Machine Learning – Time Series

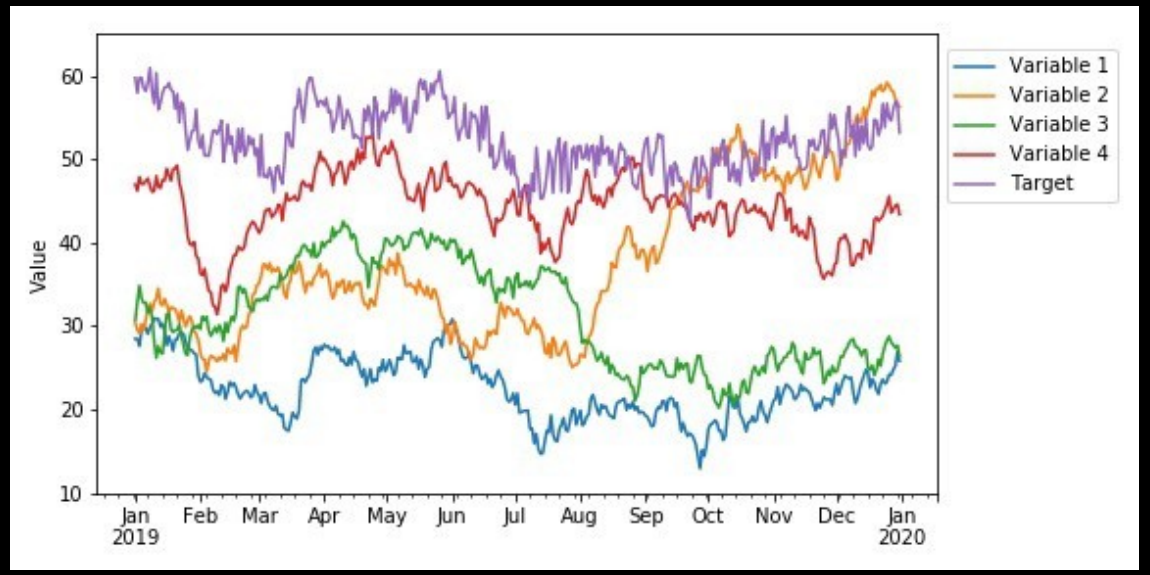
Time Series
Analítica

¿Qué es un Time Series?

Conjunto de observaciones ordenadas en el tiempo,
a través de intervalos de tiempo regulares.

El intervalo de tiempo podría ser:

- Anual (presupuesto anual)
- Trimestral (gastos)
- Mensual (tráfico aéreo)
- Semanal (ventas semanales)
- Diario (tiempo)
- Horario (precio stocks)
- Minutos (entradas en un call center)
- Segundos (Tráfico web)



Algunos ejemplos

a) Precio acciones Google

b) Diferencia diaria de precio acciones de Google

c) N° Strikes anuales en EEUU

d) Ventas mensuales de casas unifamiliares en EEUU

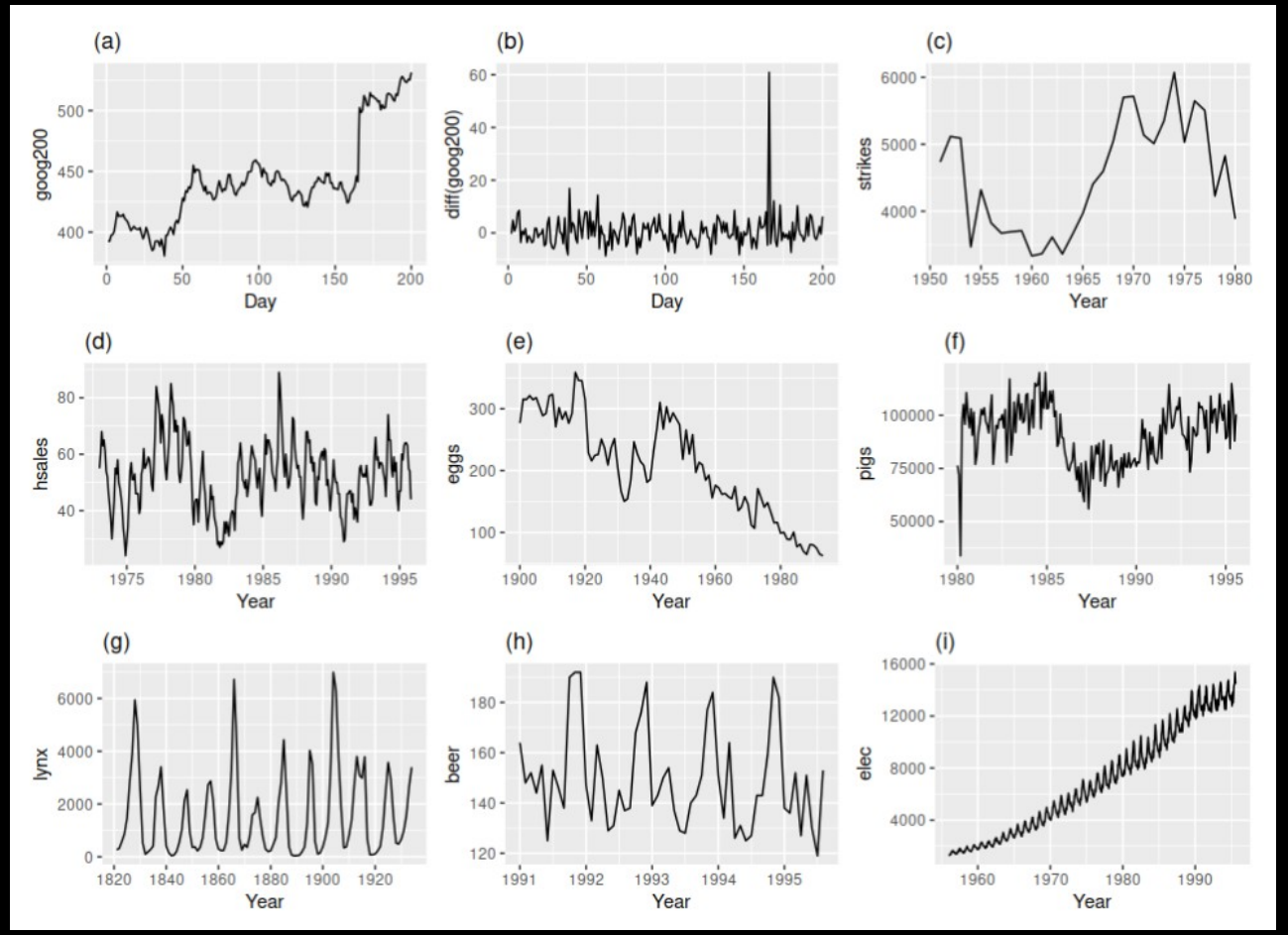
e) Precio anual de la docena de huevos en EEUU

f) Cerdos sacrificados mensualmente en Victoria, Australia

g)Total de linces atrapados en el río McKenzie (Canadá)

h) Producción cervezera mensual en Australia

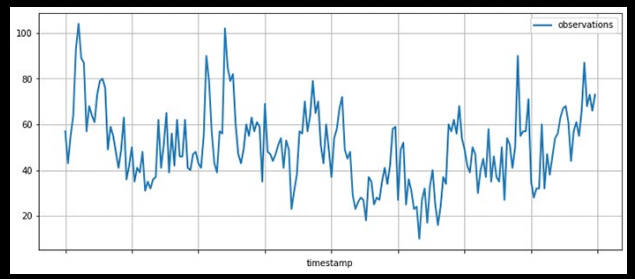
i) Producción eléctrica mensual en Australia.



