



Universidad del Bío-Bío
Facultad de Ciencias
Ingeniería Estadística

Informe Final

Docente:
Darlin Soto

Alumna:
Krishna Bustos

8 de agosto de 2025

Índice

Índice	1
1. Introducción	3
2. Serie de tiempo	3
2.1. Definición de estacionariedad	3
2.2. Función de Autocovarianza	4
2.3. Densidad Espectral	4
2.4. Periodograma	5
3. Relación entre densidad espectral y ACVF	6
4. Relación entre la densidad espectral y el periodograma	6
5. Modelo ARMA sin errores de medición	8
5.1. Función de Autocovarianza	8
5.2. Densidad Espectral de Modelo ARMA	9
6. Modelo ARMA con errores de medición	9
6.1. Relación entre W_t y Z_t	10
6.2. Función de Autocovarianza	10
6.3. Densidad Espectral de Modelo ARMA	11
7. Simulaciones	11
7.1. Simulaciones de Modelos ARMA sin errores de medición	11
7.1.1. Modelos AR(1)	11
7.1.2. Modelo AR(1)	12
7.1.3. Modelo AR(2)	14
7.1.4. Modelo ARMA(3,2)	15
7.2. Simulaciones de Modelos ARMA con errores de medición	15
7.2.1. Modelos AR(1)	16
7.2.2. Modelos AR(2)	17
7.2.3. Modelos ARMA(3,2)	18
7.3. Comparación sin error con error	19
7.3.1. Modelo AR(1)	19
7.3.2. Modelo AR(2)	20
7.3.3. Modelo ARMA(3,2)	20
8. Revisión de literatura	21
8.1. Paquetes Y funciones en R para modelos Arma	21
8.2. Modelos arma sin errores de medición	21
8.3. Modelos Arma con errores de medición	21
9. Paquetes o funciones en Python que ajusten modelo ARMA	22
9.1. Modelos Arma sin errores de medición	22
9.2. Modelos Arma Con errores de medición	22

10.Ejemplos en R modelos Arma sin errores de medición	23
10.1. Ejemplo función arima()	23
10.2. Ejemplo función Arima()	23
10.3. Ejemplo función armasubsets()	24
10.4. Ejemplo función UniGAS()	24
10.5. Ejemplo función arma()	26
11.Ejemplos en R modelos Arma con errores de medición	27
11.1. Ejemplo función MARSS	27
12.Ejemplos en python modelos Arma sin errores de medición	30
12.1. Ejemplo función SARIMAX()	30
12.2. Ejemplo función SARIMAX()	31
Referencias	33

1. Introducción

2. Serie de tiempo

Una **serie de tiempo** es una secuencia de observaciones recolectadas a lo largo del tiempo, en general en intervalos regulares. Su característica distintiva es que el orden temporal de los datos no puede ignorarse, ya que contiene información clave sobre la evolución de la variable de interés. Según Brockwell and Davis (2016),

Cada valor observado x_t corresponde a un punto específico en el tiempo t , y el análisis de series de tiempo se basa en la identificación y modelación de patrones que pueden encontrarse en los datos, tales como:

- **Tendencia:** comportamiento general creciente o decreciente a lo largo del tiempo.
- **Estacionalidad:** variaciones periódicas que se repiten en intervalos fijos, como ciclos anuales o mensuales.
- **Ciclos:** fluctuaciones que no necesariamente tienen una periodicidad fija, asociadas a fenómenos económicos o sociales.
- **Ruido:** componente aleatoria o impredecible que afecta a la serie.

El análisis de series de tiempo es fundamental en diversas disciplinas como la economía (por ejemplo, series de precios, inflación, PIB), la ingeniería (control de procesos industriales), la meteorología (temperaturas o precipitaciones), la medicina (monitorización de signos vitales), entre otras. Las técnicas utilizadas permiten modelar el comportamiento temporal, hacer inferencias estadísticas y generar pronósticos.

2.1. Definición de estacionariedad

Una serie de tiempo $\{X_t\}$ se dice que es **estrictamente estacionaria** si, para cualquier conjunto de tiempos t_1, t_2, \dots, t_n , cualquier número h , y cualquier conjunto de constantes c_1, \dots, c_n , se cumple que:

$$P(X_{t_1} < c_1, \dots, X_{t_n} < c_n) = P(X_{t_1+h} < c_1, \dots, X_{t_n+h} < c_n)$$

Esto significa que la distribución conjunta de cualquier subconjunto de variables no cambia con el tiempo, es decir, es invariante ante desplazamientos temporales.

En la práctica, esta condición suele ser difícil de verificar, por lo que se utiliza una versión más débil, conocida como **estacionariedad débil** o **de segundo orden**. Un proceso $\{X_t\}$ es débilmente estacionario si cumple las siguientes condiciones:

1. $E[X_t^2] < \infty$ para todo t ,
2. $\mu_X(t) = E[X_t]$ es constante e independiente de t ,
3. $\gamma_X(r, s) = \text{Cov}(X_r, X_s)$ depende solo de la diferencia $|s - t|$.

Es decir, la esperanza y la varianza son constantes en el tiempo, y la covarianza entre dos observaciones depende únicamente del desfase entre ellas y no de los tiempos absolutos.

2.2. Función de Autocovarianza

Como se define en (Brockwell and Davis, 2016), sea $\{X_t\}$ una serie de tiempo estacionaria. La **función de autocovarianza** (Autocovariance Function, ACVF) de $\{X_t\}$ en el desfase h se define como:

$$\gamma_X(h) = \text{Cov}(X_{t+h}, X_t).$$

La función tiene las siguientes propiedades:

- $\gamma(0) = \text{Var}(X_t)$,
- $\gamma(-h) = \gamma(h)$, es simétrica.

La **función de autocorrelación** (Autocorrelation Function, ACF) de $\{X_t\}$ en el desfase h se define como:

$$\rho_X(h) \equiv \frac{\gamma_X(h)}{\gamma_X(0)} = \text{Cor}(X_{t+h}, X_t).$$

Estas definiciones son válidas únicamente para procesos *estacionarios en sentido débil*, ya que en tales casos la covarianza entre dos observaciones sólo depende de la distancia temporal h entre ellas, y no de los tiempos absolutos t .

2.3. Densidad Espectral

La **densidad espectral** describe cómo se distribuye la varianza de un proceso estocástico a través de diferentes frecuencias. Se obtiene a partir de la función de autocovarianza (ACVF) usando la transformada de Fourier (Brockwell and Davis, 2016).

Si $\gamma(h)$ es la función de autocovarianza de un proceso estacionario en media, su densidad espectral $f(\lambda)$ se define como:

$$f(\lambda) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} e^{-i\lambda h} \gamma(h), \quad -\infty < \lambda < \infty.$$

Dado que $\gamma(h)$ es una función par (es decir, $\gamma(-h) = \gamma(h)$) para procesos reales, la densidad espectral también puede expresarse en términos de cosenos:

$$f(\lambda) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} \gamma(h) \cos(\lambda h).$$

Las propiedades fundamentales de la función de densidad espectral $f(\lambda)$ son: (Brockwell and Davis, 2016):

- **Simetría par:** $f(\lambda) = f(-\lambda)$, es decir, la función es par.
- **No negatividad:** $f(\lambda) \geq 0$ para todo $\lambda \in (-\pi, \pi]$.
- **Relación con la función de autocovarianza:**

$$\gamma(k) = \int_{-\pi}^{\pi} e^{ik\lambda} f(\lambda) d\lambda = \int_{-\pi}^{\pi} \cos(k\lambda) f(\lambda) d\lambda, \quad \text{para } k \in \mathbb{Z}.$$

2.4. Periodograma

El **periodograma** es una herramienta fundamental para estimar la densidad espectral de una serie de tiempo estacionaria a partir de una muestra finita. Su definición es análoga, en el dominio de las frecuencias, a la función de autocovarianza muestral en el dominio del tiempo. Como señalan Brockwell and Davis (2016), el periodograma de un conjunto de datos $\{x_1, \dots, x_n\}$ se define como:

$$I_n(\lambda) = \frac{1}{n} \left| \sum_{t=1}^n x_t e^{-it\lambda} \right|^2$$

Si $\{X_t\}$ es una serie de tiempo estacionaria, con media cero y función de autocovarianza absolutamente sumable, es decir,

$$\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty,$$

entonces el valor esperado del periodograma $I_n(\lambda)$ converge al valor teórico de $2\pi f(\lambda)$, es decir:

$$\mathbb{E}[I_n(\lambda)] = 2\pi f(\lambda)$$

El periodograma puede considerarse un estimador *insesgado*, aunque *no consistente*, de la función espectral $2\pi f(\lambda)$, ya que su varianza no disminuye al aumentar el tamaño muestral. Específicamente, si λ coincide con una de las frecuencias de Fourier $\omega_k = \frac{2\pi k}{n}$, entonces se cumple que:

$$I_n(\omega_k) = |a_k|^2,$$

donde a_k es el coeficiente de la transformada discreta de Fourier de la muestra. Además, la energía total de la serie (es decir, la suma de los cuadrados de las observaciones) puede expresarse como:

$$\sum_{t=1}^n x_t^2 = \sum_k |a_k|^2 = \sum_k I_n(\omega_k),$$

lo que implica que el valor del periodograma en cada frecuencia representa la contribución de esa frecuencia a la varianza total de la serie. En este sentido, el periodograma descompone la energía de la serie según sus componentes frecuenciales.

