

# Índice

<b>1. ACF, PACF y densidad espectral de algunos modelos lineales</b>	<b>3</b>
1.1. Proceso autorregresivo ARMA(p,q) . . . . .	3
1.2. Proceso ARMA(1,1) . . . . .	4
1.2.1. ARMA(1,1) (Ejemplo 1) . . . . .	5
1.2.2. ARMA(1,1) (Ejemplo 2) . . . . .	5
1.2.3. ARMA(1,1) (Ejemplo 3) . . . . .	6
1.2.4. ARMA(1,1) (Ejemplo 4) . . . . .	7
<b>2. La especificación del modelo se decide según las características de los datos.</b>	<b>7</b>
<b>3. Herramientas para desvelar propiedades de una serie temporal</b>	<b>8</b>
3.1. Análisis gráfico . . . . .	8
3.1.1. Gráfico de la serie temporal . . . . .	8
3.1.2. Gráfico rango-media . . . . .	8
3.2. Determinación del orden de integración . . . . .	9
3.2.1. Análisis gráfico . . . . .	9
3.2.2. Contrastes formales sobre el orden de integración: DF, ADF y KPSS . . . . .	13
3.3. ACF muestral . . . . .	15
3.4. PACF muestral . . . . .	15
3.5. Otras herramientas estadísticas . . . . .	15
3.5.1. Estadísticos descriptivos . . . . .	15
3.5.2. Contraste de normalidad Jarque-Bera . . . . .	16
<b>4. Nota final</b>	<b>16</b>
<b>5. Función de autocovarianzas para un ARMA(p,q)</b>	<b>16</b>

# Lección 6. Modelos ARMA. Herramientas estadísticas

Marcos Bujosa

1 de octubre de 2025

## Resumen

Examinaremos la ACF, la PACF y la densidad espectral de procesos ARMA (que son una generalización de los procesos AR y MA). Después repasaremos algunas herramientas estadísticas útiles para tratar de identificar un modelo que se ajuste adecuadamente a los datos de una serie temporal: gráficos, contrastes de raíz unitaria, ACF y PACF muestrales, estadísticos descriptivos y contrastes de normalidad.

- ([slides](#)) — ([html](#)) — ([pdf](#)) — ([mybinder](#))

## Carga de algunos módulos de python y creación de directorios auxiliares

---

```
# Para trabajar con los datos y dibujarlos necesitamos cargar algunos módulos de python
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib as mpl
# definimos parámetros para mejorar los gráficos
mpl.rcParams['text', usetex=False]
import matplotlib.pyplot as plt # data visualization
```

---

- Directorio auxiliar para albergar las figuras de la lección: para publicar la lección como pdf o página web, necesito los gráficos como ficheros .png alojados algún directorio específico:

---

```
imagenes_leccion = "./img/lecc06" # directorio para las imágenes de la lección
import os
os.makedirs(imagenes_leccion, exist_ok=True) # crea el directorio si no existe
```

---

## Gráficos para las ACF, PACF y densidades espectrales teóricas

Cargamos las funciones auxiliares (véase la carpeta `src/`)

---

```
import warnings
warnings.filterwarnings("ignore", category=UserWarning)
%run -i ./src/analisis_armas.py
```

---

# 1. ACF, PACF y densidad espectral de algunos modelos lineales

## 1.1. Proceso autorregresivo ARMA(p,q)

Sea  $\mathbf{X}$  el proceso estocástico estacionario solución de la ecuación en diferencias:

$$\phi * \mathbf{X} = \theta * \mathbf{U}$$

donde  $\mathbf{U} \sim WN(0, \sigma^2)$ ; el polinomio *autorregresivo*  $\phi$  tiene grado  $p$  con  $\phi_0 = 1$  (y raíces fuera del círculo unidad), y el polinomio *de media móvil*  $\theta$  es de grado  $q$  con  $\theta_0 = 1$ ; y donde  $\phi$  y  $\theta$  no tienen raíces comunes.

Si  $\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p$  y  $\theta(z) = 1 - \theta_1 z - \cdots - \theta_q z^q$ , entonces

$$(1 - \phi_1 B - \cdots - \phi_p B^p) X_t = (1 - \theta_1 B - \cdots - \theta_q B^q) U_t;$$

y por tanto

$$X_t = U_t + \sum_{j=1}^p \phi_j X_{t-j} + \sum_{j=1}^q -\theta_j U_{t-j}.$$

Como las raíces de  $\phi$  están fuera del círculo unidad, es decir, como *el polinomio AR es “invertible”* ( $\phi^{-\triangleright} = \phi^{-1} \in \ell^1$ ) entonces  $\mathbf{X}$  tiene una representación como MA( $\infty$ ):

$$\phi * \mathbf{X} = \theta * \mathbf{U} \Rightarrow \mathbf{X} = \frac{\theta}{\phi} * \mathbf{U} \Rightarrow X_t = U_t + \sum_{j=1}^{\infty} -\psi_j U_{t-j};$$

donde  $\psi = \phi^{-1} * \theta = (\underline{1}, -\psi_1, -\psi_2, -\psi_3, \dots)$  tiene grado  $\infty$ .

Y como las raíces de  $\theta$  están fuera del círculo unidad, es decir, como *el polinomio MA es “invertible”* ( $\theta^{-\triangleright} = \theta^{-1} \in \ell^1$ ) entonces  $\mathbf{X}$  tiene una representación como AR( $\infty$ ):

$$\phi * \mathbf{X} = \theta * \mathbf{U} \Rightarrow \frac{\phi}{\theta} * \mathbf{X} = \mathbf{U} \Rightarrow X_t = U_t + \sum_{j=1}^{\infty} \varphi_j X_{t-j};$$

donde  $\varphi = \theta^{-1} * \phi = (\underline{1}, -\varphi_1, -\varphi_2, -\varphi_3, \dots)$  tiene grado  $\infty$ .

En un ARMA( $p, q$ ), por tener representación MA( $\infty$ ):

$E(X_t) = 0$  para todo  $t \in \mathbb{Z}$  y

$$\gamma = \sigma^2 \frac{\theta(z)}{\phi(z)} * \frac{\theta(z^{-1})}{\phi(z^{-1})} = \sigma^2 \psi(z) * \psi(z^{-1}) \text{ donde } \psi = \frac{\theta}{\phi};$$

es decir,  $\gamma_k = \sigma^2 \sum_{j=0}^{\infty} \psi_{j+|k|} \psi_j$ ;  $k \in \mathbb{Z}$  (grado  $\infty$  y cogrado  $-\infty$ ).

$$\rho = \frac{1}{\gamma_0} \gamma$$

$$f(\omega) = \frac{\sigma^2}{2\pi} \frac{\theta(e^{-i\omega}) \cdot \theta(e^{i\omega})}{\phi(e^{-i\omega}) \cdot \phi(e^{i\omega})} = \frac{1}{2\pi} \sum_{h=0}^{\infty} \gamma_h \cos(h\omega); \quad \text{donde } \omega \in [-\pi, \pi].$$

(suma infinita de cosenos)

En un ARMA( $p, q$ ), por tener representación AR( $\infty$ ):

su PACF,  $\pi$ , también es una secuencia con grado  $\infty$  y cogrado  $-\infty$ .