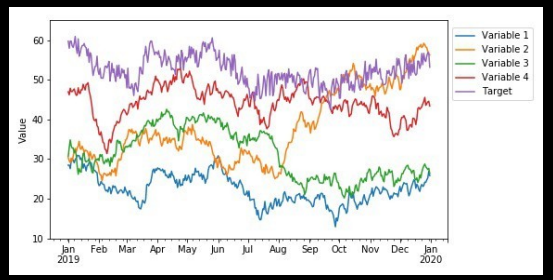
Clasificación Series Temporales

Cantidad de variables

Univariante

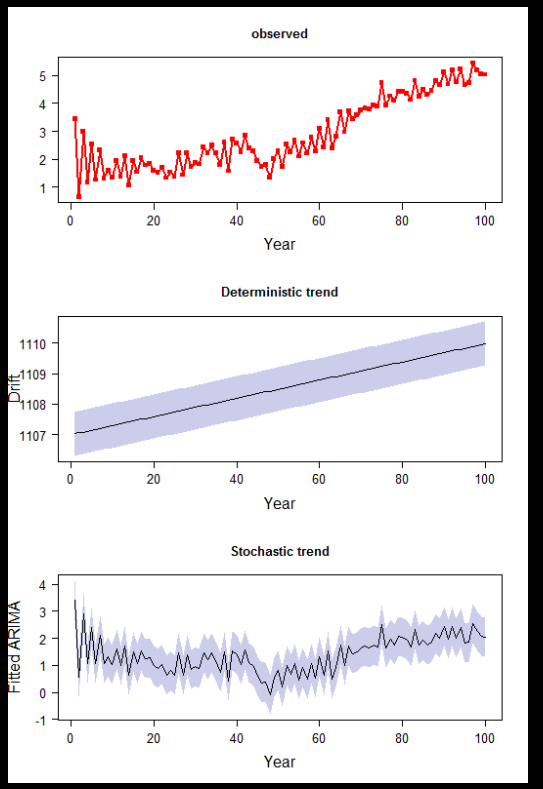


Multivariante



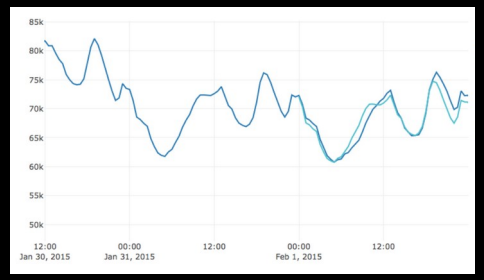
Tipo de Predicción

Determinística vs Estocástica

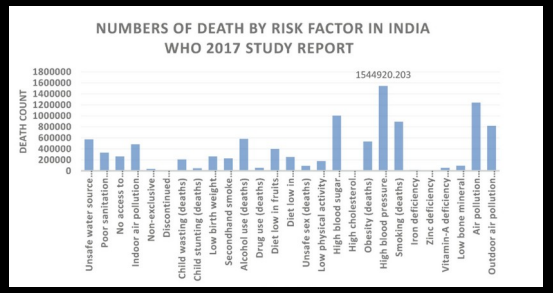


Tipo de Dato

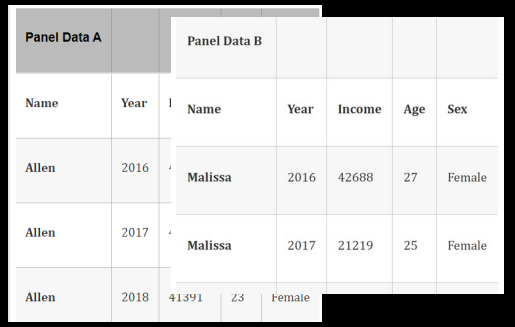
Time Series



Cross-Section



Panel



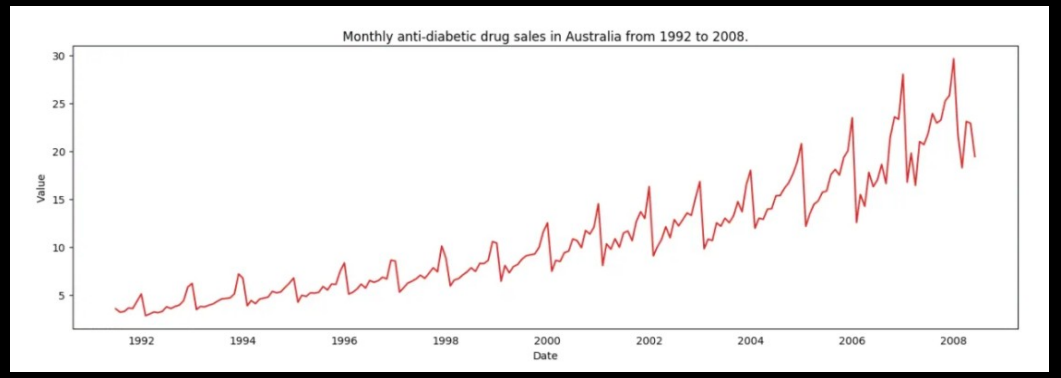
Panel Data A					
Name	Year				
Allen	2016				
Allen	2017				
Allen	2018	41391	23	Female	

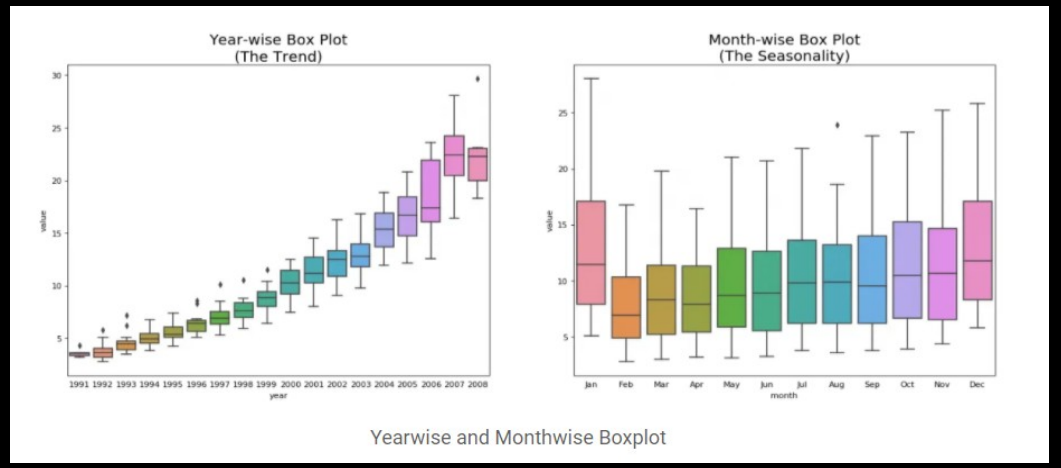
Panel Data B				
Name	Year	Income	Age	Sex
Malissa	2016	42688	27	Female
Malissa	2017	21219	25	Female

Objetivo de Time Series

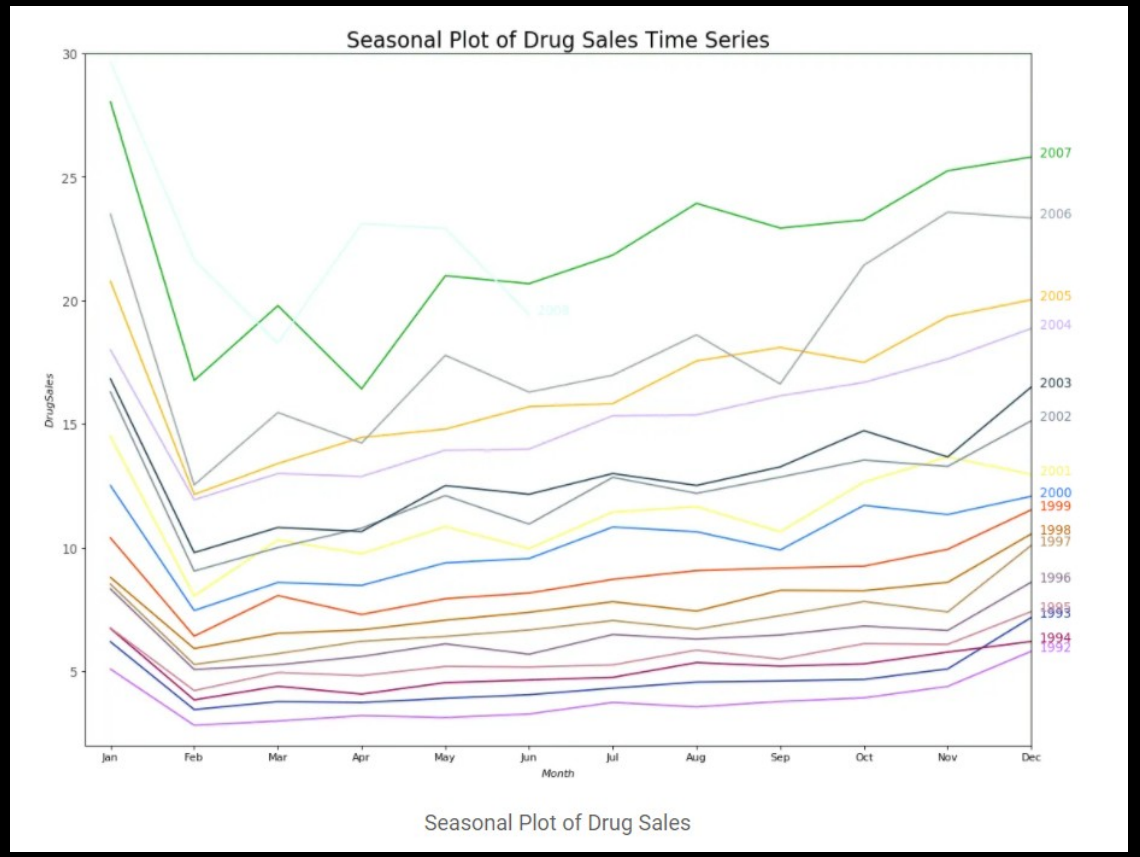
Entender el pasado mediante análisis prescriptivo para predecir el futuro

Visualización Time Series





Yearwise and Monthwise Boxplot



Seasonal Plot of Drug Sales

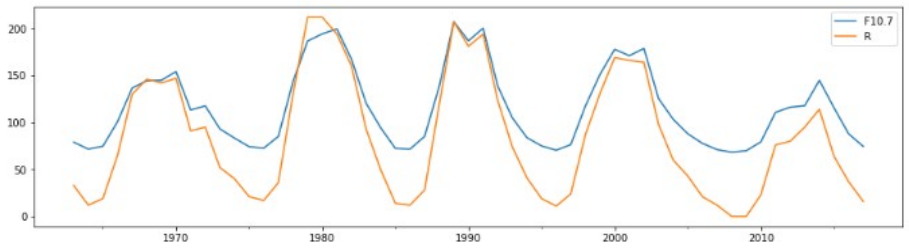
Visualización Time Series

Si tenemos muchas irregularidades en la serie temporal, podemos aplicar varias técnicas para la mejora de la visualización

