

# Análisis de Series de Tiempo

Carrera de Especialización para Inteligencia Artificial

---

Clase <sup>5</sup>

Ing. Magdalena Bouza, Ing. Carlos German Carreño Romano

# Temas

---

- Diagnóstico de modelos
- Predicción
- Análisis espectral
- Caso de estudio: pronóstico de demanda de Internet

# Diagnóstico de modelos

# Diagnóstico de los modelos

---

Se basa en determinar la bondad de la estimación del modelo, y en caso de tener un mal ajuste proponer modificaciones apropiadas.

Vamos a ver dos enfoques complementarios:

- Análisis residual
- Análisis de modelos sobre-parametrizados

# 1. Análisis de residuos

---

**Residuo:** la diferencia entre el valor verdadero y el estimado por el modelo:

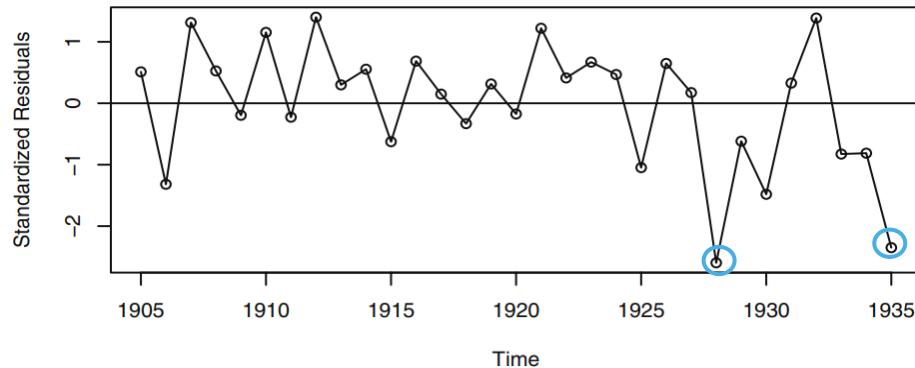
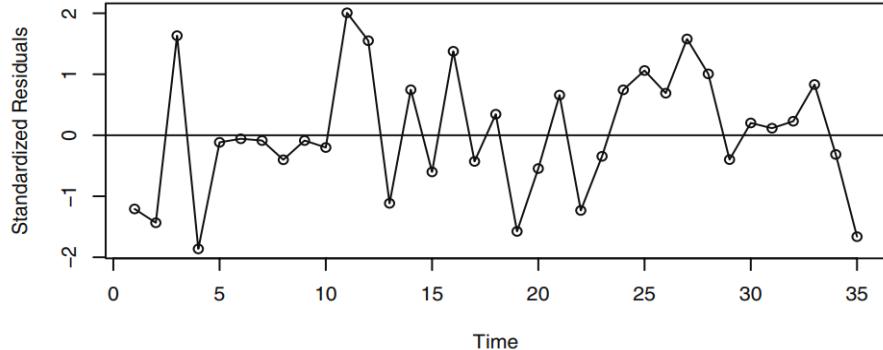
$$\hat{e}_t = Y_t - \hat{Y}_t$$

Si el modelo se encuentra bien estimado, los **residuos** deberían tener aproximadamente las propiedades del **ruido blanco**.

Las desviaciones de esta propiedad pueden ayudarnos a corregir el modelo.

# Gráfico de residuos

El primer testeo de diagnóstico consiste en graficar los residuos obtenidos a lo largo del tiempo.

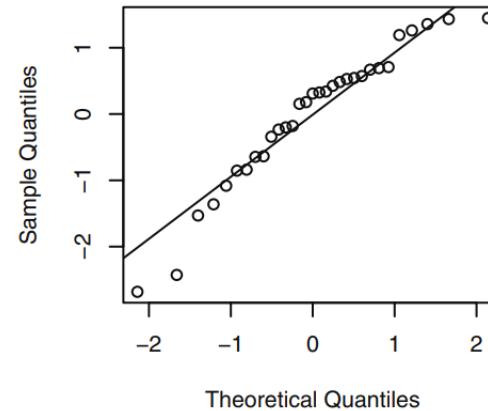
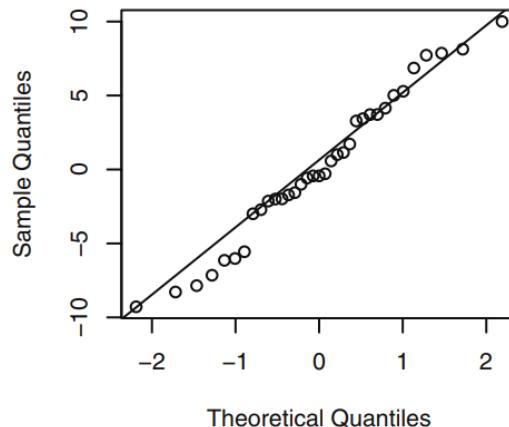


# Normalidad de los residuos

---

Una buena forma de analizar la normalidad de los residuos es mediante un QQplot.

También pueden analizarse tests estadísticos como el Shapiro-Wilk (recordar que si la cantidad de muestras es muy grande este camino puede no ser recomendable).



# Normalidad de los residuos

---

Otra forma de verificar es usando el **test de Jarque-Bera**, que busca comprobar si los datos tienen la **asimetría** y la **curtosis** de una distribución normal.

↳ 0  
↳ 3 (curtosis en exceso=0)

**H0:** los datos siguen una distribución normal → Busco no rechazar  
(p-valor>0.05)

Estadístico:  $JB = \frac{n}{6} \left( S^2 + \frac{1}{4} (K - 3)^2 \right)$   $JB \stackrel{(a)}{\sim} \chi_2^2$

$$S = \frac{\hat{\mu}_3}{\hat{\sigma}^3} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^3}{\left( \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^{3/2}}$$

$$K = \frac{\hat{\mu}_4}{\hat{\sigma}^4} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^4}{\left( \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^2}$$

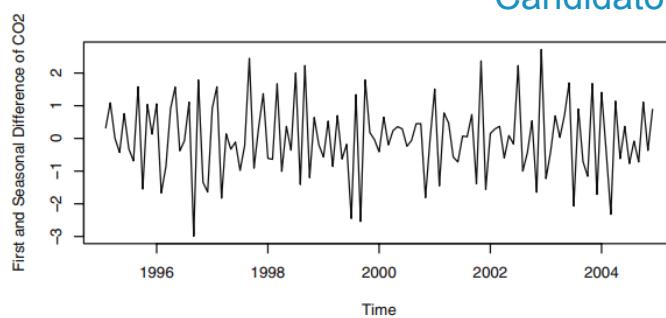
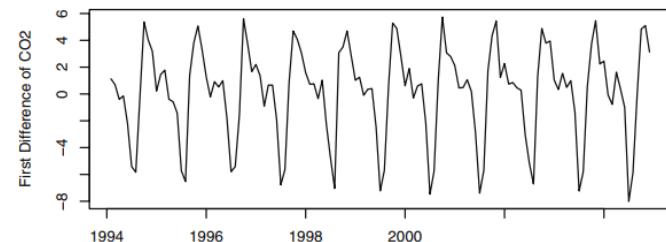
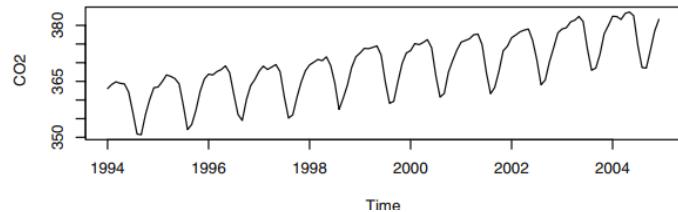
# Autocorrelación de los residuos

---

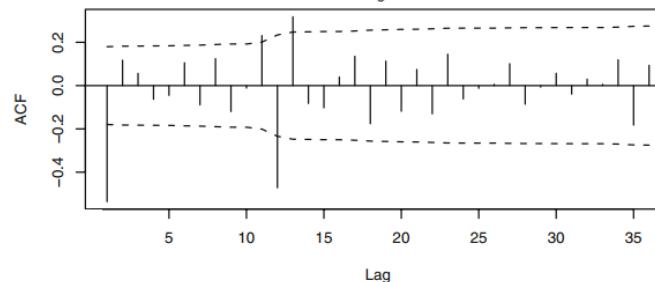
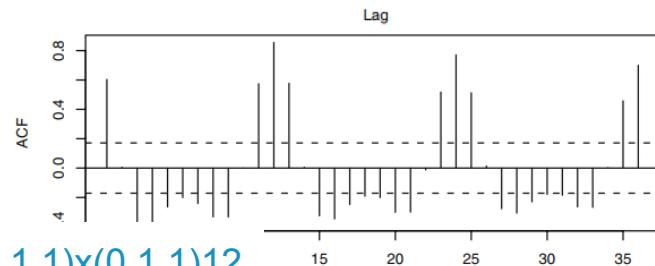
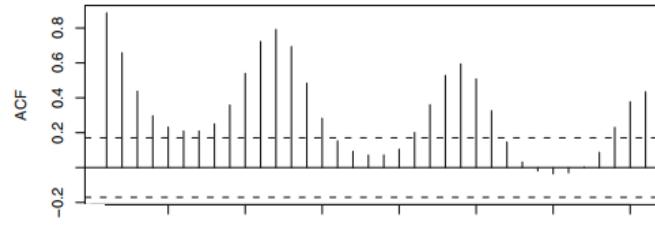
Para analizar la **independencia de los residuos** podemos estimar su función de autocorrelación.