En la sección [Función de autocovarianzas para un ARMA\(p,q\)](#) se demuestra que para un proceso ARMA( $p, q$ )

$$\phi(B)\gamma_k = \begin{cases} \sigma^2 (\theta(z) * \psi(z^{-1}))_k & \text{para } k \leq q \\ 0 & \text{para } k > q \quad (\text{como en un AR}) \end{cases}$$

Así, en cuanto a la ACF  $\rho$ :

- De  $\rho_1$  a  $\rho_q$  dependen de los  $q$  parámetros de  $\theta$  y los  $p$  parámetros de  $\phi$
- Los  $p$  valores de  $\rho_q$  a  $\rho_{q-p+1}$  son los valores iniciales para resolver la ecuación en diferencias de [Yule-Walker](#)  $\phi(B)\rho_k = 0$  (con  $k > q$ )
- Si  $q < p$  toda la secuencia  $\rho_j$  para  $j \geq 0$  decae exponencialmente o sinusoidalmente según  $\phi$  y los valores iniciales  $\rho_q$  a  $\rho_{q-p+1}$
- Si  $q \geq p$  los primeros  $q - p + 1$  valores iniciales  $\rho_0$  a  $\rho_{q-p}$  siguen una pauta diferente

En cuanto a la PACF  $\pi$ :

- Tiene grado  $\infty$  y cogrado  $-\infty$
- A partir de cierto retardo se comporta como la PACF de un MA( $q$ ), es decir, decae exponencialmente o sinusoidalmente.

## 1.2. Proceso ARMA(1,1)

Sea  $(1 - \phi z) * \mathbf{X} = (1 - \theta z) * \mathbf{U}$  con  $|\phi| < 1$  y  $|\theta| < 1$  y con  $\mathbf{U} \sim WN(0, \sigma^2)$ :

$$X_t = \phi X_{t-1} - \theta U_{t-1} + U_t.$$

- $\gamma = \sigma^2 \frac{(1-\theta z)*(1-\theta z^{-1})}{(1-\phi z)*(1-\phi z^{-1})}$  por tanto...
- $\gamma_0 = \sigma^2 \left(1 + \frac{(\theta+\phi)^2}{1+\phi^2}\right); \quad \gamma_1 = \sigma^2 \left(\phi + \theta + \frac{(\theta+\phi)^2\phi}{1+\phi^2}\right); \quad \gamma_k = \phi\gamma_{k-1} \quad \text{si } k > 1$
- $\rho = \frac{1}{\gamma_0}\gamma$
- $f(\omega) = \frac{\sigma^2}{2\pi} \frac{1+\theta^2-2\theta\cos(\omega)}{1+\phi^2-2\phi\cos(\omega)}; \quad \omega \in [-\pi, \pi]$
- $\pi$  decae geométricamente con  $\theta^k$

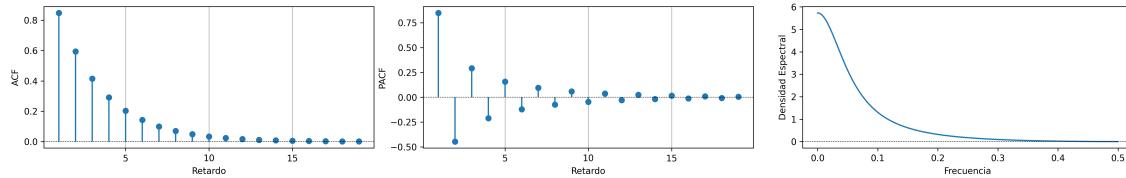
### 1.2.1. ARMA(1,1) (Ejemplo 1)

$$(1 - 0,7B)X_t = (1 + 0,8B)U_t; \quad \phi > 0; \quad \theta < 0$$

---

```
ar_params = [1, -0.7]
ma_params = [1, 0.8]
fig = plot_arma_parametric_diagnostics(ar_params, ma_params, sigma2=1, lags=20)
fig.savefig('./img/lecc06/ACF-arma11R.png', dpi=300, bbox_inches='tight')
```

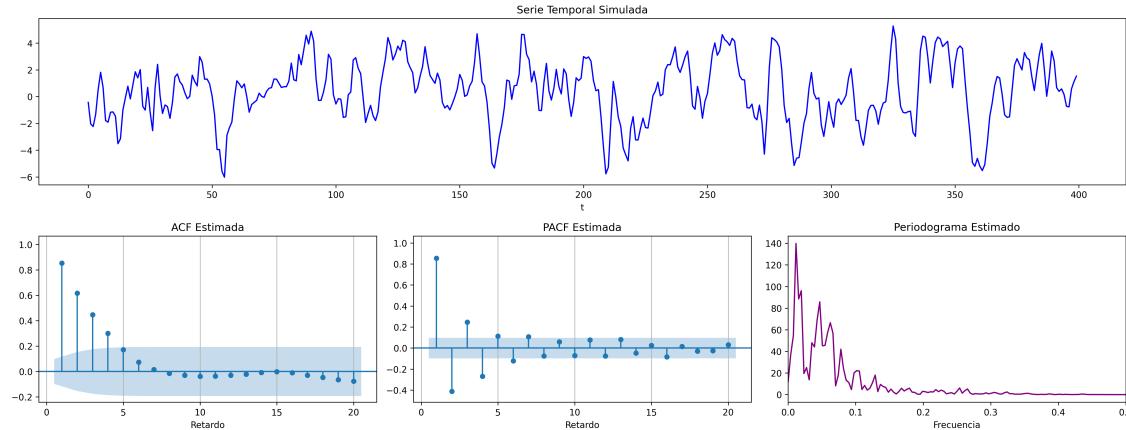
---




---

```
fig = plot_arma_analysis(ar_params, ma_params, seed=2026)
fig.savefig('./img/lecc06/Sim-arma11R.png', dpi=300, bbox_inches='tight')
```

