

Análisis de Series de Tiempo

Carrera de Especialización en Inteligencia Artificial

Clase 2

Ing. Magdalena Bouza, Esp. Ing. Carlos German Carreño Romano

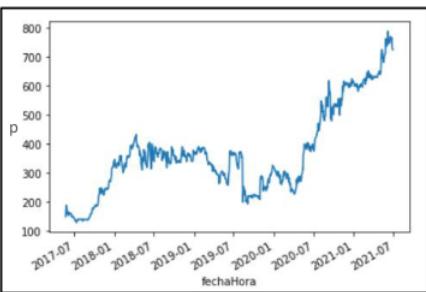
Agenda

1. Tendencia determinística.
2. Explicación del Trabajo Práctico.
3. Modelos lineales, de promedio móvil y autorregresivos (AR, MA, ARMA).
4. Autocorrelación para testear estacionariedad

Ejemplos actividad Clase 1

Ejercicio 1

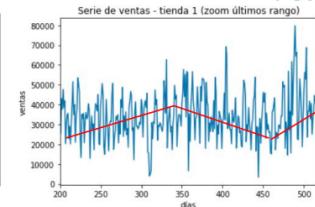
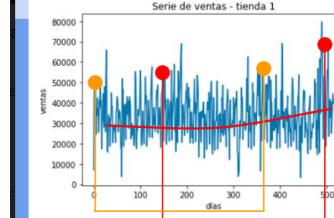
Graficar una serie a partir de un dataset relevante. Explicar observaciones



Bolsa y Mercado Argentino – Precio Cierre
23/05/2017 – 29/06/2021

La evolución del último precio presenta cierta estabilidad hasta mediados del año 2020, entorno a un valor de 300. Luego, quizás por efecto de la pandemia, podemos ver un incremento del precio en forma constante, hasta llegar a mediados del 2021, donde toca su valor máximo histórico (en el tiempo analizado).

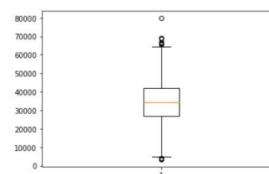
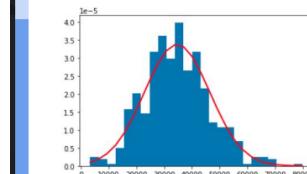
1 - Análisis visual



- La serie tiene un offset y parece tener una tendencia anual a incrementar las ventas.
- No parece estacionaria, y podría tener alguna estacionalidad del tipo semestral (verde más en una temporada que otra).

7

1 - Analizar la distribución



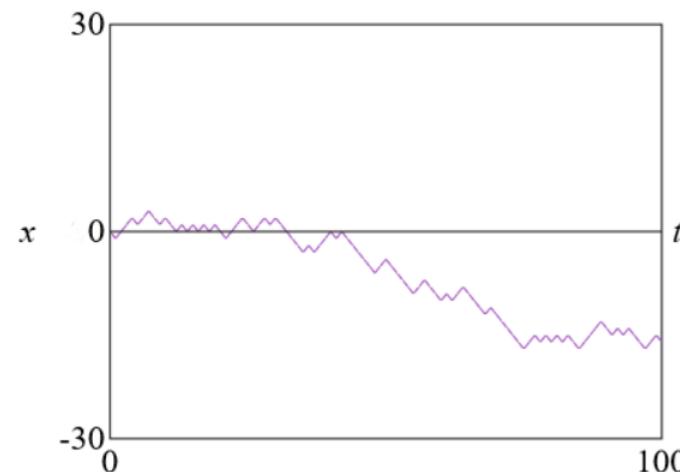
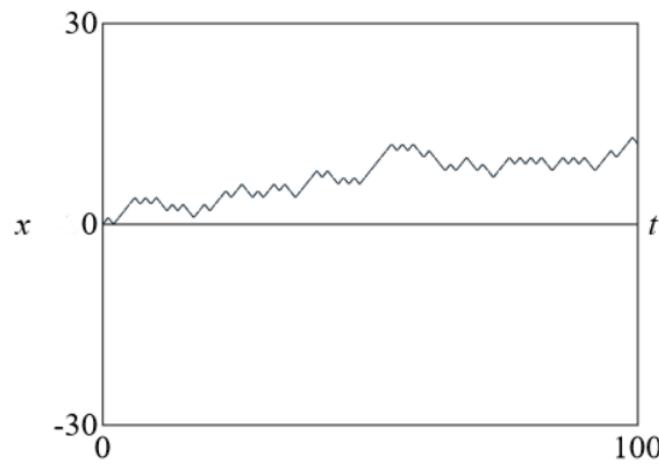
- La distribución tiene skew positiva, lo cual confirma que los valores incrementan con el tiempo.
- Dicho lo anterior, la serie no es estacionaria (varianza variable).

8

Análisis de tendencia

Tendencia estocástica vs. tendencia determinística

Si bien no hay una definición unificada de qué son las **tendencias estocásticas**, se puede decir que son aquellas que un observador podría hallar al analizar una realización de una serie de tiempo, pero que si se tiene una realización distinta esa tendencia cambia.



Tendencia estocástica vs. tendencia determinística

La **tendencia determinística** es aquella que viene dada por el modelo, y es fija para toda la serie de tiempo, sin importar qué realización se tenga. *Por ejemplo las variaciones cíclicas a lo largo de las distintas estaciones del año.*

En el caso de la tendencia determinística, la podemos estimar y descontar de la serie de tiempo. Esta situación se puede modelar como

$$Y_t = X_t + \mu_t$$

donde μ_t es la tendencia determinística y X_t es una serie de tiempo de media cero alrededor de μ_t .

Objetivo: estimar X_t con un modelo para μ_t

Si proponemos un modelo de tendencia determinística μ_t , tal que:

$$Y_t = X_t + \mu_t$$

con Y_t la serie original y X_t un proceso estacionario, los valores de X_t pueden ser estimados a partir de los residuos:

$$\hat{x}_t = y_t - \hat{\mu}_t$$

Luego vamos a poder aplicar modelos estacionarios a \hat{x}_t .

¿Cómo estimar estas tendencias?

Suele emplearse el método de cuadrados mínimos para estimar los valores de los parámetros que describen la tendencia.

