



APRENDE CON ELI

SERIES TEMPORALES

Series Temporales

TABLA DE CONTENIDO

Primeras nociones	2
Serie estacionaria	3
Serie no estacionaria	4
Prueba Dickey - Fuller de estacionariedad	5
Estacionalidad.....	6
Autocorrelación	8
Ruido Blanco.....	8
ACF.....	9
PACF.....	10
Modelos.....	11
¿Cómo seleccionar el modelo más apropiado?.....	11
Modelo Autorregresivo AR	12
¿Cómo saber si debemos incluir más retrasos y cuántos?	13
Modelo de Medias Móviles MA	13
¿Cómo saber si debemos incluir más retrasos y cuántos?	14
Modelo Autorregresivo de Medias Móviles ARMA	14
Modelo Autorregresivo Integrado de Medias Móviles ARIMA	14
Modelos con componente estacional SARMA y SARIMA.....	15
Modelos MAX	15
Modelos VAR	16
Modelos ARCH y GARCH.....	16
Auto arima.....	18
Librerías Python.....	21
Librerías R.....	22

Vamos a comenzar con la definición de *serie temporal*. Una serie de tiempo es una secuencia de información que pone en correspondencia un período de tiempo a cada valor.

El valor puede ser prácticamente cualquier cosa medible que dependa del tiempo de alguna manera. Como por ejemplo precios, humedad, o número de personas.

No existen limitaciones con respecto al tiempo total de nuestras series de tiempo. Todo lo que necesitamos es un punto de partida y un punto final.

Entre esos dos puntos, habrá muchos otros puntos. El intervalo de tiempo entre un punto en el tiempo y el siguiente, se llama *período de tiempo*.

Con qué periodicidad los valores del conjunto de datos se registran, se conoce como la *frecuencia* del conjunto de datos. Para poder analizar series temporales de una manera significativa, todos los períodos de tiempo deben ser iguales y claramente definidos, lo que daría como resultado una frecuencia *constante*. Esta frecuencia es una medida del tiempo y puede variar desde unos pocos milisegundos hasta varias décadas. Sin embargo, veremos los más usuales son cada minuto, cada hora, cada día, cada semana, cada mes, cada cuatrimestre, cada año, etc.

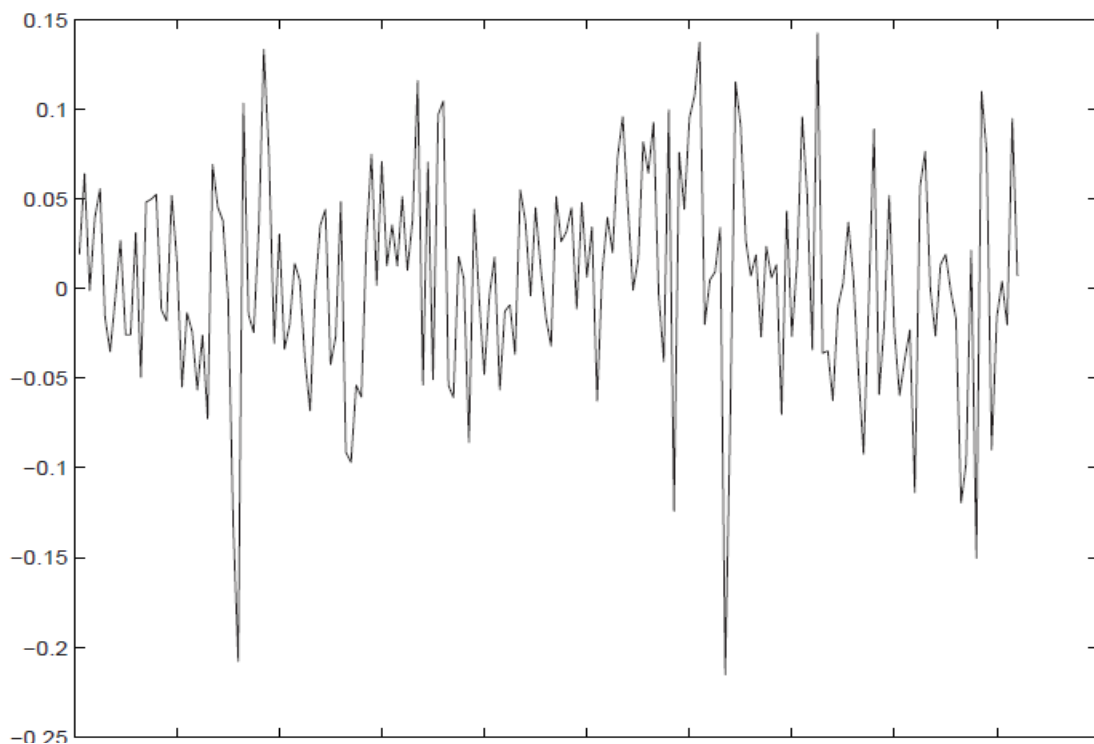
Además, se espera que los patrones que se observen en las series de tiempo persistan o continúen existiendo en el futuro.