Para mitigar la falta de consistencia del estimador, es común aplicar un *suavizado* del periodograma, promediando sobre un vecindario de frecuencias cercanas. Por ejemplo, puede utilizarse el estimador suavizado:

$$\tilde{f}(\lambda) = \frac{1}{2\pi(2m+1)} \sum_{|j| \leq m} I_n \left(g(n, \lambda) + \frac{2\pi j}{n} \right),$$

donde $m = \sqrt{n}$ y $g(n, \lambda)$ es el múltiplo de $2\pi/n$ más cercano a λ . Este estimador es *consistente*, como se demuestra en Brockwell and Davis (2016).

3. Relación entre densidad espectral y ACVF

Para una serie de tiempo estacionaria $\{X_t\}$, existe una relación fundamental entre su **función de autocovarianza** $\gamma(h)$ y su **función de densidad espectral** $f(\lambda)$. Esta relación establece que ambas son transformadas de Fourier una de la otra.

La función de densidad espectral está definida como:

$$f(\lambda) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} \gamma(h) e^{-i\lambda h}, \quad -\pi < \lambda \leq \pi,$$


donde $\gamma(h) = \text{Cov}(X_t, X_{t+h})$ es la función de autocovarianza para el desfase h , y $f(\lambda)$ describe cómo se distribuye la varianza de X_t a través de distintas frecuencias λ .

De manera inversa, también se puede recuperar la función de autocovarianza a partir de la densidad espectral mediante:

$$\gamma(h) = \int_{-\pi}^{\pi} f(\lambda) e^{i\lambda h} d\lambda.$$

Esto significa que conocer una de estas funciones implica conocer completamente la otra, y es la base del análisis espectral de series de tiempo. La función de densidad espectral es entonces la transformada de Fourier de la función de autocovarianza (Brockwell and Davis, 2016).

4. Relación entre la densidad espectral y el periodograma


El periodograma es una herramienta fundamental para estimar la densidad espectral de una serie temporal estacionaria a partir de una muestra finita. Su definición es análoga, en el dominio de las frecuencias, a la función de autocovarianza muestral en el dominio del tiempo. Según Brockwell and Davis (2016), para una muestra $\{x_1, \dots, x_n\}$, el periodograma está definido como: 

$$I_n(\lambda) = \frac{1}{2\pi n} \left| \sum_{t=1}^n x_t e^{-i\lambda t} \right|^2, \quad \lambda \in [-\pi, \pi].$$

Este estimador se evalúa en frecuencias específicas $\lambda_j = \frac{2\pi j}{n}$, para $j = 1, \dots, \lfloor \frac{n}{2} \rfloor$. Aunque $I_n(\lambda)$ es un estimador insesgado de $2\pi f(\lambda)$, es importante destacar que no es consistente, ya que su varianza no tiende a cero cuando $n \rightarrow \infty$.

En particular, Brockwell and Davis (2016) muestran que si $\{X_t\}$ es un proceso estacionario con densidad espectral $f(\lambda)$ estrictamente positiva, entonces:

- $I_n(\lambda_1), \dots, I_n(\lambda_m)$ son asintóticamente independientes, - cada $I_n(\lambda_j)$ está aproximadamente distribuido como una exponencial con media $2\pi f(\lambda_j)$, es decir:

$$I_n(\lambda_j) \approx \text{Exp}(2\pi f(\lambda_j)). \quad \text{$$

Por tanto, para cada $\lambda \in (0, \pi)$ y cualquier $\varepsilon > 0$, se cumple que:

$$\mathbb{P}(|I_n(\lambda) - 2\pi f(\lambda)| > \varepsilon) \not\rightarrow 0, \quad \text{cuando } n \rightarrow \infty.$$

Esto confirma que $I_n(\lambda)$ no es un estimador consistente de $2\pi f(\lambda)$.

Una forma de obtener un estimador consistente es suavizar el periodograma, por ejemplo, promediando $I_n(\lambda)$ en un entorno de frecuencias cercanas:

$$\tilde{f}(\lambda) = \frac{1}{2\pi(2m+1)} \sum_{|j| \leq m} I_n(\lambda + 2\pi j/n),$$

donde $m = \sqrt{n}$, como propone Brockwell and Davis (2016).

Eliminar estas secciones de la seccion de modelo ARMA con y sin error de medición



5. Modelo ARMA sin errores de medición


Un **modelo ARMA** (Modelo Autorregresivo de Media Móvil) es un modelo clásico utilizado para describir series de tiempo estacionarias. Este modelo combina dos componentes fundamentales:

- Un componente **autorregresivo** (AR), que representa la dependencia de la serie con sus propios valores pasados.
- Un componente **de media móvil** (MA), que representa la influencia de perturbaciones aleatorias pasadas (ruido blanco).

Formalmente, una serie de tiempo $\{X_t\}$ sigue un modelo ARMA(p, q) si cumple con la siguiente ecuación:

$$X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q},$$

donde:

- X_t es el valor observado de la serie en el tiempo t ,
- ϕ_1, \dots, ϕ_p son los coeficientes autorregresivos,
- $\theta_1, \dots, \theta_q$ son los coeficientes de media móvil,
- Sea $\{Z_t\}$ un proceso de ruido blanco con media cero y varianza constante, es decir, $\{Z_t\} \sim WN(0, \sigma_Z^2)$ (Brockwell and Davis, 2016) . 

El modelo ARMA permite modelar series estacionarias en las que el comportamiento presente depende tanto de valores pasados de la serie como de errores aleatorios previos. Su utilidad radica en la capacidad de capturar dinámicas temporales complejas mediante un número relativamente reducido de parámetros, siendo ampliamente utilizado en áreas como la economía, la ingeniería, la meteorología y la estadística aplicada.

5.1. Función de Autocovarianza

Un proceso $\{X_t\}$ definido por un modelo ARMA(p, q) se dice **causal** si puede expresarse como una combinación lineal infinita de los valores pasados del ruido blanco $\{Z_t\}$, es decir, si existe una secuencia de coeficientes $\{\psi_j\}$ tal que

$$X_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j}, \quad \text{con} \quad \sum_{j=0}^{\infty} |\psi_j| < \infty.$$

Esta representación existe si las raíces del polinomio autorregresivo

$$\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p$$

se encuentran fuera del círculo unitario, es decir, si $\phi(z) \neq 0$ para todo $|z| \leq 1$ (Brockwell and Davis, 2016).

Los coeficientes ψ_j se pueden calcular recursivamente a partir de los polinomios $\phi(B)$ y $\theta(B)$ mediante:

$$\psi_j = \theta_j + \sum_{k=1}^p \phi_k \psi_{j-k}, \quad j = 0, 1, 2, \dots,$$

donde se toma $\theta_j = 0$ para $j > q$, y $\psi_j = 0$ para $j < 0$ por convención.

Finalmente, la función de autocovarianza del proceso se puede obtener a partir de los coeficientes ψ_j mediante:

$$\gamma(h) = \sigma^2 \sum_{j=0}^{\infty} \psi_j \psi_{j+|h|}, \quad h \in \mathbb{Z},$$

donde σ^2 es la varianza del ruido blanco $\{Z_t\}$ (Brockwell and Davis, 2016).

5.2. Densidad Espectral de Modelo ARMA

Para un proceso ARMA(p, q) que sea **causal** y que satisfaga la ecuación:

$$\phi(B)X_t = \theta(B)Z_t,$$

La densidad espectral viene dada por:

$$f_X(\lambda) = \frac{\sigma_Z^2}{2\pi} \cdot \frac{|\theta(e^{-i\lambda})|^2}{|\phi(e^{-i\lambda})|^2}$$

siendo σ_Z^2 la varianza del proceso de ruido blanco $\{Z_t\} \sim WN(0, \sigma_Z^2)$.

Esta representación se conoce como *densidad espectral racional*, ya que se expresa como el cociente de dos polinomios trigonométricos. Sin embargo, esta expresión sólo es válida cuando se cumple la condición de **causalidad**, es decir, que el polinomio autorregresivo $\phi(z)$ no se anula en el círculo unitario Brockwell and Davis (2016).

Esta expresión muestra cómo la energía (varianza) del proceso se distribuye según la frecuencia λ . En particular, $|\theta(e^{-i\lambda})|^2$ y $|\phi(e^{-i\lambda})|^2$ representan el módulo cuadrado de las funciones de transferencia evaluadas en la frecuencia angular λ .

