Análisis de Series de Tiempo

Carrera de Especialización en Inteligencia Artificial

Clase 3

Ing. Magdalena Bouza, Ing. Carlos German Carreño Romano

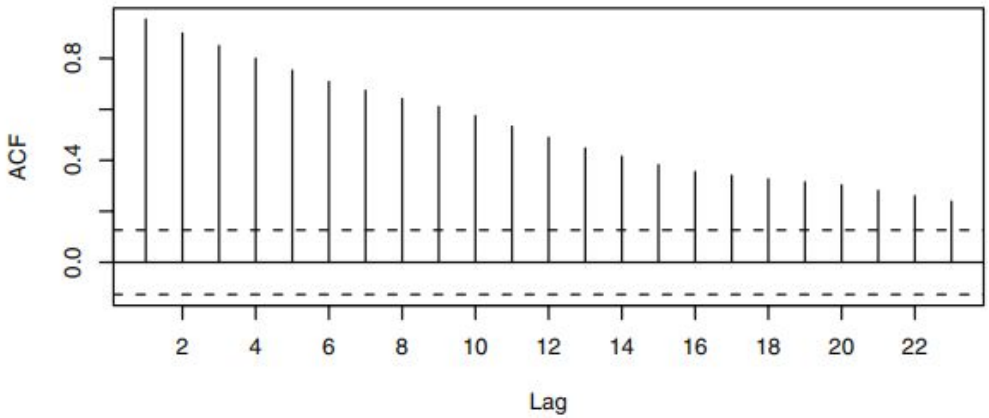
Agenda

1. Estacionariedad
2. Estacionalidad
3. Modelos ARIMA y SARIMA
4. Presentación del trabajo práctico
5. Criterios de bondad de modelos

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

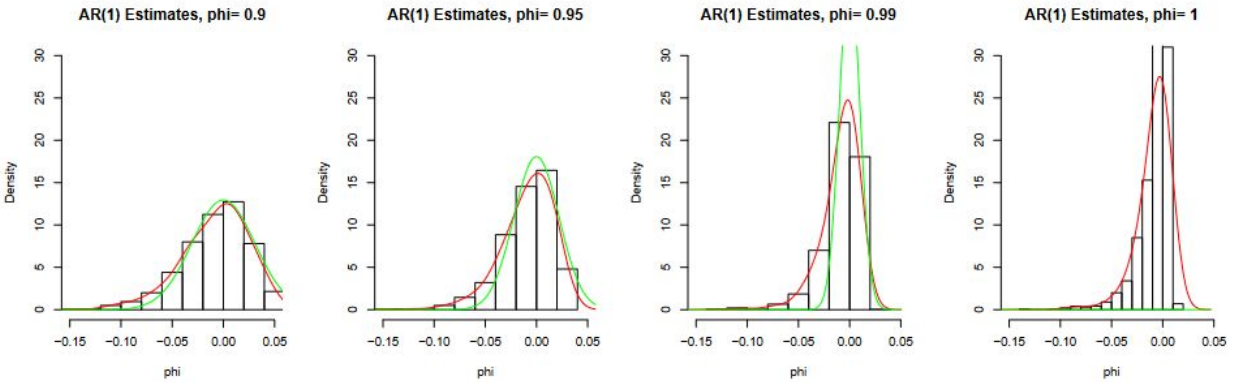
Test de Dickey-Fuller

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 $|a_1| < 1$, mientras que si $a_1 = 1$ el estimador tiene una distribución asintótica de la forma



Test de Dickey-Fuller

Consideremos el modelo $\{Y_t\}: \ Y_t = aY_{t-1} + W_t$, donde $\{X_t\}$ es un proceso estacionario. $\{Y_t\}$ será **no estacionario** si a =1.

Si miramos la serie diferenciada una vez, tenemos que

$$Y_t - Y_{t-1} = aY_{t-1} + W_t - Y_{t-1} = (a-1)Y_{t-1} + W_t$$

Dickey y Fuller proponen entonces el test

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

Su gran aporte fue hallar la distribución asintótica de $n(\widehat{a-1})$ bajo H0.