Resample

```
df[["F10.7", "R"]].resample("1y").median().plot(figsize=(15,4))
```

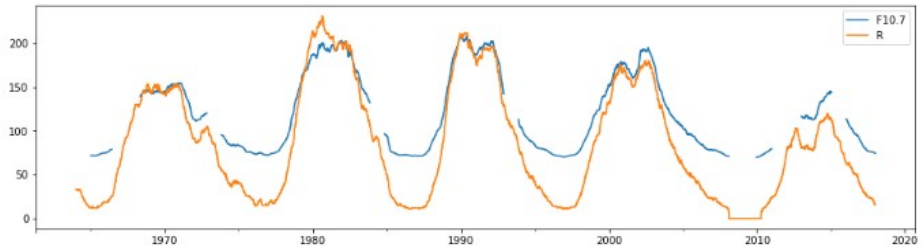
Copy contents



Rolling

```
df[["F10.7", "R"]].rolling(24*365).median().plot(figsize=(15,4))
```

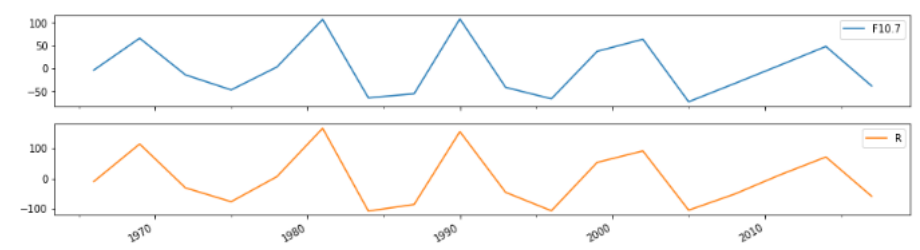
Copy contents



Differenciating

```
df[["F10.7", "R"]].resample("3y").median().diff().plot(subplots=True, figsi
ze=(15,4))
```

Copy contents



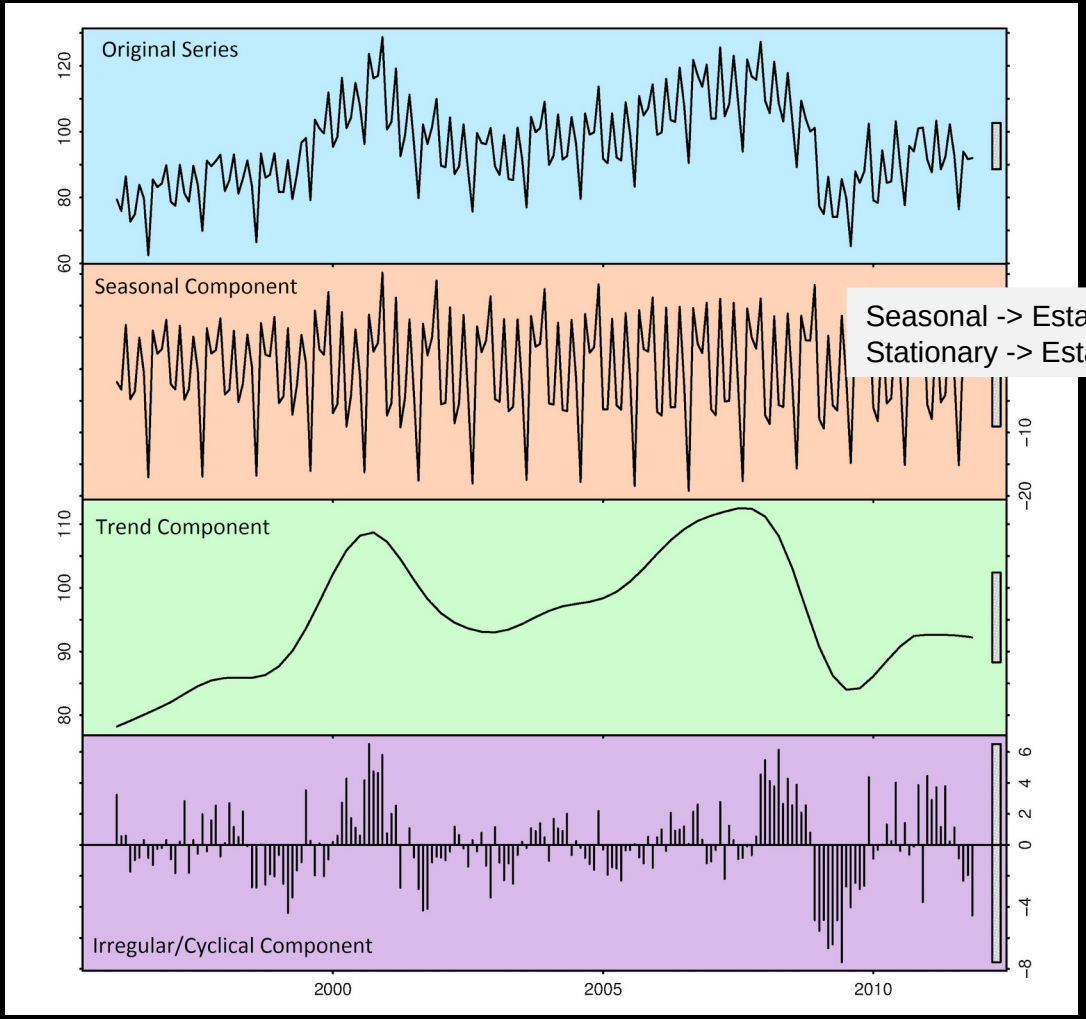
Componentes de un Time Series

Tendency: cambio a largo plazo en relación al nivel medio, o el cambio a largo plazo de la media.

Seasonality: las series presentan cierta periodicidad (mensual, anual…). Por ejemplo, el paro suele aumentar en invierno, y disminuir en verano.

Random/Residuals: si eliminamos los dos componentes anteriores, nos queda la componente aleatoria. Se pretende estudiar qué tipo de comportamiento aleatorio presentan estos residuos.

Cyclic variations: fluctuaciones producidas a lo largo de una larga tendencia. Es poco frecuente. Por ejemplo ciclos económicos de la bolsa



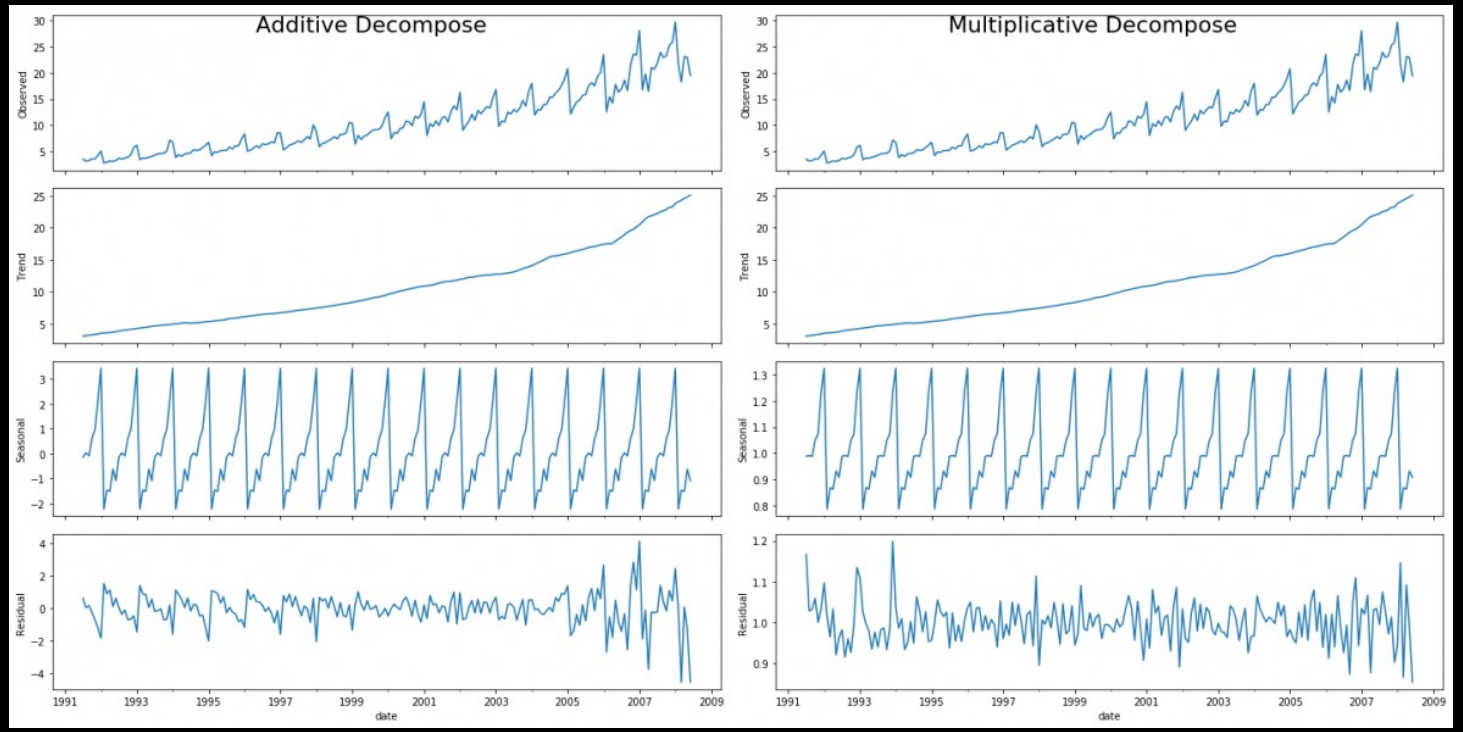
Seasonal -> Estacional
Stationary -> Estacionario

