle introduction to financial simulation. *Numerical Methods and Optimization in Finance*, 153–188. <https://doi.org/10.1016/B978-0-12-815065-8.00019-4>
- [20] Hyndman, R. J., & Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software*, 27(1), 1–22. <https://doi.org/10.18637/JSS.V027.I03>
- [21] Elwasify, A. I. (2015). A Combined Model between Artificial Neural Networks and ARIMA Models. *International Journal of Recent Research in Commerce Economics and Management (IJRRCEM)*, 2, 134–140. www.paperpublications.org

5.3. Enlaces web

- [22] Corey Weisinger. (2021). S.A.R.I.M.A. Seasonal ARIMA Models with KNIME | KNIME. Retrieved September 10, 2021, from <https://www.knime.com/blog/sarima-seasonal-arima-models-with-knime>

[23] Jayesh Salvi. (2019). Significance of ACF and PACF Plots In Time Series Analysis | by Jayesh Salvi | Towards Data Science. Retrieved September 10, 2021, from <https://towardsdatascience.com/significance-of-acf-and-pacf-plots-in-time-series-analysis-2fa11a5d10a8>

[24] 1 Year Treasury Rate - 54 Year Historical Chart | MacroTrends. (n.d.). Retrieved September 10, 2021, from <https://www.macrotrends.net/2492/1-year-treasury-rate-yield-chart>

[25] What Is an Autoregressive Model? | 365 Data Science. (n.d.). Retrieved September 10, 2021, from <https://365datascience.com/tutorials/time-series-analysis-tutorials/autoregressive-model/>

[26] Jason Fernando. (2020). Autoregressive Defined. Retrieved September 10, 2021, from <https://www.investopedia.com/terms/a/autoregressive.asp>

[27] Jason Brownlee. (2021). Autoregression Models for Time Series Forecasting With Python. Retrieved September 10, 2021, from <https://machinelearningmastery.com/autoregression-models-time-series-forecasting-python/>

[28] Eberly College of Science | Department of Statistics. (2021). 14.1 - Autoregressive Models | STAT 501. Retrieved September 10, 2021, from <https://online.stat.psu.edu/stat501/lesson/14/14.1>

[29] Marco Peixeiro. (2020). Advanced Time Series Analysis with ARMA and ARIMA | by Marco Peixeiro | Towards Data Science. Retrieved September 10, 2021, from <https://towardsdatascience.com/advanced-time-series-analysis-with-arma-and-arima-a7d9b589ed6d>

[30] Selva Prabhakaran. (2021). ARIMA Model - Complete Guide to Time Series Forecasting in Python | ML+. Retrieved September 10, 2021, from <https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/>

[31] Effective Federal Funds Rate (FEDFUNDS) | FRED | St. Louis Fed. (n.d.). Retrieved September 14, 2021, from <https://fred.stlouisfed.org/series/FEDFUNDS#>

[32] RPubS - Forecasting Time Series: Auto ARIMA, Ensemble Method and Artificial Neural Networks. (n.d.). Retrieved September 14, 2021, from <https://rpubs.com/chidungkt/428252>

[33] Indicadores de evaluación RMSE, MSE, MAE, MAPE, SMAPE, R-Squared- implementación de python + sklearn - programador clic. (n.d.). Retrieved September 14, 2021, from <https://programmerclick.com/article/7582866500/>

[34] auto.sarima: Automatic estimate of a Seasonal ARIMA model in bayesforecast: Bayesian Time Series Modeling with Stan. (n.d.). Retrieved September 11, 2021, from <https://rdrr.io/cran/bayesforecast/man/auto.sarima.html>

6. ANEXOS

En esta sección se recogen aquellos documentos utilizados durante el estudio que por su extensión no se introducen en el desarrollo de este, pero que son importantes para el mismo.

6.1. Código en R

Para la elaboración de este documento se ha requerido del uso del lenguaje estadístico R para algunos puntos. A continuación, se incluyen los scripts utilizados:

- Cargar los paquetes necesarios:

```
library('tidyverse')
library('magrittr')
library('fpp2')
library('nnfor', warn.conflicts = FALSE)
library('forecast')
library('urca')
library('tseries')
library('TSA')
library('sarima')
library('lmtest')
library('nortest')
```

- Funciones de autocorrelación teóricas

- Autocorrelation AR - ACF AR(2)

```
par(mfrow=c(2,2))
y = ARMAacf(ar = c(-0.5,0.25), lag.max = 10)
y = y[2:11]
plot(y, x = 1:10, type = "h", ylim = c(-1,1), xlab = "lag", ylab =
"Autocorrelation AR", main = "ACF AR(2)")
abline(h = 0)
```

- Partial autocorrelation AR - PACF AR(2)

```
y = ARMAacf(ar = c(-0.5,0.25), lag.max = 10, pacf=TRUE)
plot(y, x = 1:10, type = "h", ylim = c(-1,1), xlab = "lag", ylab = "Partial
autocorrelation AR", main = "PACF AR(2)")
abline(h = 0)
```

- Autocorrelation MA - ACF MA(2)

```
y = ARMAacf(ma = c(0.5,-0.25), lag.max = 10)
y = y[2:11]
```

```
plot(y, x = 1:10, type = "h", ylim = c(-1,1), xlab = "lag", ylab =
"Autocorrelation MA", main = "ACF MA(2)")
abline(h = 0)
```