Un ejemplo muy usual son los datos **meteorológicos** que suelen hacer frente a la tarea de pronosticar el clima durante varios días hacia el futuro. Para hacer predicciones precisas de manera consistente, hay que analizar datos pasados. Pero claro, si los datos no se ordenan cronológicamente, encontrar el patrón correcto sería extremadamente difícil. Por ejemplo, simplemente conocer la temperatura más alta de los últimos cinco días sería inútil, a menos que sepamos qué valor de temperatura corresponde a cada día.

En el mundo de los **negocios** los datos de series de tiempo se aplican muchísimo, en análisis de finanzas para los inversores y los propietarios de

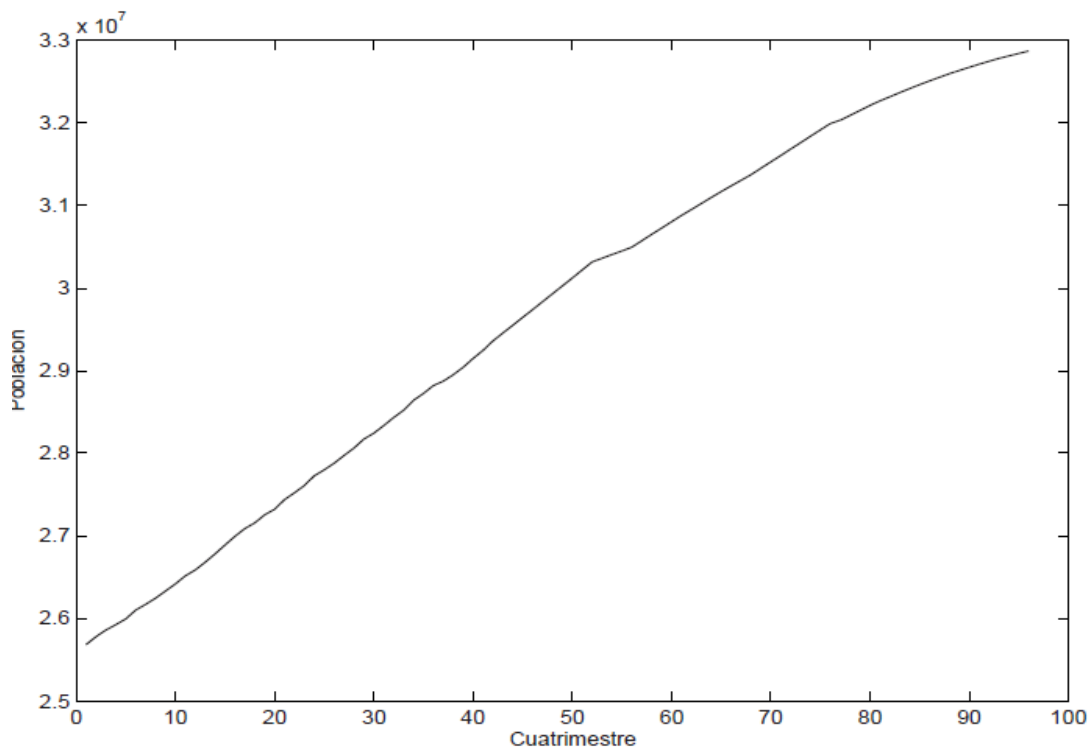
las empresas. Es crucial determinar si los precios arrojan ganancias y las ventas aumentarán o disminuirán en el futuro. Por lo tanto, un tema común en el análisis de series de tiempo es determinar la estabilidad o la volatilidad de los mercados financieros.

SERIE ESTACIONARIA



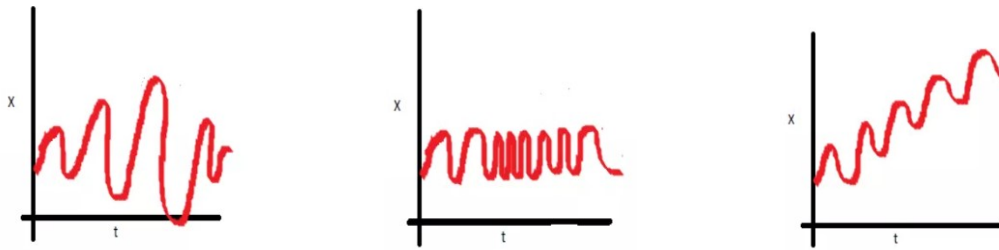
La figura es un ejemplo de una serie estacionaria. Da el rendimiento mensual promedio de la Bolsa de Madrid durante el período de 1988 a 2000 medido por el Índice General. Observe que los valores de la serie parecen moverse alrededor de un rendimiento mensual fijo. No hay tendencias crecientes o decrecientes con el tiempo.