Additive vs Multiplicative Decompose

Trend: T(t)
Seasonality: S(t)
Cyclic variation: c(t)
Residuals: e(t)

Additive decompose
Y(t) = T(t) + S(t) + c(t) + e(t)

Multiplicative decompose
Y(t) = T(t) * S(t) * c(t) * e(t)

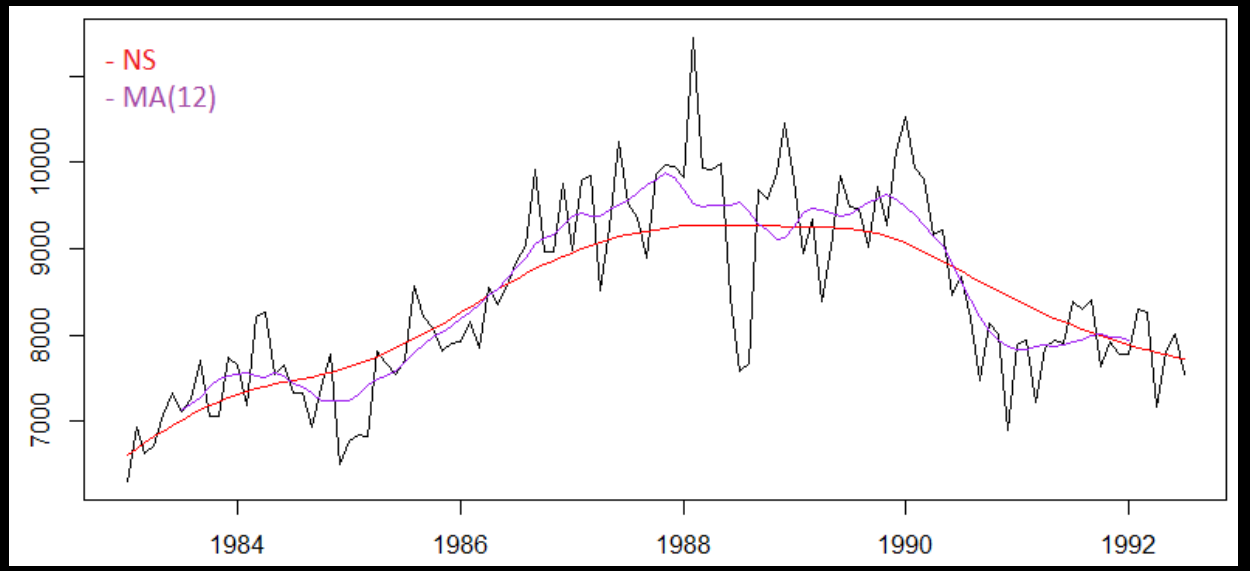


Smoothing

Smoothing

Método estadístico con el que creamos una función de aproximación para eliminar irregularidades en los datos, con el objetivo de obtener patrones significativos.

Se utiliza en modelos de predicción a corto plazo, para datos que no tienen una variación grande a lo largo del tiempo.



Modelos de smoothing

Simple Exponential Smoothing (SES)

Se emplea sobre datos que **no tienen** una tendencia o estacionalidad muy clara. Se compone de un parámetro llamado alpha (0-1), que determina cuánto suaviza la serie temporal.

$$L_t = \alpha Y_t + (1-\alpha)\, L_{t-1}$$

$$L_t = \alpha Y_t + (1-\alpha)\,[(\alpha Y_{t-1} + (1-\alpha) L_{t-2}] =$$

$$= \alpha Y_t + \alpha(1-\alpha)\, Y_{t-1} + (1-\alpha)^2\, L_{t-2} = \ldots$$

$$= \alpha Y_t + \alpha(1-\alpha)\, Y_{t-1} + \alpha\,(1-\alpha)^2\, Y_{t-2} + \ldots$$

Siendo L el nivel calculado por SES, e Y el de la propia serie temporal.
Cuanto mayor es alpha, más se asemeja a la serie original.

α	α (1- α)	α (1- α)^2	α (1- α)^3
0.9	0.089	0.0089	0.00089
0.5	0.25	0.125	0.0625
0.1	0.09	0.081	0.0729

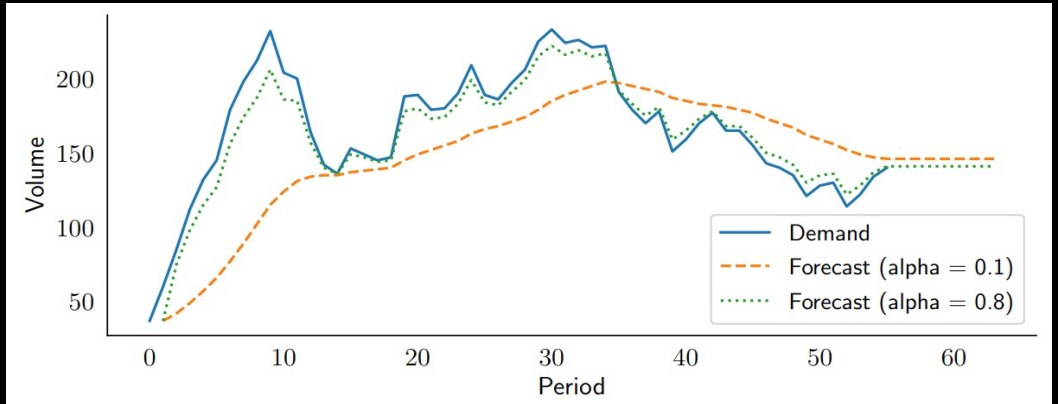


Figure 3.2: Simple smoothing

Double Exponential Smoothing (DES)

Simple Exponential Smoothing no funciona muy bien con tendencias en los Time Series. Double Exponential Smoothing maneja mejor datos **con tendencia** y sin seasonality, en comparación con otros métodos. Ahora, además de depender de alpha, dependerá también de beta.

$$L_t = \alpha Y_t + (1-\alpha)(L_{t-1} + T_{t-1})$$

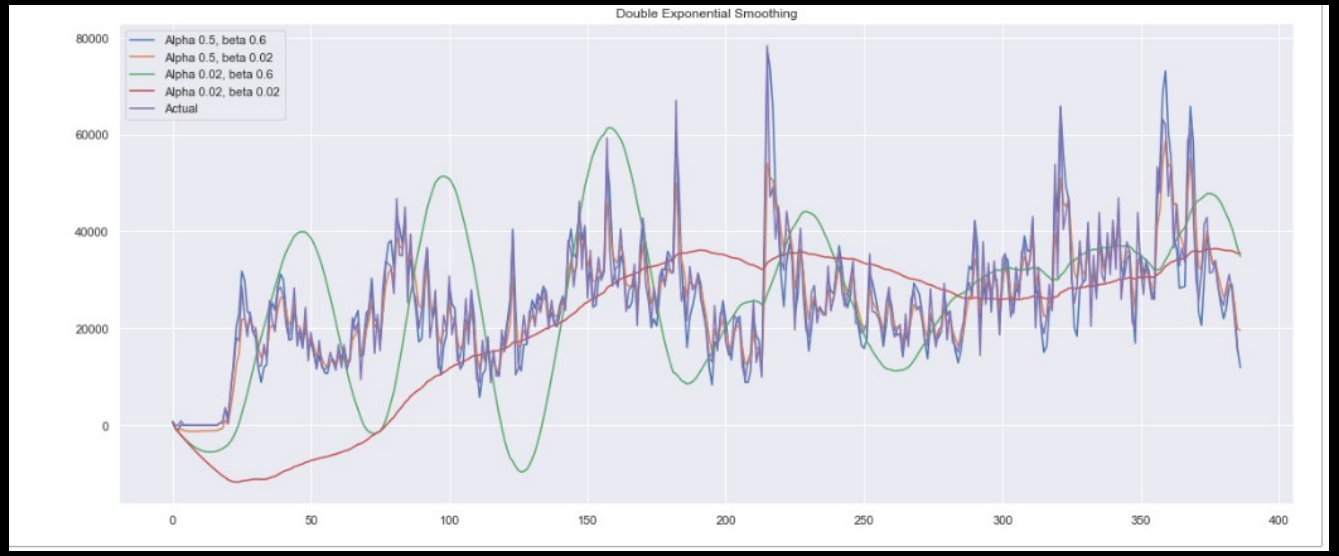
$$F_{t+k} = L_t + KT_t$$

$$T_t = \beta (L_t - L_{t-1}) + (1-\beta) T_{t-1}$$

K es el numero de puntos a predecir

En nuestro código Alpha será *smoothing_level*
Mientras que Beta será *smoothing_slope.*

Cuánto más cercano a 1, menos suavizado.