**¡Atención!** Idealmente, para tamaños grandes de muestra, las estimaciones de las correlaciones siguen una distribución normal de media 0 y varianza  $1/n$ . Sin embargo esto no resulta del todo cierto para lags pequeños  $j, k$ , donde la varianza puede resultar significativamente menor y estar altamente correlacionados

# Especificación del modelo - ejemplo



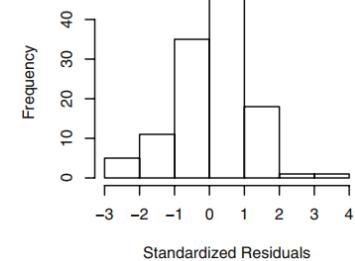
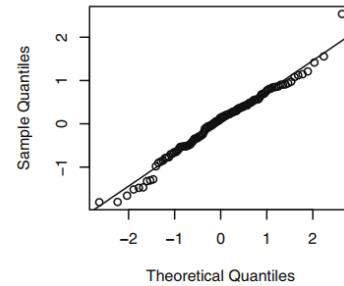
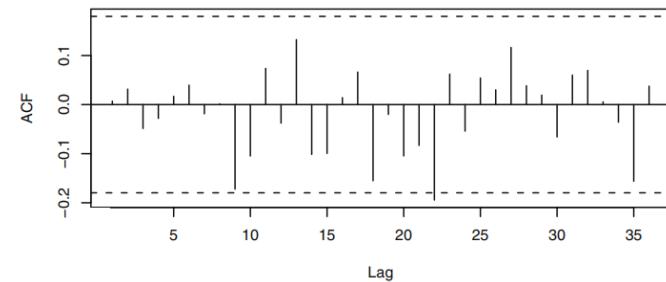
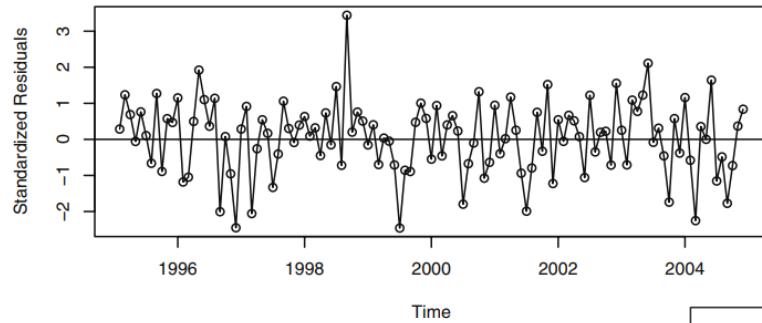
Candidato: SARIMA(0,1,1)x(0,1,1)12



# Ajuste del modelo

Una vez especificado el modelo (SARIMA(0,1,1)x(0,1,1)12) debemos ajustar los parámetros.

Nuevamente vamos a analizar los residuos de la estimación



# Test de Ljung-Box

---

Contempla las magnitudes de estas **autocorrelaciones en conjunto**.

**Box y Pierce** propusieron el estadístico  $Q = n(\hat{r}_1^2 + \hat{r}_2^2 + \dots + \hat{r}_k^2)$ . Mostraron que si los órdenes  $p$  y  $q$  del ARMA están bien estimados, y  $n$  es grande,  $Q \approx \chi^2_{k-p-q}$ . El problema es que la dist. asintótica está basada en un teorema límite.

**Ljung y Box** demostraron que esta dist. no se cumple para tamaños comunes de muestras. Propusieron  $Q_* = n(n+2)(\frac{\hat{r}_1^2}{n-1} + \frac{\hat{r}_2^2}{n-1} + \dots + \frac{\hat{r}_k^2}{n-1})$  que se asemeja mucho más a la distribución chi2.

En ambos casos el test asociado es **H0: “Los residuos están descorrelacionados”**. Busco **no rechazar** la hipótesis nula

## 2. Overfitting y redundancia de parámetros

---

Cuando tenemos un modelo ARMA, se presenta el problema de redundancia de parámetros o falta de identificabilidad. Si  $a(B)Y_t = b(B)e_t$  es el modelo correcto, luego también es correcto el modelo  $(1 - cB)a(B)Y_t = (1 - cB)b(B)e_t$  para cualquier constante  $c$ . Sin embargo, si el modelo original se correspondía con un ARMA(p,q), el segundo es un ARMA(p+1,q+1). Decimos en este caso que hay redundancia de parámetros.

1. Especificamos el modelo más sencillo que se vea factible (antes de probar alguno más complejo)
2. Al hacer overfitting, agrandamos la parte MA y la AR por separado
3. Expandir el modelo en la dirección sugerida por el análisis de residuos.

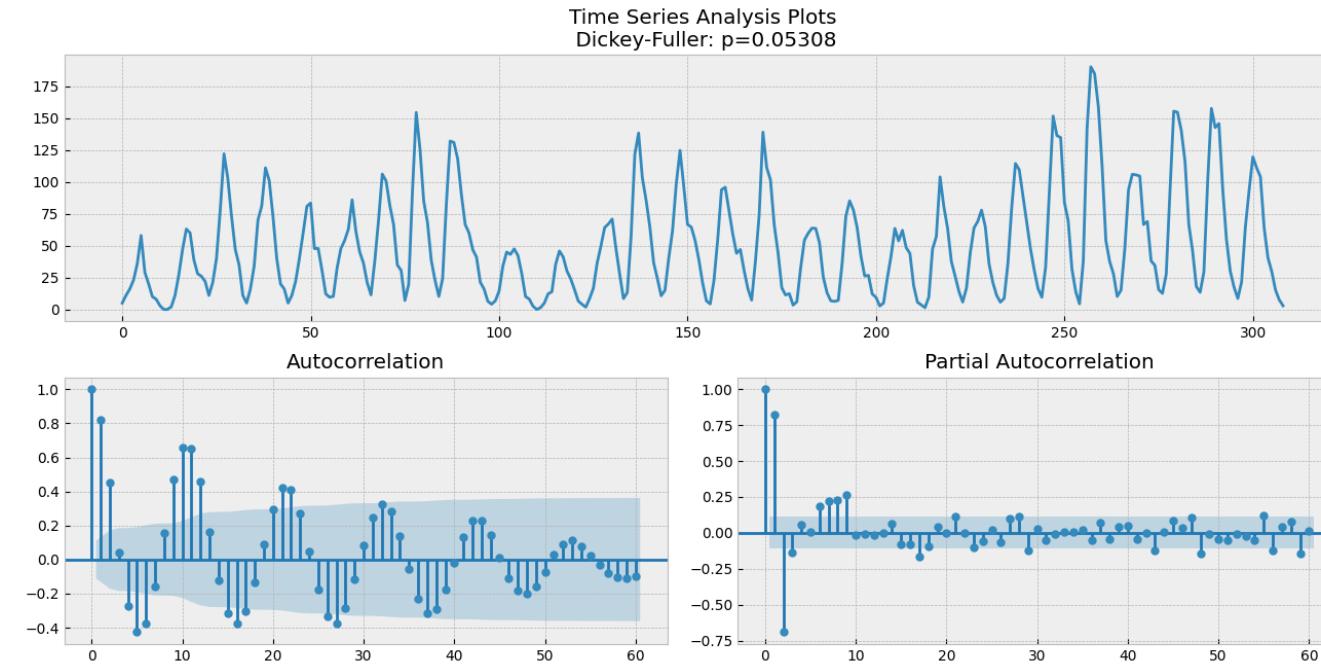
# Resultados (S) ARIMA

```
1                               SARIMAX Results
2 =====
3 Dep. Variable:                      y     No. Observations:      4840
4 Model:                 ARIMA(5, 1, 0)   Log Likelihood:   -10901.009
5 Date:                 Thu, 11 Nov 2021   AIC:                  21814.019
6 Time:                      17:51:37      BIC:                  21852.926
7 Sample:                   0 - 4840      HQIC:                 21827.678
8 Covariance Type:            opg
9
10 =====
11          coef    std err      z      P>|z|    [ 0.025   0.975]
12  -----
13 ar.L1      0.0367    0.004    9.613    0.000     0.029    0.044
14 ar.L2      0.0218    0.005    4.612    0.000     0.013    0.031
15 ar.L3      0.0451    0.005    9.124    0.000     0.035    0.055
16 ar.L4     -0.0205    0.005   -3.941    0.000    -0.031   -0.010
17 ar.L5      0.0056    0.005    1.041    0.298    -0.005    0.016
18 sigma2     5.2995    0.027  199.035    0.000     5.247    5.352
19
20 Ljung-Box (L1) (Q):            0.00  Jarque-Bera (JB):      227927.17
21 Prob(Q):                      1.00  Prob(JB):                 0.00
22 Heteroskedasticity (H):      953.07  Skew:                  -0.30
23 Prob(H) (two-sided):        0.00  Kurtosis:                 36.62
24
```

# Volvamos al ejemplo de sun activity

---

Serie de actividad del sol (SUNACTIVITY)

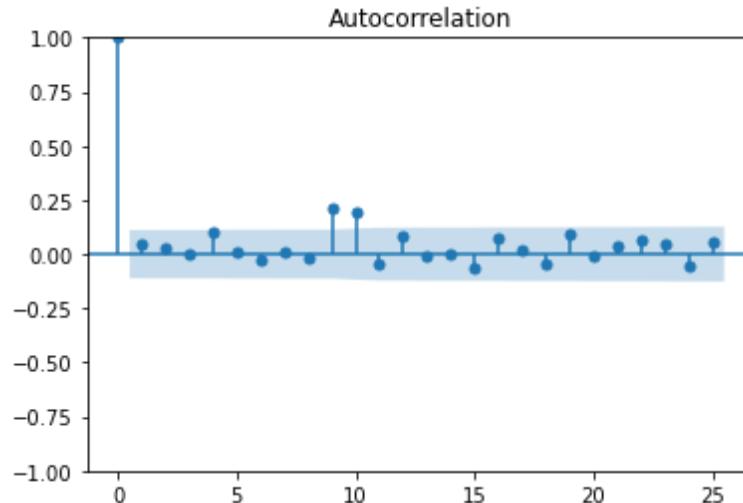
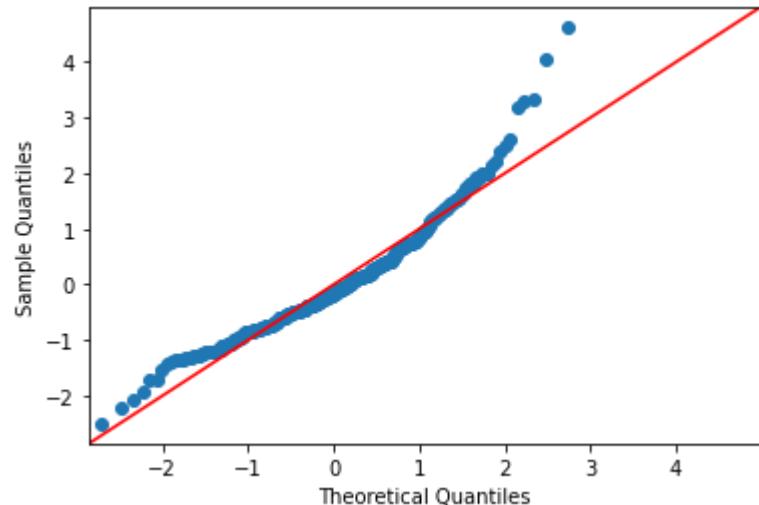