Luego estima por OLS $(a-1)$ y con eso lleva a cabo el test de hipótesis

Test de dickey-Fuller Aumentado

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

$$Y_t = aY_{t-1} + X_t \qquad X_t = \sum_{j=1}^{p} \rho_j X_{t-j} + w_t$$

Nuevamente, mirando la primera diferencia, y notando que bajo H0 $X_t = Y_t - Y_{t-1}$ tenemos que

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

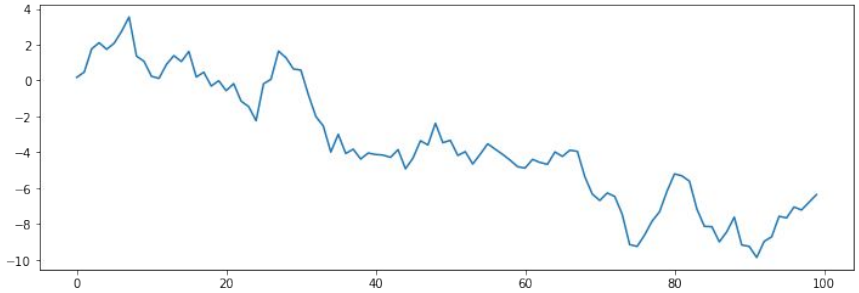
Nuevamente estimamos $a - 1$ por OLS y realizamos el test correspondiente.

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

Bibliografía extra para D-F

- Testing for unit roots
- Augmented Dickey-Fuller root tests

Cómo usar statsmodels?




```
from statsmodels.tsa.stattools import adfuller
adfuller(y)
```

✓ 0.9s

```
(-1.394757894883641,
 0.584786653359185,
 4,
 95,
 {'1%': -3.5011373281819504,
  '5%': -2.8924800524857854,
  '10%': -2.5832749307479226},
```

Estadístico

p-valor

lags usados

observaciones usadas

umbrales

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

Constant and trend order to include in regression.

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

Scripts/Dickey-Fuller_demo.ipynb

ARIMA

Modelo ARIMA

Diremos que *{Yt}* sigue un modelo ARIMA si $W_t = \nabla^d Y_t$ es un proceso ARMA(p,q) estacionario. Diremos en este caso que *Yt* sigue un proceso ARIMA(p,d,q).

Operador de *backshift*: el operador de backshift (B) opera sobre el índice temporal de la serie de tiempo y la desplaza en una unidad de tiempo:

$$BY_t = Y_{t-1}$$

El operador B es lineal: $B(aY_t + bX_t + c) = aY_{t-1} + bX_{t-1} + c$

En general escribimos que $B^d Y_t = Y_{t-d}$

ARIMA + backshift

MA(q):

$$\begin{aligned} Y_t &= e_t - b_1 e_{t-1} - \ldots - b_q e_{t-q} \\ &= e_t - b_1 B e_t - \ldots b_q B^q e_t \quad \Rightarrow Y_t = b(B)e_t \\ &= (1 - b_1 B - \ldots - b_q B^q) e_t \end{aligned}$$

ARMA(p,q)

$$a(B)Y_t = b(B)e_t$$

AR(p)

$$\begin{aligned} Y_t &= e_t + a_1 Y_{t-1} + \ldots + a_p Y_{t-p} \\ Y_t - a_1 Y_{t-1} - \ldots - a_p Y_{t-p} &= e_t \\ Y_t - a_1 B Y_t - \ldots - a_p B^p Y_t &= e_t \quad \Rightarrow a(B)Y_t = e_t \\ (1 - a_1 B - \ldots - a_p B^p) Y_t &= e_t \end{aligned}$$

ARIMA(p,d,q)

$$a(B)(1-B)^d Y_t = b(B)e_t$$

Ejemplo

https://github.com/charlieromano/TimeSeries/blob/main/Scripts/predicciones_AR_MA.py