- Partial autocorrelation MA - PACF MA(2)

```
y = ARMAacf(ma = c(0.5,-0.25), lag.max = 10, pacf=TRUE)
plot(y, x = 1:10, type = "h", ylim = c(-1,1), xlab = "lag", ylab = "Partial
autocorrelation MA", main = "PACF MA(2)")
abline(h = 0)
```

- Lectura de datos y visualización

- Leer archivo de datos '.csv' de FRED

```
df = read.csv("FEDFUNDS.csv", sep = ",")
```

- Determinar frecuencia de la serie

```
frec = 12
```

- Convertir en serie temporal

```
yt <- ts(df$FEDFUNDS, frequency = frec, start=1954)
```

- EDA frecuencia

```
#lineas
ggseasonplot(yt, year.labels=TRUE)
#polar
ggseasonplot(yt, year.labels=TRUE, year.labels.left=TRUE)
ggseasonplot(yt, polar="TRUE")
```

- Visualizar serie temporal de tipos de interes de la FED:

```
forecast::autoplot(yt) + geom_line(color = "white") + geom_point(color =
"white") +
labs(x = NULL, y = NULL, title = "Tipo de interes de la FED desde 1954
hasta la actualidad", subtitle = "Data Source: FRED Federal Reserve Bank
of St.Louis") +
theme(
axis.line = element_blank(),
axis.text.x = element_text(color = "white", lineheight = 0.9),
axis.text.y = element_text(color = "white", lineheight = 0.9),
axis.ticks = element_line(color = "white", size = 0.2),
axis.title.x = element_text(color = "white", margin = margin(0, 10, 0,
0)),
axis.title.y = element_text(color = "white", angle = 90, margin =
margin(0, 10, 0, 0)),
axis.ticks.length = unit(0.3, "lines"),
```

```

legend.background = element_rect(color = NA, fill = "gray10"),
legend.key = element_rect(color = "white", fill = "gray10"),
legend.key.size = unit(1.2, "lines"),
legend.key.height = NULL,
legend.key.width = NULL,
legend.text = element_text(color = "white"),
legend.title = element_text(face = "bold", hjust = 0, color = "white"),
legend.text.align = NULL,
legend.title.align = NULL,
legend.direction = "vertical",
legend.box = NULL,
panel.background = element_rect(fill = "gray10", color = NA),
panel.border = element_blank(),
panel.grid.major = element_line(color = "grey35"),
panel.grid.minor = element_line(color = "grey20"),
panel.spacing = unit(0.5, "lines"),
strip.background = element_rect(fill = "grey30", color = "grey10"),
strip.text.x = element_text(color = "white"),
strip.text.y = element_text(color = "white", angle = -90),
plot.background = element_rect(color = "gray10", fill = "gray10"),
plot.title = element_text(color = "white", hjust = 0, lineheight = 1.25,
margin = margin(2, 2, 2, 2)),
  plot.subtitle = element_text(color = "white", hjust = 0, margin =
margin(2, 2, 2, 2)),
  plot.caption = element_text(color = "white", hjust = 0),
  plot.margin = unit(rep(1, 4), "lines")

```

- Estacionariedad de los datos

- Estudiamos su estacionariedad con el test Dickey-Fuller.

```

adf.test(yt)
#0.1038 > 5% -> No estacionario
#Por tanto debemos usar ARIMA / SARIMA. Y no AR, MA, ARMA.

```

- Normalidad de los datos

- Gráfico Q-Q Normalidad

```

qqnorm(yt, pch = 19, col = "gray50")
qqline(yt)

```

- Test de normalidad.

```

lillie.test(x = yt)
#p-value < 2.2e-16 < 5% -> no es normal

```

- Transformación de los datos - Diferenciación

- En primer lugar se estudia la necesidad de realizar una diferenciación (parece obvia).

```
dfuller<-ur.df(yt,lags=0,type="trend")
summary(dfuller)
#No rechazo la hipótesis de raíz unitaria (compara -2.3686 con -3.41 al 5%).
```

- ¿Se deben realizar 2 diferenciaciones?

```
dfuller<-ur.df(diff(yt),lags=0,type="trend")
summary(dfuller)
#No debo de volver a diferenciar la serie (-18.9867< -3.41).
```

```
#por tanto:
dym <-diff(yt)
plot(dym)
```

- Estudiamos su estacionariedad con el test Dickey-Fuller.

```
adf.test(dym)
# 0.01 < 5% -> Estacionario
```

- Test de normalidad.

```
lillie.test(x = dym)
#p-value < 2.2e-16 < 5% -> no es normal
```

- Transformación de los datos - Box-Cox

- hacer los valores positivos (sumo 1 para los q 1)

```
ndym <-dym-min(dym)+1
plot(ndym)
```

- calculamos lambda y realizamos la transformación

```
lambda <- BoxCox.lambda(ndym)
lambda
```

```
BCym<-BoxCox(ndym,lambda)
plot(BCym)
```

- Estudiamos su estacionariedad con el test Dickey-Fuller.

```
adf.test(BCym)
# 0.01 < 5% -> Estacionario
```

- gráfico Q-Q de la transformada de Box-Cox

```
qqnorm(BCyt, pch = 19, col = "gray50")
qqline(BCyt)
```