Triple Exponential Smoothing (TES)

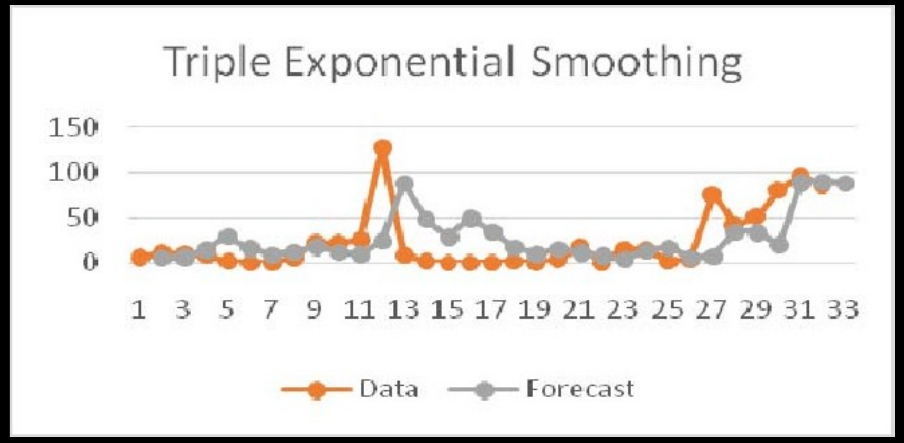
Solo nos falta un componente de los Time Series que no hemos podido modelar bien, y es seasonality. Triple Exponential Smoothing incluye un parámetro para poder modelar la periodicidad de los datos a lo largo del tiempo. Se emplea en series **con ambas** tendencia y estacionalidad clara (seasonality)

$$F_{t+k} = l_t + kT_t + S_{t+k-M}$$

Level: $Lt = \alpha (y_t/S_{t-M}) + (1- \alpha) (L_{t-1} + T_{t-1})$

Trend: $T_t = \beta (L_t - L_{t-1}) + (1- \beta) T_{t-1}$

Seasonality: $St = \gamma (Y_t/L_t) + (1+ \gamma) S_{t-M}$

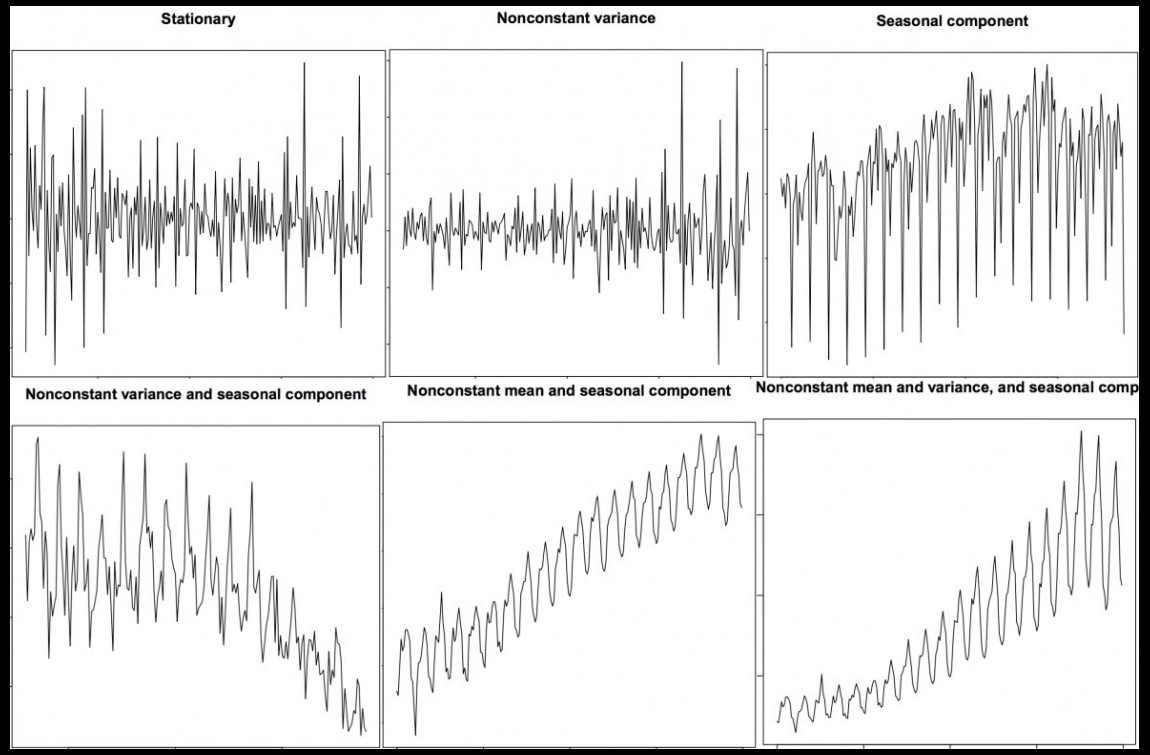


Stationary & Correlation

Stationary

Una serie es stationary (estacionaria) cuando sus propiedades estadísticas no varían con el tiempo.

Una serie NO es estacionaria cuando su tendencia cambia con el tiempo, su varianza, o tiene seasonality (patrones periódicos).



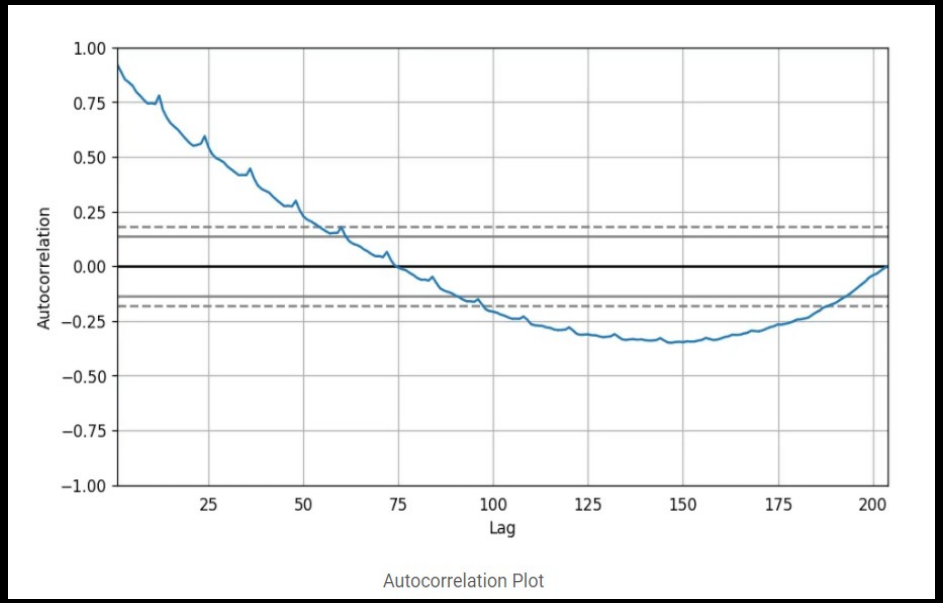
¿Por qué es importante este concepto?
La mayoría de modelos trabajan con series estacionarias. Hay que convertirlas para poder modelar los datos. Tendrán una media, varianza y autocorrelación constantes y podremos aplicar la mayoría de conceptos estadísticos.

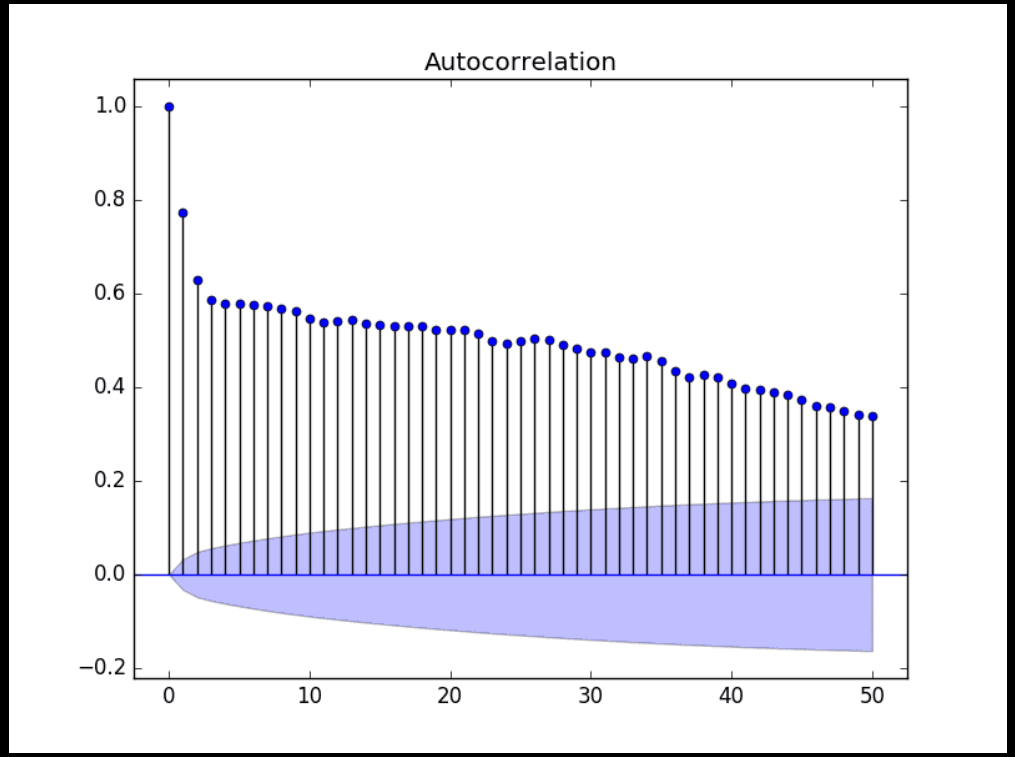
Las técnicas autoregresivas son modelos lineales que dependen de instantes anteriores del Time Series. Necesitamos que las features sean independientes y no estén correlacionadas unas con otras. Por eso quitamos la seasonality, para que no exista correlación alta. Las features tienen que ser totalmente independientes.