---



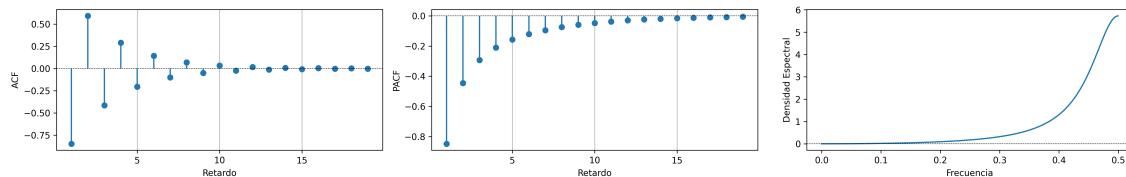
### 1.2.2. ARMA(1,1) (Ejemplo 2)

$$(1 + 0,7B)X_t = (1 - 0,8B)U_t; \quad \phi < 0; \quad \theta > 0$$

---

```
ar_params = [1, 0.7]
ma_params = [1, -0.8]
fig = plot_arma_parametric_diagnostics(ar_params, ma_params, sigma2=1, lags=20)
fig.savefig('./img/lecc06/ACF-arma11A.png', dpi=300, bbox_inches='tight')
```

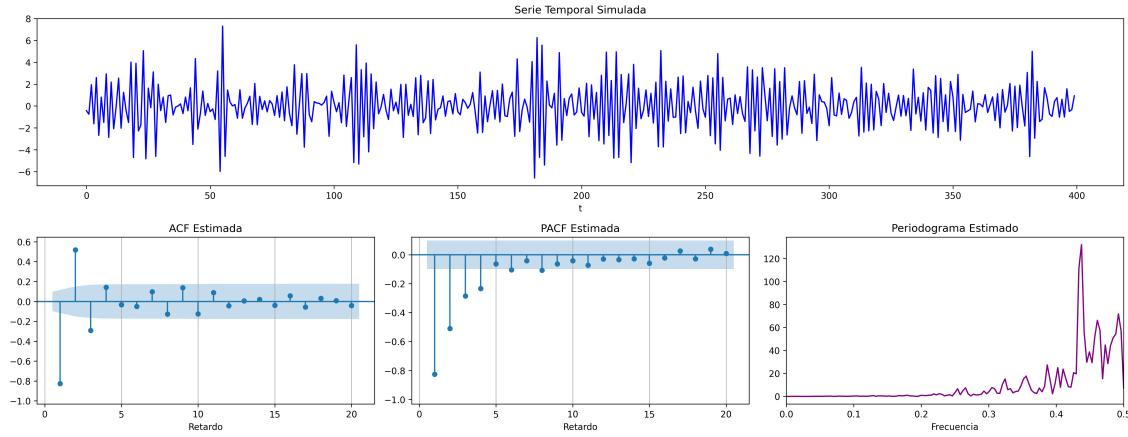
---



---

```
fig = plot_arma_analysis(ar_params, ma_params, seed=2026)
fig.savefig('./img/lecc06/Sim-arma11A.png', dpi=300, bbox_inches='tight')
```

---



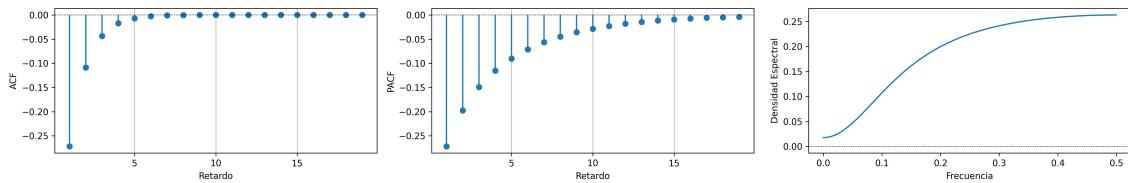
### 1.2.3. ARMA(1,1) (Ejemplo 3)

$$(1 - 0.4B)X_t = (1 - 0.8B)U_t; \quad \phi > 0; \quad \theta > 0$$

---

```
ar_params = [1, -0.4]
ma_params = [1, -0.8]
fig = plot_arma_parametric_diagnostics(ar_params, ma_params, sigma2=1, lags=20)
fig.savefig('./img/lecc06/ACF-arma11NVA.png', dpi=300, bbox_inches='tight')
```

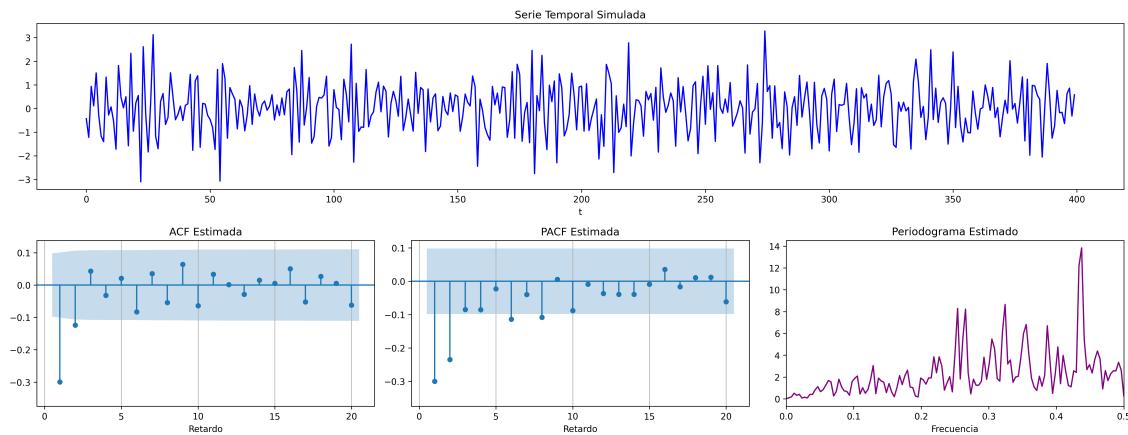
---




---

```
fig = plot_arma_analysis(ar_params, ma_params, seed=2026)
fig.savefig('./img/lecc06/Sim-arma11NVA.png', dpi=300, bbox_inches='tight')
```