- Test de normalidad.

```
lillie.test(x = BCyt)
#p-value < 2.2e-16 < 5% -> no es normal
#Podemos ver que sigue sin ser normal pese a la transformaci?n de Box-
Cox.
```

- Conjunto de datos de entrenamiento y prueba

```
p80 <- round(length(yt)*0.996) #12 datos (m?ximo de un ARIMA para
evitar variabilidad)
p80
```

- Separar datos en 80% entrenamiento y 20% test:

```
train <- yt[0:p80] %>% ts(frequency=frec)
test <- yt[(p80+1):(length(yt))] %>% ts(frequency=frec)
```

- Separar datos en 80% entrenamiento y 20% test:

```
train_nd <- ndyt[0:p80] %>% ts(frequency=1)
test_nd <- ndyt[(p80+1):(length(yt))] %>% ts(frequency=frec)
```

```
#separamos en train y test
train_bc <- BCyt[0:p80] %>% ts(frequency=1)
test_bc <- BCyt[(p80+1):(length(yt))] %>% ts(frequency=1)
```

- grafico acf

```
acf(BCyt,lag = 200)
acf(BCyt,lag = 48)
```

- grafico pacf

```
pacf(BCyt, lag = 200)
pacf(BCyt, lag = 48)
```

- Ajuste del modelo SARIMA con datos transformados

- Determinar la frecuencia de la serie temporal

```
findfrequency(yt)
```

- Incluir iteraciones

```
fit<- arima(train_bc,order=c(0,0,0),seasonal =
list(order=c(0,0,1),period=9))
```

```
#fit <- sarima(train_bc ~ 0 | ma(1) + sma(40,1) + i(0) + si(40,1) +
sma(9,1), ss.method = "sarima")
#fit <- Arima(ts(train_bc,frequency=16),order=c(0,0,0), seasonal=c(0,0,1))
#fit <- auto.arima(ts(train_bc,frequency=16),trace=TRUE, d=1, D=0)
#fit <- Arima(ts(train_bc,frequency=40),order=c(1,0,2), seasonal=c(2,0,1))
```

```
autoplot(fit)
summary(fit)
coefest(fit)
checkresiduals(fit, lag = 20)
acf(fit$residuals, lag=48)
pacf(fit$residuals, lag=48)
Box.test(fit$residuals,lag=20,type="Ljung-Box",fitdf=1)
```

- Predicción para boxcox diferenciado

```
predicted_sarima <- forecast(train_bc,h=length(test), model=fit) %>%
as.vector()
autoplot(predicted_sarima)
```

- Deshacer las transformaciones

- Deshacer Box-Cox

```
invBoxCox <- function(x, lambda){
  if (lambda == 0) exp(x) else (lambda*x + 1)^(1/lambda)
}
predBC <- invBoxCox(predicted_sarima$mean,lambda)
autoplot(predBC)
```

- Deshacer cambio de signo y diferencia

```
predBCp <- predBC+min(dyt)-1
autoplot(predBCp)
```

- Agrupo observados (primeras-obs) predicciones (?ltimas)

```
a<-0
horizonte<-length(test)
ll<-length(yt)
for (i in 1:ll){a[i]<-yt[i]}
for (j in 1:horizonte){a[ll+j]<-predBCp[j]+a[ll+j-1]}
a <- a[1:horizonte] %>% ts(frequency=1)
```

a

- Ajuste del modelo SARIMA

- DataFrame del modelo SARIMA:

```
df_sarima <- data_frame(Actual = test %>% as.vector(), Predicted = a
%>% round(2), Error = Predicted - Actual, Error_Percent = round(Error /
Actual, 2))
```

- Resultados en forma de tabla:

```
df_sarima %>% knitr::kable()
```

- Gráfico predicción del modelo SARIMA:

```
x <- cbind(seq(1,length(yt)-length(train)))
y <- cbind(df_sarima$Actual, df_sarima$Predicted)
matplot(x,y,type="l")
legend("topleft",col = 1:2, lty = 1:2, legend = colnames(df_sarima)[1:2])
```

- Ponemos el resultado de la predicción del SARIMA como serie temporal

```
predicted_sarima_ts <- ts(c(train[(length(train)-
length(a)):length(train)],a),frequency=frec)
autoplot(predicted_sarima_ts)
```

- Ponemos el resultado de los residuos de la predicción del SARIMA como serie temporal

```
previous_residuals <- ts(predicted_sarima$residuals,frequency=frec)
```

- Ajuste de la Red Neuronal - Estándar

- Generamos una red neuronal y una serie temporal con los datos de entrenamiento

```
mlp_std <- mlp(train, hd = 5, difforder = 0)
plot(mlp_std)
predicted_mlp_std <- forecast(mlp_std, h=length(test),
model=mlp_std)$mean %>% as.vector()
plot(predicted_mlp_std)
#predicted_mlp_std_ts <- ts(predicted_mlp_std, frequency = frec)
predicted_mlp_std_ts <- ts(c(train[(length(train)-
length(predicted_mlp_std)):length(train)],predicted_mlp_std), frequency =
frec)
autoplot(predicted_mlp_std_ts)
```

- Dataframe de la de la Red Neuronal - Estándar :

```
df_mlp_std <- data_frame(Actual = test %>% as.vector(), Predicted =
predicted_mlp_std %>% round(2), Error = Predicted - Actual,
Error_Percent = round(Error / Actual, 2))
```