Autocorrelation plot

- Gráfica que mide la correlación de un instante vs sus instantes anteriores.
- La autocorrelación va de -1 a 1
- Es habitual que cada instante tenga correlación alta con los mas próximos y baja con los mas lejanos

· Si es positiva > AR
· Si es negativa > MA
· Nula (aleatoria) > ARIMA





Test de hipotesis

Una hipótesis es una pregunta que acepta si/no como respuesta.

- H0: hipótesis nula
- H1: hipótesis alternativa

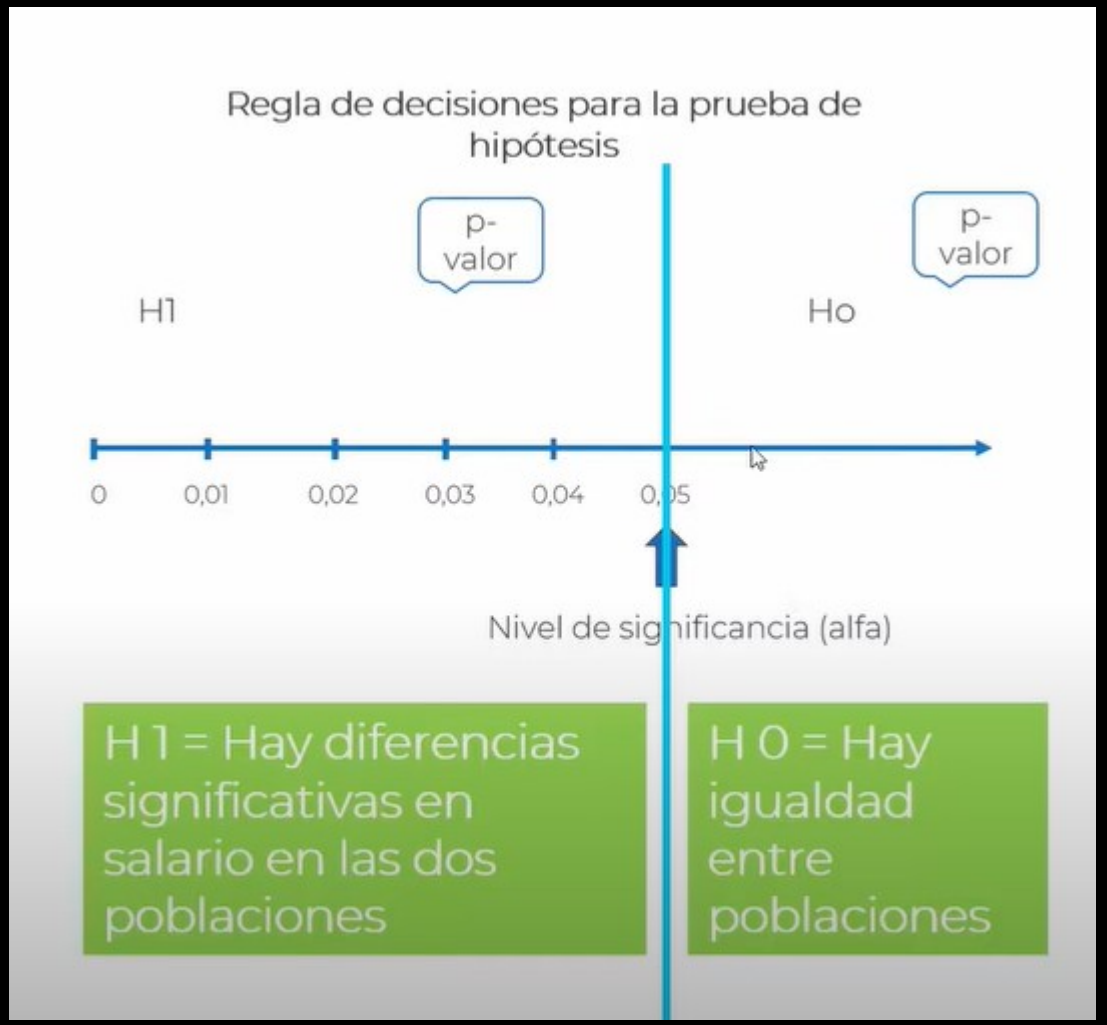
¿El salario medio de España es igual que el europeo?

- H0: El salario medio España = Salario medio europeo
- H1: El salario medio España <> Salario medio europeo

Los valores p evalúan qué tan bien los datos de la muestra apoyan el argumento de que la hipótesis nula es verdadera.

Valores p altos: los datos son probables con una hipótesis nula verdadera.
Valores p bajos: los datos son poco probables con una hipótesis nula verdadera.

Pvalor < alpha = 0,05 –> Aceptas la H1
Pvalor >= alpha = 0,05 –>“Aceptas la H0”



¿Cómo compruebo si una serie es Stationary?

1. **Representando la serie**: que no crezca/decrezca, tenga una tendencia constante.
2. **Seasonality**: se comprueba fácilmente de manera gráfica
3. **Estadísticos**: divide el Time Series en varias ventanas y calcula los estadísticos para cada ventana.
4. **Unit Root Tests**: se plantea la hipótesis nula de que los estadísticos no son constantes en el tiempo. El objetivo es rechazar la hipótesis nula. Rechazaremos la hipótesis nula si su p-value es inferior a su nivel de significación (0.05).
    a. **Augmented Dickey Fuller test (ADH Test)**
    b. Kwiatkowski-Phillips-Schmidt-Shin – KPSS test: tendencia
    c. Philips Perron test

¿Qué hago para hacerla Stationary?

1. **Seasonality**: habrá que quitar la componente estacional. Diferenciando (siguiente diapo) o utilizando un SARIMA, que tiene en cuenta la seasonality.
2. **Tendencia**: hay que quitarla. Puede servir restar la media del TS, restar la componente de la tendencia del modelo aditivo o diferenciar con un lag
3. **Autocorrelación**: diferenciar primero por un lag y luego de nuevo si es necesario para eliminar la autocorrelación.

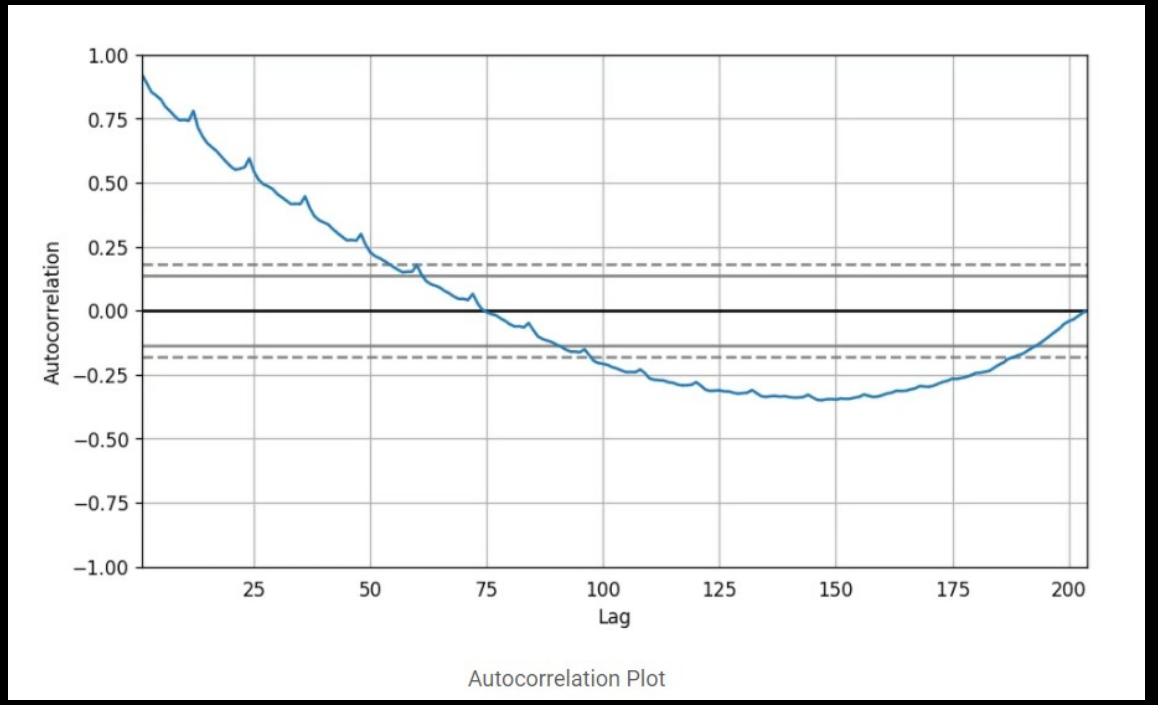
¿Cómo compruebo que el TS es seasonal?