Ejemplo con datos reales

```
inputfile = "../Datasets/TECO2.2000.2021.csv"


ts = pd.read_csv(inputfile, 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

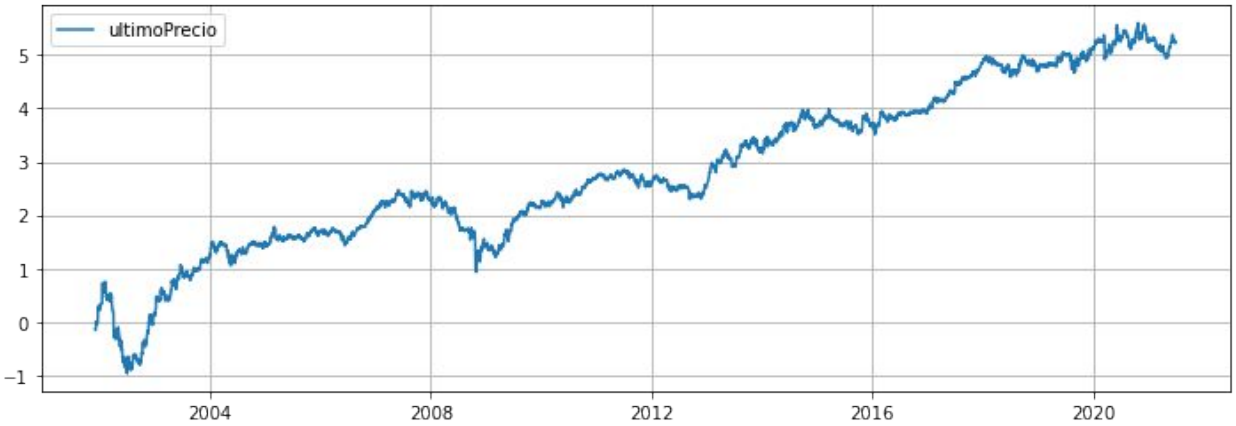
```
adfuller(ts.ultimoPrecio)
(0.57890231841091,
 0.9870838971395222,
 32,
 4807,
 {'1%': -3.431711097447145,
  '5%': -2.862141452575749,
  '10%': -2.5670901548483767},
```

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},
```

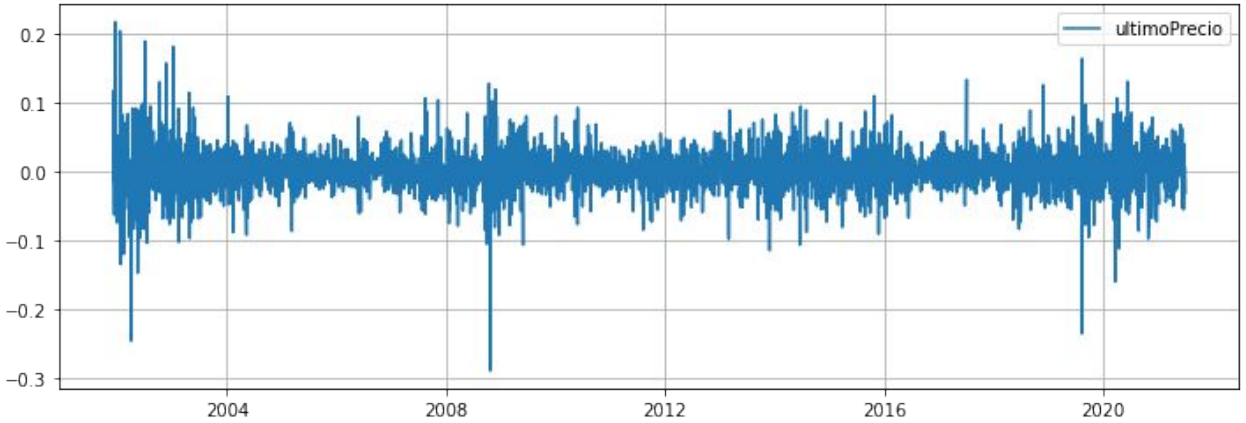


Ejemplo con datos reales

```
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)
```



Estacionalidad

Modelo SARMA multiplicativo

Cuando el proceso es estacionario podemos definir un modelo SARMA multiplicativo **ARMA(p,q)x(P,Q)s**, con período estacional **s** como un modelo AR con polinomio característico $a(x)\alpha(x)$ más un MA con polinomio característico $b(x)\beta(x)$. Con

$$
\begin{aligned}
a(x) &= 1 - a_1 x - a_2 x^2 - \ldots - a_p x^p \\
\alpha(x) &= 1 - \alpha_1 x^s - \alpha_2 x^{2s} - \ldots - \alpha_P x^{Ps}, \\
b(x) &= 1 - b_1 x - b_2 x^2 - \ldots b_q x^q \\
\beta(x) &= 1 - \beta_1 x^s - \beta_2 x^{2s} - \ldots - \beta_P x^{Ps},
\end{aligned}
$$

Modelo SARIMA multiplicativo

Una herramienta importante es el análisis de procesos estacionales **no estacionarios** es la diferenciación estacional de período s para la serie {Yt}, denotada

$$\nabla_s Y_t = Y_t - Y_{t-s}$$

Se dice que una serie estacional no estacionaria sigue un modelo SARIMA(p,d,q)x(P,D,Q)s de período s si la serie diferenciada

$$W_t = \nabla^d \nabla_s^D Y_t$$

sigue un proceso SARMA(p,q)x(P,Q)s.