SERIE NO ESTACIONARIA



Sin embargo, este otro ejemplo muestra una serie que, a diferencia de la anterior, no es claramente estable en el tiempo, y es lo que llamamos no estacionario. La serie corresponde a la población española mayor de 16 años, al final de cada trimestre durante el período de 1977 a 2000. Observe en el gráfico que la serie no es estable, ya que su nivel aumenta con el tiempo. Decimos que la serie tiene una tendencia clara y positiva. La mayoría de las series económicas y sociales no son estacionarias (estables) y muestran tendencias. En este caso, la tendencia es aproximadamente lineal, aunque la pendiente de la línea que encajamos en la primera mitad de la muestra sería algo mayor que la que obtendríamos en el segundo. Esto sugiere que quizás el crecimiento anual de la población está cambiando con el tiempo. La tendencia de la serie sería entonces variable en el tiempo en lugar de constante. Esta propiedad es típica en series reales en la práctica, donde es poco probable que observemos una tendencia constante durante largos períodos de observación.

La no estacionariedad puede ser de diferentes formas:



Una serie es estacionaria cuando es estable a lo largo del tiempo, es decir, cuando la media y la varianza son constantes en el tiempo y, además, no presenta tendencia. Esto se refleja gráficamente en que los valores de la serie tienden a oscilar alrededor de una media constante y la variabilidad con respecto a esa media también permanece constante en el tiempo.

Las series NO estacionarias son series en las cuales la tendencia y/o variabilidad cambian en el tiempo. Los cambios en la media determinan una tendencia a crecer o decrecer a largo plazo, por lo que no oscilaría alrededor de un valor constante.

PRUEBA DICKEY - FULLER DE ESTACIONARIEDAD

Para llevar a cabo un análisis adecuado de series de tiempo es vital determinar si los datos siguen un proceso estacionario o no estacionario. Por suerte para nosotros, en el siglo XX, los estadísticos David Dickey y Wayne Fuller desarrollaron una prueba para ayudarnos. El método para verificar si un conjunto de datos proviene de un proceso estacionario, se conoce como la prueba Dickey-Fuller o prueba de DF en abreviatura.

La prueba está basada en un contraste de hipótesis. Vamos a tener como hipótesis nula que la serie es No Estacionaria. Con lo cual la alternativa es que sí lo es.

Cuando hacemos el contraste de hipótesis, siempre tenemos que calcular el valor de un estadístico de prueba, que dependerá de los valores de la muestra y de nuestra hipótesis nula.

El valor del estadístico se compara con un valor crítico en la tabla Dickey Fuller. Si el estadístico de prueba o de contraste es inferior al valor crítico rechazamos la hipótesis nula. Lo que significa que entonces los datos provienen de un proceso estacionario.

Otra manera de llegar a estas conclusiones es a través del p-valor. El p-valor se puede interpretar como el valor que representa cuán probable es la hipótesis nula. Si es un valor muy cercano a cero significa que esa hipótesis es poco probable y habría que rechazarla.

¿Cómo sabemos si es un valor muy bajo, pero suficiente para rechazar la Hipótesis nula?

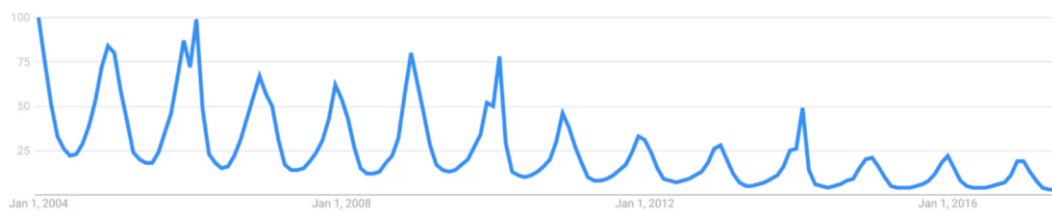
Porque siempre que hagamos algo en inferencia vamos a tener un nivel de confianza, por ejemplo, el 95%, y lo que le falta a eso para llegar al 100% es nuestro nivel de significación, es decir $5\%=0.05$. Entonces compararíamos el p-valor con el nivel de significación. Si es más pequeño diremos que es suficientemente bajo para rechazar la Hipótesis nula.

En Python lo haremos con el método *sts.adfuller* del paquete *stattools*.

En R lo haremos con la función *adf.test* del paquete *tseries*.

ESTACIONALIDAD

La estacionalidad sugiere que ciertas tendencias aparecerán de forma cíclica. Por ejemplo, las temperaturas suben y bajan según las horas del día y los meses del año. Estos son dos ejemplos distintos de patrones estacionales que observamos en nuestra vida, el cambio durante el día y el cambio durante un año completo.



Google Trends - "Snowboarding"

Hay varias formas de comprobar si existe estacionalidad.

Un enfoque es descomponer la secuencia de la serie temporal en tres efectos.

1. Tendencia
2. Estacional
3. Residual

Por supuesto, la tendencia representa el patrón consistente en todos los datos. El efecto estacional expresa todos los efectos cíclicos debido a la estacionalidad. Finalmente, los residuos son el error de predicción, o la diferencia entre los datos reales y el modelo que ajustamos.