6. Modelo ARMA con errores de medición

Un **modelo ARMA con errores de medición** extiende la formulación clásica ARMA al considerar que las observaciones están contaminadas por ruido.

Consideremos un proceso X_t que sigue un modelo ARMA dado por:

$$\phi(B)X_t = \theta(B)Z_t,$$

donde $\{Z_t\} \sim WN(0, \sigma_z^2)$. A partir de esta señal, se observa una serie ruidosa Y_t , definida como:

$$Y_t = X_t + W_t,$$

donde $W_t \sim WN(0, \sigma_w^2)$ representa ruido blanco independiente del proceso Z_t .

donde:

- Y_t es el proceso observado,
- X_t es un proceso no observable que sigue un modelo ARMA(p, q),
- $Z_t \sim WN(0, \sigma_z^2)$ es un ruido blanco,
- $W_t \sim WN(0, \sigma_w^2)$ es un error de medición independiente de Z_t ,
- $\phi(B)$ y $\theta(B)$ son los polinomios autorregresivo y de media móvil, respectivamente,
- B es el operador rezago ($BX_t = X_{t-1}$).

Este tipo de modelo es útil cuando las observaciones no representan directamente el proceso generador subyacente, debido a errores inherentes al mecanismo de medición. Brockwell and Davis (1991)

6.1. Relación entre W_t y Z_t

De acuerdo con Liu and He (2011), las variables aleatorias W_t (error de medición) y Z_t (ruido blanco del proceso subyacente) son ruidos blancos mutuamente independientes con media cero, es decir:

$$Z_t \sim WN(0, \sigma_z^2), \quad W_t \sim WN(0, \sigma_w^2),$$

y se cumple que:

$$\text{Cov}(Z_t, W_s) = 0 \quad \text{para todo } t, s.$$

Este supuesto de independencia es fundamental en el modelo de series de tiempo con errores de medición, ya que permite obtener la densidad espectral del proceso observado como la suma de las densidades espectrales del proceso verdadero y del error de medición. En particular, si X_t satisface una ecuación AR(p) con ruido blanco Z_t , y si $Y_t = X_t + W_t$ es la serie observada con error, entonces la densidad espectral de Y_t está dada por:

$$f_Y(\lambda) = f_X(\lambda) + f_W(\lambda),$$

donde $f_X(\lambda) = \frac{\sigma_z^2}{2\pi|A(e^{i\lambda})|^2}$ y $f_W(\lambda) = \frac{\sigma_w^2}{2\pi}$, siguiendo la notación de Liu and He (2011).

6.2. Función de Autocovarianza

Sea $\{X_t\}$ un proceso ARMA con errores de medición, como se define en la sección 6. Entonces, la función de autocovarianza del proceso observado $\{Y_t\}$ está dada por:

$$\gamma_Y(h) = \begin{cases} \gamma_X(0) + \sigma_w^2, & \text{si } h = 0, \\ \gamma_X(h), & \text{si } h \neq 0, \end{cases}$$

donde $\gamma_X(h)$ es la función de autocovarianza del proceso $\{X_t\}$ verdadero (Brockwell and Davis, 1991).

6.3. Densidad Espectral de Modelo ARMA

Sea X_t un proceso ARMA causal con errores de medición, como se definió en la sección 6. La densidad espectral del proceso observado Y_t está dada por:

$$f_Y(\lambda) = f_X(\lambda) + f_W(\lambda),$$

donde:

- $f_X(\lambda) = \frac{\sigma_z^2}{2\pi|A(e^{i\lambda})|^2}$ es la densidad espectral del proceso ARMA verdadero,
- $f_W(\lambda) = \frac{\sigma_w^2}{2\pi}$ es la densidad espectral del error de medición.

Por lo tanto, el efecto del error de medición es adicionar una componente constante a la densidad espectral del proceso observado en todas las frecuencias (Liu and He, 2011).

7. Simulaciones

En esta sección se busca evaluar el comportamiento del periodograma promedio y la estimación de la densidad espectral en comparación con la densidad espectral teórica de modelos ARMA, a través de simulaciones Monte Carlo para distintos modelos ARMA estacionarios. A continuación, se describe el procedimiento seguido y los resultados obtenidos.

7.1. Simulaciones de Modelos ARMA sin errores de medición

Para cada modelo considerado, se generaron $B = 300$ réplicas de longitud $n = 500$, utilizando la función `arma.sim()` de R. En cada réplica, se ajustó un modelo ARMA mediante máxima verosimilitud, y se calcularon:

- El periodograma $I_n(\lambda)$ para un conjunto denso de frecuencias $\lambda \in [-\pi, \pi]$,
- La función de densidad espectral estimada $\hat{f}(\lambda)$ a partir de los parámetros ajustados,
- La función de densidad espectral teórica $f(\lambda)$ del modelo verdadero.

Se utilizó una rejilla de 1000 frecuencias igualmente espaciadas en el intervalo $[-\pi, \pi]$. La varianza del término de ruido fue fijada en $\sigma^2 = 2$ para todos los casos.

Para comparar los estimadores, se graficó el promedio del periodograma y de la PSD estimada, junto con la PSD teórica.

7.1.1. Modelos AR(1)

Se simularon $B = 300$ réplicas de un proceso autorregresivo de primer orden (AR(1)) con parámetro $\phi = 0,7$ y longitud $n = 500$, se fijó $\sigma^2 = 2$.

Para cada réplica, se estimaron los parámetros $\hat{\phi}$ y $\hat{\sigma}^2$ mediante máxima verosimilitud. Con dichos estimadores, se construyó la función de densidad espectral (PSD) estimada $\hat{f}(\lambda)$ y el periodograma $I_n(\lambda)$, evaluados en una rejilla de 1000 frecuencias igualmente espaciadas en el intervalo $[-\pi, \pi]$.

Posteriormente, se calculó el promedio de ambas estimaciones espectrales a lo largo de las 300 réplicas y se compararon con la función de densidad espectral teórica del modelo donde se utilizaron los valores reales de los parámetros: $\phi = 0,5$ y $\sigma^2 = 2$.

Esta comparación permite evaluar el comportamiento promedio de los estimadores espectrales bajo el supuesto de correcta especificación del modelo.

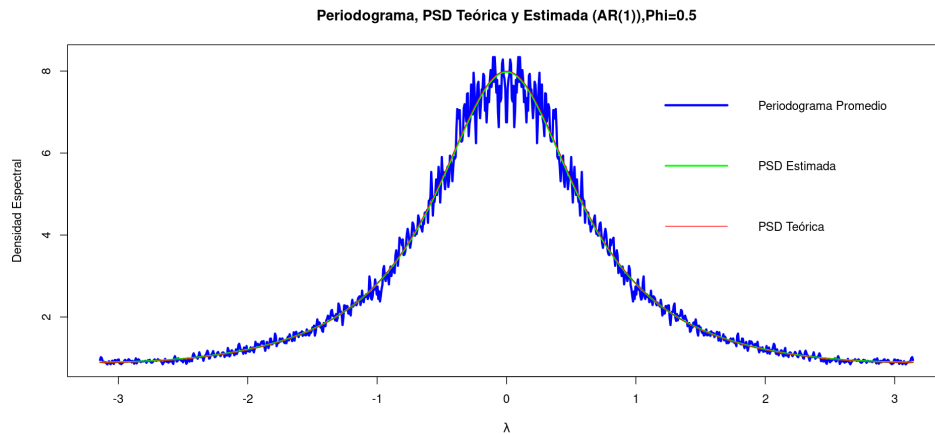


Figura 1: Comparación entre el periodograma promedio, la PSD estimada y la PSD teórica para el modelo AR(1) con $\phi = 0,5$.

Se presentan los resultados obtenidos a partir de un experimento de simulación diseñado para evaluar el desempeño de los estimadores espectrales bajo un modelo AR(1) con parámetro autorregresivo $\phi = 0,5$. En la figura se observa:

- El **periodograma promedio** (línea azul), que exhibe la habitual variabilidad alta en torno a la PSD teórica. Aunque sigue correctamente la forma general, evidencia su ineficiencia como estimador directo debido a su falta de consistencia.
- La **PSD estimada promedio** (línea verde), construida a partir de los parámetros ajustados en cada réplica, muestra una concordancia notable con la **PSD teórica** (línea roja), validando empíricamente la consistencia de los estimadores paramétricos bajo el supuesto de correcta especificación del modelo.

Este resultado es coherente con la teoría: en el caso de un modelo correctamente especificado y ruido blanco gaussiano, el estimador MLE de los parámetros del modelo AR(1) es consistente y asintóticamente normal, lo que se traduce en una estimación precisa de la función espectral. La diferencia entre el periodograma promedio y la PSD estimada pone de manifiesto las ventajas de la estimación paramétrica frente al uso directo del periodograma.

Finalmente, la precisión observada en las bajas frecuencias se justifica por la mayor concentración de energía del proceso AR(1) en torno a $\lambda = 0$ cuando $\phi > 0$, lo que también permite observar cómo las estimaciones espectrales logran capturar correctamente el patrón de dependencia temporal.