---



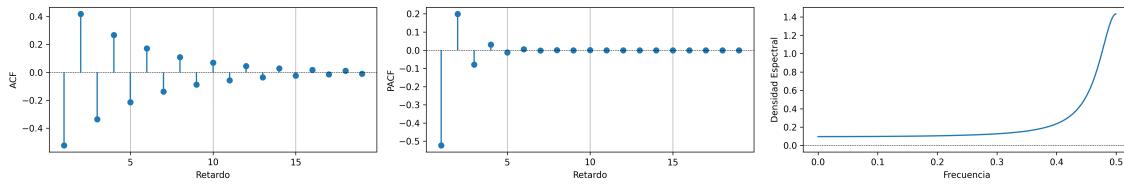
#### 1.2.4. ARMA(1,1) (Ejemplo 4)

$$(1 + 0,8B)X_t = (1 + 0,4B)U_t; \quad \phi < 0; \quad \theta < 0$$

---

```
ar_params = [1, 0.8]
ma_params = [1, 0.4]
fig = plot_arma_parametric_diagnostics(ar_params, ma_params, sigma2=1, lags=20)
fig.savefig('./img/lecc06/ACF-arma11RNV.png', dpi=300, bbox_inches='tight')
```

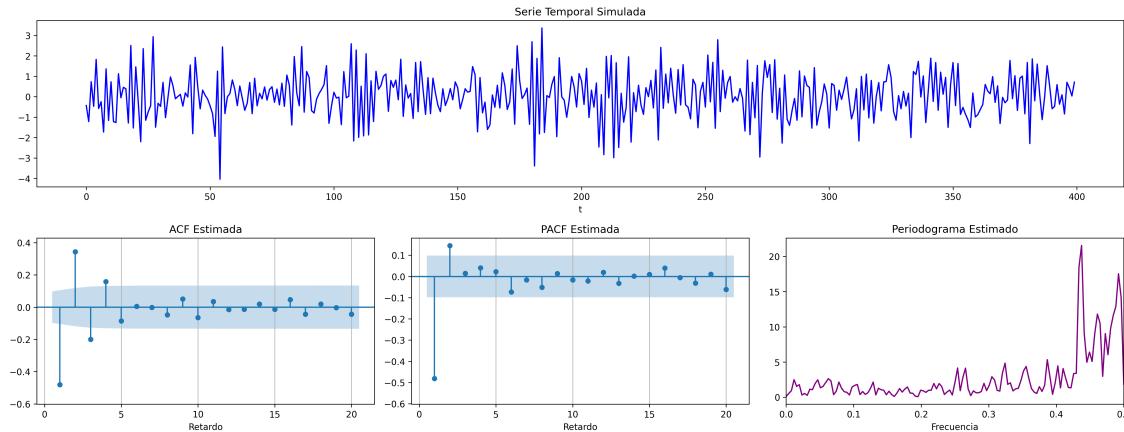
---




---

```
fig = plot_arma_analysis(ar_params, ma_params, seed=2026)
fig.savefig('./img/lecc06/Sim-arma11RNV.png', dpi=300, bbox_inches='tight')
```

---



## 2. La especificación del modelo se decide según las características de los datos.

¿Es la serie

- “estacionaria en media”?

- (y si lo es, ¿cuál es su media?)
- (y si no lo es, ¿cómo cambia o evoluciona su media?)

- “estacionaria en varianza”? (*homocedástica*)

- (y si lo es, ¿cuál es su varianza?)
- (y si es *heterocedástica*, ¿cómo cambia o evoluciona su varianza?)

¿Están sus valores correlados con su historia pasada (autocorrelados)? (ACF)

¿Qué correlación tienen los datos con los datos de  $k$  períodos atrás una vez descontado el efecto de los datos intermedios? (PACF)

Veamos algunas herramientas estadísticas para poder desvelar estas características.

### 3. Herramientas para desvelar propiedades de una serie temporal

#### 3.1. Análisis gráfico

##### 3.1.1. Gráfico de la serie temporal

Representa sus valores en el eje vertical ( $y$ ) frente a una escala temporal en el horizontal ( $x$ ). Es útil para detectar visualmente:

- tendencias y/o estacionalidad
- cambios de variabilidad
- valores atípicos (*outliers*)
  - el 95 % aprox. de una muestra de valores generados por una distribución normal debería estar comprendido entre  $\mu \pm 2\sigma$
  - la probabilidad de que una variable normal genere un valor fuera de las bandas de  $\mu \pm 3\sigma$  es 0,0023

(véase gráfico IBEX 35)

Es importante escalar y rotular adecuadamente los ejes y asegurar la comparabilidad entre series y gráficos distintos (si los hubiere).

##### 3.1.2. Gráfico rango-media

Cambios de variabilidad de una serie pueden evidenciarse en su gráfico temporal.

Pero también suelen verse bien en un gráfico rango-media, donde se representa:

- en eje  $x$ : nivel de la serie (normalmente la media de submuestras no solapadas).
- en eje  $y$ : dispersión de la serie (normalmente el rango de dichas submuestras).

Si los puntos se sitúan alrededor de una recta de pendiente positiva y la serie temporal no toma valores negativos, se deben tomar logaritmos.

Veamos el gráfico de la serie de pasajeros de líneas aéreas junto a su gráfico de rango mediana:

```
# Leer los datos
df = pd.read_csv('../datos/airline-passengers.csv')
df['Month'] = pd.to_datetime(df['Month'])
df = df.set_index(['Month'])

# Crear la figura y los subplots
fig, axs = plt.subplots(1, 2, figsize=(15, 5))
```

```

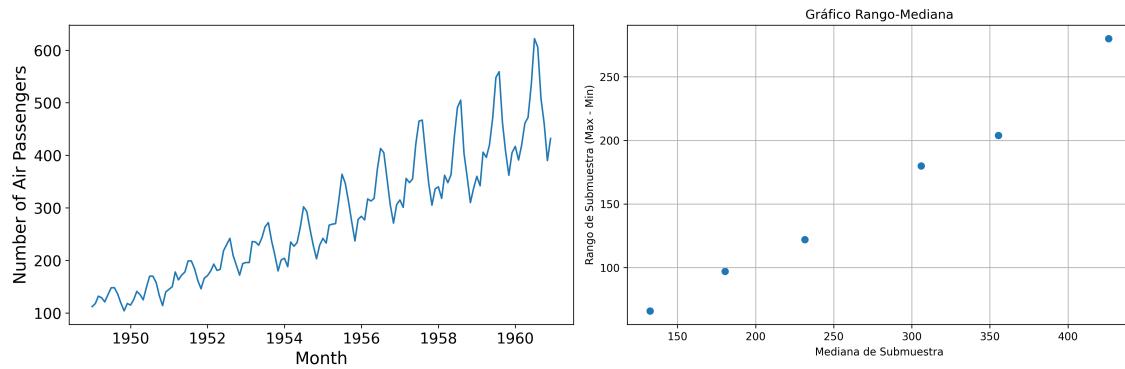
# Graficar la serie de tiempo
axs[0].plot(df['Passengers'])
axs[0].set_xlabel("Month", fontsize=16)
axs[0].set_ylabel(r"Number of Air Passengers", fontsize=16)
axs[0].tick_params(axis='both', labelsize=14)

# Graficar el rango-mediana
RangoMediana(df['Passengers'], ax=axs[1])

plt.tight_layout()
plt.savefig('./img/lecc06/rangoMedianaPasajeros.png', dpi=300, bbox_inches='tight')

```

---



El gráfico de rango media a veces se acompaña de una regresión de la dispersión sobre los niveles para medir la relación nivel-dispersión.