Si llamamos $f(t; \theta)$ a los modelos presentados anteriormente, buscamos hallar los parámetros θ que minimicen el ECM. Es decir, buscamos θ que minimice

$$Q(\theta) = \frac{1}{n} \sum_{i=1}^n (y_t - f(t; \theta))^2$$

Algunos modelos comunes para tendencia

1. Constante: $\mu_t = \mu \quad \forall t$
2. Lineal: $\mu_t = \beta_0 + \beta_1 t \quad \forall t$
3. Cuadrática: $\mu_t = \beta_0 + \beta_1 t + \beta_2 t^2 \quad \forall t$
4. Cíclicas: $\mu_t = \mu_{t-T} \quad \forall t$, donde T es el período del ciclo.
 - a. *Ejemplo: temperatura a lo largo del año tiene un período de T=12 meses*
5. Coseno:
$$\mu_t = \beta \cos(2\pi ft + \phi) \quad \forall t$$

Caso constante:

$$\begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \mu + \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_1 \end{bmatrix}$$

Luego, la solución por c.m. es

$$\mu = \underbrace{\left([1 \ \dots \ 1] \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right)^{-1}}_n [1 \ \dots \ 1] \begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_1 \end{bmatrix} = \frac{1}{n} \sum_{t=1}^n y_t$$

Caso lineal

$$\begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & n \\ 1 & n-1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_1 \end{bmatrix}$$

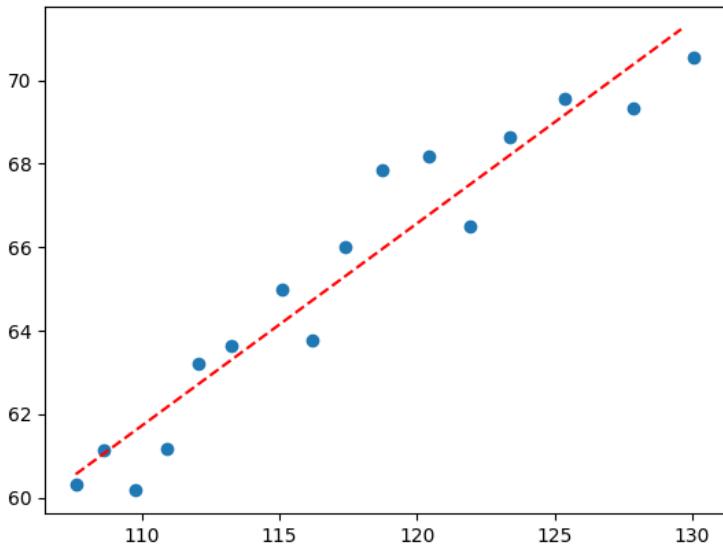
Y la solución por c.m resulta

$$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} = \left(\begin{bmatrix} 1 & n \\ 1 & n-1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix}^T \begin{bmatrix} 1 & n \\ 1 & n-1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & n \\ 1 & n-1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix}^T \begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_1 \end{bmatrix} =$$

$$\hat{\beta}_1 = \frac{\sum_{t=1}^n (y_t - \bar{y})(t - \frac{n+1}{2})}{\sum_{t=1}^n (t - \frac{n+1}{2})^2} \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \frac{n+1}{2}$$

Caso lineal

```
from scipy.optimize import curve_fit
# define the true objective function
def objective(x, a, b):
    return a * x + b
```



```
dataframe = read_csv(url, header=None)
data = dataframe.values
# choose the input and output variables
x, y = data[:, 4], data[:, -1]
# curve fit
popt, _ = curve_fit(objective, x, y)
# summarize the parameter values
a, b = popt
print('y = %.5f * x + %.5f' % (a, b))
# plot input vs output
pyplot.scatter(x, y)
# define a sequence of inputs between the smallest and largest observed
x_line = arange(min(x), max(x), 1)
# calculate the output for the range
y_line = objective(x_line, a, b)
# create a line plot for the mapping function
pyplot.plot(x_line, y_line, '--', color='red')
pyplot.show()
```

Caso cuadrático

$$\begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & n & n^2 \\ 1 & n-1 & (n-1)^2 \\ \vdots & \vdots & \\ 1 & 1 & 1 \end{bmatrix}}_A \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_1 \end{bmatrix}$$

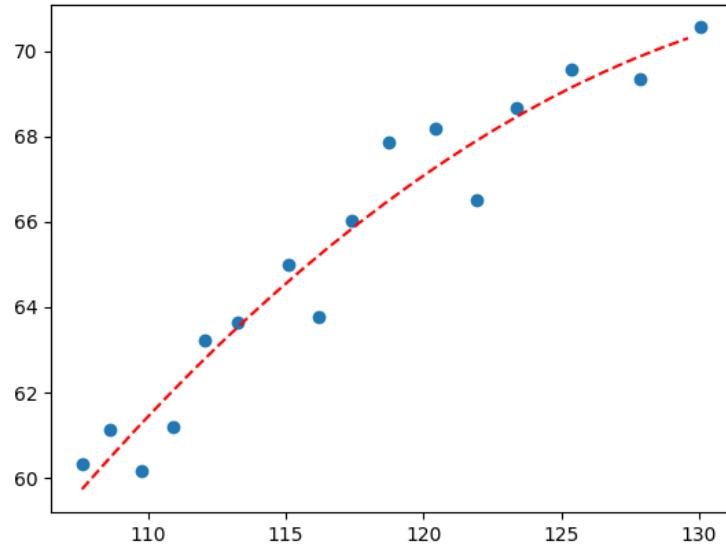
Y la solución por c.m. queda

$$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix} = (A^T A)^{-1} A^T \begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_1 \end{bmatrix}$$

Caso cuadrático

```
# define the true objective function
def objective(x, a, b, c):
    return a * x + b * x**2 + c

# choose the input and output variables
x, y = data[:, 4], data[:, -1]
# curve fit
popt, _ = curve_fit(objective, x, y)
# summarize the parameter values
a, b, c = popt
print('y = %.5f * x + %.5f * x^2 + %.5f' % (a, b, c))
# plot input vs output
pyplot.scatter(x, y)
# define a sequence of inputs between the smallest and
# largest observed values
x_line = arange(min(x), max(x), 1)
# calculate the output for the range
y_line = objective(x_line, a, b, c)
```

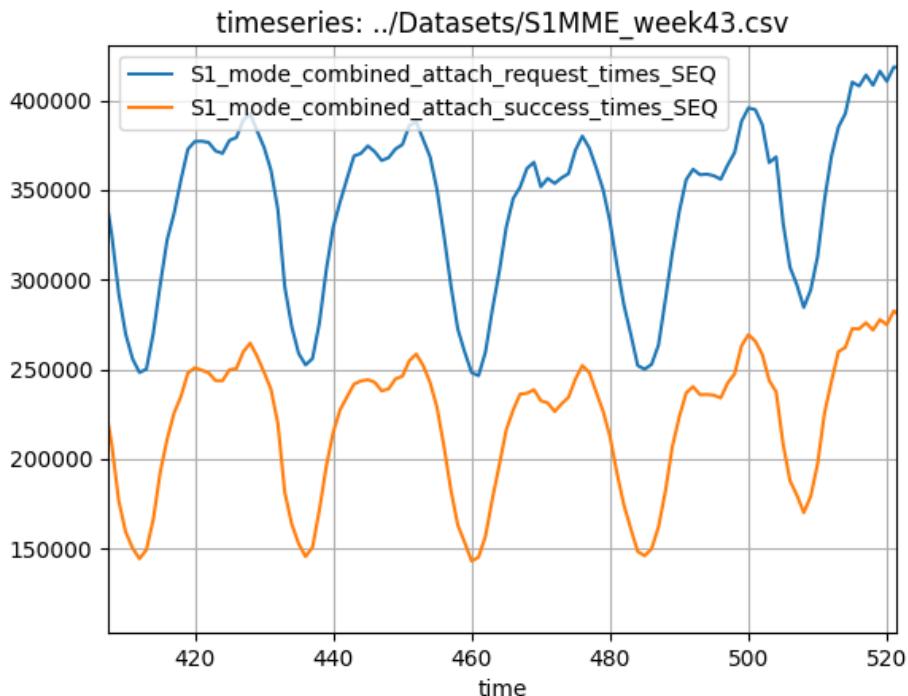


Caso cíclico

$$\mu_t = \begin{cases} \beta_1 & t = 1, 1+T, 1+2T, \dots \\ \beta_2 & t = 2, 2+T, 2+2T, \dots \\ \vdots \\ \beta_T & t = T, 2T, 3T, \dots \end{cases}$$