# Ajuste de modelo SARIMAX(2,0,3)x(1,1,1)10

No. Observations:	309
Log Likelihood	-1257.865
AIC	2531.730
BIC	2561.307
HQIC	2543.569

	coef	std err	z	P> z	[0.025	0.975]
ar.L1	1.2853	0.183	7.037	0.000	0.927	1.643
ar.L2	-0.6071	0.108	-5.632	0.000	-0.818	-0.396
ma.L1	-0.0998	0.187	-0.533	0.594	-0.467	0.267
ma.L2	0.0945	0.138	0.685	0.493	-0.176	0.365
ma.L3	0.0224	0.122	0.184	0.854	-0.217	0.261
ar.S.L11	0.2089	0.074	2.826	0.005	0.064	0.354
ma.S.L11	-0.9780	0.144	-6.787	0.000	-1.260	-0.696
sigma2	247.3620	30.231	8.182	0.000	188.110	306.614
Ljung-Box (L1) (Q): 0.08				Jarque-Bera (JB): 155.97		
Prob(Q): 0.78				Prob(JB): 0.00		
Heteroskedasticity (H): 1.43				Skew: 1.12		
Prob(H) (two-sided): 0.08				Kurtosis: 5.74		

# ¿Qué pasa con los residuos?



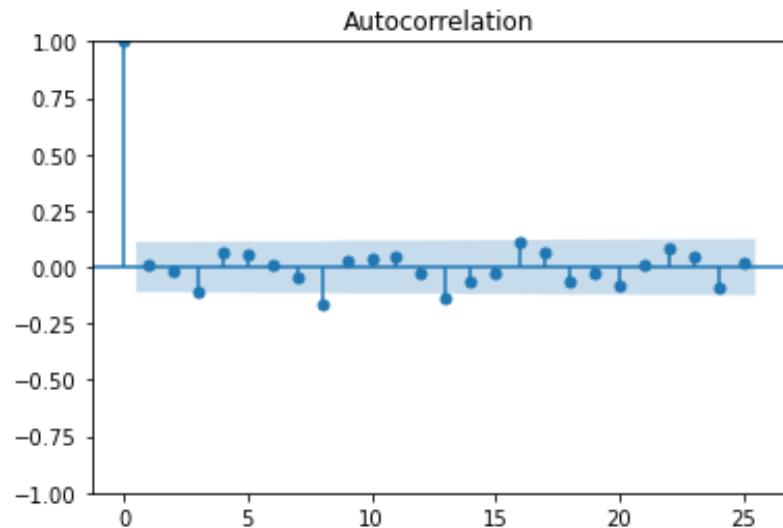
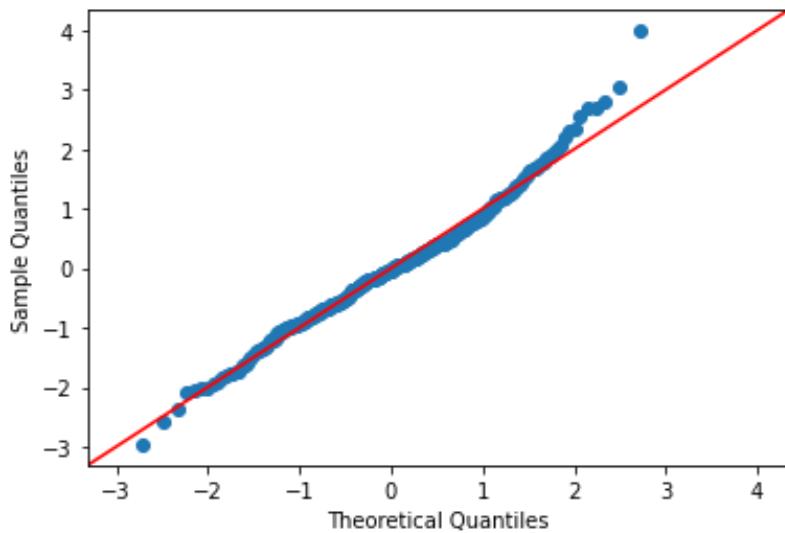
El modelo  
todavía se  
puede mejorar

## Probemos con otro: SARIMA(2,0,3)x(1,1,1)10

No. Observations:	309
Log Likelihood	-1251.934
AIC	2517.868
BIC	2543.701
HQIC	2528.211

	coef	std err	z	P> z	[0.025	0.975]
ar.L1	1.6084	0.029	55.070	0.000	1.551	1.666
ar.L2	-0.9328	0.024	-38.205	0.000	-0.981	-0.885
ma.L1	-1.4432	0.061	-23.733	0.000	-1.562	-1.324
ma.L2	0.4691	0.101	4.651	0.000	0.271	0.667
ma.L3	0.1220	0.064	1.902	0.057	-0.004	0.248
ma.S.L12	-0.9987	2.759	-0.362	0.717	-6.406	4.409
sigma2	241.3748	662.225	0.364	0.715	-1056.563	1539.313
Ljung-Box (L1) (Q):		0.00	Jarque-Bera (JB):		24.02	
Prob(Q):		0.97	Prob(JB):		0.00	
Heteroskedasticity (H):		1.41	Skew:		0.42	
Prob(H) (two-sided):		0.09	Kurtosis:		4.11	

## Probemos con otro: SARIMA(2,0,3)x(1,1,1)10



¡Este se ve  
mejor!

# Predicciones

# Predicciones

---

Uno de los principales motivos de entrenar un modelo es para poder hacer predicciones acerca de los valores que va a tomar la serie de tiempo.

Es sumamente importante también conocer la precisión de estas estimaciones.

 Para la mayor parte de lo que veamos esta clase vamos a suponer que conocemos perfectamente el modelo verdadero. Si bien es una suposición que en la práctica no se cumple, el uso de parámetros estimados cuando se cuenta con una cantidad grande de muestras no modifica significativamente los resultados presentados.

# Objetivo

---

Dadas la historia disponible en un instante  $t$ ,  $Y_1, \dots, Y_t$  queremos predecir el valor de  $Y_{t+k}$  que va a ocurrir dentro de  $k$  instantes de tiempo.

De la materia Probabilidad y Estadística, sabemos que el mejor predictor de  $Y_{t+k}$  basado en las muestras  $Y_1, \dots, Y_t$  es la esperanza condicional

$$\hat{Y}_t(\ell) = \mathbb{E}[Y_{t+\ell} | Y_1, \dots, Y_t]$$

## Predicción con tendencias determinísticas

---

Consideremos el caso visto en la clase 3 donde  $Y_t$  podía modelarse como

$$Y_t = X_t + \mu_t$$

donde  $X_t$  es una serie de tiempo de media nula y  $\mu_t$  es la tendencia determinística.

Si además  $X_t$  tiene las propiedades de ruido blanco de varianza  $C_0^2$ , ocurre que  $X_t, X_l$  son independientes  $\forall t \neq l$  y

$$\hat{Y}_{t+\ell} = \mathbb{E}[Y_{t+\ell} | Y_1, \dots, Y_t] = \mathbb{E}[\mu_{t+\ell} + X_{t+\ell} | Y_1, \dots, Y_t] = \mu_{t+\ell}$$

El error de predicción en este caso resulta

$$e_t(\ell) = Y_{t+\ell} - \hat{Y}_t(\ell) = X_{t+\ell}$$

con

$$\mathbb{E}[e_t(\ell)] = 0 \text{ y } \text{var}(e_t(\ell)) = C_0$$

# Predicción de ARIMA - ejemplo AR(1)

---

Consideremos un AR(1) con media distinta de cero:

$$Y_t - \mu = a_1(Y_{t-1} - \mu) + e_t$$

Luego, la predicción a un paso resulta

$$\hat{Y}_t(1) - \mu = \mathbb{E}[Y_{t+1} | Y_1, \dots, Y_t] = \mathbb{E}[a_1(Y_t - \mu) + e_t | Y_1, \dots, Y_t] = a_1(Y_t - \mu)$$

Podemos generalizar la predicción a  $\ell$  pasos como

$$\hat{Y}_t(\ell) = \mu + a_1^\ell(Y_t - \mu)$$

Observar que si  
 $|a_1| < 1$  y  $\ell$  es  
grande  
 $\mathbb{E}[\hat{Y}_t(\ell)] \approx \mu$

# Predicción de ARIMA - ejemplo AR(1)

Siguiendo la misma lógica de antes,

$$\begin{aligned} \boxed{e_t(1)} &= Y_{t+1} - \hat{Y}_t(1) = Y_{t+1} - (a_1(Y_t - \mu) + \mu) \\ &= a_1(Y_t - \mu) + \mu + e_{t+1} - (a_1(Y_1 - \mu) + \mu) = \boxed{e_{t+1}} \end{aligned}$$

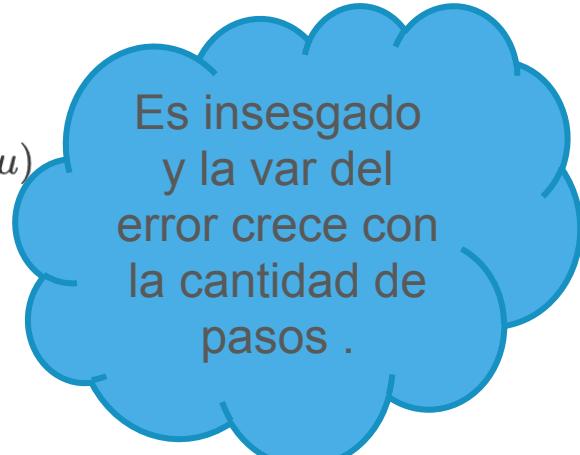
De donde obtenemos que

$$\mathbb{E}[e_t(1)] = 0 \text{ y } \text{var}(e_t(1)) = C_0$$

Para el caso general:  $e_t(\ell) = Y_{t+\ell} - \hat{Y}_t(\ell) = Y_{t+\ell} - (a_1^\ell(Y_t - \mu) + \mu)$

$$\begin{aligned} &= e_{t+\ell} + a_1 e_{t+\ell-1} + \dots + a_1^{\ell-1} e_t \\ &= e_{t+\ell} + \psi_1 e_{t+\ell-1} + \dots + \psi_{\ell-1} e_{t+1} \end{aligned}$$

$$\begin{aligned} \mathbb{E}[e_t(\ell)] &= 0 \text{ y } \text{var}(e_t(\ell)) = (1 + \psi_1 + \dots + \psi_{\ell-1})\sigma_e^2 \\ &= \sigma_e^2 \left( \frac{1 - a_1^{2\ell}}{1 - a_1^2} \right) \end{aligned}$$



Es insesgado  
y la var del  
error crece con  
la cantidad de  
pasos .