El tipo más simple de descomposición es la descomposición clásica. Con ella esperamos una relación lineal entre las tres partes y la serie de tiempo observada. Hay dos enfoques principales en la descomposición clásica:

1. Aditivo
2. Multiplicativo

El aditivo supone que, para cualquier período de tiempo, el valor observado es la suma de la tendencia, más el efecto estacional, más el residual, para ese período. Del mismo modo, la descomposición multiplicativa supone que la serie original es un producto de los tres efectos.

El Python, el paquete de modelos de estadística incluye un método llamado *seasonal_decompose* que significa descomposición estacional, que coge una serie de tiempo y la divide en esos tres efectos.

En R usaremos la función *decompose* del paquete *stats*.

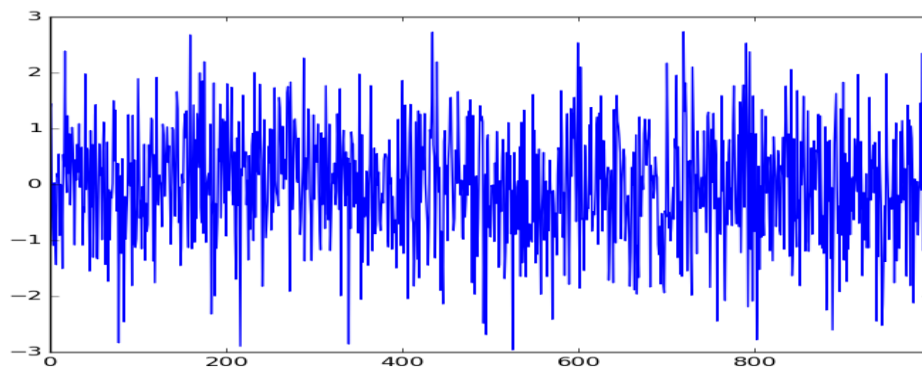
AUTOCORRELACIÓN

Como su nombre lo indica, la autocorrelación representa la correlación entre una secuencia y sí misma. Más precisamente, mide el nivel de semejanza entre una secuencia de hace varios períodos y los datos reales.

La secuencia de hace varios períodos atrás se llama “retraso” o “lag” en inglés, porque es una versión retrasada de la original. Por ejemplo, si calculamos la autocorrelación para una serie de tiempo con frecuencia diaria, estamos determinando cuántos de los valores de ayer se parecen a los valores de hoy. Si la frecuencia es, en cambio, anual, entonces la autocorrelación medirá las similitudes de año en año.

RUIDO BLANCO

El ruido blanco es un tipo especial de serie temporal donde los datos no siguen un patrón. Como en el ruido blanco no encontramos ningún patrón, no se puede predecir en el futuro. Es decir, el ruido blanco no es predecible.



Además, para que una serie se pueda considerar como ruido blanco, necesita satisfacer las siguientes tres condiciones:

1. Media constante

2. Varianza constante
3. No tener autocorrelaciones significativas en ningún período.

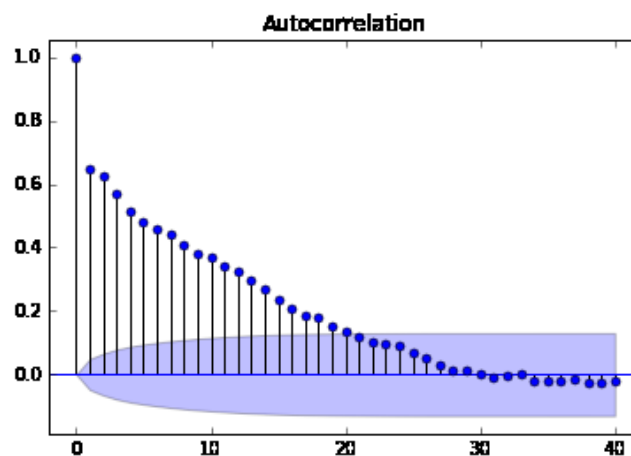
ACF

En series temporales es vital poder calcular y comparar los valores de autocorrelación entre diferentes retrasos. Para ello, necesitamos introducir la función de autocorrelación o ACF (autocorrelation function).

La función de autocorrelación proporciona la autocorrelación para cualquier retraso que consideremos. La alternativa sería encontrar manualmente la correlación entre nuestros datos originales y múltiples retrasos de sí misma.

En Python usaremos la función *plot_acf* y en R la función *acf*.

Un ejemplo sería:



En el gráfico, los valores en el eje x representan los retrasos, que en este caso van hasta 40 ya que eso es lo que se estableció.

Los números a la izquierda en el eje vertical representan los posibles valores para el coeficiente de autocorrelación. La correlación solo puede tomar valores entre -1 y 1.

Las líneas en el gráfico representan la autocorrelación entre la serie de tiempo y una versión retrasada de sí misma. La primera línea indica la autocorrelación con retraso de 1 período de tiempo, la segunda línea

representa la autocorrelación para hacer dos períodos y así sucesivamente. La primera línea es uno porque la correlación entre un valor y sí mismo siempre será uno.