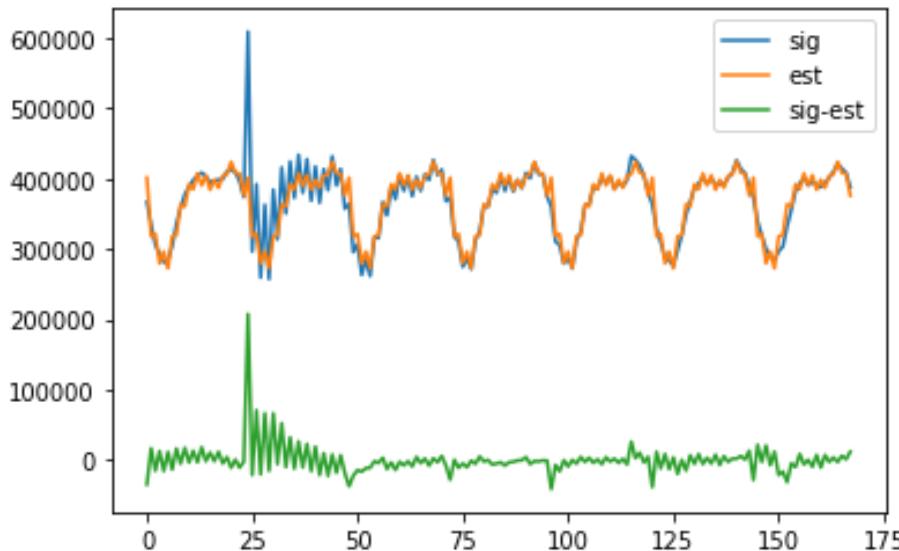
$$\begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_1 \end{bmatrix} = A \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_T \end{bmatrix} + \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_1 \end{bmatrix}, \quad a_{i,j} = \begin{cases} 1 & (i \bmod j) = 0 \\ 0 & (i \bmod j) \neq 0 \end{cases}$$

Análisis de series de tiempo



Cada período presenta picos y valles bien marcados. La serie cae en sus valles a valores mínimos que suelen mantener un valor absoluto regular. En cada período se pueden observar dos picos a una distancia regular, donde el valor del segundo pico supera siempre al primero. Ambos picos tienen variaciones en su valor absoluto, a diferencia de los valles.

Ajuste por cuadrados mínimos: caso cíclico



[ver ejemplo: linearRegression24hs.py](#)

```
dataset=sig
interval=1
diff = list()
for i in range(interval, len(dataset)):

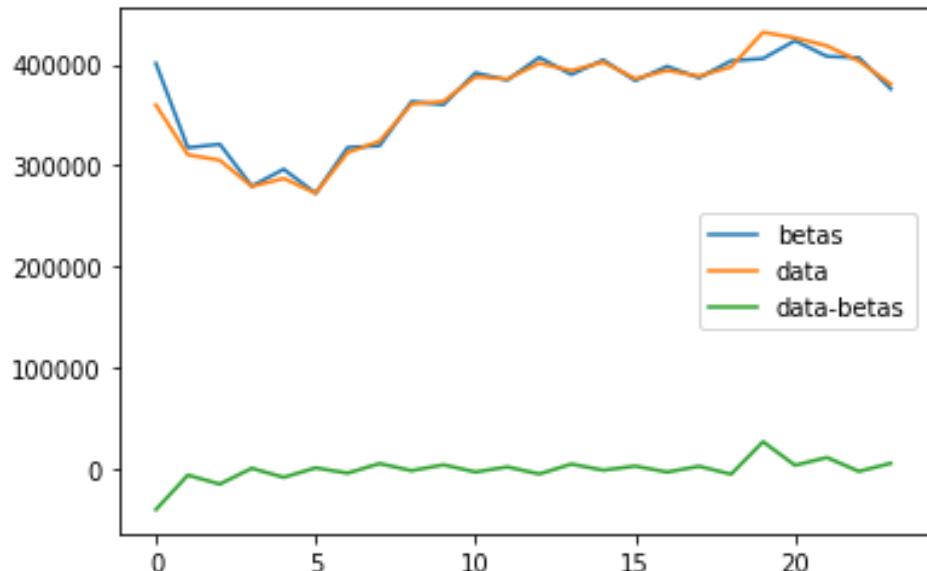
    value = dataset[i] - dataset[i - interval]
    diff.append(value)

plt.plot(sig)
plt.plot(diff)
plt.legend(['sig','diff'])
plt.show()

#repito la estimacion pero para la diferenciada
N=24 #hours
dataframe = pd.Series(pd.concat([pd.Series(diff[0]),p
ts=pd.DataFrame(dataframe.values)
rows=int(len(ts)/N)
data = ts.values.reshape(rows,N)

betas=data.mean(axis=0)
```

Ajuste por cuadrados mínimos: caso cíclico



Usando la técnica de cuadrados mínimos se pueden estimar los valores promedio de los períodos.

Luego si se resta a la serie original la serie estimada por cuadrados mínimos se obtiene una serie que puede servir para analizar estacionariedad.

Caso coseno

Reescribiendo la tendencia podemos llevarla a una expresión lineal:

$$\mu_t = \beta \cos(2\pi ft + \phi) = \beta_1 \cos(2\pi ft) + \beta_2 \sin(2\pi ft),$$

$$\beta_1 = \beta \cos(\phi), \quad \beta_2 = \beta \sin(\phi)$$

$$\begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \cos(2\pi fn) & \sin(2\pi fn) \\ 1 & \cos(2\pi f(n-1)) & \sin(2\pi f(n-1)) \\ \vdots & \vdots & \vdots \\ 1 & \cos(2\pi f1) & \sin(2\pi f1) \end{bmatrix}}_A \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_1 \end{bmatrix}$$

¡Cuidado!

Los resultados por cuadrados mínimos son válidos estrictamente sólo cuando X_t es un proceso blanco (las muestras temporales son independientes entre sí).

Sin embargo, se puede demostrar que si el proceso es estacionario, los resultados son asintóticamente válidos.