## Predicción de ARIMA - ejemplo MA(1)

---

Consideremos ahora un MA con media no nula:  $Y_t = \mu + e_t - b_1 e_{t-1}$

En este caso resulta que la predicción a un paso se puede aproximar como

$$\hat{Y}_t(1) = \mathbb{E}[\mu + e_{t+1} - b_1 e_t | Y_1, \dots, Y_t] = \mu - b_1 \mathbb{E}[e_t | Y_1, \dots, Y_t] \approx \mu - b_1 e_t$$

Nuevamente, podemos obtener una expresión más general

$$\hat{Y}_t(\ell) = \mathbb{E}[\mu + e_{t+\ell} - b_1 e_{t+\ell-1} | Y_1, \dots, Y_t] = \mu \quad \ell > 1$$

Para el caso de  $\ell = 1$ , tenemos que el error resulta

$$e_t(1) = e_{t+1}$$

## Predicción de ARIMA - ejemplo Random Walk

---

Consideremos el caso del caminante aleatorio con deriva (*drift*)

$$Y_t = Y_{t-1} + \theta - e_t$$

donde la predicción a un paso resulta

$$\hat{Y}_t(1) = \mathbb{E}[Y_t + \theta - e_{t+1} | Y_1, \dots, Y_t] = Y_t + \theta$$

Similar al caso AR(1) la predicción a  $\ell$  pasos se puede obtener a partir de una expresión recursiva como

$$\hat{Y}_t(\ell) = Y_t + \theta\ell, \quad \ell \geq 1$$

Observar que si  $\theta \neq 0$  la predicción no converge a un valor

## Predicción de ARIMA - ejemplo Random Walk

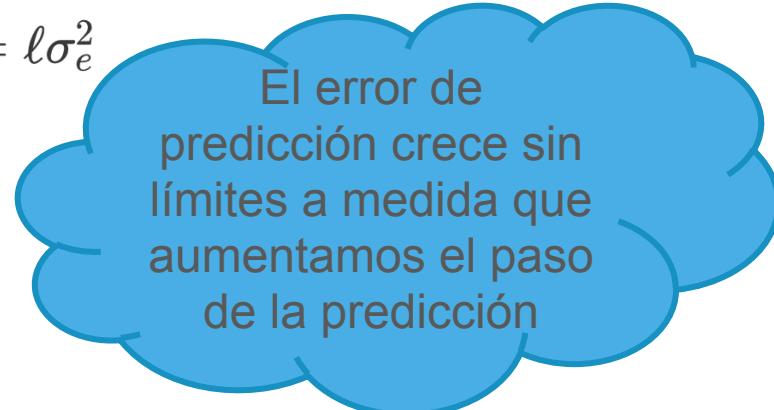
---

El error para este ejemplo del caminante aleatorio resulta

$$\begin{aligned} e_t(\ell) &= Y_{t+1} - \hat{Y}_t(\ell) = (Y_t + e_{t+1} + \dots + e_{t+\ell} + \ell\theta) - (Y_t + \theta\ell) \\ &= e_{t+1} + \dots + e_{t+\ell} \end{aligned}$$

Se obtiene entonces que

$$\mathbb{E}[e_t(\ell)] = 0 \text{ y } \text{var}(e_t(\ell)) = \ell\sigma_e^2$$



El error de predicción crece sin límites a medida que aumentamos el paso de la predicción

## Predicción de ARIMA - ejemplo ARMA(p,q)

---

Consideremos ahora un modelo ARMA(p,q)

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

En este caso, se puede ver que

$$\begin{aligned}\hat{Y}_t(\ell) = & \mathbb{E}[Y_{t+\ell} | Y_1, \dots, Y_t] = a_1 \hat{Y}_t(\ell-1) + \dots + a_p \hat{Y}_t(p) + \theta_0 - \\ & b_1 \mathbb{E}[e_{t-1} | Y_1, \dots, Y_t] + \dots + b_q \mathbb{E}[e_{t-q} | Y_1, \dots, Y_t]\end{aligned}$$

donde  $\theta_0 = \mu(1 - a_1 - \dots - a_p)$ .

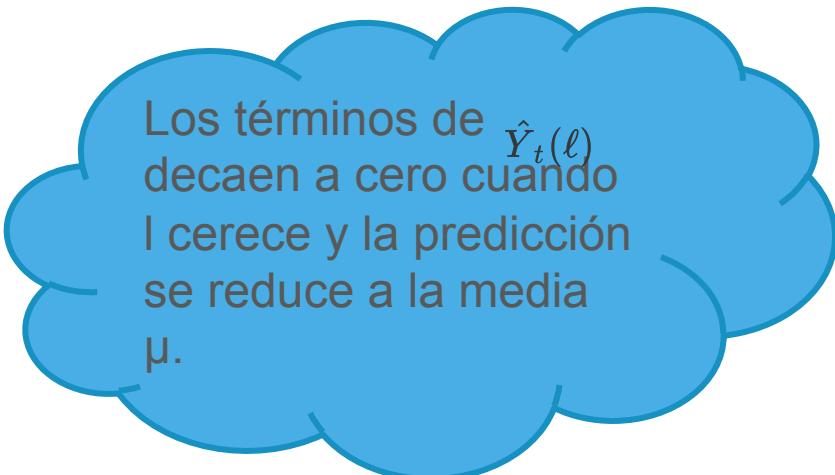
Se observa que si  $\ell > q$  los términos asociados al MA desaparecen y el comportamiento queda regido por la componente AR del modelo. Con lo cual tenemos que

$$\hat{Y}_t(\ell) = a_1(\hat{Y}_t(\ell-1) - \mu) + \dots + a_p(\hat{Y}_t(p) - \mu) + \mu \quad \ell > q$$

## Predicción de ARIMA - ejemplo ARMA(p,q)

---

Observar que la expresión hallada de  $\hat{Y}_t(\ell)$  sigue la misma recursión de Yule-Walker que la función de autocorrelación del proceso, y por lo tanto las raíces de la ec. característica determinarán el comportamiento para  $l$  grande. En particular, se puede escribir a  $\hat{Y}_t(\ell)$  como ~~sumas~~  $\sum_{i=1}^q$  de términos que decaen exponencialmente con  $l$ , y senoidales amortiguadas.



Los términos de  $\hat{Y}_t(\ell)$  decaen a cero cuando  $l$  cerece y la predicción se reduce a la media  $\mu$ .

# Modelo ARMA como proceso lineal truncado

Se puede demostrar (ver. apéndice G de Time Series Analysis) que todo modelo ARMA(p,q) se puede representar usando un proceso lineal truncado:

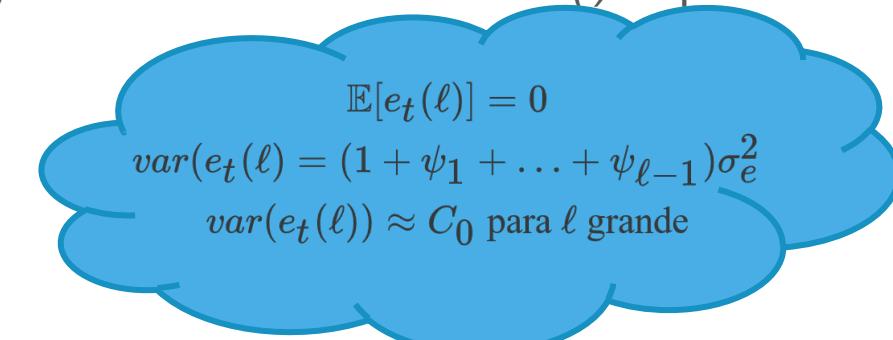
$$Y_{t+\ell} = C_t(\ell) + I_t(\ell), \quad \ell > 1$$

donde  $C_t(\ell)$  es una función de  $Y_t, Y_{t-1}, \dots$  y

$$I_t(\ell) = e_{t+\ell} + \psi_1 e_{t+\ell-1} + \dots + \psi_{\ell-1} e_{t+1}$$

Además, si  $\ell$  es lo suficientemente grande y el sistema es invertible  $C_t(\ell)$  depende sólo de  $Y_t, Y_{t-1}, \dots, Y_1$ , y

$$\hat{Y}_t(\ell) = C_t(\ell) \text{ y } e_t(\ell) = Y_{t+\ell} - \hat{Y}_t(\ell) = I_t(\ell)$$



## Predicción de ARIMA - ARIMA(p,d,q)

---

Se puede ver que un modelo ARIMA(p,d,q) se puede reescribir como un modelo ARMA(p+d,q) no estacionario:

$$Y_t = \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \dots + \varphi_{t-p-d} + e_t - b_1(e_t - 1) - \dots - b_q e_{t-q}$$

y las predicciones pueden calcularse igual que para el caso ARMA(p+d,q) y

$$e_t(\ell) = e_{t+\ell} + \psi_1 e_{t-1} + \dots + \psi_{\ell-1} e_{t+\ell-1}$$

Nuevamente es válido que

$$\mathbb{E}[e_t(\ell)] = 0 \text{ y } \text{var}(e_t(\ell)) = \sigma_e^2 \sum_{i=1}^{\ell-1} \psi_i^2$$

La diferencia es que al ser un proceso no estacionario los coeficientes no convergen a cero a medida que aumenta  $i$ , y la varianza aumenta sin cota a medida que  $\ell$  aumenta.

## Límites de la predicción

---

Como siempre, queremos saber la bondad de nuestra estimación. Hasta ahora analizamos la media y la varianza del error para cada instante predicho.