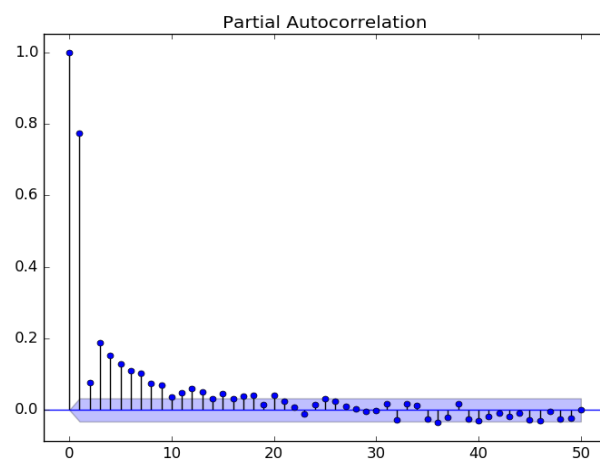
El área azul alrededor del eje x representa la significación de los valores de autocorrelación. Es decir, mide si son significativamente distintos de cero, lo que sugiere la existencia de autocorrelación para ese retraso específico.

Si los coeficientes son significativos, es un indicador de la dependencia del tiempo en los datos.

La función ACF se utilizará para investigar cuántos retrasos es necesario en el modelo de medias móviles. Y para investigar si los residuos son ruido blanco, es decir, si las correlaciones son no significativas para los residuos.

PACF

La autocorrelación mide la similitud entre una serie temporal y una versión anterior de sí misma. Sin embargo, los coeficientes también capturan efectos de los momentos anteriores de manera indirecta. Por indirecta nos referimos a todos los demás canales a través de los cuales los datos del pasado afectan a los datos actuales. Si deseamos determinar solo la relación directa entre la serie de tiempo y su versión retrasada, necesitamos calcular la autocorrelación parcial



La función PACF se utilizará para investigar cuántos retrasos es necesario en el modelo autorregresivo.

En Python usaremos la función *plot_pacf* y en R la función *pacf*.

MODELOS

Después de un análisis descriptivo y exploratorio de la serie temporal, el objetivo siguiente, va a ser predecir el futuro en base a un modelo de series temporales. El procedimiento general es el siguiente:

1. Escoger el modelo
2. Dividir los datos en datos de entrenamiento y datos de prueba.
3. Ajustar el modelo a los datos de entrenamiento.
4. Evaluar el modelo en los datos de prueba.
5. Re-ajustar el modelo con todos los datos
6. Predecir los datos futuros.

¿CÓMO SELECCIONAR EL MODELO MÁS APROPIADO?

Usualmente comenzamos con un modelo simple y luego lo expandimos siempre que se cumplan varias condiciones. Obviamente podemos complejizar el modelo introduciendo más retrasos, pero este modelo más complejo tendrá más coeficientes para estimar. Cuando hagamos esto, puede ser que el modelo no sea apropiado, por eso una de las cosas que debemos comprobar es que los nuevos coeficientes sean significativamente diferentes de cero.

Si eso no es cierto, las estimaciones para estos parámetros son esencialmente cero. Por lo tanto, no nos ayudan a estimar valores futuros. Esto significa que no tienen poder predictivo, por lo que debemos desecharlo y volver al modelo más simple.

Una vez que tengamos varios posibles modelos, debemos compararlos entre sí. Para determinar si las predicciones de un modelo son significativamente mejores que las de otro, utilizaremos la inferencia. Donde tendremos una prueba estadística llamada prueba de razón de log-verosimilitudes o log-likelihood test. Sin embargo, la prueba solo se puede aplicar a modelos con diferentes grados de libertad (número de parámetros que estima).

Por lo tanto, al comparar varios modelos con el mismo número total de parámetros estimados, vamos a usar los criterios de información de cada uno. Los dos criterios de información más usados son el AIC y el BIC

AIC: Akaike Information Criteria

BIC: Bayesian Information Criteria

El mejor modelo será el que tenga menor AIC o BIC.

Además se mirará el valor de la **log-verosimilitud** (LL: log likelihood) de ambos modelos en el resumen estadístico. El mejor modelo será el que tenga mayor LL.

Por otro lado, si nuestro modelo se ajusta bien, no debería haber un efecto de tendencia que no hayamos tenido en cuenta, esto se traduce en que los residuos del modelo deben parecerse al ruido blanco, por lo que podemos concluir que no hay otro patrón por ahí escondido. Esto debe comprobarse a través de la prueba DF de estacionariedad, que debe dar positiva a favor de los residuos siendo estacionarios. Y a través de la función de autocorrelación de los residuos, en la que se debería obtener que no hay correlaciones significativas.

MODELO AUTORREGRESIVO AR

Los modelos autorregresivos predicen el valor actual de nuestra serie temporal en función de los valores pasados. Un modelo autorregresivo de

orden simple, de orden uno, sólo consideraría el valor del período anterior, y el modelo se expresaría de la siguiente manera:

$$x_t = c + \varphi x_{t-1} + \epsilon_t$$

Donde x_t es el valor de interés, del período actual, de nuestra serie de tiempo, c es la constante, φ es el coeficiente que debemos estimar, ϵ_t es el residuo en el período actual, y x_{t-1} es el valor de la serie en el período anterior.