Trabajo práctico

Trabajo Práctico

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

2: Descomponer una serie de tiempo usando el modelo aditivo y el modelo multiplicativo.

3: Aplicar los modelos vistos en clase: (entrega clase 5, re-entrega clase 6)

- para la tendencia usar cuadrados mínimos y expresar los coeficientes. Sacar conclusiones acerca de la validez del modelo
- componente cíclica: usar análisis espectral y hallar las frecuencias principales*
- para la componente estacional usar ARIMA
- para la componente de error obtener R_k, C_k

3: Predicciones: **(entrega clase 8)**

- realizar predicciones usando (S)ARIMA
- realizar predicciones usando redes neuronales LSTM
- extraer conclusiones

Criterios de Bondad de modelos

Criterios de bondad de modelo

Uno de los objetivos cuando analizamos series de tiempo es poder modelarlas. Esto incluye tanto elegir la familia de modelos correcta como hallar todos los parámetros de la misma.

Es necesario entonces poder contar con algún criterio que nos permita saber cuán bueno es nuestro modelo, es decir cuán cerca se encuentra la distribución especificada por el modelo a la distribución verdadera de los datos.

Divergencia de Kullback-Leibler (DK-L)

Supongamos que los datos se encuentran generados por un proceso **desconocido** $g(y)$, y que consideramos un modelo $f(y)$ para aproximarnos a $g(y)$.

Si conociéramos $f(y)$, podríamos utilizar la divergencia de Kullback-Leibler

$$D_{\mathrm{KL}}(g \parallel f) = \int_{-\infty}^{\infty} g(y) \log\left(\frac{g(y)}{f(y)}\right) dy = \mathbb{E}\left[\log\left(\frac{g(Y)}{f(Y)}\right)\right]$$

que nos da una medida de la pérdida de información que tenemos al aproximar $f(y)$ por $g(y)$. Cuanto más pequeño sea el valor de la divergencia K-L, más cerca estarán $g(y)$ y $f(y)$. El mejor modelo será entonces aquel minimice la DK-L.

Estimación de la divergencia K-L

Dado que en general no se conoce el valor verdadero de la distribución g(y), debemos estimar el valor de $D_{\mathrm{KL}}(g \parallel f)$, por lo cual debemos estimar su valor a partir de las muestras $y_1, \ldots, y_n$. Para ello, se asume que las muestras son observadas de forma independiente de g(y).

En primer lugar, vemos que podemos reescribir

$$D_{\mathrm{KL}}(g \parallel f) = \mathbb{E}\left[\log\left(\frac{g(Y)}{f(Y)}\right)\right] = \mathbb{E}[\log(g(Y))] - \mathbb{E}[\log(f(Y))]$$

El primer término no puede ser calculado, pero si lo que nos interesa es comparar modelos, podemos descartarlo pues será una constante común a todos.

Estimación de la divergencia K-L

El término $\mathbb{E}[\log(f(Y))] = \int \log(f(y))g(y)dy$ y tampoco se puede calcular. Sin embargo la ley de los grandes números garantiza que

$$\frac{1}{n}\sum_{i=1}^{n} \log(f(y_i)) \to \mathbb{E}[\log(f(Y))]$$

Recordemos que $\ell = \sum_{i=1}^{n} \log(f(y_i)) = \log(L)$ es el logaritmo de la verosimilitud (*log-likelihood*), donde L es la función de verosimilitud.

Obs: Minimizar$D_{\mathrm{KL}}(g \parallel f)$ equivale a maximizar la log-verosimiliud. Por lo tanto una forma natural de hallar los mejores parámetros para un modelo dado es utilizando el estimador de máxima verosimilitud.

Criterio de Información de Akaike (AIC)

El modelo de información de Akaike sirve para comparar la bondad de dos modelos propuestos, y se basa en los resultados analizados previamente.

$$\mathrm{AIC} = 2k - 2\ell(\hat{\theta})$$

cantidad de parámetros parámetros estimados por MV

Se elegirá entonces el modelo que alcance el menor AIC.