7.1.2. Modelo AR(1)

Se simularon $B = 300$ réplicas de un proceso autorregresivo de primer orden (AR(1)) con parámetro $\phi = 0,7$ y longitud $n = 500$, se fijó $\sigma^2 = 2$.

Para cada réplica, se estimaron los parámetros $\hat{\phi}$ y $\hat{\sigma}^2$ mediante máxima verosimilitud. Con dichos estimadores, se construyó la función de densidad espectral (PSD) estimada $\hat{f}(\lambda)$ y el periodograma $I_n(\lambda)$, evaluados en una rejilla de 1000 frecuencias igualmente espaciadas en el intervalo $[-\pi, \pi]$.

Posteriormente, se calculó el promedio de ambas estimaciones espectrales a lo largo de las 300 réplicas y se compararon con la función de densidad espectral teórica del modelo donde se utilizaron los valores reales de los parámetros: $\phi = 0,7$ y $\sigma^2 = 2$.

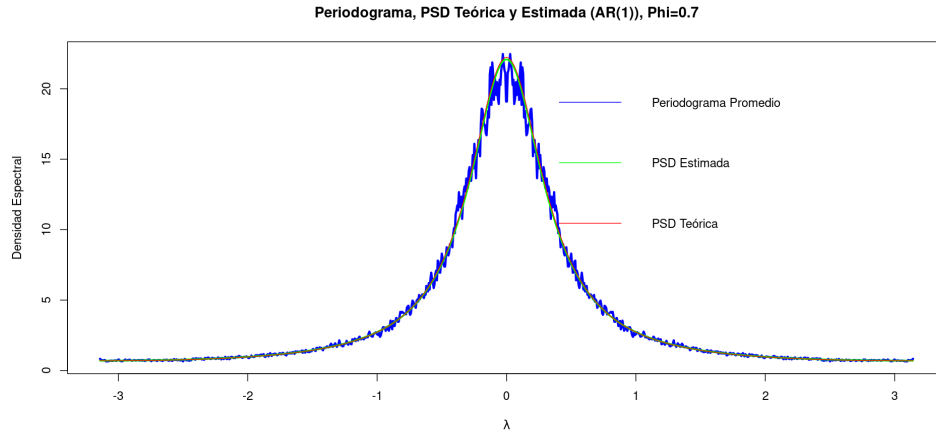


Figura 2: Comparación entre el periodograma promedio, la PSD estimada y la PSD teórica para el modelo AR(1) con $\phi = 0,7$.

Se presenta una simulación diseñada para evaluar el desempeño de los estimadores espectrales bajo un modelo AR(1) con parámetro autorregresivo $\phi = 0,7$.

En la figura se observa:

- El **periodograma promedio** (línea azul), calculado como promedio sobre las 300 réplicas simuladas. Muestra una elevada variabilidad en torno a la forma teórica de la densidad espectral, en especial cerca de $\lambda = 0$, donde se concentra la mayor energía. Esta variabilidad es consistente con la conocida falta de suavización y alta varianza del periodograma como estimador no paramétrico de la PSD.
- La **PSD estimada promedio** (línea verde), obtenida a partir de los parámetros estimados por máxima verosimilitud en cada réplica, evidencia una notable concordancia con la **PSD teórica** (línea roja), validando la consistencia de los estimadores paramétricos en el contexto de modelos correctamente especificados.

Desde un punto de vista teórico, cuando el modelo es correctamente especificado y el ruido es gaussiano, el estimador MLE de los parámetros del modelo AR(1) es consistente y asintóticamente eficiente. En consecuencia, la estimación paramétrica de la función de densidad espectral resulta más precisa que el uso directo del periodograma, lo cual queda demostrado gráficamente en la figura.

Cabe destacar que, en este caso, el valor de $\phi = 0,7$ implica una dependencia temporal más fuerte que en el caso anterior ($\phi = 0,5$), lo que se refleja en una mayor concentración de energía en las frecuencias bajas. Esta característica es capturada correctamente tanto por la PSD estimada como por el periodograma, aunque con diferencias en la suavidad y precisión. En particular, la densidad espectral presenta un pico más alto y estrecho en torno a $\lambda = 0$, lo que es esperable en procesos AR(1) con alta persistencia temporal.

7.1.3. Modelo AR(2)

En este caso se consideró un proceso autorregresivo de orden 2 (AR(2)) con parámetros $\phi_1 = 1,9634$ y $\phi_2 = -0,984644$, los cuales corresponden a una estructura oscilatoria acentuada en la serie.

El ajuste se realizó utilizando modelos AR(2), y se estimaron los parámetros en cada una de las réplicas generadas mediante simulación.

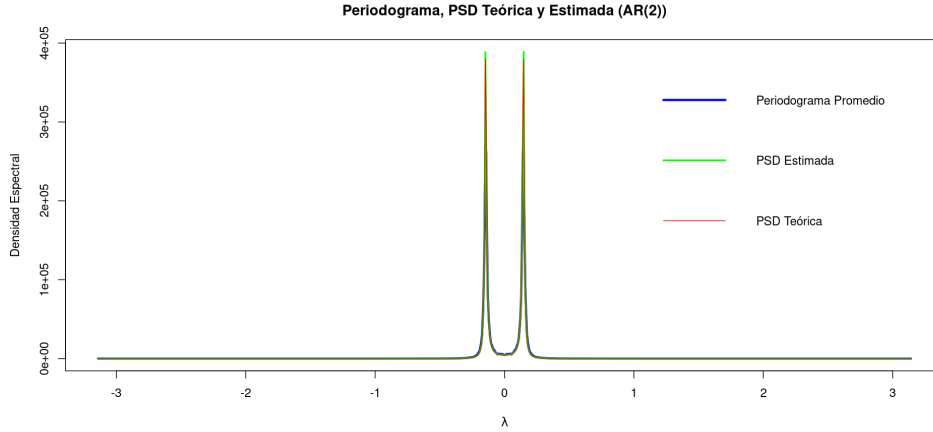


Figura 3: Comparación entre el periodograma promedio, la PSD estimada y la PSD teórica para el modelo AR(2) con coeficientes $\phi_1 = -1,2$ y $\phi_2 = 0,3$.

Se Muestran los resultados del experimento de simulación realizado bajo un modelo autorregresivo de orden dos, AR(2), cuyos coeficientes se fijaron como $\phi_1 = -1,2$ y $\phi_2 = 0,3$. Este conjunto de parámetros genera un proceso oscilatorio, con un espectro de potencia caracterizado por la presencia de dos picos espectrales bien definidos, en frecuencias distintas de cero.

- El **periodograma promedio** (línea azul) evidencia dos picos espectrales marcados, lo cual es característico de procesos con comportamiento oscilatorio. Aunque el periodograma captura la localización general de los picos, presenta la habitual variabilidad que lo convierte en un estimador inconsistente de la densidad espectral.
- La **PSD estimada promedio** (línea verde) coincide estrechamente con la **PSD teórica** (línea roja) en toda la banda de frecuencias, incluso en las zonas de alta energía espectral. Esto confirma la alta precisión del enfoque paramétrico cuando el modelo es correctamente especificado, en especial para procesos con estructura compleja como el AR(2).

Este resultado es coherente con la teoría espectral: los modelos AR(2) pueden generar densidades espectrales con múltiples picos (bandas dominantes de frecuencia), reflejo de un comportamiento cíclico en la serie temporal. En este caso, la consistencia de los estimadores MLE permite capturar con gran precisión esta estructura, a diferencia del periodograma, cuya varianza alta lo hace menos fiable.

En síntesis, los resultados muestran que las estimaciones paramétricas logran una aproximación altamente precisa a la PSD real.

7.1.4. Modelo ARMA(3,2)

Finalmente, se analizó un proceso ARMA(3,2) con parámetros $\phi = (0,04, -0,1, -0,3)$ y $\theta = (0,5, 0,3)$.

En cada simulación se estimaron todos los parámetros del modelo, y posteriormente se comparó el promedio del periodograma con la densidad espectral de potencia (PSD) estimada y la PSD teórica.

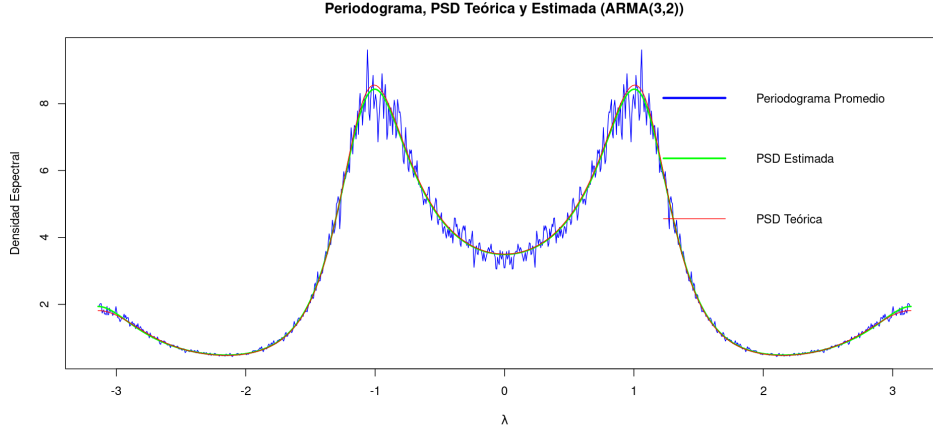


Figura 4: Comparación entre el periodograma promedio, la PSD estimada y la PSD teórica para el modelo ARMA(3,2) con coeficientes $\phi = (0,04, -0,1, -0,3)$ y $\theta = (0,5, 0,3)$.