- Resultados en forma de tabla:

```
df_mlp_std %>% knitr::kable()
```

- Ajuste de la Red Neuronal - Residuos

- Generamos una red neuronal y una serie temporal con las predicciones de los residuos anteriores

```
mlp_residuals <- mlp(previous_residuals,hd = 5, difforder = 0)
plot(mlp_residuals)
predicted_residuals <- forecast(mlp_residuals, h=length(test),
model=mlp_residuals)$mean %>% as.vector()
plot(predicted_residuals)
predicted_residuals_ts <-
ts(c(previous_residuals[(length(previous_residuals)-
length(test)):length(previous_residuals)],predicted_residuals), frequency =
frec)
#predicted_residuals_ts <- ts(predicted_residuals, frequency = frec)
autoplot(predicted_residuals_ts)
```

- Dataframe de la red neuronal pura (datos entrenamiento):

```
df_mlp_res <- data_frame(Actual = test %>% as.vector(), Predicted =
predicted_residuals %>% round(2), Error = Predicted - Actual,
Error_Percent = round(Error / Actual, 2))
```

- Resultados en forma de tabla:

```
df_mlp_res %>% knitr::kable()
```

- Ajuste del Híbrido - Estándar

- Generamos una red neuronal cuyo input son las predicciones del SARIMA

```
mlp_ensembled_std <- (predicted_sarima_ts+predicted_mlp_std_ts)/2
plot(mlp_ensembled_std)
mlp_predicted_std <- mlp_ensembled_std[(1+length(mlp_ensembled_std)-
length(test)):length(mlp_ensembled_std)] %>% ts(frequency=1)
autoplot(mlp_predicted_std)
```

- Dataframe de la red neuronal híbrida:

```
df_ensemble_std <- data_frame(Actual = test %>% as.vector(), Predicted =
mlp_predicted_std %>% round(2), Error = Predicted - Actual,
Error_Percent = round(Error / Actual, 2))
```

- Resultados en forma de tabla:

```
df_ensemble_std %>% knitr::kable()
```

- Ajuste del Híbrido - Residuos

- Descomposición tendencia-ciclo, estacionalidad y ruido.

```
yt %>% stl(s.window=12, robust=TRUE) %>% autoplot()
```

```
y2 <- decompose(yt)
plot(y2,xaxt = "n")
```

- Generamos una red neuronal que se alimenta de los residuos del SARIMA y las previsiones de la red neuronal de los residuos

```
mlp_res <- (predicted_residuals_ts*0.21+predicted_sarima_ts*0.79)
plot(mlp_res)
mlp_predicted_res <- mlp_res[(1+length(mlp_res)-
length(test)):length(mlp_res)] %>% ts(frequency=1)
```

- Dataframe de la red neuronal híbrida:

```
df_ensemble_res <- data_frame(Actual = test %>% as.vector(), Predicted
= mlp_predicted_res %>% round(2), Error = Predicted - Actual,
Error_Percent = round(Error / Actual, 2))
```

- Resultados en forma de tabla:

```
df_ensemble_res %>% knitr::kable()
```

- Ajuste del Híbrido - Residuos + Entrenamiento

- Generamos una red neuronal que se alimenta de los residuos del SARIMA y las previsiones de la red neuronal de los residuos

```
mlp_res_train <-
(predicted_residuals_ts*0.21+(predicted_sarima_ts+predicted_mlp_std_ts)
/2*0.79)
plot(mlp_res_train)
mlp_predicted_res_train <- mlp_res_train[(1+length(mlp_res_train)-
length(test)):length(mlp_res_train)] %>% ts(frequency=1)
```

- Dataframe de la red neuronal híbrida:

```
df_ensemble_res_train <- data_frame(Actual = test %>% as.vector(),
Predicted = mlp_predicted_res_train %>% round(2), Error = Predicted -
Actual, Error_Percent = round(Error / Actual, 2))
```

- Resultados en forma de tabla:

```
df_ensemble_res_train %>% knitr::kable()
```

- Gráfico resumen modelos

- Gráfico predicción que incluye al modelo SARIMA y al híbrido

```
x <- cbind(seq(1,length(yt)-length(train)))
y <- cbind(df_sarima$Actual, df_sarima$Predicted, df_mlp_std$Predicted,
df_ensemble_std$Predicted, df_ensemble_res$Predicted,
df_ensemble_res_train$Predicted)
matplot(x,y,type="l")
legend("topleft",col = 1:6, lty = 1:6, legend = c("Valor", "SARIMA",
"MLP-STD", "HIB-STD", "HIB-RES", "HIB-STD+RES"))
```

- Función de precisión

- Función que calcula los errores:

```
get_accuracy_measures <- function(your_result_df) {
  act <- your_result_df %>% pull(Actual) # Real
  pred <- your_result_df %>% pull(Predicted) # Predecido
  err <- act - pred # Error
  per_err <- abs(err / act) # % Error
  mae <- err %>% abs() %>% mean() # Mean Absolute Error (MAE)
  mape <- 100*mean(per_err) # Mean Absolute Percentage Error (MAPE)
  mse <- mean(err^2) # Mean Squared Error (MSE)
  rmse <- sqrt(mse) # Root Mean Squared Error (RMSE)
  return(data_frame(MAE = mae, MAPE = mape, MSE = mse, RMSE =
rmse, N = length(act))) # Resultados
}
```