Observaciones:

- El resultado de AIC es asintótico pues se basa en la LGN
- AIC no nos dice cuán bueno es cada modelo sino cuál es el mejor de todos (podría ser que sean todos malos)

Criterio de información Bayesiano

Similar a la definición de AIC, el criterio de información Bayesiano también se basa en la log-verosimilitud, pero penaliza más una mayor cantidad de parámetros.

$$\mathrm{BIC} = k\log(n) - 2\ell(\hat{\theta})$$

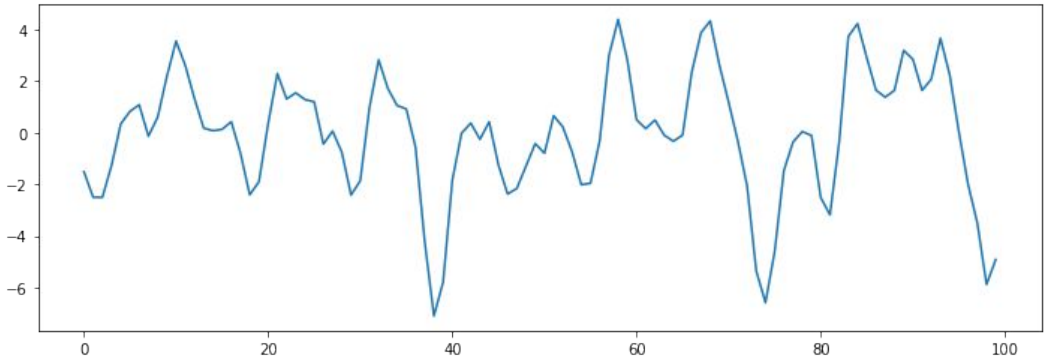
Nuevamente, se preferirán los modelos con BIC más bajo.

Si bien la expresión es similar a AIC, la derivación de BIC es totalmente distinta. AIC aparece como consecuencia de un enfoque Bayesiano, donde lo que se busca es el modelo f que maximice $\mathbb{P}(f|y_1, \ldots, y_n)$.

Para una explicación más detallada ver “Elements of Statistical Learning”, Trevor Hastie, Robert Tibshirani, Jerome Friedman.

Ejemplo

```
ar_coef  = np.array([0.8,-0.2])
ma_coef = np.array([0.9,0.5,-.3])
arma_ts = arma_generate_sample(
ar=np.r_[1,-ar_coef], ma=np.r_[1,ma_coef], nsample =100)
plt.figure(figsize=(8,3))
plt.plot(arma_ts)
```



compare_model.ipynb

Ejemplo

```
model_good = ARIMA(arma_ts, order=(2, 3, 0))
model_good_res = model_good.fit()
print(model_good_res.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  100
Model:                 ARIMA(2, 3, 0)   Log Likelihood                -206.926
Date:                Fri, 13 May 2022   AIC                            419.851
Time:                        23:31:29   BIC                            427.575
Sample:                             0   HQIC                           422.974
                                  - 100
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.3483      0.117     -2.977      0.003      -0.578      -0.119
ar.L2         -0.1755      0.115     -1.531      0.126      -0.400       0.049
sigma2         4.1663      0.783      5.318      0.000       2.631       5.702
===================================================================================
Ljung-Box (L1) (Q):                   1.11   Jarque-Bera (JB):                 2.72
Prob(Q):                              0.29   Prob(JB):                         0.26
Heteroskedasticity (H):               1.00   Skew:                             0.14
Prob(H) (two-sided):                  1.00   Kurtosis:                         2.23
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step)
```

```
model_bad = ARIMA(arma_ts, order=(5, 5, 0))
model_bad_res = model_bad.fit()
print(model_bad_res.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  100
Model:                 ARIMA(5, 5, 0)   Log Likelihood                -234.720
Date:                Fri, 13 May 2022   AIC                            481.439
Time:                        23:31:32   BIC                            496.762
Sample:                             0   HQIC                           487.631
                                  - 100
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.4774      0.105    -14.112      0.000      -1.683      -1.272
ar.L2         -1.4495      0.172     -8.426      0.000      -1.787      -1.112
ar.L3         -1.3962      0.182     -7.674      0.000      -1.753      -1.040
ar.L4         -0.8958      0.173     -5.189      0.000      -1.234      -0.557
ar.L5         -0.3412      0.104     -3.283      0.001      -0.545      -0.138
sigma2         7.8979      1.543      5.119      0.000       4.874      10.921
===================================================================================
Ljung-Box (L1) (Q):                   3.41   Jarque-Bera (JB):                 3.88
Prob(Q):                              0.06   Prob(JB):                         0.14
Heteroskedasticity (H):               0.99   Skew:                             0.27
Prob(H) (two-sided):                  0.98   Kurtosis:                         2.17
===================================================================================
```

¿preguntas?