En cada réplica se ajustó un modelo ARMA(3,2) mediante máxima verosimilitud (MLE), y a partir de los parámetros estimados se calcularon dos estimadores espectrales:

- El **periodograma promedio** (línea azul), que representa un estimador no paramétrico conocido por su alta varianza.
- La **densidad espectral estimada** (línea verde), construida a partir de los parámetros estimados.

Ambos se comparan con la **densidad espectral teórica** del modelo (línea roja), que se obtiene analíticamente a partir de los verdaderos parámetros.

- La **PSD estimada** se ajusta muy bien a la **PSD teórica**, lo que confirma que los estimadores paramétricos logran recuperar correctamente la forma de la densidad espectral cuando el modelo está bien especificado y se cumplen los supuestos.
- El **periodograma promedio** sigue la forma general, pero presenta oscilaciones y ruido, especialmente cerca de las frecuencias donde la densidad espectral alcanza sus máximos.
- El gráfico muestra **dos picos importantes** en la PSD, lo que indica que el modelo ARMA(3,2) genera una estructura cíclica más compleja que los modelos AR(1).

7.2. Simulaciones de Modelos ARMA con errores de medición

Con el objetivo de analizar el impacto que tienen los errores de medición sobre la estimación de parámetros y sobre el comportamiento espectral de los modelos, se realizaron

experimentos de simulación para tres procesos distintos: un modelo AR(1), un modelo AR(2) y un modelo ARMA(3,2). En todos los casos, se introdujo un componente de error aditivo gaussiano a la serie observada.

Cada simulación se desarrolló bajo el siguiente esquema general:

- Se generaron $B = 300$ réplicas de longitud $n = 500$ para cada modelo, utilizando parámetros reales conocidos.
- Para cada réplica, se simuló una serie latente $\{X_t\}$ según el modelo especificado, y se agregó un error de medición $\{w_t\} \sim \mathcal{N}(0, \sigma_w^2)$, con $\sigma_w^2 = 2$, obteniendo así una serie observada $Y_t = X_t + w_t$.
- Luego, se ajustó un modelo del mismo tipo (AR(1), AR(2) o ARMA(3,2)) a los datos contaminados $\{Y_t\}$, sin considerar explícitamente la presencia del error de medición en el ajuste.
- Se registraron los parámetros estimados $\hat{\phi}$ y $\hat{\theta}$, los cuales se analizaron posteriormente para evaluar el sesgo inducido por la presencia del error.

Se busca evidenciar cómo la contaminación de las observaciones afecta la calidad de las estimaciones paramétricas y, en consecuencia, la estimación de la densidad espectral del proceso. En las siguientes secciones se presentan los resultados para cada uno de los modelos evaluados.

7.2.1. Modelos AR(1)

Se simula un modelo AR(1) con $\phi = 0,5$, al cual se le agrega un error de medición gaussiano con varianza $\sigma_e^2 = 2$. A partir de la serie observada $Y_t = X_t + w_t$, se ajusta nuevamente un modelo AR(1) y se estiman los parámetros $\hat{\phi}$ y $\hat{\sigma}^2$.

Con esas estimaciones se calcula la densidad espectral de potencia (PSD) para cada réplica. También se calcula el periodograma de cada serie contaminada. Ambos se promedian al final del experimento.

Además, se incluye la PSD teórica con error de medición, que considera el aumento de energía espectral introducido por el ruido aditivo. Finalmente, se comparan gráficamente las tres curvas: periodograma promedio, PSD estimada y PSD teórica corregida, para analizar cómo el error de medición afecta la estimación espectral.

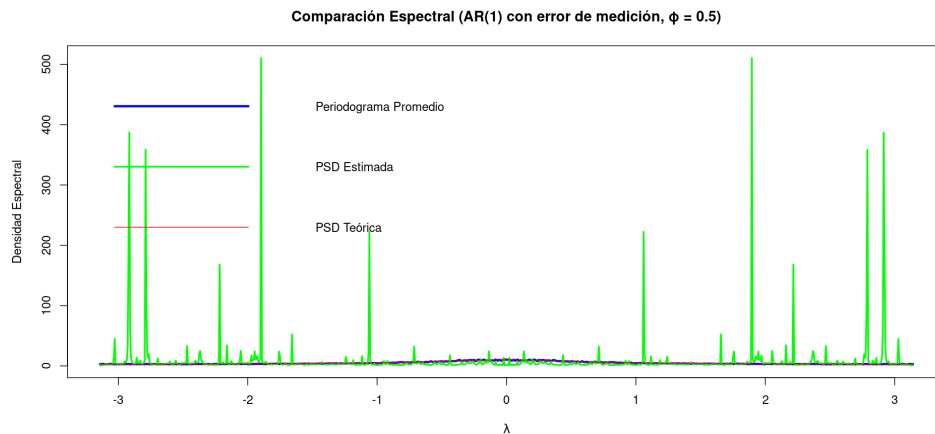


Figura 5: Comparación entre el periodograma promedio, la PSD estimada y la PSD teórica para el modelo AR(1) con coeficiente $\phi = 0,5$ y error de medición aditivo.

Se presenta la comparación entre tres estimaciones espectrales obtenidas a partir de múltiples simulaciones del proceso Y_t :

- El **periodograma promedio** (línea azul),
 - La **densidad espectral estimada** (verde),
 - Y la **densidad espectral teórica** del proceso verdadero X_t (línea roja).
- El **periodograma promedio** logra capturar de manera global la forma del espectro verdadero, aunque elevado debido a la contribución aditiva del término $\frac{\sigma_w^2}{2\pi}$.
 - La **PSD estimada** presenta fluctuaciones más pronunciadas, producto de la varianza del error y del comportamiento estocástico del proceso. Esto es esperable, ya que el error de medición introduce ruido adicional en todo el rango de frecuencias.
 - La **PSD teórica**, calculada directamente a partir de la fórmula del AR(1), representa la verdadera estructura espectral del proceso sin la influencia del error.

Los resultados de la simulación son coherentes con la teoría. La adición de un error de medición genera un incremento uniforme en la densidad espectral, lo cual se refleja en la diferencia vertical entre la PSD teórica y las otras estimaciones. A pesar de este efecto, la estructura del espectro se conserva, lo que valida el procedimiento de simulación y análisis espectral bajo la presencia de errores de medición.

7.2.2. Modelos AR(2)

Se presenta un estudio de simulación cuyo objetivo es analizar el impacto de los errores de medición en la estimación espectral de un proceso $AR(2)$,

Se realizaron $B = 300$ simulaciones de longitud $n = 500$ con parámetros verdaderos $\phi = (1,9634, -0,984644)$ y $\sigma_z^2 = 2$. En cada iteración, se generó una realización del proceso X_t y se le añadió ruido blanco independiente con varianza $\sigma_w^2 = 2$ para obtener la serie Y_t .

Posteriormente, se estimaron los coeficientes $\hat{\phi}_1$ y $\hat{\phi}_2$ a partir de la serie ruidosa Y_t mediante el ajuste de un modelo $AR(2)$ con `arima`. También se estimó la densidad espectral de potencia (PSD) utilizando los coeficientes estimados, y se comparó con el periodograma promedio y la densidad espectral teórica del modelo con error.

Este experimento permite analizar gráficamente cómo el error de medición afecta la forma del espectro, y cuantitativamente cómo distorsiona los parámetros del modelo. El gráfico resultante muestra que el efecto del error se traduce en un aumento constante en el nivel espectral, como predice la teoría, y que los parámetros estimados se alejan de sus valores reales.