El objetivo es poder brindar un intervalo de confianza alrededor del valor predicho para cada valor de  $l$ .

Si las innovaciones ( $e_t$ ) siguen una distribución gaussiana, luego el error  $e_t(l)$  también va a seguir una distribución gaussiana, de media y varianza ya calculadas. Luego,

$$\mathbb{P}\left(z_{\alpha/2} \leq \frac{Y_{y+\ell} - \hat{Y}_{t(\ell)}}{\sqrt{\text{var}(e_t(\ell))}} \leq z_{1-\alpha/2}\right) = 1 - \alpha$$

resultando que

$$Y_{t+\ell} \in [\hat{Y}_{t(\ell)} - z_{1-\alpha/2} \sqrt{\text{var}(e_t(\ell))}, \hat{Y}_{t(\ell)} + z_{1-\alpha/2} \sqrt{\text{var}(e_t(\ell))}]$$

# Actualizando las predicciones ARIMA

---

¿Cómo actualizamos las predicciones realizadas una vez que nos llega una nueva muestra de la serie?

Supongamos que tengo la predicción  $\hat{Y}_t(\ell + 1)$ , que es la predicción del instante  $t+1$ , basados en las observaciones hasta tiempo  $t$ . Supongamos que de pronto me llega información acerca del instante  $t+1$ . Luego, quiero actualizar mi estimación de  $Y_{t+1}$ :  $\hat{Y}_{t+1}(\ell)$

$$\hat{Y}_{t+1}(\ell) = \hat{Y}_t(\ell + 1) + \psi_\ell [Y_{t+1} - \hat{Y}_t(1)]$$

Observar que se parece mucho a la predicción de Kalman!

## Predicción de series transformadas

---

Pongamos el ejemplo de la transformación logarítmica. En este caso tenemos  $Y_t$  la serie original y  $W_t = \log(Y_t)$ . Se puede demostrar que

$$\mathbb{E}[Y_{t+\ell} | Y_1, \dots, Y_t] \geq e^{\mathbb{E}[W_{t+\ell} | Y_1, \dots, Y_t]}$$

con lo cual el estimador naive  $\hat{Y}_t(\ell) = e^{\hat{W}_t(\ell)}$  no es el de ECM. Si  $W_t$  tiene distribución normal, vale que predictor de ECM es

$$\hat{Y}_t(\ell) = e^{\hat{W}_t(\ell) + 0.5 \text{var}(e_t(\ell))}$$

# Ejemplos

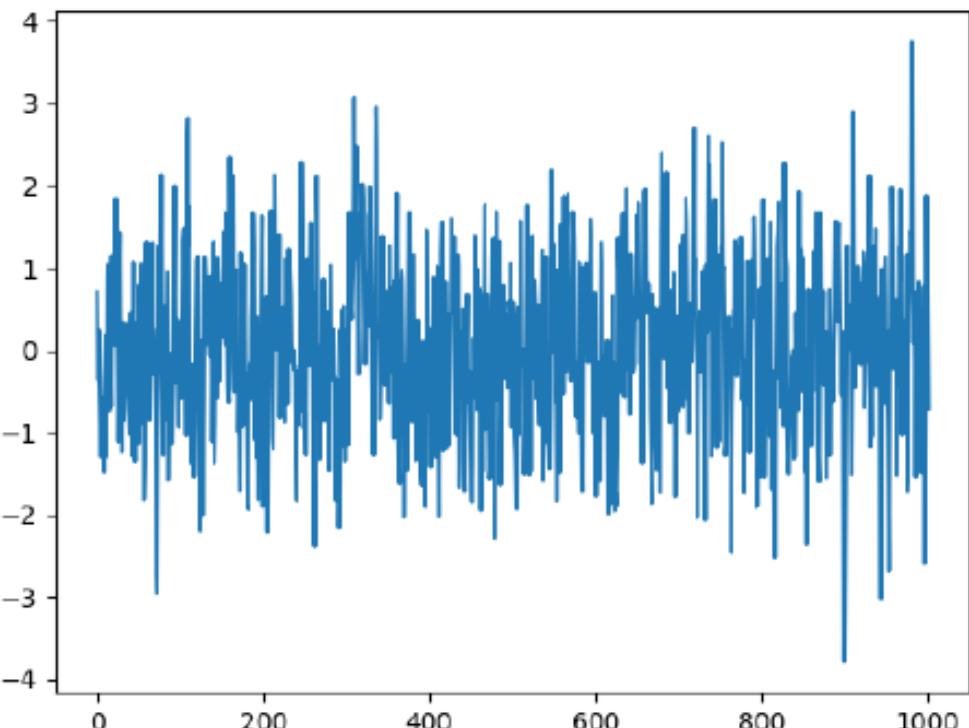
# Modelo de Promedio Móvil MA(1)

$$y_t = b_1 \epsilon_{t-1} + \epsilon_t$$

```
# MA(1)
N = 1000
b1 = 0.25
e_t0 = np.random.normal(0,1)
e_t1 = np.random.normal(0,1)
y = np.append(e_t0,b1*e_t0+e_t1) #
```

```
for i in range(N):
    e_t0 = e_t1
    e_t1 = np.random.normal(0,1)
    y = np.append(y,b1*e_t0+e_t1)
```



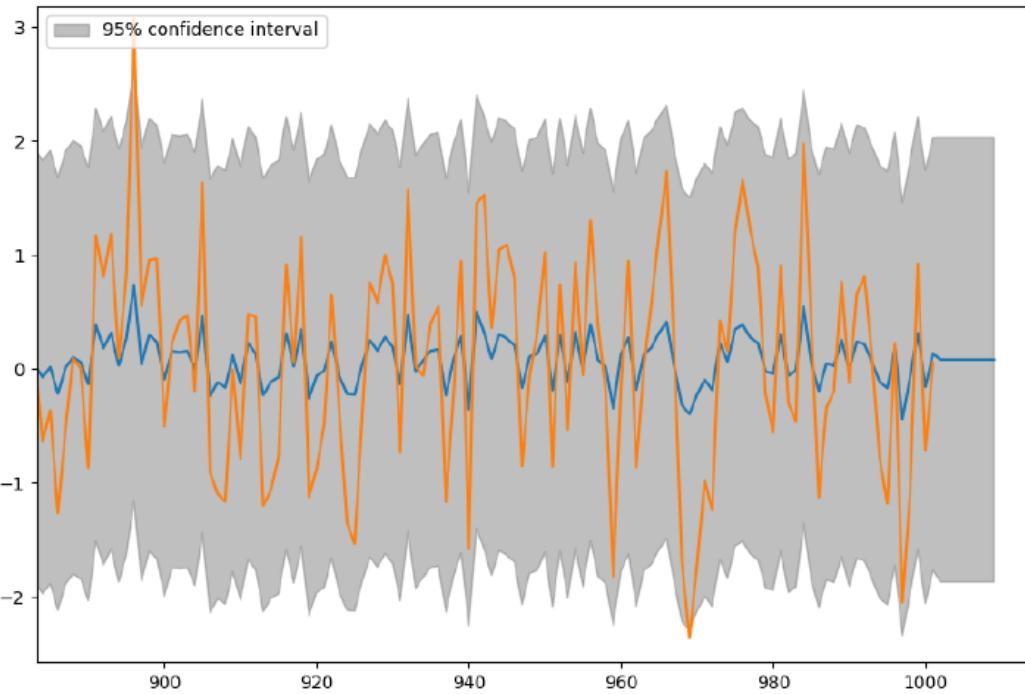
## Modelo de Promedio Móvil MA(1)

$$y_t = b_1 \epsilon_{t-1} + \epsilon_t$$

```
# Predict
ma1 = ARIMA(y, order=(0, 0, 1))
ma1_res = ma1.fit()
print(ma1_res.summary())

fig, ax = plt.subplots(figsize=(10, 8))
fig = plot_predict(ma1_res, start=1, end=1000)
plt.plot(y)
legend = ax.legend(loc="upper left")
plt.show()
```