No funcionan bien si los datos no son estacionarios.

¿CÓMO SABER SI DEBEMOS INCLUIR MÁS RETRASOS Y CUÁNTOS?

Sabemos que la ACF captura los efectos directos e indirectos del valor anterior sobre el valor presente. Como queremos un modelo eficiente solo queremos considerar aquellos retrasos que tengan un efecto directo y significativo sobre el período presente. Por lo tanto, debemos examinar la PACF antes de construir un modelo con demasiados coeficientes de retraso.

MODELO DE MEDIAS MÓVILES MA

Los modelos de medias móviles predicen el valor actual de nuestra serie temporal en función de los residuos pasados. Un modelo de medias móviles de orden simple, de orden uno, sólo consideraría el valor del residuo en el período anterior, y el modelo se expresaría de la siguiente manera:

$$x_t = c + \theta \epsilon_{t-1} + \epsilon_t$$

Donde x_t es el valor de interés, del período actual, de nuestra serie de tiempo, c es la constante, θ es el coeficiente que debemos estimar, ϵ_t es el residuo en el período actual, y ϵ_{t-1} es el valor del residuo en el período anterior.

No funcionan bien si los datos no son estacionarios.

¿CÓMO SABER SI DEBEMOS INCLUIR MÁS RETRASOS Y CUÁNTOS?

En este caso, vamos a mirar la ACF.

MODELO AUTORREGRESIVO DE MEDIAS MÓVILES ARMA

Es la combinación de ambos modelos AR y MA, con lo cual un modelo ARMA tendría dos órdenes (p,q), donde p es el orden de la parte autorregresiva y q es el orden de la parte de medias móviles. Un modelo ARMA de orden simple, un ARMA(1,1) sería:

$$x_t = c + \varphi x_{t-1} + \theta \epsilon_{t-1} + \epsilon_t$$

Los modelos ARMA funcionan mejor si la serie es estacionaria.

MODELO AUTORREGRESIVO INTEGRADO DE MEDIAS MÓVILES ARIMA

Es un modelo ARMA aplicado al resultado de integrar la serie temporal un número de veces determinado, por el orden de integración. Es decir, un modelo ARIMA de orden (p,d,q) consiste en integrar la serie original d veces, y luego ajustar un modelo ARMA(p,q) a esa serie integrada. El objetivo de la integración es obtener una serie estacionaria, ya que los modelos ARMA funcionan peor para series no estacionarias.

En este modelo no podemos mirar la ACF ni la PACF para más o menos saber el orden. Entonces habría que hacer lo siguiente:

1. Empezar con el ARIMA(1,1,1)
2. Examinar los residuos
3. Complejizar el modelo según lo que hayamos visto en los residuos
4. Ir bajando el número de componentes hasta el modelo simple.

Cuidado cuando integramos la serie, porque por cada integración, perdemos una observación.

Vamos a recordar que la estacionalidad ocurre cuando ciertos patrones aparecen periódicamente, por ejemplo, algo que se repite cada año. Por lo tanto, en este caso, un modelo simple autorregresivo AR no describirá bien los datos. ¿Por qué? Porque sólo considera el valor del período anterior para estimar el periodo actual.

¿Entonces qué haríamos en este caso? Necesitaríamos incluir los valores registrados durante el período de hace un año.

Entonces primero necesitamos definir el orden estacional que es ese ciclo que sospechamos que se repite. Por ejemplo, si nuestros datos son diarios, y el ciclo se repite cada semana, un buen orden sería 7. Si nuestros datos son mensuales y observamos una repetición anual, un buen orden sería 12.

Tendremos una variante estacional para todos los modelos que hemos estudiado: SARMA, SARIMA. Y, por ejemplo, un SARIMA tendría orden $(p,d,q)(P,D,Q,s)$, donde p,d,q son los órdenes no estacionales del modelo y P,D,Q son los órdenes estacionales, que se refieren a los retrasos que se añaden respecto al orden estacional s .

Ejemplo de $SARIMA(1,0,2)(2,0,1,5)$:

$$x_t = c + \varphi_1 x_{t-1} + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \Phi_1 (x_{t-5} + \varphi_1 x_{t-6}) \\ + \Phi_2 (x_{t-10} + \varphi_1 x_{t-11}) + \Theta_1 (\epsilon_{t-5} + \theta_1 \epsilon_{t-6} + \theta_2 \epsilon_{t-7}) + \epsilon_t$$

Los modelos MAX son modelos que consideran información exógena para explicar la variable endógena (la serie que estamos estudiando). Vamos a

tener versiones MAX de todos los modelos estudiados: ARMAX, ARIMAX, SARMAX y SARIMAX.

Ejemplo, modelo ARMAX de orden simple, donde Y es la variable exógena:

$$x_t = c + \beta Y + \varphi_1 x_{t-1} + \theta_1 \epsilon_{t-1} + \epsilon_t$$

La variable exógena puede ser una serie o varias series (varias variables).