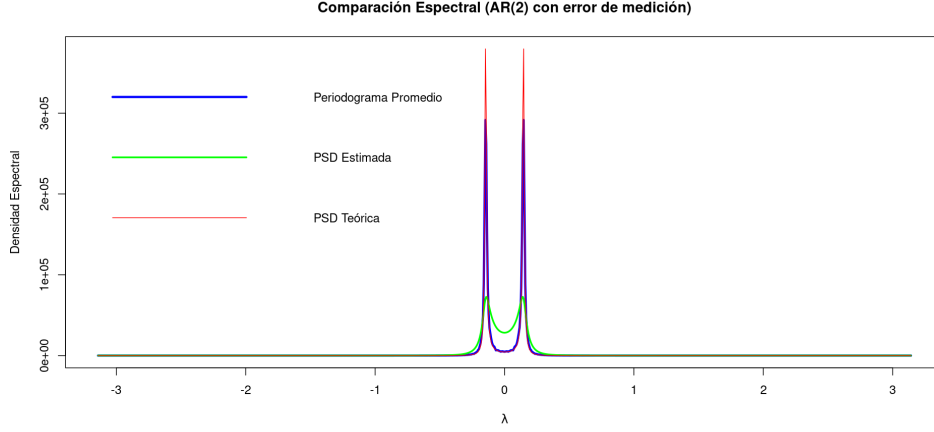


Figura 6: Comparación entre el periodograma promedio, la PSD estimada y la PSD teórica para el modelo $AR(2)$ con coeficientes $\phi = (1,9634, -0,9846)$ y error de medición aditivo.

En el gráfico se observa que las tres curvas presentan una forma característica con dos picos simétricos alrededor de $\lambda = 0$, lo cual es coherente con la estructura oscilatoria de un proceso $AR(2)$ con raíces complejas.

Tal como predice la teoría, la presencia del error de medición provoca un **aumento uniforme** en los niveles espectrales. Este desplazamiento vertical es visible en la distancia constante entre la PSD teórica (roja) y la curva base del proceso (aproximada por la PSD estimada y el periodograma promedio).

- El **periodograma promedio** logra capturar razonablemente bien la forma de la PSD, aunque presenta una leve subestimación de los picos debido al suavizado inherente al promediado.
- La **PSD estimada** muestra un ajuste más preciso en la zona de los picos, pero también se ve ligeramente afectada por la variabilidad introducida por el error.
- La **PSD teórica** se ubica sistemáticamente por encima de las otras curvas, reflejando la suma teórica del espectro del proceso original más la contribución plana del ruido blanco.

El error de medición actúa como un componente aditivo que eleva la densidad espectral de forma constante en todas las frecuencias, sin modificar la estructura oscilatoria fundamental del proceso $AR(2)$. Las estimaciones obtenidas permiten recuperar la forma espectral general, validando el procedimiento de ajuste y estimación bajo error.

7.2.3. Modelos ARMA(3,2)

Con el objetivo de analizar el impacto de los errores de medición en la estimación espectral de un proceso ARMA, se llevó a cabo una simulación basada en un modelo ARMA(3,2) con parámetros verdaderos $\phi = (0,04, -0,1, -0,3)$ y $\theta = (0,5, 0,3)$, asumiendo una varianza del ruido innovador de $\sigma^2 = 2$.

Primero, se simularon 300 trayectorias del proceso X_t , generado bajo el modelo ARMA(3,2), sin presencia de errores de medición. Para cada trayectoria, se ajustó un modelo ARMA(3,2) y se almacenaron los parámetros estimados, así como las estimaciones de la densidad espectral utilizando los coeficientes obtenidos.

Posteriormente, se incorporó un error de medición aditivo $W_t \sim \mathcal{WN}(0, \sigma_w^2)$ con $\sigma_w^2 = 2$ a cada trayectoria, obteniendo así una serie observada $Y_t = X_t + W_t$. Se ajustaron nuevamente modelos ARMA(3,2) a las series contaminadas y se compararon los periodogramas promedio, las densidades espectrales estimadas y la densidad espectral teórica ajustada para este caso.

Este procedimiento permite observar cómo la presencia de ruido blanco aditivo afecta la estimación espectral, así como los parámetros del modelo subyacente. Además, se presenta una comparación entre los coeficientes estimados en ambas situaciones (con y sin error) respecto de los valores reales.

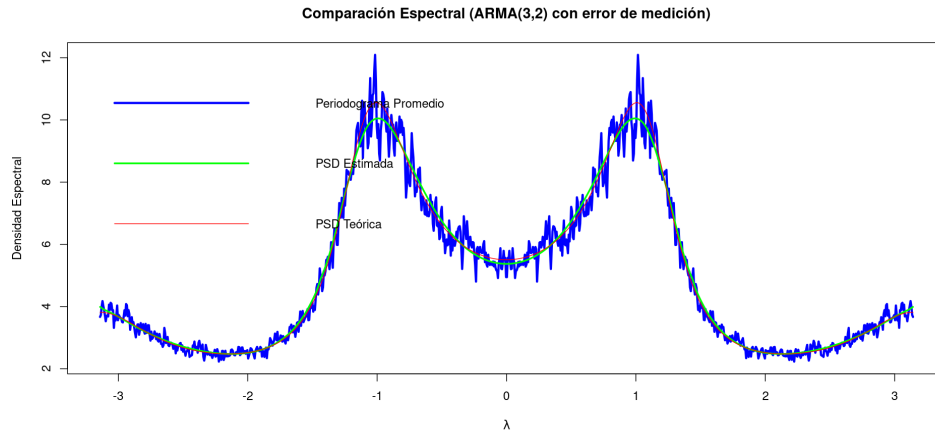


Figura 7: Comparación entre el periodograma promedio, la PSD estimada y la PSD teórica para el modelo AR(2) con coeficientes $\phi = (1,9634, -0,9846)$, bajo la presencia de error de medición aditivo con varianza $\sigma_w^2 = 2$.

Se observa que la curva roja representa la verdadera densidad espectral del proceso subyacente X_t , ajustada por la presencia de ruido blanco aditivo de varianza $\sigma_w^2 = 2$. La curva verde, correspondiente a la PSD estimada a partir de los modelos ajustados sobre las series contaminadas Y_t , muestra una excelente aproximación a la densidad teórica, lo que valida la capacidad del modelo ARMA para recuperar la estructura espectral incluso bajo la presencia de ruido.

El periodograma promedio (línea azul), como es característico, exhibe una mayor variabilidad debido a su naturaleza no suavizada. No obstante, su forma general sigue el patrón de la PSD, destacando los principales picos de energía espectral en las frecuencias dominantes.

7.3. Comparación sin error con error



7.3.1. Modelo AR(1)

```
> cat(" Real          :", round(phi_real, 4), "\n")
Real          : 0.5
> cat(" Sin error de medición :", round(prom_phi_xt, 4), "\n")
Sin error de medición : 0.4953
> cat(" Con error de medicion :", round(prom_phi_yt, 4), "\n")
Con error de medicion : 0.2825
```

7.3.2. Modelo AR(2)

```
> cat(" Real      :", round(phi_real, 4), "\n")
Real      : 1.9634 -0.9846
> cat(" Sin error de medición :", round(prom_phi_xt, 4), "\n")
Sin error de medición : 1.9594 -0.9806
> cat(" Con error de medicion :", round(prom_phi_yt, 4), "\n")
Con error de medicion : 1.8561 -0.8777
```

7.3.3. Modelo ARMA(3,2)

```
Coeficientes (phi):
> cat(" Real      :", round(phi_real, 4), "\n")
Real      : 0.04 -0.1 -0.3
> cat(" Sin error :", round(prom_phi_xt, 4), "\n")
Sin error : 0.0357 -0.0991 -0.2966
> cat(" Con error :", round(prom_phi_yt, 4), "\n\n")
Con error : 0.2592 -0.2313 -0.1388

> cat("Coeficientes (theta):\n")
Coeficientes (theta):
> cat(" Real      :", round(theta_real, 4), "\n")
Real      : 0.5 0.3
> cat(" Sin error :", round(prom_theta_xt, 4), "\n")
Sin error : 0.5012 0.2977
> cat(" Con error :", round(prom_theta_yt, 4), "\n")
Con error : 0.0101 0.215
```

8. Revisión de literatura

8.1. Paquetes Y funciones en R para modelos Arma

8.2. Modelos arma sin errores de medición

Cuadro 1: Paquetes y funciones en R para ajuste de modelos ARMA sin error de medición

Paquete	Función(es)	Descripción	Ejemplo
stats	arima()	Función base de R para ajustar modelos ARMA(p,q) mediante máxima verosimilitud o suma de cuadrados condicionales (Cryer and Chan, 2008).	Sí
forecast	Arima()	Versión moderna del ajuste ARMA con interfaz amigable, selección automática o manual de órdenes y opciones adicionales para predicción (Cuartas Rendón, 2022).	Sí
TSA	armasubsets()	No ajusta un modelo final, sino que explora posibles combinaciones de órdenes AR y MA (Cryer and Chan, 2008).	Sí
GAS	UniGAS(), MultiGAS()	Los modelos GAS proveen una estructura unificada para modelar la dinámica en los parámetros condicionales de series temporales. Se pueden usar para capturar comportamientos similares a ARMA, GARCH y otros.” (Ardia et al., 2021).	Sí
tseries	arma()	Ajuste de modelos ARMA con opciones avanzadas. Alternativa a <code>arima()</code> del paquete <code>stats</code> (Trapletti and Hornik, 2003).	Sí

8.3. Modelos Arma con errores de medición

Cuadro 2: Paquetes y funciones en R para modelos ARMA con error de medición

Paquete	Función(es)	Descripción
MARSS	MARSS(), MARSSkfas()	Permite estimar modelos ARMA con error de medición bajo una estructura de espacio de estados. Usa filtro de Kalman y EM para la estimación (Holmes et al., 2012).