$$\gamma_0 = (1 + \theta_1^2 + \theta_2^2 + \dots + \theta_q^2) \sigma_e^2$$

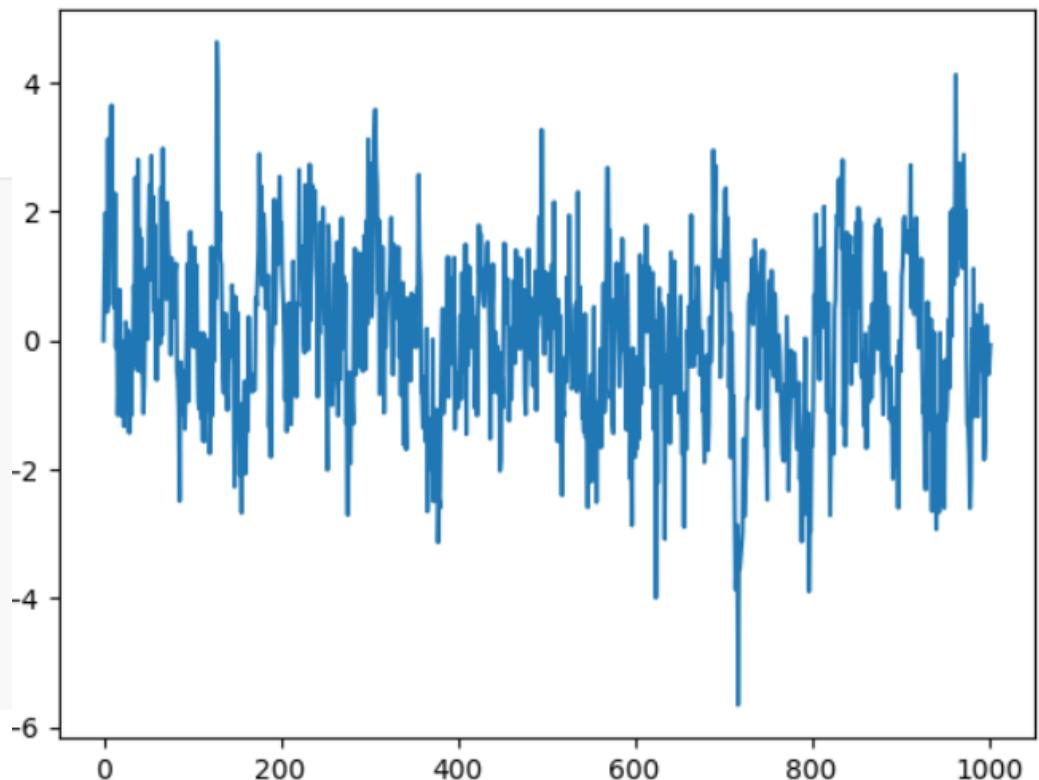


# Modelo Autoregresivo AR(2)

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \epsilon_t$$

```
# AR(2)
N=1000
a1=0.4
a2=0.35
x=np.arange(2)
e_t=np.random.normal(0,1)
y=np.append(x,a1*x[1]+a2*x[0]+e_t)

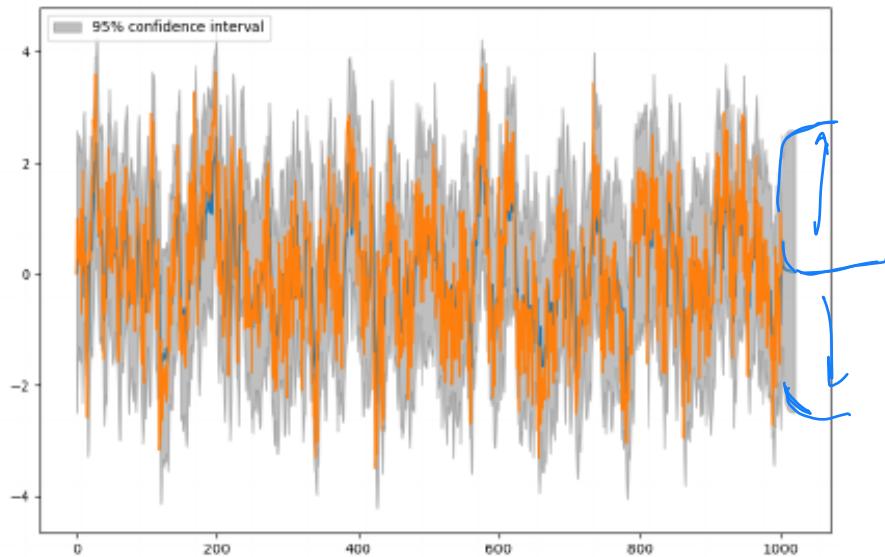
for i in range(N):
    e_t=np.random.normal(0,1)
    y=np.append(y,a1*y[-1]+a2*y[-2]+e_t)
```



# Modelo Autoregresivo AR(2)

```
# Predict
l=20
ar2 = ARIMA(y, order=(2, 0, 0))
ar2_res = ar2.fit()
print(ar2_res.summary())

fig, ax = plt.subplots(figsize=(10, 8))
fig = plot_predict(ar2_res, start=0, end=N+l, ax=ax)
plt.plot(y)
legend = ax.legend(loc="upper left")
plt.show()
```

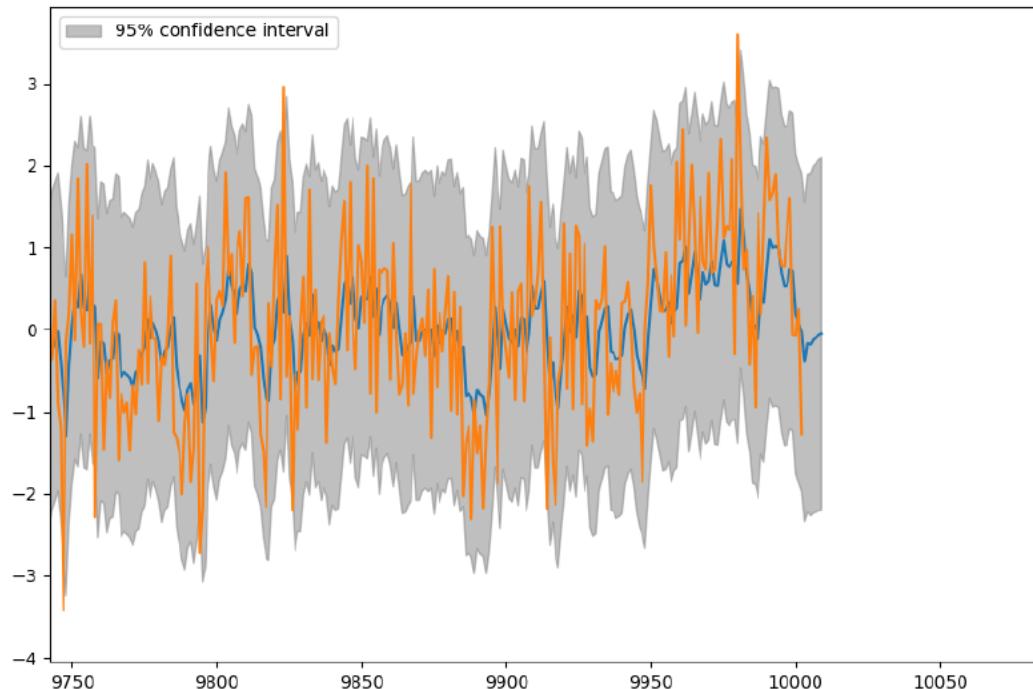


→ ver resultados del modelo

# Modelo ARMA(2,1)

```
# Predict
y_arma = ARIMA(y, order=(2, 0, 1))
y_arma_res = y_arma.fit()
print(y_arma_res.summary())

fig, ax = plt.subplots(figsize=(10, 8))
fig = plot_predict(y_arma_res, start=1, end=N
plt.plot(y)
```



# Modelo ARIMA(2,1,1)

---

```
y=np.append(y, a1*y[-1]+a2*y[-2]+ b1*e_0 + e_t)
```

```
t = np.arange(int(N*0.01), step=0.01)  
y=y[2:N+2]
```

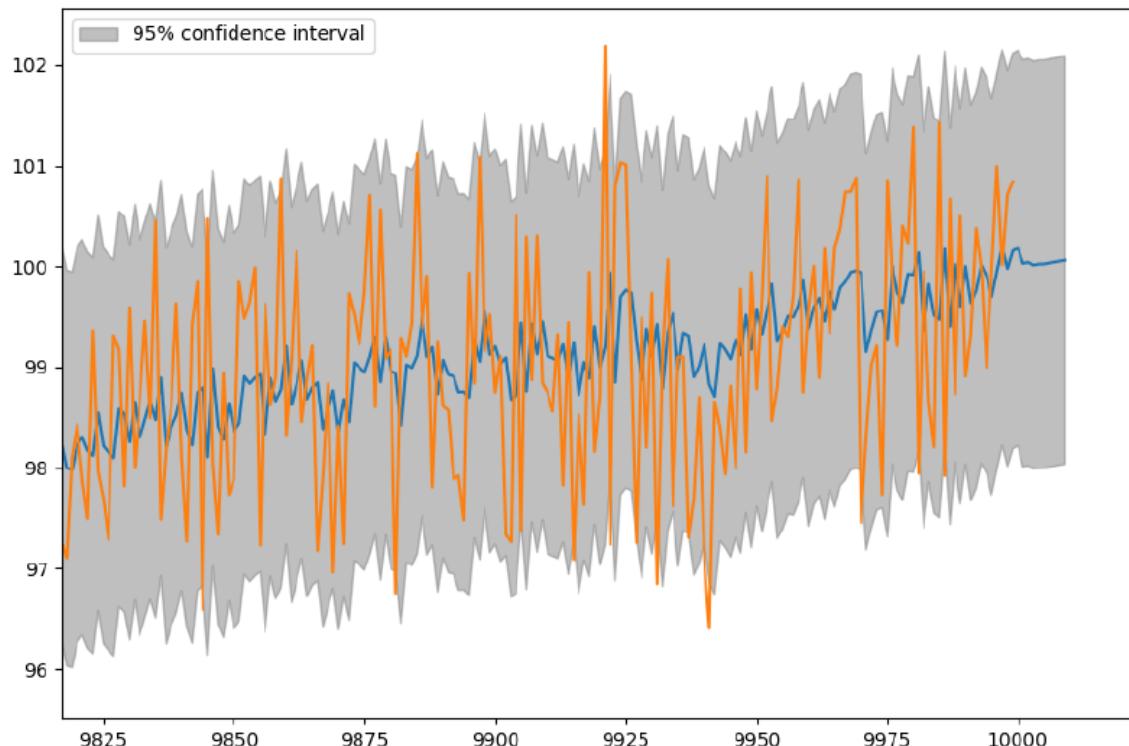
```
y = y+t
```

```
# Predict  
y_arma = ARIMA(y, order=(2, 1, 1), trend='t')  
y_arma_res = y_arma.fit()  
print(y_arma_res.summary())  
  
fig, ax = plt.subplots(figsize=(10, 8))  
fig = plot_predict(y_arma_res, start=1, end=N+10, ax=ax)
```

# Modelo ARIMA(2,1,1)

```
1                               SARIMAX Results
2 =====
3 Dep. Variable:                      y   No. Observations:                 10000
4 Model:                 ARIMA(2, 1, 1)   Log Likelihood:            -14245.820
5 Date:                 jue, 18 nov 2021   AIC:                         28501.641
6 Time:                     19:40:41      BIC:                         28537.692
7 Sample:                      0   HQIC:                         28513.844
8                               - 10000
9 Covariance Type:                opg
10 =====
11          coef    std err      z   P>|z|    [0.025    0.975]
12 -----
13 x1      0.0100  9.94e-06  1006.340      0.000    0.010    0.010
14 ar.L1    0.0335      0.010      3.470      0.001    0.015    0.052
15 ar.L2    0.2474      0.010     25.610      0.000    0.228    0.266
16 ma.L1   -0.9996      0.001    -1875.605     0.000   -1.001   -0.999
17 sigma2   1.0100      0.014      70.300      0.000    0.982    1.038
18 =====
19 Ljung-Box (L1) (Q):                  4.66   Jarque-Bera (JB):                0.24
20 Prob(Q):                           0.03   Prob(JB):                  0.89
21 Heteroskedasticity (H):                0.98   Skew:                      0.00
22 Prob(H) (two-sided):                0.61   Kurtosis:                  2.98
23 =====
```