- Tabla resumen modelos

- Comparación de los errores de los tres:

```
bind_rows(df_sarima %>% get_accuracy_measures(),
df_mlp_std %>% get_accuracy_measures(),
df_ensemble_std %>% get_accuracy_measures(),
df_ensemble_res %>% get_accuracy_measures(),
df_ensemble_res_train %>% get_accuracy_measures()
) %>%
mutate(Approach = c("SARIMA", "MLP-STD", "HIB-STD", "HIB-
RES", "HIB-STD+RES")) %>%
select(Approach, everything()) %>%
mutate_if(is.numeric, function(x) {round(x, 2)}) %>%
knitr::kable()
```

6.2. Otros anexos

- Datos utilizados, se agrupan cada cincuenta registros de izquierda a derecha primero, y después de arriba abajo, por motivos de espacio:

DATE	FED FUNDS						
1/7/54	0.80	1/9/58	1.76	1/11/62	2.94	1/1/67	4.94
1/8/54	1.22	1/10/58	1.80	1/12/62	2.93	1/2/67	5.00
1/9/54	1.07	1/11/58	2.27	1/1/63	2.92	1/3/67	4.53
1/10/54	0.85	1/12/58	2.42	1/2/63	3.00	1/4/67	4.05
1/11/54	0.83	1/1/59	2.48	1/3/63	2.98	1/5/67	3.94
1/12/54	1.28	1/2/59	2.43	1/4/63	2.90	1/6/67	3.98
1/1/55	1.39	1/3/59	2.80	1/5/63	3.00	1/7/67	3.79
1/2/55	1.29	1/4/59	2.96	1/6/63	2.99	1/8/67	3.90
1/3/55	1.35	1/5/59	2.90	1/7/63	3.02	1/9/67	3.99
1/4/55	1.43	1/6/59	3.39	1/8/63	3.49	1/10/67	3.88
1/5/55	1.43	1/7/59	3.47	1/9/63	3.48	1/11/67	4.13
1/6/55	1.64	1/8/59	3.50	1/10/63	3.50	1/12/67	4.51
1/7/55	1.68	1/9/59	3.76	1/11/63	3.48	1/1/68	4.61
1/8/55	1.96	1/10/59	3.98	1/12/63	3.38	1/2/68	4.71
1/9/55	2.18	1/11/59	4.00	1/1/64	3.48	1/3/68	5.05
1/10/55	2.24	1/12/59	3.99	1/2/64	3.48	1/4/68	5.76
1/11/55	2.35	1/1/60	3.99	1/3/64	3.43	1/5/68	6.12
1/12/55	2.48	1/2/60	3.97	1/4/64	3.47	1/6/68	6.07
1/1/56	2.45	1/3/60	3.84	1/5/64	3.50	1/7/68	6.03
1/2/56	2.50	1/4/60	3.92	1/6/64	3.50	1/8/68	6.03
1/3/56	2.50	1/5/60	3.85	1/7/64	3.42	1/9/68	5.78
1/4/56	2.62	1/6/60	3.32	1/8/64	3.50	1/10/68	5.91
1/5/56	2.75	1/7/60	3.23	1/9/64	3.45	1/11/68	5.82
1/6/56	2.71	1/8/60	2.98	1/10/64	3.36	1/12/68	6.02
1/7/56	2.75	1/9/60	2.60	1/11/64	3.52	1/1/69	6.30
1/8/56	2.73	1/10/60	2.47	1/12/64	3.85	1/2/69	6.61
1/9/56	2.95	1/11/60	2.44	1/1/65	3.90	1/3/69	6.79
1/10/56	2.96	1/12/60	1.98	1/2/65	3.98	1/4/69	7.41
1/11/56	2.88	1/1/61	1.45	1/3/65	4.05	1/5/69	8.67
1/12/56	2.94	1/2/61	2.54	1/4/65	4.09	1/6/69	8.90
1/1/57	2.84	1/3/61	2.02	1/5/65	4.10	1/7/69	8.61
1/2/57	3.00	1/4/61	1.49	1/6/65	4.05	1/8/69	9.19
1/3/57	2.96	1/5/61	1.98	1/7/65	4.09	1/9/69	9.15
1/4/57	3.00	1/6/61	1.73	1/8/65	4.12	1/10/69	9.00
1/5/57	3.00	1/7/61	1.17	1/9/65	4.02	1/11/69	8.85