Por lo general mejorar mucho al modelo, pero tienen la desventaja de que para predecir datos de futuro, como en el futuro no conocemos los valores de esas variables exógenas, no podríamos usar esto, a no ser que estimemos de alguna forma los valores de esas variables extra. Esto tiene que ver con otros modelos que se llaman VAR.

MODELOS VAR

Es un modelo multivariante al que le pasamos varias variables para estimar cada serie en dependencia de las demás. Por ejemplo para estimar los precios de mercado, cada uno, en dependencia del otro conjunto de precios.

MODELOS ARCH Y GARCH

Son modelos para estimar la **volatilidad** (variabilidad y cambios) en una serie temporal. Cuando los inversores están interesados en hacer grandes compras a largo plazo, a menudo buscan estabilidad. Es por eso por lo que surge la necesidad de modelos especiales que midan la volatilidad. Para empezar, cuando hablamos de volatilidad, estamos hablando de la magnitud de los residuos. No nos importa si son positivos o negativo. Solo nos importa cuán volátiles sean nuestras predicciones. Dado que buscamos estabilidad, eso explica por qué la volatilidad es sinónimo de variación y por qué queremos para evitarlo una varianza más baja, lo que indicaría mayor

estabilidad que a su vez sugiere un riesgo menor. Por lo tanto, medir la volatilidad es clave cuando queremos seguridad en nuestras inversiones.

Una solución es propuesta por los modelos autorregresivos con heterocedasticidad condicional, que en inglés se diría *Autoregressive Conditional Heteroscedasticity Model*, o modelo **ARCH**.

Lo primero, es que, a diferencia de la familia ARIMA, los ARCH consisten en varias ecuaciones, una para la media y otra para la varianza.

La ecuación para la varianza de un ARCH(q), que mide la varianza condicional en función de residuos pasados, es:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2 + \dots + \alpha_q \epsilon_{t-q}^2$$

Y la ecuación para la media junto con la que estima el valor actual de la serie es:

$$x_t = c_0 + \varphi_1 \mu_{t-1} + \epsilon_t$$

Para este método se investiga la PACF.

Una posible mejora de los modelos ARCH es incluir también valores pasados, además de los residuos al cuadrado. Pero ¿cuáles serían exactamente estos valores pasados? Pensemos, incluir valores pasados de los retornos en este caso no tendría mucho sentido ya que generalmente contamos con ellos en la ecuación media. Si queremos trazar una idea paralela a lo que hacíamos con el modelo ARMA, debemos incluir algunos componentes autorregresivos, y como estamos trabajando con varianzas, vamos a incluir esas variaciones condicionales pasadas para ayudarnos a explicar las variaciones condicionales actuales. Ya que los períodos de alta volatilidad son seguidos por períodos de alta volatilidad y los períodos de baja volatilidad son seguidos por tales con baja volatilidad, eso significa que agregar variaciones condicionales pasadas debería servir como un gran punto de referencia para las estimaciones del presente. Una varianza condicional pasada ya incluiría en sí misma los efectos de los residuos al

cuadrado anteriores. Esto haría que agregar más residuos al cuadrado anteriores sea redundante.

Es por eso por lo que el siguiente modelo se llama *Generalized Autorregresive Conditional Heteroscedasticity Model* o simplemente **GARCH**. El modelo GARCH es simplemente una extensión del modelo ARCH, donde agregamos varianzas condicionales pasadas a la ecuación de varianza. La intuición detrás de esto proviene que la volatilidad pasada nos da información patrones consecutivos, por lo tanto, si incluimos la varianza condicional del último período tendremos una especie de punto de referencia sobre qué debemos esperar.

Un GARCH(1,1) tendría como ecuación de la varianza:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

Donde β_1 es el coeficiente del nuevo término que hemos incluido explicando la varianza condicional del período anterior.

Cuando se trata de la variación de precios del mercado, ningún modelo GARCH de orden superior supera al modelo GARCH(1,1) simple.

AUTO ARIMA

Es el método automático para estimar el mejor modelo que se ajusta a la serie temporal. Por defecto tiene parámetros fijados, como por ejemplo el orden máximo que puede tomar el modelo. En la práctica, se recomienda hacer un análisis exploratorio antes para ver si hay un orden máximo superior que deberíamos considerar, superior al que viene por defecto en la función.

Pros:

1. Ahorra tiempo
2. Elimina ambigüedades
3. Reduce el riesgo de error humano

Contras:

1. Poner ciegamente la fe en un solo modelo
2. No se ve realmente si los demás modelos funcionan mejor o no
3. Requiere cierta experiencia en el tema
4. Error humano

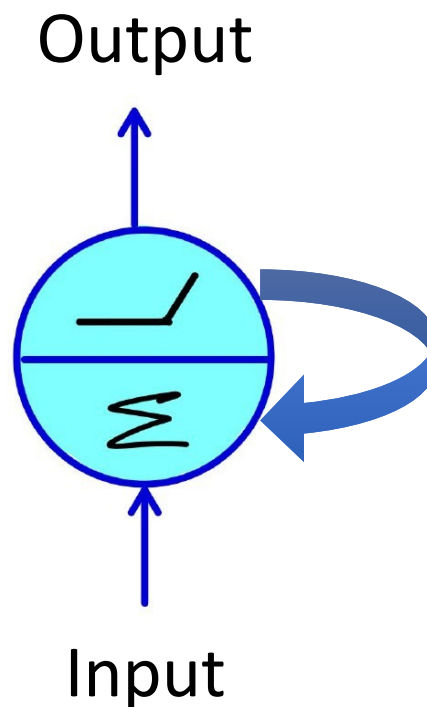
Python: [auto.arima](#) del paquete *pmdarima*.