### 3.2. Determinación del orden de integración

Un proceso estocástico sin componentes deterministas es  $I(0)$  si tiene una representación ARMA estacionaria e invertible

Más formalmente, un proceso estocástico  $\mathbf{Y}$  que satisface  $Y_t - E(Y_t) = \sum_{j=0}^{\infty} \psi_j U_{t-j}$  se dice que es  $I(0)$  si  $\psi$  es absolutamente sumable y  $\sum_{j=0}^{\infty} \psi_j \neq 0$  (la primera condición garantiza que el proceso es estacionario, y la última condición excluye que el proceso no sea invertible, por ejemplo, excluye que  $\psi(z)$  sea  $1 - 1z$ ).

El *orden de integración* de un proceso estocástico  $\mathbf{Y}$  es el número de diferencias necesarias para transformarlo en un proceso  $I(0)$ .

Decidir adecuadamente el orden de integración es crucial en el análisis de series temporales.

Las herramientas utilizadas para tomar la decisión son

- el análisis gráfico
- los contrastes formales

#### 3.2.1. Análisis gráfico

---

```

# Leer los datos
df1 = pd.read_csv('..../datos/PoblacionAustralia.csv')

```

```

# Calcular las diferencias
df1['primera_diferencia'] = df1['pob'].diff()
df1['segunda_diferencia'] = df1['primera_diferencia'].diff()

# Crear la figura y los ejes, compartiendo el eje x
fig, axs = plt.subplots(3, 1, figsize=(15, 9), sharex=True) # 3 filas, 1 columna

# Graficar la serie temporal
axs[0].plot(df1['obs'], df1['pob'])
axs[0].set_ylabel('Habitantes', fontsize=14)
axs[0].set_title('Población australiana', fontsize=18)
axs[0].grid()

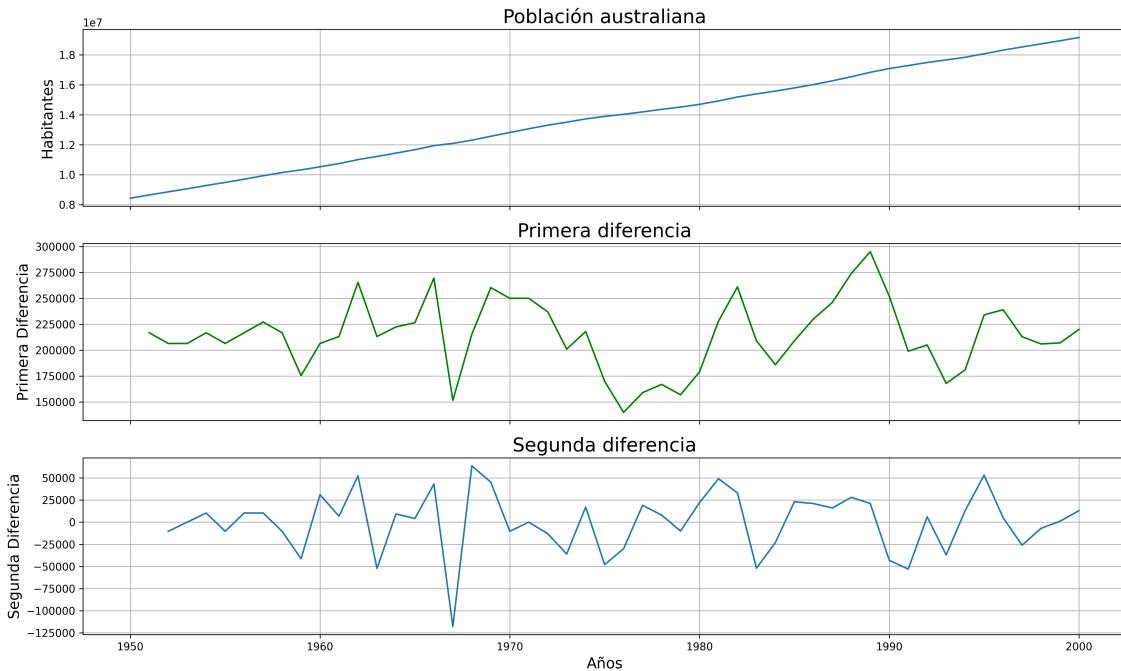
# Graficar la primera diferencia
axs[1].plot(df1['obs'], df1['primera_diferencia'], color='green')
axs[1].set_ylabel('Primera Diferencia', fontsize=14)
axs[1].set_title('Primera diferencia', fontsize=18)
axs[1].grid()

# Graficar la segunda diferencia
axs[2].plot(df1['obs'], df1['segunda_diferencia'])
axs[2].set_xlabel('Años', fontsize=14)
axs[2].set_ylabel('Segunda Diferencia', fontsize=14)
axs[2].set_title('Segunda diferencia', fontsize=18)
axs[2].grid()

# Ajustar el layout
plt.tight_layout()
plt.savefig('./img/lecc06/diferenciasPoblacion.png', dpi=300, bbox_inches='tight')

```

---



La serie de población  $y$  tiene una clara tendencia creciente (primer gráfico), que desaparece al tomar una diferencia ordinaria,

$$\nabla y = (1 - B) * y$$

(segundo gráfico). Bastar con tomar una primera diferencia de la serie de población para obtener

una nueva serie que se asemeja a la realización de un proceso estacionario.

No obstante, ¿qué pasa si tomamos una segunda diferencia ordinaria?

$$\nabla\nabla \mathbf{y} = \nabla^2 \mathbf{y} = (1 - B)^2 * \mathbf{y}$$

(segundo gráfico). Pues que la serie obtenida también es estacionaria, pero ojo, es un grave error tomar más diferencias de las necesarias al modelizar los datos. Se debe tomar el mínimo número de transformaciones que arrojen una serie “estacionaria” (recuerde que decir que una serie temporal es *estacionaria* es un abuso del lenguaje).