1/6/57	3.00	1/8/61	2.00	1/10/65	4.08	1/12/69	8.97
1/7/57	2.99	1/9/61	1.88	1/11/65	4.10	1/1/70	8.98
1/8/57	3.24	1/10/61	2.26	1/12/65	4.32	1/2/70	8.98
1/9/57	3.47	1/11/61	2.61	1/1/66	4.42	1/3/70	7.76
1/10/57	3.50	1/12/61	2.33	1/2/66	4.60	1/4/70	8.10
1/11/57	3.28	1/1/62	2.15	1/3/66	4.66	1/5/70	7.95
1/12/57	2.98	1/2/62	2.37	1/4/66	4.67	1/6/70	7.61
1/1/58	2.72	1/3/62	2.85	1/5/66	4.90	1/7/70	7.21
1/2/58	1.67	1/4/62	2.78	1/6/66	5.17	1/8/70	6.62
1/3/58	1.20	1/5/62	2.36	1/7/66	5.30	1/9/70	6.29
1/4/58	1.26	1/6/62	2.68	1/8/66	5.53	1/10/70	6.20
1/5/58	0.63	1/7/62	2.71	1/9/66	5.40	1/11/70	5.60
1/6/58	0.93	1/8/62	2.93	1/10/66	5.53	1/12/70	4.90
1/7/58	0.68	1/9/62	2.90	1/11/66	5.76	1/1/71	4.14
1/8/58	1.53	1/10/62	2.90	1/12/66	5.40	1/2/71	3.72
1/3/71	3.71	1/4/73	7.12	1/5/75	5.22	1/6/77	5.39
1/4/71	4.16	1/5/73	7.84	1/6/75	5.55	1/7/77	5.42
1/5/71	4.63	1/6/73	8.49	1/7/75	6.10	1/8/77	5.90
1/6/71	4.91	1/7/73	10.40	1/8/75	6.14	1/9/77	6.14
1/7/71	5.31	1/8/73	10.50	1/9/75	6.24	1/10/77	6.47
1/8/71	5.57	1/9/73	10.78	1/10/75	5.82	1/11/77	6.51
1/9/71	5.55	1/10/73	10.01	1/11/75	5.22	1/12/77	6.56
1/10/71	5.20	1/11/73	10.03	1/12/75	5.20	1/1/78	6.70
1/11/71	4.91	1/12/73	9.95	1/1/76	4.87	1/2/78	6.78
1/12/71	4.14	1/1/74	9.65	1/2/76	4.77	1/3/78	6.79
1/1/72	3.51	1/2/74	8.97	1/3/76	4.84	1/4/78	6.89
1/2/72	3.30	1/3/74	9.35	1/4/76	4.82	1/5/78	7.36
1/3/72	3.83	1/4/74	10.51	1/5/76	5.29	1/6/78	7.60
1/4/72	4.17	1/5/74	11.31	1/6/76	5.48	1/7/78	7.81
1/5/72	4.27	1/6/74	11.93	1/7/76	5.31	1/8/78	8.04
1/6/72	4.46	1/7/74	12.92	1/8/76	5.29	1/9/78	8.45
1/7/72	4.55	1/8/74	12.01	1/9/76	5.25	1/10/78	8.96
1/8/72	4.81	1/9/74	11.34	1/10/76	5.02	1/11/78	9.76
1/9/72	4.87	1/10/74	10.06	1/11/76	4.95	1/12/78	10.03
1/10/72	5.05	1/11/74	9.45	1/12/76	4.65	1/1/79	10.07
1/11/72	5.06	1/12/74	8.53	1/1/77	4.61	1/2/79	10.06
1/12/72	5.33	1/1/75	7.13	1/2/77	4.68	1/3/79	10.09
1/1/73	5.94	1/2/75	6.24	1/3/77	4.69	1/4/79	10.01
1/2/73	6.58	1/3/75	5.54	1/4/77	4.73	1/5/79	10.24
1/3/73	7.09	1/4/75	5.49	1/5/77	5.35	1/6/79	10.29
1/7/79	10.47	1/8/81	17.82	1/9/83	9.45	1/10/85	7.99
1/8/79	10.94	1/9/81	15.87	1/10/83	9.48	1/11/85	8.05