R: [auto.arima](#) del paquete *forecast*.

REDES NEURONALES

Las redes neuronales son un mecanismo que nos permite obtener también modelos predictivos bajo un procedimiento similar a los métodos que hemos visto antes: entrenar el modelo, obtenerlo y luego usarlo para predecir el futuro. Sin embargo, son más difíciles de interpretar que los modelos ARIMA.

Los modelos basados en ARIMA son mucho más fáciles de entender porque realmente se puede ver los valores de los coeficientes y cuántos retrasos se necesitan, etc. Es decir, vamos a poder ver si fueron necesarios 12 meses de información o solo tres meses de información histórica. En las redes neuronales, ese tipo de información está técnicamente oculta en algún lugar de la red, no podemos verla directamente porque tiene tantas neuronas y tantos pesos que en realidad no podemos determinar qué peso va a ser relevante para qué parámetro o característica en particular en la serie de tiempo. Es por eso que yo recomendaría que siempre que tengamos un nuevo problema de series temporales comencemos primero con los modelos basados en ARIMA, que a menudo funcionan mejor que las redes neuronales y la otra gran ventaja es que podemos interpretarlos con mayor facilidad. Las redes neuronales son solo cajas negras donde lo que ves es lo que obtienes, así que mejor usarlos como un último recurso, ya que realmente no se puede interpretar tan bien los resultados.



Las redes neuronales recurrentes están específicamente diseñadas para trabajar con datos de secuencia, por ejemplo, series de tiempo.

La novedad en este tipo de redes neuronales es que tiene ciclos, envía el output como input para sí misma.

Si desenrollamos esto a lo largo del tiempo, una neurona recurrente particular tendría una entrada en el tiempo $t-1$, y daría una salida en $t-1$. Pero esta salida luego pasaría como entrada de la neurona en el tiempo t , donde también tendría su entrada en ese tiempo t , y una salida en ese tiempo t que luego podemos tomar como entrada en el tiempo $t + 1$ y así sucesivamente. A esto se le llama desenrollar esa neurona recurrente. Algo que es importante tener en cuenta aquí es que la neurona en realidad está recibiendo tanto entradas de una marca de tiempo anterior como entradas del tiempo actual.

REDES LSTM

Un problema común que enfrentan las redes neuronales recurrentes es que después de un tiempo, especialmente si estamos entrenando la red en una secuencia muy grande o larga, la red neuronal recurrente comenzará a olvidar las primeras entradas. Las últimas entradas van a comenzar a sobrescribir los pesos de las entradas iniciales. Por lo tanto, queremos asegurarnos de que no estamos comenzando a olvidar esas primeras entradas y que no perdemos esa información. Es decir, realmente lo que necesitamos es algún tipo de memoria a largo plazo para nuestras redes.

Es decir, necesitamos un equilibrio, un balance, entre la memoria a corto plazo de las redes, los datos con los que se entrenó recientemente, y la memoria a largo plazo de las redes, es decir, los primeros datos con los que se entrenó.

Las LSTM (Long Short Term Memory) se crean para ayudar a abordar estos problemas en las redes neuronales recurrentes. La librería de Keras, en Python, nos permite modelar redes neuronales recurrentes para pronosticar una secuencia de series de tiempo en el futuro, para los períodos que queramos. En R podemos hacer uso de la librería nnet.

LIBRERÍAS PYTHON

1. [ARMA](#)
2. [ARIMA](#)
3. [SARIMAX](#)
4. [ARCH](#)
5. [Auto arima](#)
6. [Prophet](#) de Facebook.
7. [Yfinance](#)
8. [Keras](#)

1. Paquete [lubridate](#), que no es específico de series temporales sino más bien de pre-procesado de datos. Lo usaremos para formatear la fecha y el tiempo.
2. Paquete [tseries](#), que también nos permitirá trabajar con series temporales, modelar y graficar. También lo usaremos para testear por estacionariedad.
3. Paquete [forecast](#) que tiene muchas funciones, como la función Auto Arima que es un método de modelado automático. También hay un montón de herramientas para limpiar el dataset de datos faltantes, de atípicos o outliers, tiene muchas herramientas gráficas para modelar series de tiempo y permite modelar los pronósticos hacia el futuro.
4. Paquete [zoo](#), nos permite trabajar con series temporales, definirlas, trabajar más fácilmente con diferentes frecuencias o fechas no igualmente espaciadas (series de tiempo irregulares).
5. Paquete [nnet](#) nos permite crear modelos basados en redes neuronales en R.