Si se toman más diferencias de las necesarias se obtiene un proceso *no invertible* (raíces unitarias en la parte MA que imposibilitan que el proceso tenga una representación como AR( $\infty$ ) causal), lo cual conduce a problemas de identificación y estimación.

---

```
# Leer los datos
df = pd.read_csv('../datos/airline-passengers.csv')

# Asegurarse de que la columna Month sea tipo datetime
df['Month'] = pd.to_datetime(df['Month'])

# Establecer la columna Month como índice
df.set_index('Month', inplace=True)

# Calcular el logaritmo de los pasajeros
df['log_passengers'] = np.log(df['Passengers'])

# Calcular la primera diferencia
df['primera_diferencia'] = df['log_passengers'].diff()

# Calcular la diferencia estacional (de orden 12)
df['diferencia_estacional'] = df['log_passengers'] - df['log_passengers'].shift(12)

# Calcular la primera diferencia de la diferencia estacional
df['primera_diferencia_estacional'] = df['diferencia_estacional'].diff()

# Crear la figura y los ejes, compartiendo el eje x
fig, axs = plt.subplots(4, 1, figsize=(15, 12), sharex=True) # 4 filas, 1 columna

# Graficar el logaritmo de la población
axs[0].plot(df.index, df['log_passengers'])
axs[0].set_ylabel(r'$\log(y)$', fontsize=14)
axs[0].set_title('Log de los Pasajeros', fontsize=18)
axs[0].grid()

# Graficar la primera diferencia
axs[1].plot(df.index, df['primera_diferencia'], color='black')
axs[1].set_ylabel(r'$\nabla y$)', fontsize=14)
axs[1].set_title('Primera diferencia (Log)', fontsize=18)
axs[1].grid()

# Graficar la diferencia estacional
axs[2].plot(df.index, df['diferencia_estacional'], color='red')
axs[2].set_ylabel(r'$\nabla_{12}(y) = (1 - B^{12})y$', fontsize=14)
axs[2].set_title('Diferencia estacional (Log)', fontsize=18)
axs[2].grid()

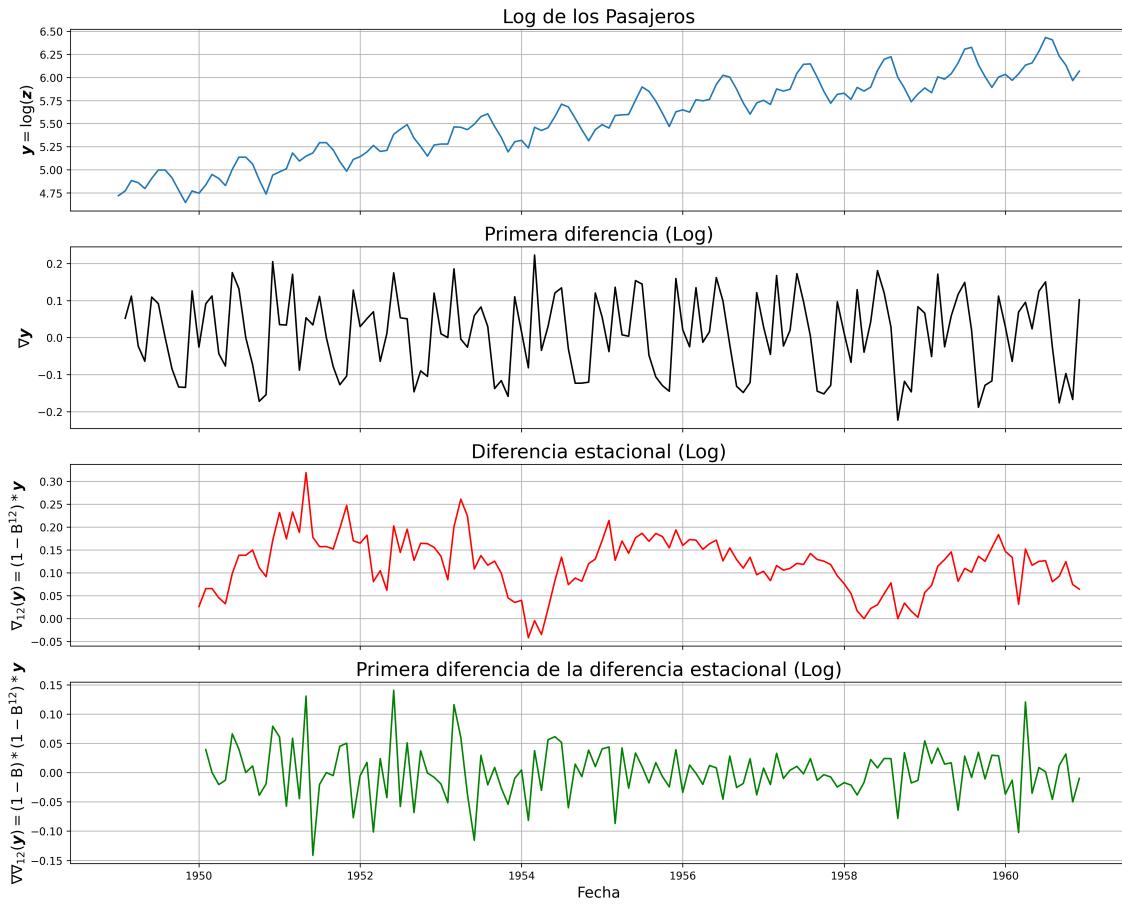
# Graficar la primera diferencia de la diferencia estacional
axs[3].plot(df.index, df['primera_diferencia_estacional'], color='green')
axs[3].set_xlabel('Fecha', fontsize=14)
axs[3].set_ylabel(r'$\nabla_{12}(\nabla_{12}(y)) = (1 - B^2)(1 - B^{12})y$', fontsize=14)
```

```

axs[3].set_title('Primera diferencia de la diferencia estacional (Log)', fontsize=18)
axs[3].grid()

# Ajustar el layout
plt.tight_layout()
plt.savefig('./img/lecc06/diferenciasPasajeros.png', dpi=300, bbox_inches='tight')

```



Como ya vimos, la serie pasajeros en logaritmos tiene tendencia y estacionalidad muy evidentes. No basta con tomar solo una diferencia ordinaria

$$\nabla \mathbf{y} = (1 - B) * \mathbf{y};$$

pues el resultado muestra una pauta estacional. Ni tampoco basta con tomar solo una diferencia estacional

$$\nabla_{12}(\mathbf{y}) = (1 - B^{12}) * \mathbf{y};$$

pues resulta una serie que “deambula”, i.e., que no es “estacionaria” en media.