## Modelo ARIMA(2,1,1)



## #ARIMA

N=10000

a1=0.35

$a^2 = 0.25$

$$b_1 = -0.35$$

```
t = np.arange(int(N*0.01), step=0.01)
```

```
10  =====
11  |          |          |          |          |  coef
12  -----
13  x1          0.0100
14  ar.L1       0.0335
15  ar.L2       0.2474
16  ma.L1      -0.9996
17  sigma2      1.0100
18  =====
```

# Modelos con tendencia determinística

---

- Procesos estocásticos con tendencia determinística
- Si puedo extraer la tendencia con un modelo analítico, entonces voy a poder usar modelos de forma adecuada

$$Y_t = X_t + \mu_t$$

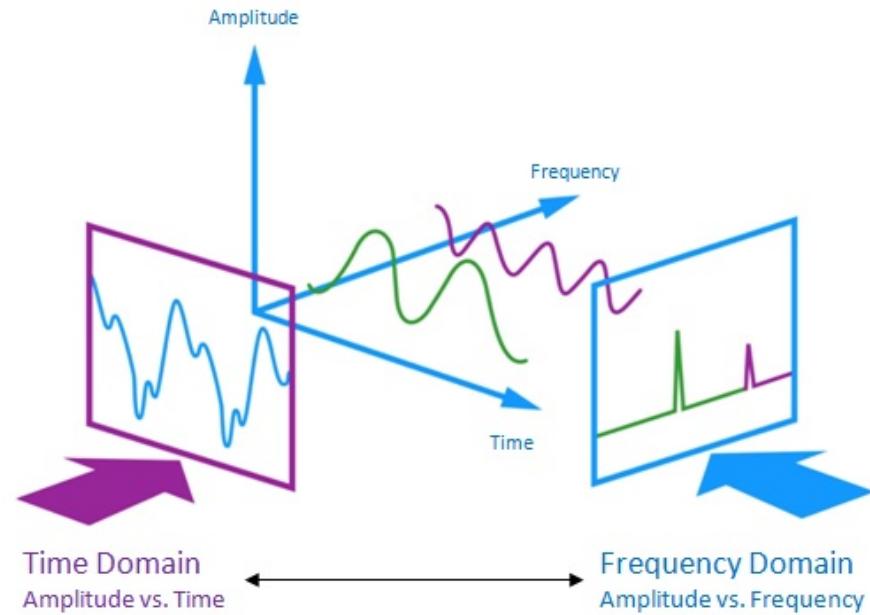
$$\cdot \mu_t := \begin{cases} \text{Constante} & \mu_t = \mu \\ \text{Lineal} & \mu_t = \beta_0 + \beta_1 t \\ \text{Cuadrática} & \mu_t = \beta_0 + \beta_1 t + \beta_2 t^2 \\ \text{Cíclica} & \mu_t = \mu_{t-T} \\ \text{Senoidal} & \mu_t = \beta \cos(\omega t + \phi) \end{cases}$$

→ ver extensión al modelo SARIMA

# Introducción al análisis espectral

# Introducción al análisis espectral

- Series periódicas
- Análisis en el dominio del tiempo
- Densidad espectral de potencia
- Periodograma
- Análisis en el dominio de frecuencias



# Series periódicas

---

Consideremos un proceso dado por la expresión

$$Y_t = R \cos (2\pi ft + \Phi)$$

Conceptos clave en este tipo de series son:

- Amplitud
- Frecuencia
- Fase

Usando trigonometría podemos pasar a una expresión distinta:

$$Y_t = R \cos (2\pi ft + \Phi) = A \cos (2\pi ft) + B \sin (2\pi ft)$$

$$Y_t = R \cos (2\pi ft + \Phi) = A \cos (2\pi ft) + B \sin (2\pi ft)$$



$$\begin{aligned} A &= R \cos (\Phi) & B &= -R \sin (\Phi) \\ R &= \sqrt{A^2 + B^2} & \Phi &= \arctan (-B/A) \end{aligned}$$

Si tenemos la frecuencia, entonces podemos ajustar los parámetros A y B a partir de, por ejemplo, cuadrados mínimos.

Una generalización de lo anterior nos permite definir un proceso 'Y' a partir de una combinación lineal de senos y cosenos de distintas frecuencias, a partir de la expresión:

$$Y_t = A_0 + \sum_{j=1}^m (A_j \cos(2\pi f_j t) + B_j \sin(2\pi f_j t))$$

Nuevamente, podemos hacer estimaciones por cuadrados mínimos para encontrar los coeficientes  $A_j$  y  $B_j$ .

# Fourier

---

Suponiendo que  $n$  es impar, esto es  $n=2k+1$ , se tiene que las frecuencias de la forma  $1/n, 2/n, \dots, k/n$  se denominan frecuencias de Fourier. Las variables predictoras del seno y coseno a tales frecuencias son ortogonales, y la estimación por cuadrados mínimos resulta simplemente:

$$\hat{A}_0 = \bar{Y}$$

$$\hat{A}_j = \frac{2}{n} \sum_{t=1}^n Y_t \cos (2\pi t j / n) \quad \hat{B}_j = \frac{2}{n} \sum_{t=1}^n Y_t \sin (2\pi t j / n)$$

El resultado de la serie de suma de cosenos y senos es general. Esto es, cualquier serie de cualquier longitud  $n$ , sea determinística o estocástica y con o sin periodicidades verdaderas puede ajustarse perfectamente por este modelo eligiendo  $m=n/2$  si  $n$  es par y  $m=(n-1)/2$  si  $n$  es impar. Luego habrá  $n$  parámetros a estimar para ajustar la serie de largo  $n$ .

## Periodograma

---

Para muestras de tamaño impar (**n=2k+1**), el periodograma se define como:

$$I(j/n) = \frac{n}{2}(\hat{A}_j^2 + \hat{B}_j^2)$$

donde  $j/n$  es la frecuencia para  $j=1, 2, \dots, k$ .

Si la muestra tiene tamaño par (**n=2k**), en la frecuencia de corte  $f=k/n=1/2$  resulta:

$$I(1/2) = n\hat{A}_k^2$$

# Periodograma - Interpretación

---

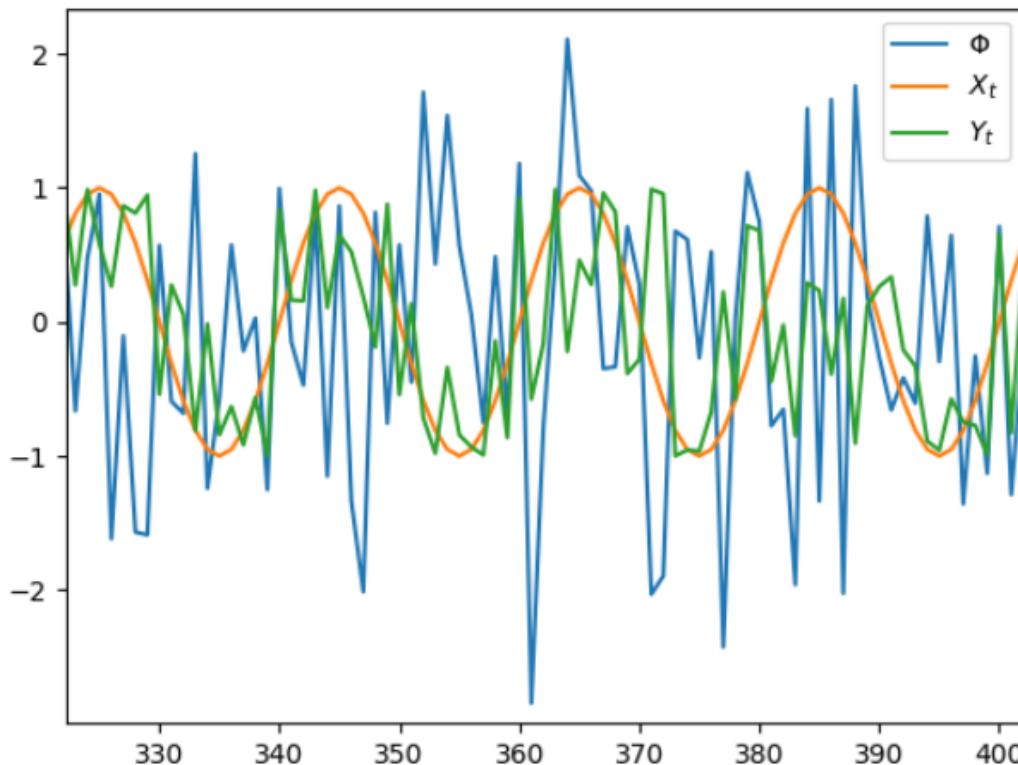
- 1. En términos de intensidad relativa:** Como el periodograma es proporcional a la suma de cuadrados de los coeficientes de regresión asociados a la frecuencia  $f=j/n$ , la altura del periodograma muestra la intensidad relativa del par sin-cos a distintas frecuencias en el comportamiento general de la serie
- 2. En términos de análisis de varianza:**  $I(j/n)$  es suma de cuadrados de asociadas al par  $(A_j, B_j)$  en la frecuencia  $j/2m$  luego

$$\sum_{j=1}^n (Y_j - \hat{Y})^2 = \sum_{j=1}^k I(j/n)$$

## Introducción al análisis espectral

```
# example 1a
N=1000
f=50
T=1/f
t=np.arange(N)
Phi=np.random.normal(0,1,N)
X=sin(2*pi*f*t/N)
Y=sin(2*pi*f*t/N + Phi)

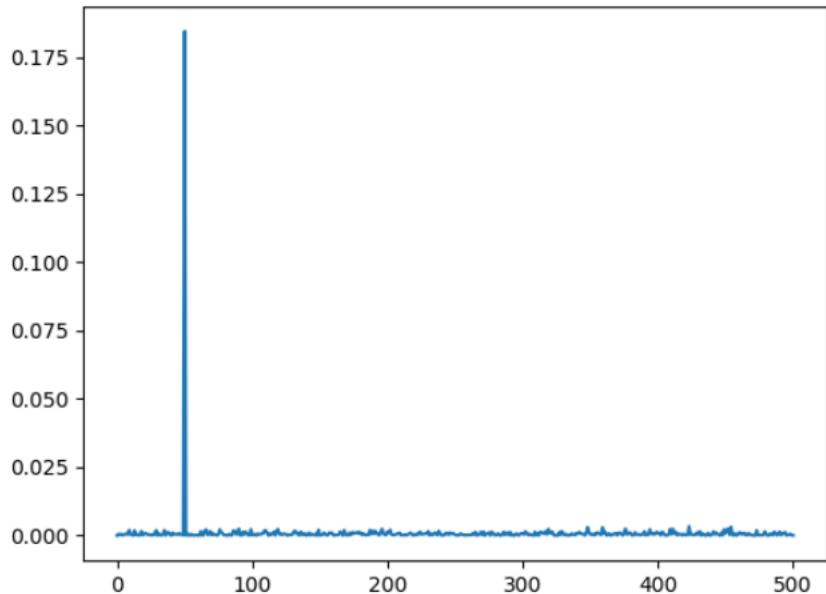
plt.plot(Phi)
plt.plot(X)
plt.plot(Y)
plt.legend(['$\\Phi$', '$X_t$',
plt.show()
```



# Periodograma

---

```
G, Pyy_den = signal.periodogram(Y, N)  
plt.plot(G, Pyy_den)
```



Se puede observar el piso de ruido y una componente marcada en la frecuencia 50.