9. Paquetes o funciones en Python que ajusten modelo ARMA

9.1. Modelos Arma sin errores de medición

Cuadro 3: Paquetes y funciones en Python para ajuste de modelos ARMA

Paquete	Función(es)	Descripción	Ejemplo
statsmodels	SARIMAX()	Biblioteca estadística más utilizada para series de tiempo en Python. Permite ajustar modelos ARMA, ARIMA y SARIMAX mediante máxima verosimilitud (Seabold and Perktold, 2010).	Sí
pmdarima	auto_arima()	Interfaz automática inspirada en R para seleccionar órdenes p y q de modelos ARMA/ARIMA. Basado en statsmodels, orientado a usuarios de R (Smith, 2022).	Sí
arch	ARX(), ARXResults	Diseñado para econometría financiera; permite ajuste de modelos ARMA con regresores exógenos, además de modelos ARCH/GARCH (Sheppard, 2022).	?
skforecast	ForecasterAutoreg()	Utiliza regresores scikit-learn para ajustar modelos autoregresivos. Útil para extender AR/ARMA con aprendizaje supervisado (Palacios & Salinas, 2023).	?

9.2. Modelos Arma Con errores de medición



10. Ejemplos en R modelos Arma sin errores de medición

10.1. Ejemplo función arima()

```
# serie ARMA(2,1) con 200 observaciones
set.seed(123)

# Parámetros del modelo ARMA(2,1)
phi <- c(0.5, -0.3)
theta <- 0.4
sigma <- 1
n <- 200

# Simular serie con arima.sim()
serie <- arima.sim(n = n,
                   model = list(ar = phi, ma = theta),
                   sd = sigma)

# Graficar la serie simulada
plot.ts(serie, main = "Serie simulada ARMA(2,1)", ylab = "Valor", xlab = "Tiempo")

# Ejemplo 1: stats::arima()

# Ajustar modelo ARMA(2,1) con la función base
modelo_stats <- arima(serie, order = c(2, 0, 1)) # AR=2, MA=1, sin diferencia
summary(modelo_stats)
Call:
arima(x = serie, order = c(2, 0, 1))

Coefficients:
          ar1          ar2          ma1  intercept
      0.4978   -0.3316   0.3668      0.0344
s.e.  0.1182    0.0917   0.1183    0.1142

sigma^2 estimated as 0.971:  log likelihood = -281.32,  aic = 570.63
```

10.2. Ejemplo función Arima()

```
#Ejemplo 2: forecast::Arima()

library(forecast)

# Ajustar el modelo ARMA(2,1)
modelo_forecast <- Arima(serie, order = c(2, 0, 1))
summary(modelo_forecast)
Series: serie
```


ARIMA(2,0,1) with non-zero mean

Coefficients:

	ar1	ar2	ma1	mean
	0.4978	-0.3316	0.3668	0.0344
s.e.	0.1182	0.0917	0.1183	0.1142

sigma^2 = 0.9908: log likelihood = -281.32

AIC=572.63 AICc=572.94 BIC=589.12

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.001212093	0.9854058	0.7772929	20.40507	163.1105	0.7430036	0.003642729

10.3. Ejemplo función armasubsets()

```
set.seed(92397)
```

```
library(TSA)
```

```
test <- arima.sim(model = list(ar = c(rep(0,11), .8),  
                                ma = c(rep(0,11), .7)),  
                  n = 120)
```

```
res <- armasubsets(y = test, nar = 14, nma = 14, ar.method = "ols")  
plot(res)
```

10.4. Ejemplo función UniGAS()

```
> library(GAS)
```

```
> # Simular serie de volatilidad condicional con distribución t
```

```
> set.seed(786)
```

```
> T.sim <- 1000
```

```
> Dist <- "std" # Student-t
```

```
> kappa <- c(0.0, log(1.5), log(7 - 2.01))
```

```
> A <- matrix(c(0.001 , 0.0 , 0.0 ,  
               0.0 , 0.01 , 0.0 , 0.0 , 0.0 , 0.0 ), 3, byrow = TRUE)
```

```
> B = matrix(c(0.7 , 0.0 , 0.0 ,  
               0.0 , 0.98, 0.0 ,  
               0.0 , 0.0 , 0.0), 3, byrow = TRUE) # Matrix of autoregressive parameters
```

```
> Sim = UniGASSim(fit = NULL, T.sim, kappa, A, B, Dist, ScalingType = "Identity")
```

```
> Sim
```

```
-----  
-          Univariate GAS Sim          -  
-----
```

Model Specification:

```

T = 1000
Conditional distribution: std
Score scaling type: Identity
-----
Parameters:
vKappa:
  location      scale      shape
0.0000000 0.4054651 1.6074359
mA:
      [,1] [,2] [,3]
[1,] 0.001 0.00  0
[2,] 0.000 0.01  0
[3,] 0.000 0.00  0

```

10.5. Ejemplo función arma()

```
> y2 <- arima.sim(n = 100, model = list(ar = c(0.6, -0.4), ma = 0.3))
> ajuste <- arma(y2, order = c(2, 1))
> print(ajuste)
```

Call:

```
arma(x = y2, order = c(2, 1))
```

Coefficient(s):

ar1	ar2	ma1	intercept
0.4899	-0.5312	0.2374	-0.1082

11. Ejemplos en R modelos Arma con errores de medición

11.1. Ejemplo función MARSS

```
# ----- CREAR MATRIZ A -----
crear_matriz_A <- function(r, p, param) {
  col_phi = matrix(nrow = r, ncol = 1)
  for (i in 1:r) {
    if (i <= p) {
      col_phi[i, 1] = param[i]
    } else {
      col_phi[i, 1] = 0
    }
  }
  m_identidad <- diag(rep(1, r - 1))
  v_ceros <- rep(0, r - 1)
  relleno_A <- rbind(m_identidad, v_ceros)
  A <- cbind(col_phi, relleno_A)
  return(A)
}
```

```
# ----- CREAR MATRIZ G -----
crear_matriz_G <- function(r, p, q, param) {
  col_theta = matrix(nrow = r - 1, ncol = 1)
  fila_1 = c(1)
  for (i in 1:(r - 1)) {
    if (i <= q) {
      col_theta[i, 1] = param[i + p]
    } else {
      col_theta[i, 1] = 0
    }
  }
  G <- rbind(fila_1, col_theta)
  return(G)
}
```

```
# ----- FUNCIÓN DE VEROSIMILITUD -----
logLikelihood <- function(param, y, p, q) {
  y1 = y - mean(y)
  r = max(p, q + 1)
  A = crear_matriz_A(r, p, param)
  C = matrix(c(1, rep(0, r - 1)), nrow = 1, ncol = r)
  G = crear_matriz_G(r, p, q, param)
  R = 0
  Q = (param[p + q + 1])^2 * G %*% t(G)
  m1 = matrix(0, nrow = r, ncol = 1)
```

```

if (r == 1) {
  P1 = matrix(var(y1), nrow = r)
} else {
  P1 = matrix(diag(rep(var(y1), r)), nrow = r)
}

ny = 1
nx = r

f_kalman = enssm::ACQR_kfilter(y1, A, C, Q, R, m1, P1, nx, ny)
return(-f_kalman$loglik)
}

# ----- ESTIMACIÓN FINAL -----
estim_ARMA <- function(v_iniciales, serie, p, q) {
  r = max(p, q + 1)

  estim = optim(v_iniciales, logLikelihood, serie, p, q,
               gr = NULL, method = "BFGS",
               hessian = TRUE,
               control = list(trace = 1, REPORT = 1, maxit = 100))

  param_est = estim$par
  M_ARMA = matrix(nrow = p + q, ncol = 1)
  M_ARMA[1:p] = estim$par[1:p]

  for (i in 1:p) {
    rownames(M_ARMA)[i] = paste0("phi_", i)
  }

  if (q > 0) {
    for (i in 1:q) {
      rownames(M_ARMA)[p + i] = paste0("theta_", i)
    }
  }

  colnames(M_ARMA) = "estimate"

  M_var = matrix(estim$par[p + q + 1]^2, ncol = 1, nrow = 1)
  rownames(M_var) = "var(w_t)"
  colnames(M_var) = "estimate"

  standard_errors = sqrt(diag(solve(estim$hessian)))
  MLE = cbind(M_ARMA, "standard_error" = standard_errors)

  z = qnorm(1 - 0.05 / 2)
  IC_1 = MLE[, 1] - z * standard_errors
  IC_2 = MLE[, 1] + z * standard_errors