1. **Plot**: lo más probable es que con un simple plot, veamos los patrones repetidos en la serie temporal

2. **Auto Correlation Function (ACF)**: Cuánto de correlacionados están los valores con instantes anteriores.

   La correlación lineal nos da un indicador de cuánto se relaciona linealmente una variable con otra. La autocorrelación sirve para ver si hay relación entre cada instante y sus lags, es decir, instantes anteriores.

   Si existe una estacionalidad mensual, habrá correlación entre cada instante y su valor 12 instantes más atrás. 24 instantes más atrás también estará relacionado, pero no tanto.



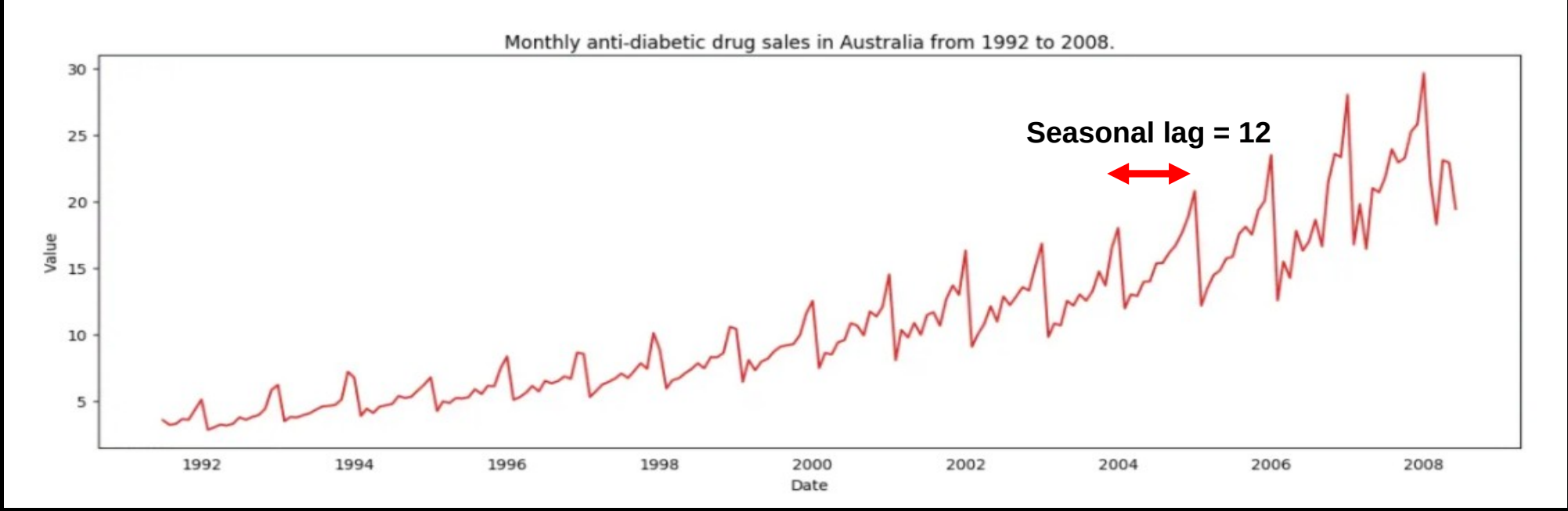
Deseasonalize

Moving Average: aplicar moving average sobre la ventana estacional (media de time series para cada ventana)

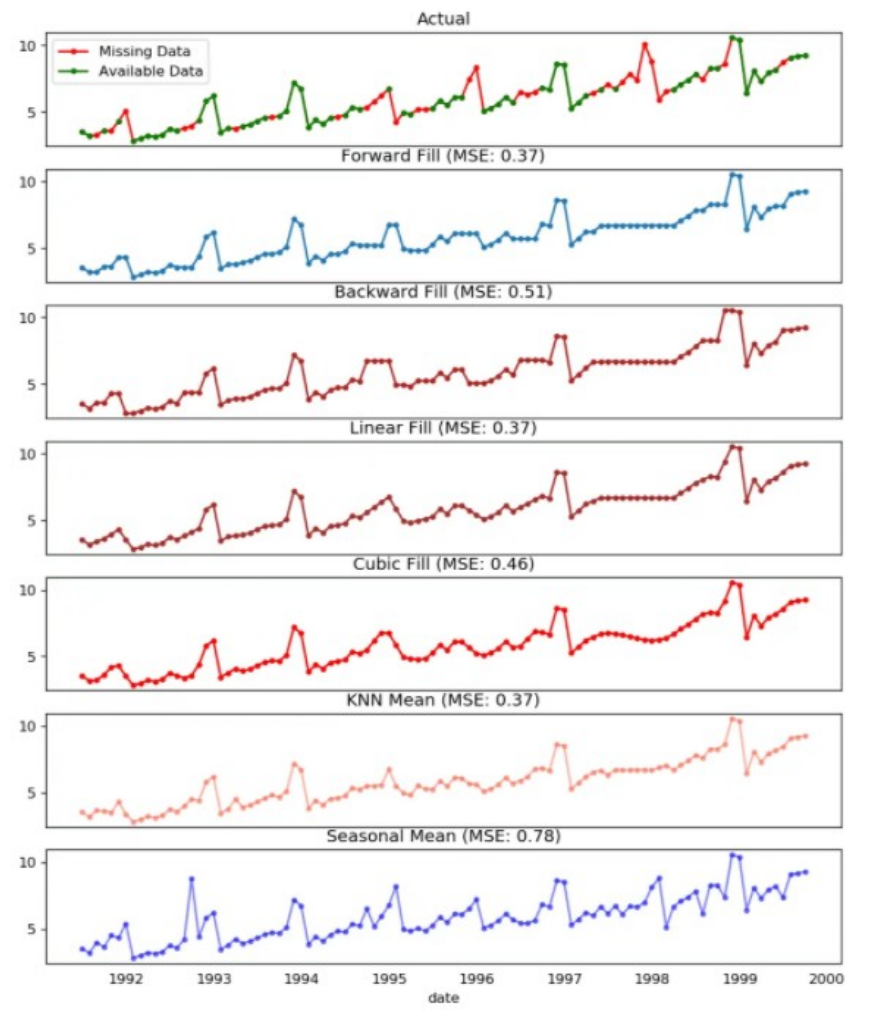
Seasonal difference: resta de Y(t) - Y(t-s), siendo s el periodo estacional: 12 (meses), 4 (semanas)...

Restar Seasonal component: descomponer la serie mediante un modelo aditivo, y restarle la componente de seasonality a la serie.

SARIMA: usar un modelo que integre automáticamente este componente



Missings



Missings en los valores

1. **Backward fill**: lo más probable es que con un simple plot, veamos los patrones repetidos en la serie temporal
2. **Forward fill**: Cuánto de correlacionados están los valores con instantes anteriores.
3. **Interpolate**: interpolar valores en función del anterior y posterior. Se suele usar la media.
4. **KNN**: similitud con los n últimos valores de la serie.
5. Linear interpolation
6. Quadratic/Cubic interpolation

Missings en las fechas

Podría ocurrir que falte alguna fecha. Se podría solucionar obteniendo un range desde el primer valor al último, equiespaciandolo como nuestros datos (diario, horario...), y a este vector le aplicamos un left join de la serie original. Obtendremos como resultado un TS sin huecos en el tiempo y con algunos missings en los valores, que habrá que rellenar con las técnicas anteriores.

Time Series

Técnicas de Regresión

Auto Regressive (AR)

Modelo que predice valores futuros basados en los datos pasados. Para utilizar un modelo AR necesitamos un TS stationary.

En la fórmula beta es un parámetro que podremos ir modificando, de -1 a 1, y epsilon es ruido.
p es el orden del modelo de autoregresión. Se suele usar 1 o 2.
Alpha es el intercept.

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \epsilon_1$$

Moving Average (MA)

Básicamente el MA predice valores futuros utilizando los errores cometidos en el pasado. También necesitamos un TS stationary.
q es el orden del modelo MA.

$$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

ARIMA

AutoRegressive Integrated Moving Average

Modelo que utiliza valores y errores pasados para realizar predicciones en una serie temporal.

Seasonality

La serie temporal NO puede tener seasonality. ARIMA lo modelará con el parámetro d. O diferencio por el lag de su seasonality o utilizo SARIMA.

¿Por qué no puede tener seasonality?

Este modelo es una REGRESIÓN basada en instantes anteriores, por lo que no puede existir correlación alta entre instantes de la serie temporal. Seasonality implica repetición de patrones cada X periodo (12 meses, 24h...), por lo que NO podemos usar datos con seasonality. Necesitamos garantizar la independencia de los lags de la serie.

SARIMA

Podemos utilizar el modelo SARIMA para series temporales con seasonality.

Stationary

El TS tiene que ser stationary. Lo haremos con el parámetro d

p

Auto Regressive (AR)
Cantidad de lags usados como predictores

d

Integrated (I)
Número de veces que hay que diferenciar el TS

q

Moving Average (MA)
Cantidad de errores utilizados en la predicción

¿Cómo consigo un TS stationary?