Tomar una diferencia ordinaria y otra estacional

$$\nabla\nabla_{12}(\mathbf{y}) = (1 - B) * (1 - B^{12}) * \mathbf{y}$$

arroja una serie que sí parece ser “estacionaria”.

---

```

df3 = pd.read_csv('../datos/Retiro.txt')
# Generar una columna de fechas desde enero de 1985
df3['Fecha'] = pd.date_range(start='1985-01', periods=len(df3), freq='M') # cambiar por 'ME' en el futuro

# Configurar la columna de fechas como índice
df3.set_index('Fecha', inplace=True)

# Calcular la diferencia estacional (de orden 12)
df3['diferencia_estacional'] = df3['TemperaturaMedia'] - df3['TemperaturaMedia'].shift(12)

# Crear la figura y los ejes, compartiendo el eje x
fig, axs = plt.subplots(2, 1, figsize=(15, 8), sharex=True) # 2 filas, 1 columna

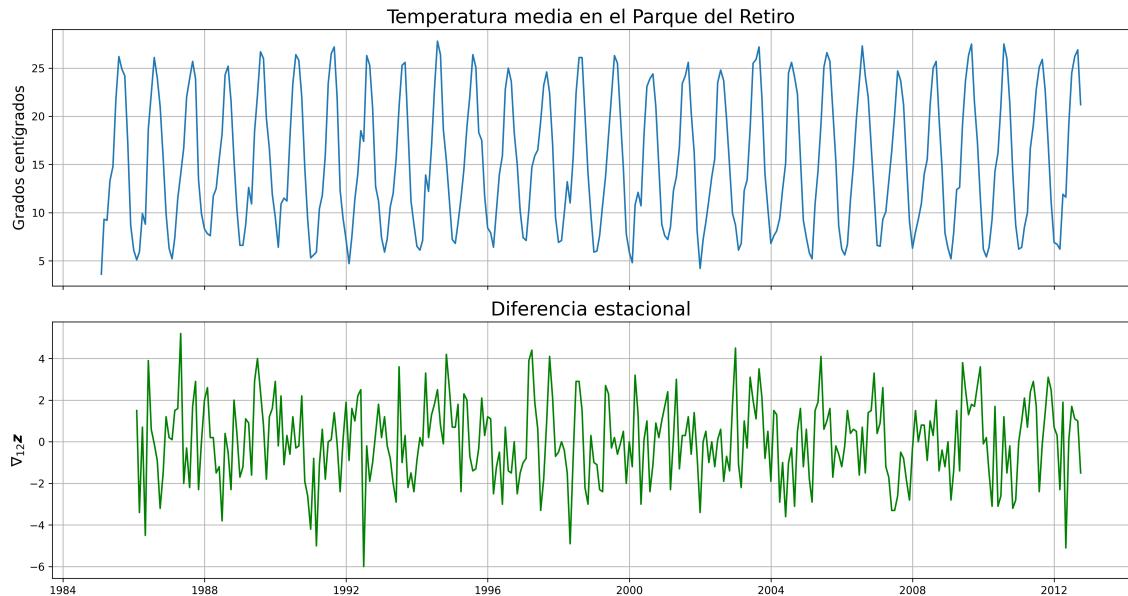
# Graficar el logaritmo de la población
axs[0].plot(df3.index, df3['TemperaturaMedia'])
axs[0].set_ylabel(r'Grados centígrados', fontsize=14)
axs[0].set_title('Temperatura media en el Parque del Retiro', fontsize=18)
axs[0].grid()

# Graficar la primera diferencia
axs[1].plot(df3.index, df3['diferencia_estacional'], color='green')
axs[1].set_ylabel(r' $\nabla_{12}z$ ', fontsize=14)
axs[1].set_title('Diferencia estacional', fontsize=18)
axs[1].grid()

# Ajustar el layout
plt.tight_layout()
plt.savefig('../img/lecc06/diferenciasTemperaturasRetiro', dpi=300, bbox_inches='tight')

```

---



En el caso de la serie de temperaturas en el Parque del Retiro, parece que es suficiente con tomar solo una diferencia estacional.

### 3.2.2. Contrastes formales sobre el orden de integración: DF, ADF y KPSS

#### 1. Test de Dickey-Fuller (DF)

$H_0$  la serie es  $I(1)$

$H_1$  la serie es  $I(0)$ .

Sea el siguiente modelo donde  $\phi$  es un parámetro autorregresivo y  $U_t$  es ruido blanco

$$Y_t = \phi Y_{t-1} + U_t,$$

Habrá una raíz unitaria (será no-estacionario) si  $\phi = 1$ . Restando  $Y_{t-1}$  a ambos lados

$$\nabla Y_t = (\phi - 1)Y_{t-1} + U_t = \delta Y_{t-1} + U_t$$

donde  $\delta = \phi - 1$  y, por tanto, la  $H_0$  se reduce a  $\delta = 0$ .

Pasos del contraste:

- se estima el último modelo por MCO
- se calcula el ratio  $t$  de significación de  $\delta$
- se compara dicho ratio con la distribución específica del contraste  $DF$  (pues bajo la nula, el ratio  $t$  no se distribuye de la forma habitual).

2. Test de Dickey-Fuller aumentado (ADF) Una variante habitual del test DF es el test de Dickey-Fuller aumentado (ADF), que consiste en estimar por MCO el modelo:

$$\nabla Y_t = c + \delta Y_{t-1} + \pi_1 \nabla Y_{t-1} + \pi_2 \nabla Y_{t-2} + \cdots + \pi_p \nabla Y_{t-p} + U_t$$

que añade  $p$  retardos del regresando como regresores para permitir autocorrelación. Por lo demás, el test se calcula de la forma habitual (comparando de ratio  $t$  de  $\hat{\delta}$  con las tablas del test ADF).

Otra variante consiste en incluir una tendencia temporal determinista:

$$\nabla Y_t = \underbrace{c + \beta t}_{\text{tendencia}} + \delta Y_{t-1} + \pi_1 \nabla Y_{t-1} + \pi_2 \nabla Y_{t-2} + \cdots + \pi_p \nabla Y_{t-p} + U_t$$

3. Test Kwiatkowski–Phillips–Schmidt–Shin (KPSS) El contraste KPSS es un contraste alternativo (y complementario) al test DF/ADF

Las hipótesis del test KPSS están invertidas respecto a las del Test DF

$H_0$  la serie es  $I(0)$ .