```

```

MLE = cbind(MLE, IC_1, IC_2)

p_value = 2 * (1 - pnorm(abs(MLE[, 1] / standard_errors)))
MLE = cbind(MLE, p_value)

return(MLE)
}
library(fpp2)

# Ver el dataset 'uschange'
data("uschange")
head(uschange)

# Extraer solo la serie de consumo
consum <- uschange[, "Consumption"]

# Graficar la serie
autoplot(consum) + ggtitle("Serie original de consumo") +
  ylab("Cambio porcentual") + xlab("Tiempo")

# Ya que es una serie de cambios porcentuales por trimestre, es estacionaria
# (como dice el texto de tu imagen), pero también podemos hacerle la prueba ADF:

library(tseries)
adf.test(consum)

# Definir valores iniciales para estimar un modelo ARMA(1,3)
v_iniciales <- c(0.3, 0.1, 0.1, 0.1, 1)

# Estimar el modelo con tu función estim_ARMA()
resultado <- estim_ARMA(v_iniciales, consum, p = 1, q = 3)
Error en loadNamespace(x): no hay paquete llamado 'enssm'
print(round(resultado, 6))

```

A pesar de que en la literatura se presentan funciones como `estim_ARMA()` y `simul_ARMA()` para estimar modelos ARMA con error de medición bajo una formulación de espacio de estados (?), dichas funciones dependen de rutinas internas como `ACQR_kfilter()` del paquete `enssm`, el cual no está disponible públicamente en CRAN ni en otros repositorios conocidos. Esto impide la reproducción directa del procedimiento en R, ya que la función clave para implementar el filtro de Kalman no puede ser cargada ni utilizada. Por esta razón, no fue posible ajustar el modelo siguiendo exactamente dicha implementación.

12. Ejemplos en python modelos Arma sin errores de medición

12.1. Ejemplo función SARIMAX()

```
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.arima_process import ArmaProcess

# Simulación de una serie ARMA(2,2)
np.random.seed(42)

# Coeficientes AR y MA
ar_params = np.array([0.75, -0.25])
ma_params = np.array([0.65, 0.35])

# Estilo requerido para ArmaProcess: AR con signo negativo
ar = np.r_[1, -ar_params]
ma = np.r_[1, ma_params]

arma_process = ArmaProcess(ar, ma)
y = arma_process.generate_sample(nsample=300)

# Ajuste con SARIMAX como ARMA(2,2): d=0 indica sin integración
modelo = SARIMAX(y, order=(2, 0, 2), enforce_stationarity=True, enforce_invertibility=True)
ajuste = modelo.fit(dispatch=False)

# Mostrar resumen
print(ajuste.summary())
```

SARIMAX Results

```
=====
Dep. Variable:          y      No. Observations:          300
Model:                SARIMAX(2, 0, 2)  Log Likelihood      -419.130
Date:                 Thu, 24 Jul 2025  AIC                  848.259
Time:                 02:34:06          BIC                  866.778
Sample:               0              HQIC                  855.671
                   - 300
Covariance Type:      opg
=====
```

	coef	std err	z	P> z	[0.025	0.975]
ar.L1	0.8308	0.151	5.515	0.000	0.536	1.126
ar.L2	-0.3411	0.112	-3.056	0.002	-0.560	-0.122
ma.L1	0.4812	0.156	3.089	0.002	0.176	0.787
ma.L2	0.2612	0.105	2.488	0.013	0.055	0.467
sigma2	0.9513	0.070	13.557	0.000	0.814	1.089

```
=====
```

Ljung-Box (L1) (Q):	0.06	Jarque-Bera (JB):	6.51
Prob(Q):	0.81	Prob(JB):	0.04
Heteroskedasticity (H):	1.38	Skew:	0.21
Prob(H) (two-sided):	0.11	Kurtosis:	3.59

=====

12.2. Ejemplo función SARIMAX()

```
import numpy as np
import pmdarima as pm
from pmdarima.datasets import load_wineind

# Cargar una serie de tiempo (índice de producción de vino)
wineind = load_wineind().astype(np.float64)

# Ajustar modelo auto-ARIMA (estacional con period 12)
model = pm.auto_arima(
    wineind,
    start_p=1, start_q=1,
    max_p=3, max_q=3,
    m=12,
    start_P=0,
    seasonal=True,
    d=1, D=1,
    trace=True,
    error_action='ignore',
    suppress_warnings=True,
    stepwise=True
)

print(model.summary())
Performing stepwise search to minimize aic
ARIMA(1,1,1)(0,1,1)[12] : AIC=3066.492, Time=0.62 sec
ARIMA(0,1,0)(0,1,0)[12] : AIC=3131.408, Time=0.05 sec
ARIMA(1,1,0)(1,1,0)[12] : AIC=3097.884, Time=0.15 sec
ARIMA(0,1,1)(0,1,1)[12] : AIC=3066.329, Time=0.14 sec
ARIMA(0,1,1)(0,1,0)[12] : AIC=3089.456, Time=0.06 sec
ARIMA(0,1,1)(1,1,1)[12] : AIC=3067.457, Time=0.26 sec
ARIMA(0,1,1)(0,1,2)[12] : AIC=3067.481, Time=0.38 sec
ARIMA(0,1,1)(1,1,0)[12] : AIC=3071.631, Time=0.16 sec
ARIMA(0,1,1)(1,1,2)[12] : AIC=inf, Time=1.93 sec
ARIMA(0,1,0)(0,1,1)[12] : AIC=3117.921, Time=0.27 sec
ARIMA(0,1,2)(0,1,1)[12] : AIC=3065.533, Time=0.20 sec
ARIMA(0,1,2)(0,1,0)[12] : AIC=3087.883, Time=0.07 sec
ARIMA(0,1,2)(1,1,1)[12] : AIC=3066.239, Time=0.34 sec
ARIMA(0,1,2)(0,1,2)[12] : AIC=3066.373, Time=0.53 sec
ARIMA(0,1,2)(1,1,0)[12] : AIC=3070.728, Time=0.16 sec
ARIMA(0,1,2)(1,1,2)[12] : AIC=inf, Time=1.95 sec
```



```

ARIMA(1,1,2)(0,1,1)[12]          : AIC=3066.424, Time=0.41 sec
ARIMA(0,1,3)(0,1,1)[12]          : AIC=3066.351, Time=0.45 sec
ARIMA(1,1,3)(0,1,1)[12]          : AIC=3068.295, Time=0.84 sec
ARIMA(0,1,2)(0,1,1)[12] intercept : AIC=3066.787, Time=0.23 sec

```

Best model: `ARIMA(0,1,2)(0,1,1)[12]`

Total fit time: 9.212 seconds

SARIMAX Results

```

=====
Dep. Variable:          y      No. Observations:
Model:          SARIMAX(0, 1, 2)x(0, 1, [1], 12)  Log Likelihood      -1
Date:              Thu, 24 Jul 2025      AIC              3
Time:              02:48:04      BIC              3
Sample:              0      HQIC              3
                  - 176

Covariance Type:      opg
=====

```

	coef	std err	z	P> z	[0.025	0.975]
ma.L1	-0.5756	0.041	-13.952	0.000	-0.656	-0.495
ma.L2	-0.1065	0.048	-2.224	0.026	-0.200	-0.013
ma.S.L12	-0.3848	0.054	-7.156	0.000	-0.490	-0.279
sigma2	7.866e+06	7.01e+05	11.228	0.000	6.49e+06	9.24e+06

```

=====
Ljung-Box (L1) (Q):          2.84      Jarque-Bera (JB):          18.05
Prob(Q):                    0.09      Prob(JB):              0.00
Heteroskedasticity (H):      1.17      Skew:                 -0.55
Prob(H) (two-sided):         0.56      Kurtosis:             4.21
=====

```

Referencias

- Ardia, D., Boudt, K., Catania, L., and Mullen, S. (2021). *GAS: Generalized Autoregressive Score Models*. CRAN. R package version 0.2.2.
- Brockwell, P. J. and Davis, R. A. (1991). *Time Series: Theory and Methods*. Springer Series in Statistics. Springer, New York, 2nd edition.
- Brockwell, P. J. and Davis, R. A. (2016). *Introduction to Time Series and Forecasting*. Springer, 3rd edition.
- Cryer, J. D. and Chan, K.-S. (2008). *Time Series Analysis: With Applications in R*. Springer, 2nd edition.
- Cuartas Rendón, S. (2022). Modelación de modelos arma(p, q) en r. https://rpubs.com/scuartasr/ST_03. RPubs.
- Holmes, E. E., Ward, E. J., and Wills, K. (2012). Marss: Multivariate autoregressive state-space modeling. *The R Journal*, 4(1):11–19.
- Liu, Z. and He, S. (2011). On estimation and selection for autoregressive models with measurement errors. *Statistica Sinica*, 21(3):1041–1065.
- Seabold, S. and Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, pages 57–61.
- Smith, T. G. (2022). pmdarima: Arima estimators for python. Python package 'pmdarima', <https://alkaline-ml.com/pmdarima/>. Doc. Quickstart Examples.
- Trapletti, A. and Hornik, K. (2003). *tseries: Time Series Analysis and Computational Finance*. R package version 0.10-54.