1/9/79	11.43	1/10/81	15.08	1/11/83	9.34	1/12/85	8.27
1/10/79	13.77	1/11/81	13.31	1/12/83	9.47	1/1/86	8.14
1/11/79	13.18	1/12/81	12.37	1/1/84	9.56	1/2/86	7.86
1/12/79	13.78	1/1/82	13.22	1/2/84	9.59	1/3/86	7.48
1/1/80	13.82	1/2/82	14.78	1/3/84	9.91	1/4/86	6.99
1/2/80	14.13	1/3/82	14.68	1/4/84	10.29	1/5/86	6.85
1/3/80	17.19	1/4/82	14.94	1/5/84	10.32	1/6/86	6.92
1/4/80	17.61	1/5/82	14.45	1/6/84	11.06	1/7/86	6.56
1/5/80	10.98	1/6/82	14.15	1/7/84	11.23	1/8/86	6.17
1/6/80	9.47	1/7/82	12.59	1/8/84	11.64	1/9/86	5.89
1/7/80	9.03	1/8/82	10.12	1/9/84	11.30	1/10/86	5.85
1/8/80	9.61	1/9/82	10.31	1/10/84	9.99	1/11/86	6.04
1/9/80	10.87	1/10/82	9.71	1/11/84	9.43	1/12/86	6.91
1/10/80	12.81	1/11/82	9.20	1/12/84	8.38	1/1/87	6.43
1/11/80	15.85	1/12/82	8.95	1/1/85	8.35	1/2/87	6.10
1/12/80	18.90	1/1/83	8.68	1/2/85	8.50	1/3/87	6.13
1/1/81	19.08	1/2/83	8.51	1/3/85	8.58	1/4/87	6.37
1/2/81	15.93	1/3/83	8.77	1/4/85	8.27	1/5/87	6.85
1/3/81	14.70	1/4/83	8.80	1/5/85	7.97	1/6/87	6.73
1/4/81	15.72	1/5/83	8.63	1/6/85	7.53	1/7/87	6.58
1/5/81	18.52	1/6/83	8.98	1/7/85	7.88	1/8/87	6.73
1/6/81	19.10	1/7/83	9.37	1/8/85	7.90	1/9/87	7.22
1/7/81	19.04	1/8/83	9.56	1/9/85	7.92	1/10/87	7.29
1/11/87	6.69	1/12/89	8.45	1/1/92	4.03	1/2/94	3.25
1/12/87	6.77	1/1/90	8.23	1/2/92	4.06	1/3/94	3.34
1/1/88	6.83	1/2/90	8.24	1/3/92	3.98	1/4/94	3.56
1/2/88	6.58	1/3/90	8.28	1/4/92	3.73	1/5/94	4.01
1/3/88	6.58	1/4/90	8.26	1/5/92	3.82	1/6/94	4.25
1/4/88	6.87	1/5/90	8.18	1/6/92	3.76	1/7/94	4.26
1/5/88	7.09	1/6/90	8.29	1/7/92	3.25	1/8/94	4.47
1/6/88	7.51	1/7/90	8.15	1/8/92	3.30	1/9/94	4.73
1/7/88	7.75	1/8/90	8.13	1/9/92	3.22	1/10/94	4.76
1/8/88	8.01	1/9/90	8.20	1/10/92	3.10	1/11/94	5.29
1/9/88	8.19	1/10/90	8.11	1/11/92	3.09	1/12/94	5.45
1/10/88	8.30	1/11/90	7.81	1/12/92	2.92	1/1/95	5.53
1/11/88	8.35	1/12/90	7.31	1/1/93	3.02	1/2/95	5.92
1/12/88	8.76	1/1/91	6.91	1/2/93	3.03	1/3/95	5.98
1/1/89	9.12	1/2/91	6.25	1/3/93	3.07	1/4/95	6.05
1/2/89	9.36	1/3/91	6.12	1/4/93	2.96	1/5/95	6.01
1/3/89	9.85	1/4/91	5.91	1/5/93	3.00	1/6/95	6.00
1/4/89	9.84	1/5/91	5.78	1/6/93	3.04	1/7/95	5.85
1/5/89	9.81	1/6/91	5.90	1/7/93	3.06	1/8/95	5.74

1/6/89	9.53	1/7/91	5.82	1/8/93	3.03	1/9/95	5.80
1/7/89	9.24	1/8/91	5.66	1/9/93	3.09	1/10/95	5.76
1/8/89	8.99	1/9/91	5.45	1/10/93	2.99	1/11/95	5.80
1/9/89	9.02	1/10/91	5.21	1/11/93	3.02	1/12/95	5.60
1/10/89	8.84	1/11/91	4.81	1/12/93	2.96	1/1/96	5.56
1/11/89	8.55	1/12/91	4.43	1/1/94	3.05	1/2/96	5.22
1/3/96	5.31	1/4/98	5.45	1/5/00	6.27	1/6/02	1.75
1/4/96	5.22	1/5/98	5.49	1/6/00	6.53	1/7/02	1.73
1/5/96	5.24	1/6/98	5.56	1/7/00	6.54	1/8/02	1.74
1/6/96	5.27	1/7/98	5.54	1/8/00	6.50	1/9/02	1.75
1/7/96	5.40	1/8/98	5.55	1/9/00	6.52	1/10/02	1.75
1/8/96	5.22	1/9/98	5.51	1/10/00	6.51	1/11/02	1.34
1/9/96	5.30	1/10/98	5.07	1/11/00	6.51	1/12/02	1.24
1/10/96	5.24	1/11/98	4.83	1/12/00	6.40	1/1/03	1.24
1/11/96	5.31	1/12/98	4.68	1/1/01	5.98	1/2/03	1.26
1/12/96	5.29	1/1/99	4.63	1/2/01	5.49	1/3/03	1.25
1/1/97	5.25	1/2/99	4.76	1/3/01	5.31	1/4/03	1.26
1/2/97	5.19	1/3/99	4.81	1/4/01	4.8	1/5/03	1.26
1/3/97	5.39	1/4/99	4.74	1/5/01	4.21	1/6/03	1.22
1/4/97	5.51	1/5/99	4.74	1/6/01	3.97	1/7/03	1.01
1/5/97	5.50	1/6/99	4.76	1/7/01	3.77	1/8/03	1.03
1/6/97	5.56	1/7/99	4.99	1/8/01	3.65	1/9/03	1.01
1/7/97	5.52	1/8/99	5.07	1/9/01	3.07	1/10/03	1.01
1/8/97	5.54	1/9/99	5.22	1/10/01	2.49	1/11/03	1.00
1/9/97	5.54	1/10/99	5.20	1/11/01	2.09	1/12/03	0.98
1/10/97	5.50	1/11/99	5.42	1/12/01	1.82	1/1/04	1.00
1/11/97	5.52	1/12/99	5.30	1/1/02	1.73	1/2/04	1.01
1/12/97	5.50	1/1/00	5.45	1/2/02	1.74	1/3/04	1.00
1/1/98	5.56	1/2/00	5.73	1/3/02	1.73	1/4/04	1.00
1/2/98	5.51	1/3/00	5.85	1/4/02	1.75	1/5/04	1.00
1/3/98	5.49	1/4/00	6.02	1/5/02	1.75	1/6/04	1.03
1/7/04	1.26	1/8/06	5.25	1/9/08	1.81	1/10/10	0.19
1/8/04	1.43	1/9/06	5.25	1/10/08	0.97	1/11/10	0.19
1/9/04	1.61	1/10/06	5.25	1/11/08	0.39	1/12/10	0.18
1/10/04	1.76	1/11/06	5.25	1/12/08	0.16	1/1/11	0.17
1/11/04	1.93	1/12/06	5.24	1/1/09	0.15	1/2/11	0.16
1/12/04	2.16	1/1/07	5.25	1/2/09	0.22	1/3/11	0.14
1/1/05	2.28	1/2/07	5.26	1/3/09	0.18	1/4/11	0.10
1/2/05	2.50	1/3/07	5.26	1/4/09	0.15	1/5/11	0.09
1/3/05	2.63	1/4/07	5.25	1/5/09	0.18	1/6/11	0.09
1/4/05	2.79	1/5/07	5.25	1/6/09	0.21	1/7/11	0.07
1/5/05	3.00	1/6/07	5.25	1/7/09	0.16	1/8/11	0.10