Recordar: el objetivo de estimar tendencia determinística es obtener una serie estacionaria en sentido débil a partir de las mediciones de una serie de tiempo.

Trabajo práctico

Trabajo Práctico

- Graficar una serie a partir de un dataset relevante. **Describir** observaciones.
- **Descomponer** una serie de tiempo usando el modelo aditivo de cuatro componentes.
- Extraer la **tendencia** y ajustar un **modelo determinístico**. Explicar su relación con el contexto. Obtener conclusiones acerca de la validez del modelo.
- **Entrega clase 3:** Evaluar si la serie original es **estacionaria**. Aplicar transformaciones (**preprocesamiento**), graficar autocorrelación, modelar tendencia determinística, extraer conclusiones.

Trabajo Práctico

- **Entrega clase 5:** A partir de las transformaciones propuestas ajustar distintos **modelos (S)ARIMA**. Extraer orden, parámetros, coeficientes numéricos y análisis de la bondad del modelo.
- Ajustar y predecir usando **redes neuronales LSTM**. Comparar con **predicciones** usando SARIMA y extraer conclusiones.
- Realizar el **análisis espectral** de la serie original. Hallar las frecuencias principales y comparar con las **componentes cíclica y estacional** usando la descomposición.
- **Entrega clase 8:** Presentación incluyendo introducción, gráficos, modelos propuestos, expresiones analíticas y conclusiones.

Modelos para series de tiempo estacionarias

Proceso lineal general

Sean $\{Y_t\}$ la serie de tiempo observada y $\{e_t\}$ una serie de ruido blanco no observable. $\{e_t\}$ es una secuencia de v.a. i.i.d, con media cero y varianza σ_e^2 .

Un proceso lineal general es aquel que puede representarse como una combinación lineal de términos presentes y pasados del proceso de ruido blanco:

$$Y_t = e_t + a_1 e_{t-1} + a_2 e_{t-2} + \dots$$

Si la cantidad de términos a sumar es infinita se pide que $\sum_{i=1}^{\infty} a_i^2 < \infty$.

Se puede demostrar que:

- $\mathbb{E}[Y_t] = 0$
- $C_k = cov(Y_t, Y_{t-k}) = \sigma_e^2 \sum_{i=1}^{\infty} a_i a_{i+k}, \quad k \geq 0$

$$Y_t = e_t + \alpha_1 e_{t-1} + \alpha_2 e_{t-2} \dots$$

$$\text{Cov}(Y_t, Y_{t-2}) = \text{Cov} \left(e_t + \alpha_1 e_{t-1} + \alpha_2 e_{t-2} + \alpha_3 e_{t-3} + \alpha_4 e_{t-4} + \alpha_5 e_{t-5}, e_{t-2} + \alpha_1 e_{t-3} + \alpha_2 e_{t-4} + \alpha_3 e_{t-5} + \dots \right)$$

$$\alpha_5 = 1$$

$$\sum_{k=0}^{\infty} \alpha_k \alpha_{k+2} \sigma_e^2 + \alpha_3 \cdot \alpha_1 \sigma_e^2 + \alpha_1 \alpha_2 \sigma_e^2 + \dots$$

Modelo de promedios móviles

Un **modelo de promedio móvil (MA)** es un modelo lineal general donde existe una cantidad finita términos $a_i \neq 0$.

Diremos que $\{Y_t\}$ es un modelo de promedio móvil de orden q (MA(q)) si

$$Y_t = e_t - a_1 e_{t-1} - \dots - a_q e_{t-q}$$

Los parámetros de este modelo son los pesos a_1, \dots, a_q .

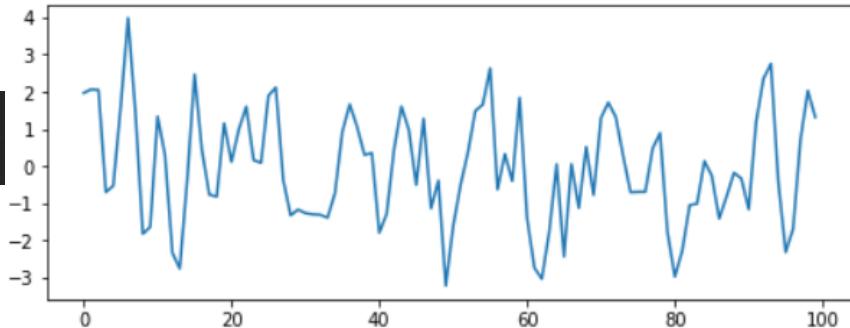
Para estos modelos, $C_0 = (1 + a_1^2 + \dots + a_q^2) \sigma_e^2$

$$R_k = \begin{cases} \frac{-a_k + a_1 a_{k+1} + a_2 a_{k+2} + \dots + a_{1-k} a_q}{1 + a_1^2 + \dots + a_q^2} & k = 1, 2, \dots, q \\ 0 & k > q \end{cases}$$

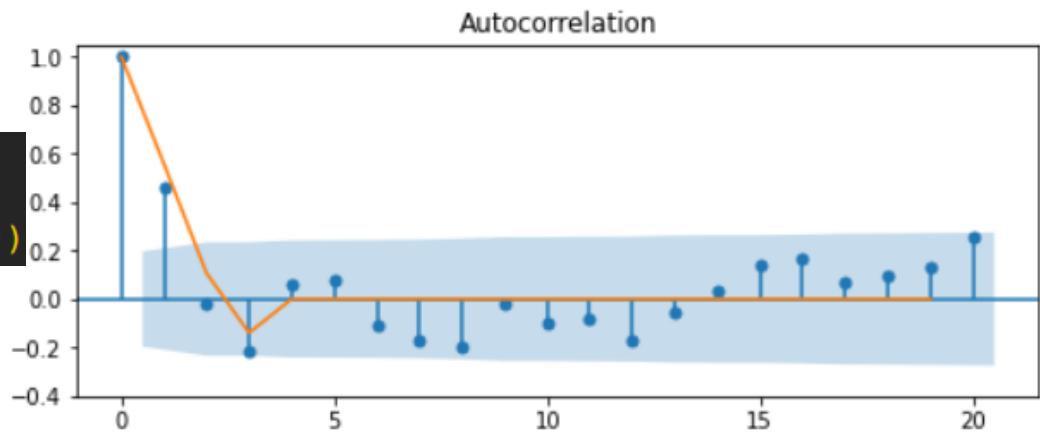


Ejemplo!

```
ma_ts = arma_generate_sample(  
    ar=[1], ma=np.r_[1,ma_coef], nsample =100)
```



```
plot_acf(ma_ts, ax= ax)  
plt.plot(arma_acf(ar=[1],  
    ma=np.r_[1,ma_coef], lags=20))
```



Modelo autoregresivo

Los modelos autorregresivos incluyen regresiones sobre sí mismos.