Ejemplos hands-on  
(ver spectral\_intro.py)

## Densidad espectral de potencia

---

Si la autocovarianza decrece rápido a medida que aumentan los retardos y se satisface que tienen energía finita, esto es:

$$\sum_{k=-\infty}^{\infty} |C_k| < \infty$$

se puede definir la transformada de Fourier de la autocovarianza. La función definida entre  $-\frac{1}{2} < f < \frac{1}{2}$  como sigue se denomina ***densidad espectral de potencia (PSD)***:

$$p(f) = \sum_{k=-\infty}^{\infty} C_k e^{-2\pi j k f}$$

# Densidad espectral de potencia

---

Se puede ver que:

$$p(f) = \sum_{k=-\infty}^{\infty} C_k e^{-2\pi h k f}$$

$$= \sum_{k=-\infty}^{\infty} C_k \cos(2\pi k f) - j \sum_{k=-\infty}^{\infty} C_k \sin(2\pi k f)$$

p(f) es par! 

$$= \sum_{k=-\infty}^{\infty} C_k \cos(2\pi k f)$$

cos es par! 

$$= C_0 + 2 \sum_{k=0}^{\infty} C_k \cos(2\pi k f)$$

## Densidad espectral de potencia

---

Entonces una serie de tiempo se puede expresar en función de componentes trigonométricas y la PSD es una representación en esos términos.

También podemos partir de una PSDm y querer encontrar la función de autocovarianza, que se puede calcular a partir de la transformada inversa de Fourier:

$$C_k = \int_{-1/2}^{1/2} p(f) e^{2\pi j k f} = \int_{-1/2}^{1/2} p(f) \cos(2\pi k f) df$$

# Periodograma y Densidad espectral de Potencia

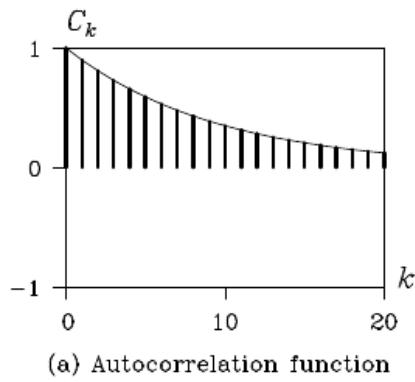
---

Las expresiones que vimos recién corresponden a expresiones **teóricas**, que se calculan a partir del conocimiento del modelo.

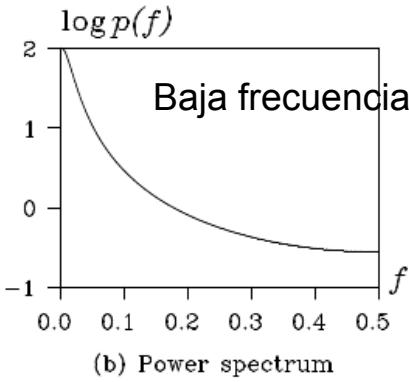
Observar que la PSD es la transformada de fourier de la función de covarianza, y por lo tanto podemos **estimarla** haciendo un periodograma de la misma

En python existe la función **scipy.signal.welch**, que devuelve la estimación de la PSD

# Ejemplo 1: densidad espectral de un proceso AR(1)



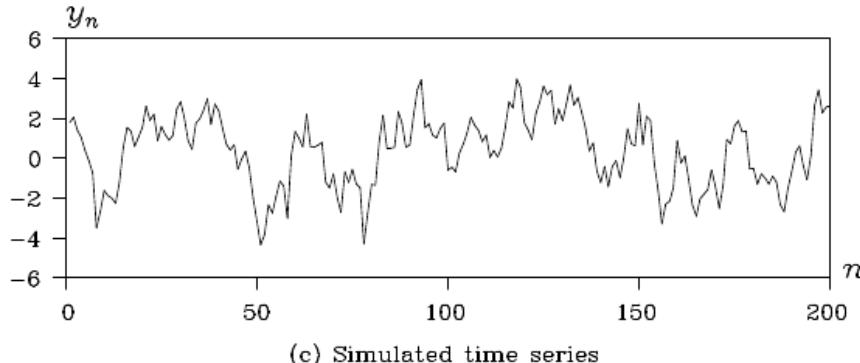
(a) Autocorrelation function



AR(1),  $a = 0.9$

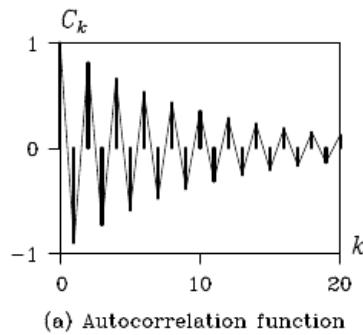
$$Y_t = a_1 Y_{t-1} + e_t$$

$$C_k = \sigma^2 \frac{a^{|k|}}{1 - a^2}$$

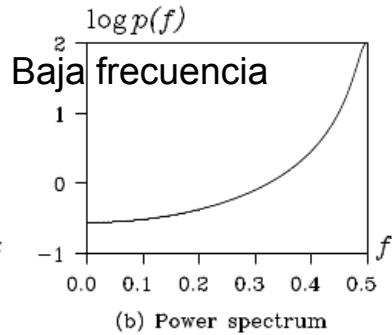


$$p(f) = \frac{\sigma^2}{|1 - ae^{2\pi j f}|^2}$$
$$= \frac{\sigma^2}{1 - 2a \cos 2\pi f + a^2}$$

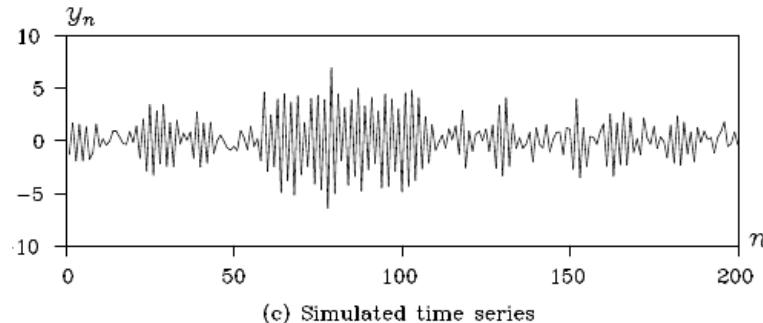
## Ejemplo 2: densidad espectral de un proceso AR(1)



(a) Autocorrelation function



(b) Power spectrum



(c) Simulated time series

AR(1),  $a = -0.9$

$$Y_t = a_1 Y_{t-1} + e_t$$

$$C_k = \sigma^2 \frac{a^{|k|}}{1 - a^2}$$

$$\begin{aligned} p(f) &= \frac{\sigma^2}{|1 - ae^{2\pi j f}|^2} \\ &= \frac{\sigma^2}{1 - 2a \cos 2\pi f + a^2} \end{aligned}$$

# Análisis en el dominio espectral

---

Muchas veces la información en el dominio del tiempo para distintas series parece ser muy distinta y sin embargo en el dominio de frecuencias son similares. Cuando la representación espectral es continua, surge la necesidad de estimar el espectro.

Algunas técnicas de estimación espectral se llaman estimaciones no-paramétricas y asumen muy poco acerca de la densidad espectral “verdadera”.

La estimación paramétrica asume que un modelo autorregresivo de orden adecuado sirve para ajustar un modelo a la serie de tiempo. Luego la densidad espectral estimada va a tener las mismas propiedades que las del autorregresivo.

# Análisis en el dominio espectral

---

- densidad espectral suavizada → <https://hakanyurdakul.com/smoothing-power-spectral-density/>
- sesgo
- varianza
- fugas
- ancho de banda
- ensanchamiento

# Resumen

---

- Modelos de series de tiempo como combinaciones lineales de senos y cosenos, introducción al análisis espectral
- El Periodograma es una herramienta para encontrar la contribución de varias frecuencias en la representación espectral
- Se puede extender el modelado a rangos continuos de frecuencias
- Se exploran las densidades espectrales de procesos ARMA
- Algunas propiedades de la densidad espectral muestral sugieren que la densidad espectral no es un estimador consistente, por lo que hay que buscar más para estimar espectros...

# Conclusiones

---

- Los modelos **AR, MA, ARMA** se basan en procesos estacionarios y están bien estudiados.
- Si un proceso no estacionario pueda ser **diferenciado** 'i' veces y volverse estacionario, entonces podemos usar la extensión del modelo **ARIMA**
- Para procesos con **tendencia constante o lineal**, el framework funciona normalmente bien con **ARIMA.fit()**
- Con la misma idea, procesos a los que se les puede extraer una tendencia determinística conviene tratarlos con modelos conocidos.
- Vimos que **Cuadrados Mínimos** puede ser útil para ajustar tendencias determinísticas y obtener los coeficientes del modelo
- La extensión **SARIMA** usa estos métodos con la misma idea de extraer componentes que se pueden modelar por descomposición
- Los **coeficientes** son **todo** a la hora de modelar para predecir. Una buena predicción se da cuando el modelo es el adecuado y los coeficientes están bien ajustados.