1/6/05	3.04	1/7/07	5.26	1/8/09	0.16	1/9/11	0.08
1/7/05	3.26	1/8/07	5.02	1/9/09	0.15	1/10/11	0.07
1/8/05	3.50	1/9/07	4.94	1/10/09	0.12	1/11/11	0.08
1/9/05	3.62	1/10/07	4.76	1/11/09	0.12	1/12/11	0.07
1/10/05	3.78	1/11/07	4.49	1/12/09	0.12	1/1/12	0.08
1/11/05	4.00	1/12/07	4.24	1/1/10	0.11	1/2/12	0.10
1/12/05	4.16	1/1/08	3.94	1/2/10	0.13	1/3/12	0.13
1/1/06	4.29	1/2/08	2.98	1/3/10	0.16	1/4/12	0.14
1/2/06	4.49	1/3/08	2.61	1/4/10	0.20	1/5/12	0.16
1/3/06	4.59	1/4/08	2.28	1/5/10	0.20	1/6/12	0.16
1/4/06	4.79	1/5/08	1.98	1/6/10	0.18	1/7/12	0.16
1/5/06	4.94	1/6/08	2.00	1/7/10	0.18	1/8/12	0.13
1/6/06	4.99	1/7/08	2.01	1/8/10	0.19	1/9/12	0.14
1/7/06	5.24	1/8/08	2.00	1/9/10	0.19	1/10/12	0.16
1/11/12	0.16	1/12/14	0.12	1/1/17	0.65	1/2/19	2.40
1/12/12	0.16	1/1/15	0.11	1/2/17	0.66	1/3/19	2.41
1/1/13	0.14	1/2/15	0.11	1/3/17	0.79	1/4/19	2.42
1/2/13	0.15	1/3/15	0.11	1/4/17	0.90	1/5/19	2.39
1/3/13	0.14	1/4/15	0.12	1/5/17	0.91	1/6/19	2.38
1/4/13	0.15	1/5/15	0.12	1/6/17	1.04	1/7/19	2.40
1/5/13	0.11	1/6/15	0.13	1/7/17	1.15	1/8/19	2.13
1/6/13	0.09	1/7/15	0.13	1/8/17	1.16	1/9/19	2.04
1/7/13	0.09	1/8/15	0.14	1/9/17	1.15	1/10/19	1.83
1/8/13	0.08	1/9/15	0.14	1/10/17	1.15	1/11/19	1.55
1/9/13	0.08	1/10/15	0.12	1/11/17	1.16	1/12/19	1.55
1/10/13	0.09	1/11/15	0.12	1/12/17	1.30	1/1/20	1.55
1/11/13	0.08	1/12/15	0.24	1/1/18	1.41	1/2/20	1.58
1/12/13	0.09	1/1/16	0.34	1/2/18	1.42	1/3/20	0.65
1/1/14	0.07	1/2/16	0.38	1/3/18	1.51	1/4/20	0.05
1/2/14	0.07	1/3/16	0.36	1/4/18	1.69	1/5/20	0.05
1/3/14	0.08	1/4/16	0.37	1/5/18	1.70	1/6/20	0.08
1/4/14	0.09	1/5/16	0.37	1/6/18	1.82	1/7/20	0.09
1/5/14	0.09	1/6/16	0.38	1/7/18	1.91	1/8/20	0.10
1/6/14	0.10	1/7/16	0.39	1/8/18	1.91	1/9/20	0.09
1/7/14	0.09	1/8/16	0.40	1/9/18	1.95	1/10/20	0.09
1/8/14	0.09	1/9/16	0.40	1/10/18	2.19	1/11/20	0.09
1/9/14	0.09	1/10/16	0.40	1/11/18	2.20	1/12/20	0.09
1/10/14	0.09	1/11/16	0.41	1/12/18	2.27	1/1/21	0.09
1/11/14	0.09	1/12/16	0.54	1/1/19	2.40	1/2/21	0.08
1/3/21	0.07	1/9/21	0.08	1/3/22	0.33		
1/4/21	0.07	1/10/21	0.08	1/4/22	0.33		
1/5/21	0.06	1/11/21	0.08	1/5/22	0.83		

1/6/21	0.08	1/12/21	0.08	1/6/22	1.33		
1/7/21	0.10	1/1/22	0.08				
1/8/21	0.09	1/2/22	0.08				