Diremos que Y_t es un modelo autorregresivo de orden p (AR(p)) si cumple que

$$Y_t = a_1 Y_{t-1} + \dots + a_p Y_{t-p} + e_t \rightarrow \text{innovación}$$

Suponemos que para cada t, e_t es independiente de Y_{t-1}, Y_{t-2}, \dots

El modelo AR(p) tiene asociado su polinomio característico definido como:

$$\phi(x) = 1 - a_1 x + a_2 x^2 + \dots + a_p x^p$$

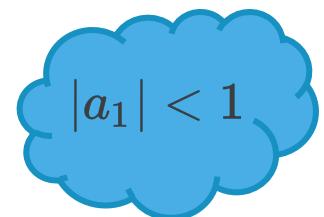
Se puede demostrar que el proceso es AR(q) es estacionario si las raíces $\phi(x)$ se encuentran fuera del círculo unitario (módulo mayor a 1)

Modelo AR(1)

Si $p=1$, tenemos que $Y_t = a_1 Y_{t-1} + e_t$, luego $\mathbb{E}[Y_t] = 0$.

Puedo tomar la varianza miembro a miembro para obtener

$$\text{var}(Y_t) = a_1^2 \text{var}(Y_{t-1}) + \sigma_e^2 \rightarrow C_0 = a_1^2 C_0 + \sigma_e^2 \Rightarrow C_0 = \frac{\sigma_e^2}{1-a_1^2}$$



Además,

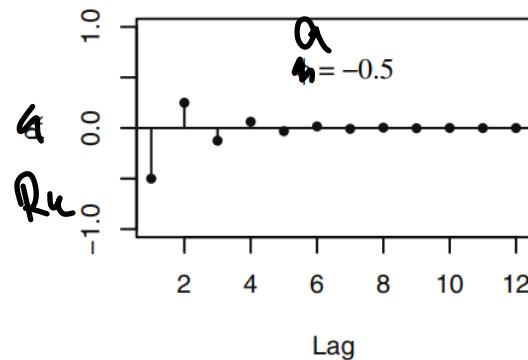
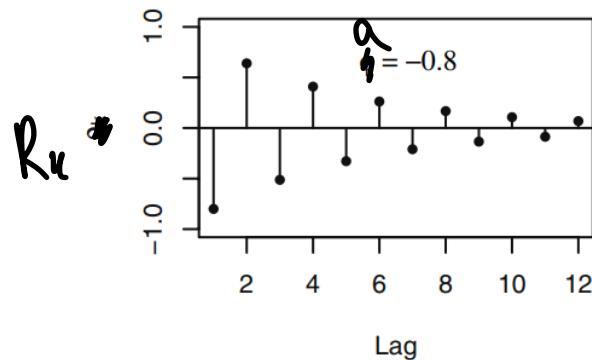
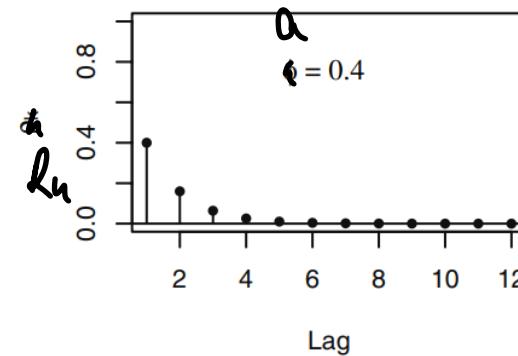
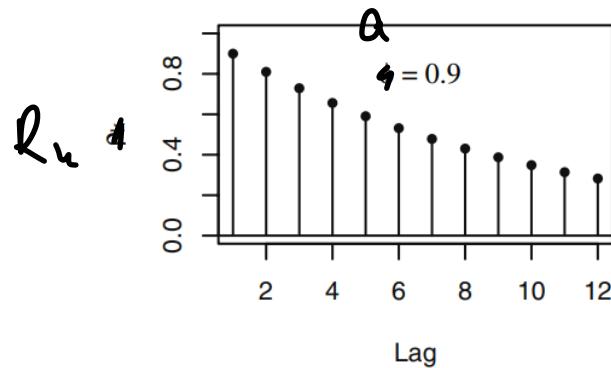
$$\begin{aligned} C_k &= \text{cov}(Y_{t-k}, Y_t) = \text{cov}(Y_{t-k}, a_1 Y_{t-1} + e_t) \\ &= a_1 \underbrace{\text{cov}(Y_{t-k}, Y_{t-1})}_{C_{k-1}} + \underbrace{\text{cov}(Y_{t-k}, e_t)}_0 \\ &= a_1 C_{k-1} \end{aligned}$$

A partir del valor semilla de C_0 obtenemos de forma recursiva que

$$C_k = a_1^k \frac{\sigma_e^2}{1-a_1^2}$$



Modelo AR(1)



Modelo AR(1) como un proceso lineal general

Para comprender el modelo, la expresión dada para el modelo AR(1) es muy útil, sin embargo para muchas otras cosas es necesario llevarlo a la forma de un proceso general lineal.

Para eso comenzamos reemplazando Y_{t-1} por $a_1 Y_{t-2} + e_{t-1}$:

$$Y_t = a_1(a_1 Y_{t-2} + e_{t-1}) + e_t = a_1^2 Y_{t-2} + a_1 e_{t-1} + e_t$$

aplicando la misma idea ($k-1$) veces:

$$Y_t = a_1^k e_{t-k} + a_1^{k-1} e_{t-k+1} + \dots + a_1 e_{t-1} + e_t$$

Asumiendo $|a_1| < \infty$ e incrementando k sin límite tenemos que

$$\begin{aligned} Y_t &= e_t + \boxed{a_1} e_{t-1} + \boxed{a_1^2} e_{t-2} + \boxed{a_1^3} e_{t-3} + \dots \\ Y_t &= e_t + \boxed{\psi_1} e_{t-1} + \boxed{\psi_2} e_{t-2} + \boxed{\psi_3} e_{t-3} + \dots \end{aligned}$$

Modelo AR(p)

Se puede mostrar que una condición necesaria (pero no suficiente) para que el proceso sea estacionario es que $a_1 + a_2 + \dots + a_p < 1$ y $|a_p| < 1$

Al igual que como hicimos para el AR(1), podemos calcular la función de autocovarianza y autocorrelación:

$$\begin{aligned} C_k &= cov(Y_t, Y_{t-k}) = cov(a_1 Y_{t-1}, Y_{t-k}) + \dots + cov(a_p Y_{t-p}, Y_{t-k}) + cov(e_t, Y_{t-k}) \\ &= a_1 C_{k-1} + \dots + a_p C_{k-p} \end{aligned}$$

$$R_k = \frac{cov(Y_t, Y_{t-k})}{C_0} = a_1 R_{k-1} + \dots + a_p R_{k-p}$$

Modelo AR(p)

Evaluando para $k=1, \dots, p$ y recordando que $R_0 = 1$ y $R_k = R_{-k}$ obtenemos las **ecuaciones de Yule-Walker (Y-W)**:

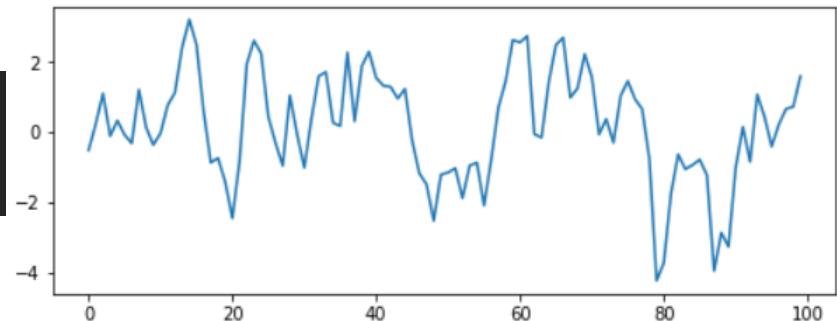
$$\begin{cases} R_1 = a_1 + a_2 R_1 + \dots + a_p R_{p-1} \\ \vdots \\ R_p = a_1 R_{p-1} + a_2 R_{p-2} + \dots + a_p R_p \end{cases}$$

Dado los valores de a_1, \dots, a_p , se puede resolver el sistema de ecuaciones para hallar R_1, \dots, R_p

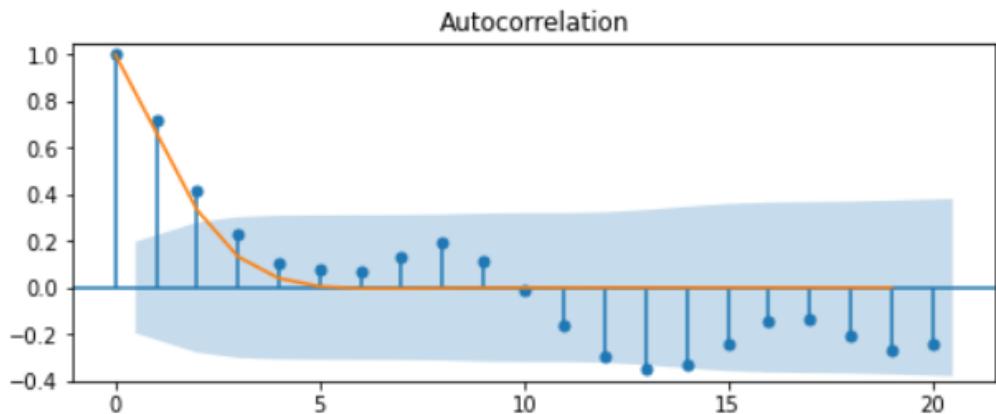
Finalmente, podemos usar estos valores para hallar $C_0 = a_1 C_1 + \dots + a_p C_p + \sigma_e^2$ observando que $C_0 = a_1 R_1 C_0 + \dots + a_p R_p C_0 + \sigma_e^2 \Rightarrow C_0 = \frac{\sigma_e^2}{1 - a_1 R_1 - \dots - a_p R_p}$

Ejemplo!

```
ar_coef = np.array([0.8,-0.2])
ar_ts = arma_generate_sample(ar=np.r_[1,-ar_coef],
                             ma=[1], nsample =100)
```



```
plot_acf(ar_ts, ax=ax)
plt.plot(arma_acf(ar=np.r_[1,-ar_coef],
                  ma=[1], lags=20))
```



ARMA(p,q)

Modelo ARMA

El modelo arma es una combinación de un proceso AR con un MA. Diremos que $\{Y_t\}$ sigue un modelo ARMA(p,q) si

$$Y_t = a_1 Y_{t-1} + \dots + a_p Y_{t-p} + e_t - b_1 e_{t-1} - \dots - b_q e_{t-q}$$

Si se satisfacen las condiciones de estacionariedad, el modelo ARMA(p,q) puede reescribirse como un proceso lineal general con coeficientes ψ_1, ψ_2, \dots dados por:

$$\begin{cases} \psi_0 = 1 \\ \psi_1 = -b_1 + a_1 \\ \psi_2 = -b_2 + a_2 + a_1 \psi_1 \\ \vdots \\ \psi_j = -b_j + a_p \psi_{j-p} + a_{p-1} \psi_{j-p+1} + \dots + a_1 \psi_{j-1} \end{cases} \quad \psi_j = 0, \text{ si } j < 0 \text{ y } b_j = 0 \text{ si } j > q$$

ARMA(p,q)

Se puede ver que la función de autocorrelación está dada por:

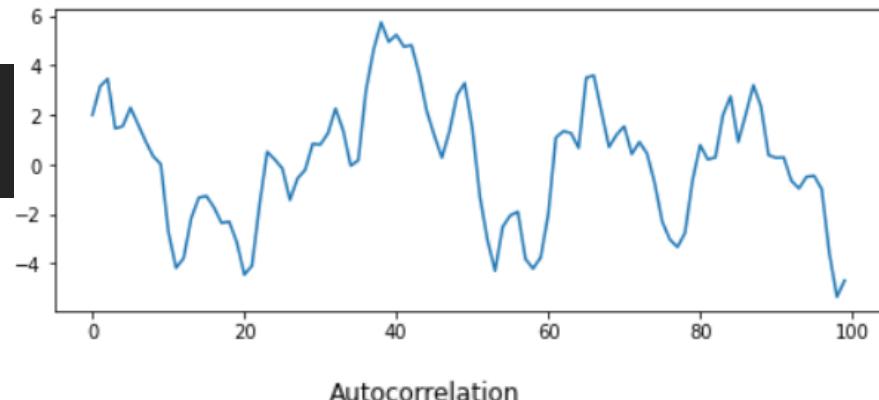
$$\begin{cases} C_0 = a_1C_1 + a_2C_2 + \dots - \sigma_e^2(b_0 + b_1\psi_1 + \dots + b_q\psi_q) \\ C_1 = a_1C_0 + a_2C_1 + \dots + a_pC_{p-1} - \sigma_e^2(b_1 + b_2\psi_1 + \dots + b_1\psi_{q-1}) \\ \vdots \\ C_p = a_1C_{p-1} + a_2C_{p-2} + \dots + a_pC_0 - \sigma_e^2(b_p + b_{p+1}\psi_1 + \dots + b_q\psi_{q-p}) \end{cases}$$

Si $k > q$ entonces la expresión puede simplificarse como:

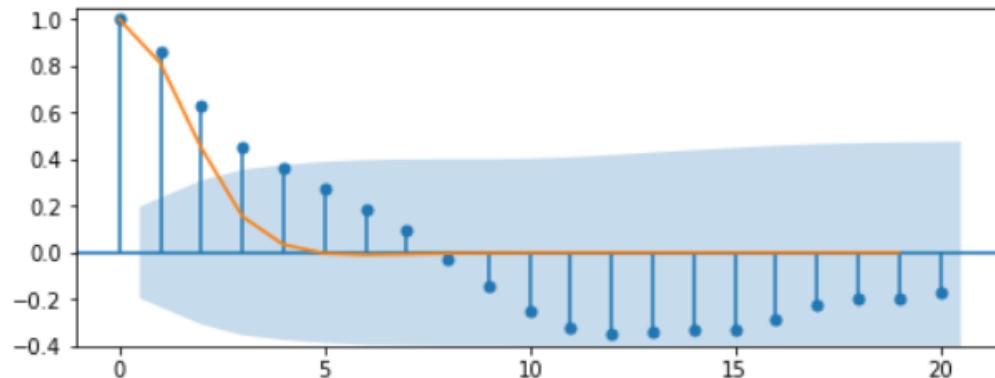
$$C_k = a_1C_{k-1} + a_2C_{k-2} + \dots + a_pC_{k-p}$$

Ejemplo!

```
arma_ts = arma_generate_sample(  
    ar=np.r_[1,-ar_coef], ma=np.r_[1,ma_coef],  
    nsample =100)
```



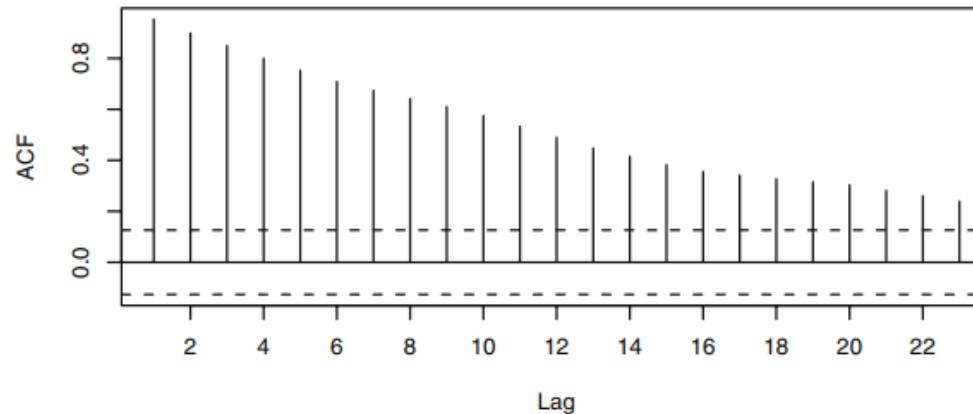
```
plot_acf(arma_ts, ax=ax)  
plt.plot(arma_acf(ar=np.r_[1, -ar_coef],  
    ma=np.r_[1,ma_coef], lags=20))
```



Primera idea para testear estacionariedad

Autocorrelación

Un método relativamente fácil, aunque bastante a ojo, para verificar que una serie **no es estacionaria** es a través de su función de autocorrelación muestral.



Si la gráfica no alcanza valores nulos para lags grandes es porque posiblemente no sea estacionaria

Tests para determinar estacionariedad

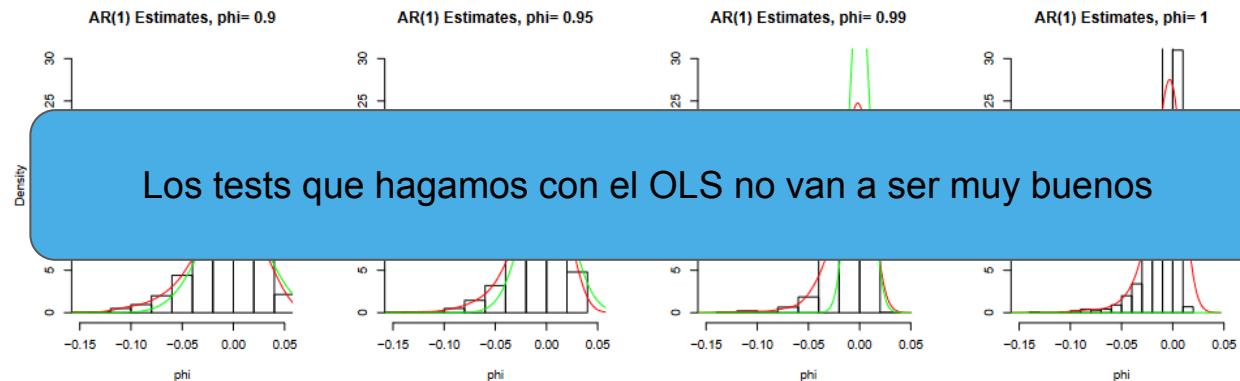
Tengo un modelo AR(1) de la forma $Y_t = a_1 Y_{t-1} + e_t$ y quiero saber si $a_1 = 1$.

Si desconocemos la estacionariedad del modelo, sólo podemos estimar el parámetro por OLS:

$$\hat{a}_1 = \frac{\sum_{t=1}^n Y_t Y_{t-1}}{\sum_{t=1}^n Y_{t-1}^2}$$

Se puede demostrar que $\sqrt{n}(\hat{a}_1 - a_1) \sim \mathcal{N}(0, 1 - a_1^2)$ si $|a_1| < 1$.

Cuando $a_1 \approx 1$, esta aproximación deja de ser válida.



Test de Dickey-Fuller

Propone el modelo $Y_t = aY_{t-1} + X_t$ ← Proceso estacionario

Si $a=1 \rightarrow$ El proceso es **no estacionario** (hay caminante aleatorio)

Si miramos la serie diferenciada una vez, tenemos que

$$Y_t - Y_{t-1} = aY_{t-1} + \cancel{X_t} - Y_{t-1} = (a - 1)Y_{t-1} + \cancel{X_t}$$

Dickey y Fuller proponen entonces el test

$$H_0 : (a - 1) = 0 \quad vs. \quad H_1 : (a - 1) \neq 0$$

Busco rechazar el
test

Su gran aporte fue hallar la distribución asintótica de $\widehat{n(a-1)}$ bajo H_0 .

Es un test para determinar si la serie (**a una diferenciación**), posee una componente de RW

Test de dickey-Fuller Aumentado

Incorpora al modelo un término de ruido dependiente (pero estacionario)

$$Y_t = aY_{t-1} + X_t \quad X_t = \sum_{j=1}^p \rho_j X_{t-j} + w_t \quad \text{Ruido Blanco}$$

Si tomamos la primera diferencia, y observamos que bajo H0

$$X_t = Y_t - Y_{t-1} \text{ tenemos que}$$

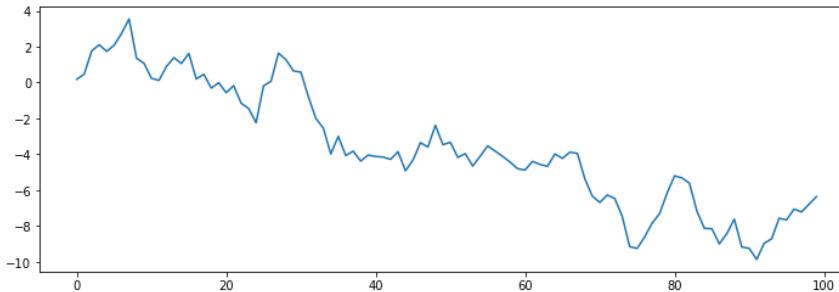
$$Y_t - Y_{t-1} = (a - 1)Y_{t-1} + \sum_{j=1}^p \rho_j (X_{t-j} - X_{t-j-1}) + W_t$$

Nuevamente se definen las hipótesis:

$$H_0 : (a - 1) = 0 \quad vs. \quad H_1 : (a - 1) \neq 0$$

p habría que estimarlo, pero podemos dejar que se encargue el software.