Diferenciándolo una o dos veces. Restando cada valor por su anterior.
Esto sería el parámetro “d” de ARIMA. Si la serie ya es stationary, d=0.

¿Cómo elimino la seasonality?

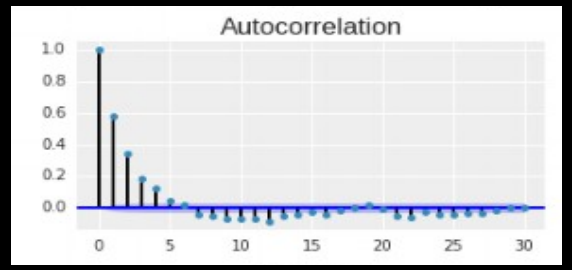
Diferenciándolo por el periodo de la estacionalidad. Por ejemplo, diferenciando de 12 en 12, si tenemos datos mensuales.
O usando un SARIMA

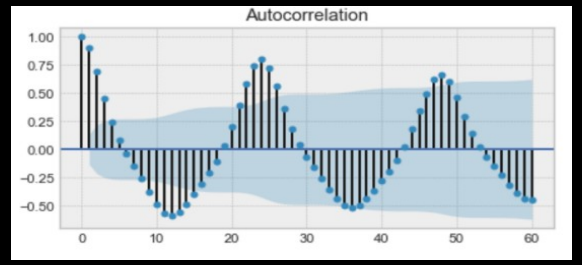
Valor differencing (d)

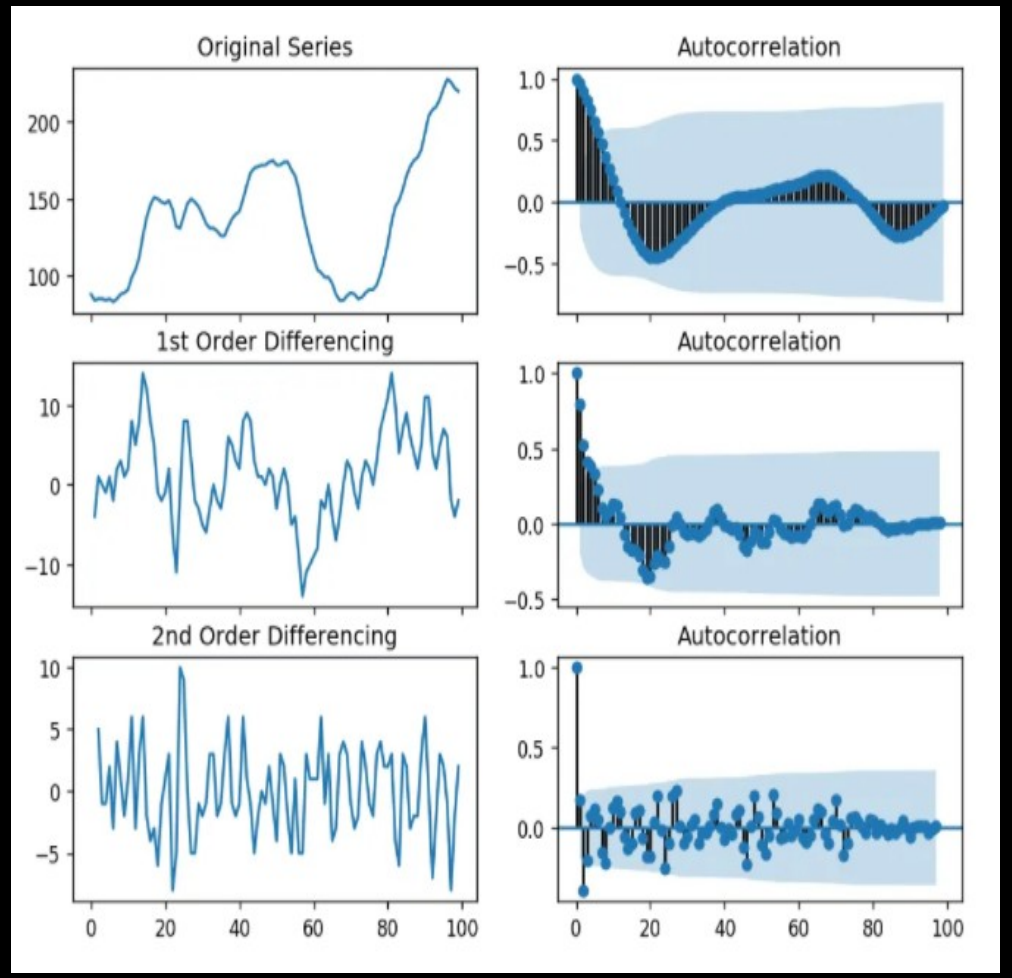
¿Para qué diferenciamos? Para convertir un non-stationary TS en un stationary. Por tanto, primero habrá que comprobar que efectivamente el TS es stationary. Para ello podemos aplicar el **test Augmented Dickey Fuller.**

Para obtener el valor óptimo de diferenciación hay que fijarse en la ACF (Autocorrelation Function). Buscamos que los lags no tengan autocorrelación. Posibles escenarios:

1. A partir de cierto lag, tiene una caída brusca en correlación: diferenciaremos por ese lag
2. Una caída exponencial: sería una buena situación. Diferenciaríamos por el lag 1
3. 10 lags o más con autocorrelación alta: tiene pinta de que tendremos que diferenciar dos veces.

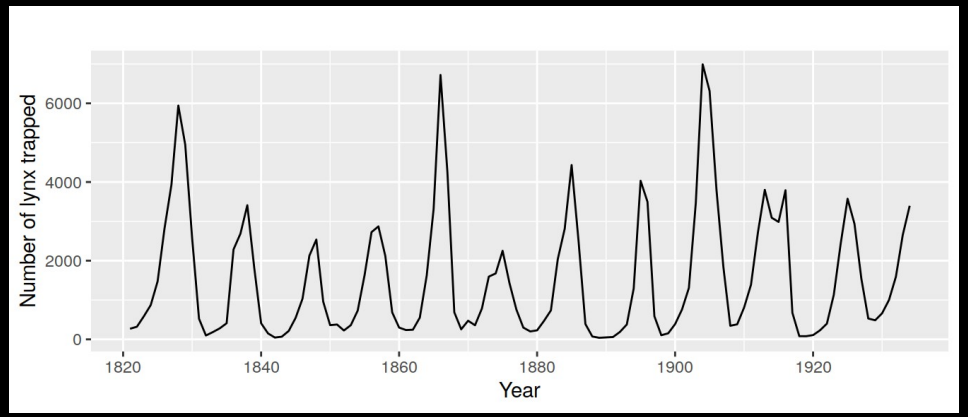






SARIMA

Modelo ARIMA que permite modelizar los comportamientos de seasonalidad, es decir, patrones repetitivos en los datos



SARIMAX

Añadimos una variable exógena, que básicamente es una variable externa

Otros modelos

Podemos transformar la serie temporal en un conjunto de lags, que serían las features del modelo, e intentaríamos predecir como si fuese un modelo clásico supervisado de regresión.

En este caso podríamos aplicar otros algoritmos: DecisionTreeRegressor, RandomForestRegressor...

date	data	t-12	t-11	t-10	t-9	t-8	t-7	t-6	t-5	t-4	t-3	t-2	t-1
1963-01-01	0.83	0.710000	0.630000	0.850000	0.440000	0.610000	0.69	0.92	0.55	0.72	0.77	0.92	0.60
1963-04-02	0.80	0.630000	0.850000	0.440000	0.610000	0.690000	0.92	0.55	0.72	0.77	0.92	0.60	0.83
1963-07-02	1.00	0.850000	0.440000	0.610000	0.690000	0.920000	0.55	0.72	0.77	0.92	0.60	0.83	0.80
1963-10-01	0.77	0.440000	0.610000	0.690000	0.920000	0.550000	0.72	0.77	0.92	0.60	0.83	0.80	1.00
1964-01-01	0.92	0.610000	0.690000	0.920000	0.550000	0.720000	0.77	0.92	0.60	0.83	0.80	1.00	0.77
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1979-10-01	9.99	6.840000	9.540000	10.260000	9.540000	8.729999	11.88	12.06	12.15	8.91	14.04	12.96	14.85
1980-01-01	16.20	9.540000	10.260000	9.540000	8.729999	11.880000	12.06	12.15	8.91	14.04	12.96	14.85	9.99
1980-04-01	14.67	10.260000	9.540000	8.729999	11.880000	12.060000	12.15	8.91	14.04	12.96	14.85	9.99	16.20
1980-07-02	16.02	9.540000	8.729999	11.880000	12.060000	12.150000	8.91	14.04	12.96	14.85	9.99	16.20	14.67
1980-10-01	11.61	8.729999	11.880000	12.060000	12.150000	8.910000	14.04	12.96	14.85	9.99	16.20	14.67	16.02

Facebook Prophet

Librería open source de Facebook que permite realizar forecast de series temporales

pip install fbprophet

Clean your Time Series Data

@daansan_ml

MACHINE LEARNING PILLS

Handle missing values

Remove trend

Remove seasonality

Check for stationarity

Normalize the data

Remove outliers

Smooth the data

Build your model