$H_1$  la serie es  $I(1)$  o estacionaria en torno a una tendencia determinista

**DF/ADF y KPSS se complementan.** Si la serie es:

- *estacionaria*: el test DF/ADF debería rechazar su  $H_0$  : la serie es  $I(1)$
- *NO estacionaria*: el test KPSS debería rechazar su  $H_0$  : la serie es  $I(0)$

Estos test son poco potentes y son frágiles ante incumplimientos (heterocedasticidad o no-normalidad); por ello debemos complementarlos con el análisis gráfico.

### 3.3. ACF muestral

La  $k$ -ésima autocorrelación muestral simple ( $\widehat{\rho}_k$ ) se define como:

$$\widehat{\rho}_k = \frac{\widehat{\gamma}_k}{\widehat{\gamma}_0}; \quad \widehat{\gamma}_k = \frac{1}{n} \sum_{t=k+1}^n \widetilde{X}_t \widetilde{X}_{t-k}, \quad \text{para } k = 1, 2, \dots$$

donde  $\widetilde{X}_t = X_t - \bar{X}$ .

Para valorar la significatividad individual de estas autocorrelaciones puede usarse el error es-tándar asintótico:  $s.e.(\widehat{\rho}_k) = 1/\sqrt{n}$ .

Para contrastar la  $H_0 : \text{los } k \text{ primeros retardos son conjuntamente no significativos}$  (es decir, para contrastar si el proceso es *ruido blanco*) se emplea el test de Ljung-Box

$$Q = n(n+2) \sum_{k=1}^h \frac{\widehat{\rho}_k^2}{n-k}$$

(usaremos el test de Ljung-Box para evaluar los modelos).

### 3.4. PACF muestral

La  $k$ -ésima autocorrelación muestral  $\widehat{\pi}_k$  se puede estimar mediante el algoritmo Levinson-Durbin sustituyendo las autocorrelaciones teóricas por las muestrales.

O bien, calculando el  $k$ -ésimo coeficiente MCO de una autorregresión de orden  $k$

$$\tilde{X}_t = \widehat{\phi}_{k1}\tilde{X}_{t-1} + \widehat{\phi}_{k2}\tilde{X}_{t-2} + \dots + \widehat{\phi}_{kk}\tilde{X}_{t-k} + U_t; \quad k = 1, 2, \dots$$

donde  $\tilde{X}_t = X_t - \bar{X}$  y donde  $\widehat{\pi}_k = \widehat{\phi}_{kk}$

## 3.5. Otras herramientas estadísticas

### 3.5.1. Estadísticos descriptivos

- el nivel de la serie (media, mediana),
- valores extremos (máximo y mínimo). A veces son outliers o errores de registro
- dispersión de la variable (desviación típica, coeficiente de variación, rango, rango interpercentiles, rango intercuartílico)
- otros momentos (asimetría, exceso de curtosis)

También es frecuente contrastar si la media es significativa ( $H_0 : \mu = 0$ )

$$\frac{\widehat{\mu}}{\widehat{dt}(\widehat{\mu})} \underset{H_0}{\sim} t_{n-1}; \quad \widehat{dt}(\widehat{\mu}) = \frac{\widehat{\sigma}}{\sqrt{n}}.$$

---

`df3['TemperaturaMedia'].describe()`

---

### 3.5.2. Contraste de normalidad Jarque-Bera

[Jarque-Bera test \(Wikipedia\)](#)

Si los datos son mayores que cero a menudo la transformación logarítmica ayuda a conseguir normalidad

Si hay valores atípicos el mejor tratamiento es *intervenirlos*.

---

```
from scipy import stats
rng = np.random.default_rng()
x = rng.normal(0, 1, 100000)
jarque_bera_test = stats.jarque_bera(x)
jarque_bera_test
#jarque_bera_test.statistic
#jarque_bera_test.pvalue
```

---

## 4. Nota final

Se puede demostrar que

- si un proceso tiene una ACF  $\rho$  con cogrado  $-q$  y grado  $q$ , es un proceso MA( $q$ )
- si un proceso tiene una PACF  $\pi$  con cogrado  $-p$  y grado  $p$ , es un proceso AR( $p$ )

Véase *Pourahmadi M. (2001)*.

Consecuentemente, que la ACF o la PACF se corten “bruscamente” nos indica que el modelo es AR o MA (además de su grado).

Desgraciadamente para los modelos ARMA tanto la ACF como la PACF tienen infinitos términos no nulos, por lo que su identificación no es tan sencilla.

	ACF finita	ACF persistente
PACF finita	Ruido blanco: retardos conjuntamente NO significativos	AR: orden indicado por la PACF
PACF persistente	MA: orden indicado por la ACF	ARMA

## 5. Función de autocovarianzas para un ARMA( $p,q$ )

Sea un ARMA( $p, q$ ) estacionario:  $\phi(B)X_t = \theta(B)U_t$  donde  $\phi$  y  $\theta$  no tienen raíces comunes. Multiplicando por  $X_{t-k}$ , tomando esperanzas y sustituyendo  $X_{t-k}$  por su representación MA( $\infty$ ), donde  $\psi = \frac{\theta}{\phi}$ :

$$\underbrace{E[(\phi(B)X_t) \cdot X_{t-k}]}_{\phi(B)\gamma_k \text{ (por ??)}} = E[(\theta(B)U_t) \cdot X_{t-k}] = \underbrace{E[(\theta(B)U_t) \cdot (\psi(B)U_{t-k})]}_{\gamma_{W,Y}(k)}$$

Donde hemos usando (??) y renombrando  $\boldsymbol{\theta}(\mathbb{B})U_t = \mathbf{W}$  y  $\boldsymbol{\psi}(\mathbb{B})U_t = \mathbf{Y}$ . Así:

$$\begin{aligned}\phi(\mathbb{B})\gamma_k &= \boldsymbol{\gamma}_{\mathbf{W}, \mathbf{Y}}(k) \\ &= \sigma^2 \left( \boldsymbol{\theta}(z) * \boldsymbol{\psi}(z^{-1}) \right)_k \quad \text{por (5 - Lecc3)}\end{aligned}$$

Por Y como  $\boldsymbol{\theta}(z) * \boldsymbol{\psi}(z^{-1})$  tiene grado  $q$  y cogrado  $-\infty$

$$\phi(\mathbb{B})\gamma_k = \begin{cases} 0 & k > q \quad (\text{como en un AR}) \\ \sigma^2 \left( \boldsymbol{\theta}(z) * \boldsymbol{\psi}(z^{-1}) \right)_k & k \leq q \quad (\text{que depende de } \boldsymbol{\theta} \text{ y } \boldsymbol{\phi}) \end{cases} \quad (1)$$

(Donde (5-Lecc 3) se refiere a la [última ecuación de la Lección 3](#))