¿Cómo implementar en statsmodels?



```
from statsmodels.tsa.stattools import adfuller  
adfuller(y)  
✓ 0.9s  
  
(-1.394757894883641, ← Estadístico  
 0.584786653359185, ← p-valor  
 4, ← # lags usados  
 95, ← # observaciones usadas  
 {'1%': -3.5011373281819504,  
  '5%': -2.8924800524857854, ← umbrales  
  '10%': -2.5832749307479226},
```

regression : {"c","ct","cct","n"}

Constant and trend order to include in regression.

- "c" : constant only (default).
- "ct" : constant and trend.
- "cct" : constant, and linear and quadratic trend.
- "n" : no constant, no trend.

Ejemplo con datos reales

```
infile = "../Datasets/TEC02.2000.2021.csv"  
  
ts = pd.read_csv(infile, header=0, index_col=0, squeeze=True)  
ts.fechaHora = pd.to_datetime(ts.fechaHora)  
ts.fechaHora=pd.to_datetime(ts.fechaHora).dt.date  
ts.fechaHora=pd.DatetimeIndex(ts.fechaHora)  
ts=ts.sort_index(ascending=False)
```

Scripts/Dickey-Fuller_dataset_example.ipynb

```
adfuller(ts.ultimoPrecio)  
(0.57890231841091,  
 0.9870838971395222,  
 32,  
 4807,  
 {'1%': -3.431711097447145,  
 '5%': -2.862141452575749,  
 '10%': -2.5670901548483767},
```

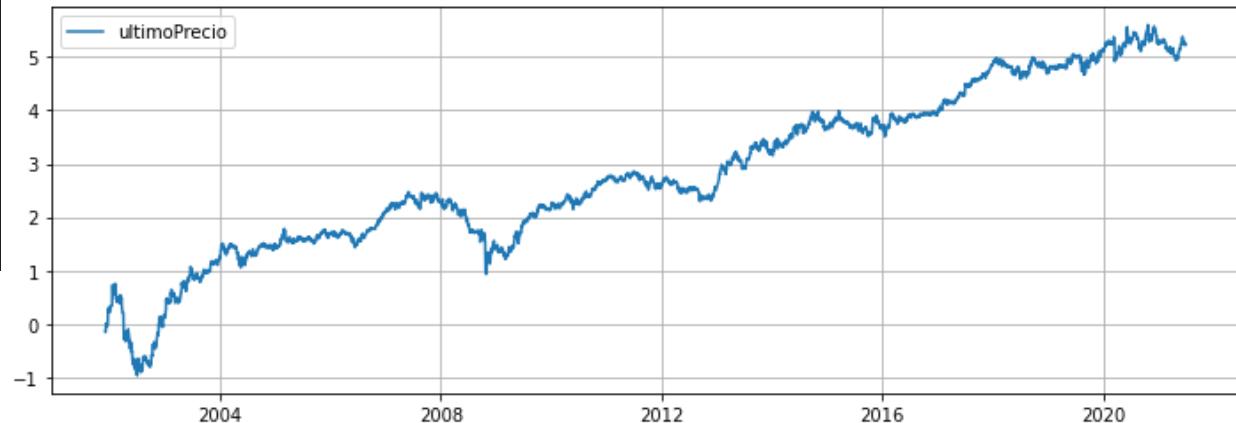


Ejemplo con datos reales

```
log_ultimoPrecio = np.log(ts.ultimoPrecio)
```

```
adfuller(log_ultimoPrecio, regression='ct')
```

```
(-3.037699968528972,  
 0.1218440464079108,  
 2,  
 4837,  
 {'1%': -3.9606428515465306,  
 '5%': -3.4113980566548285,  
 '10%': -3.127584714163724},
```

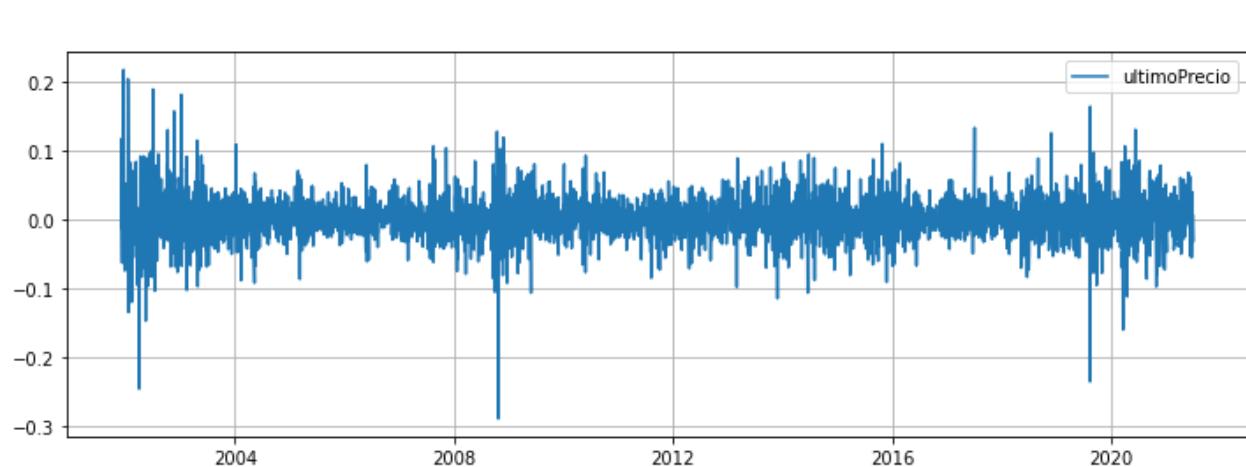


Ejemplo con datos reales

```
diff_log_ultimoPrecio = log_ultimoPrecio - log_ultimoPrecio.shift(1)
```

```
adfuller(diff_log_ultimoPrecio.dropna())
```

```
(-46.783753322231384,
 0.0,
 1,
 4837,
 {'1%': -3.4317026511738518,
 '5%': -2.862137721117907,
 '10%': -2.567088168437432},
 -20357.65641243886)
```



Conclusiones

- Vimos modelos con **tendencia determinística** con ejemplos: **lineal, cuadrático, cílico y coseno**
- Hicimos ajustes por **cuadrados mínimos** para encontrar los **valores de los parámetros** a estimar
- En la segunda parte, vimos **modelos estocásticos: MA, AR y ARMA**