

# **DIPLOMATURA EN MACHINE LEARNING CON PYTHON**

## **Series Temporales**

## Series temporales

- Taxonomía
- Separación de componentes
- Predicciones
- Implementación con statsmodel
- Predicción en series con único período

## Series temporales

### Introducción

Una serie temporal es una colección de observaciones de una variable aleatoria tomadas en forma secuencial a medida que transcurre el tiempo. Normalmente estas observaciones se toman en instantes de tiempo equiespaciados.

Permite predecir el comportamiento de variables a lo largo del tiempo. Como modelo predictivo, tratarán de analizar el comportamiento de una serie temporal en el pasado para inferir su comportamiento en el futuro.

Se pueden señalar ejemplos tomados de numerosos campos de la actividad humana:

#### Demografía

- Población de un país año por año.
- Mortalidad infantil año por año.

#### Economía

- Precio del alquiler de departamentos a lo largo de varios meses.
- Evolución diaria del precio de la soja.
- Ganancias de cierta empresa tomadas mes por mes.
- Precio del barril de petróleo.

#### Medioambiente

- Niveles de anhídrido carbónico en una ciudad, hora por hora, durante varios años.
- Precipitaciones por día en una localidad.
- Temperatura media mensual a lo largo de decenas de años.
- Concentración de plomo en un río.

En las series temporales las sucesivas observaciones no pueden asumirse como independientes entre sí, y, por eso, el análisis se lleva a cabo respetando el orden temporal correspondiente.

Los métodos estadísticos basados en la independencia de las observaciones no son válidos para el análisis de series temporales porque las observaciones en un instante de tiempo dependen de los valores de la serie en el pasado.

### ¿Qué se propone el análisis de las series temporales?

Podemos contemplar varios posibles objetivos:

### 1. Descripción

Al estudiar una serie temporal, lo primero es graficarla y así ver con claridad sus características y medidas descriptivas básicas.

Por ejemplo:

- Si los datos presentan forma creciente (tendencia).
- Si existe influencia de ciertos periodos de cualquier unidad de tiempo (estacionalidad).
- Si aparecen outliers (observaciones extrañas o discordantes).

### 2. Predicción

Simplemente queremos, nunca es fácil, inferir los valores futuros (desconocidos) a partir de los valores pasados. Como se trata de una extrapolación siempre resulta más difícil.

#### Clasificaciones de las series temporales

- Si las observaciones se toman a intervalos regulares diremos la serie temporal es discreta.
- Si existe un valor para cada instante de tiempo diremos, contrariamente, que es continua.
- Si los valores que va a tomar la serie pueden predecirse en forma precisa diremos que la serie es determinística.
- Si, por el contrario, los valores futuros dependen de las observaciones pasadas pero sólo se pueden calcular en forma aproximada ya que también intervienen factores aleatorios entonces la serie la llamamos estocástica.

#### Series temporales deterministas

- Tendencia lineal: consiste en la extrapolación hacia el futuro. La variable objeto del estudio es una función determinista del tiempo.

$$X_t = a + b t$$

- Medias móviles: la variable en el momento t es un promedio simple de un determinado número de valores pasados.

$$X_t = (X_{t-1} + \dots + X_{t-12})/12$$

- Alisado exponencial: utilizan toda la información del pasado y dan mayor peso a valores recientes

$$X_{t+1} = X_t + (1 - \alpha)X_{t-1} + (1 - \alpha)^2 X_{t-2} + \dots \quad 0 \leq \alpha \leq 1$$

### Series temporales estocásticas

- Enfoque ARIMA: parte de la consideración general de que la serie temporal que se trata de predecir es generada por un proceso estocástico o aleatorio cuya naturaleza puede ser caracterizada y descripta mediante un modelo. Se trata de identificar el proceso estocástico que ha generado estos datos, estimar los parámetros que caracterizan dicho proceso, contrastar que se cumplan las hipótesis y, cuando se satisfagan las condiciones, podemos utilizar el modelo para predecir.

### Aspectos de una serie temporal

El análisis descriptivo de series temporales busca descomponer la variación de una serie en componentes básicas. Este enfoque no siempre resulta el más adecuado, pero siempre es interesante intentarlo, sobre todo, cuando en la serie se observa cierta tendencia o cierta periodicidad.

Ninguna descomposición no es en única. Se parece a tratar de representar un problema en distintos sistemas de coordenadas. Todas las representaciones pueden ser correctas pero, normalmente sólo una es la más útil.

Buscamos entonces encontrar componentes que representen las tendencias de largo plazo, los comportamientos periódicos o estacionales y la aleatoriedad.

- Tendencia: Se relaciona con el comportamiento de la serie a largo plazo. Se vislumbra respecto de un nivel medio que involucre, si los hubiera, a varios períodos. Una media móvil que tome una cantidad entera de períodos representa la tendencia de largo plazo.
- Estacionalidad: Varias series temporales muestran comportamientos periódicos. Por ejemplo, el desempleo aumenta en invierno y disminuye en verano por los empleos del sector turístico.
- Estos efectos suelen ser fáciles de entender y se pueden medir con facilidad. Sobre este particular veremos que es posible incluso desestacionalizar una serie.
- Componente Aleatoria: Da cuenta del resto del comportamiento de la serie. Tendrá sentido tratar de descubrir qué tipo de distribución sigue este resto aleatorio mediante algún modelo probabilístico que nos ayude a estimar los desvíos que podrán sufrir nuestras predicciones.

De estas tres componentes, las dos primeras, resultan determinísticas mientras que la tercera es estocástica.

Lo vamos a expresar como:

$$S(t) = L(t) + P(t) + R(t)$$

Donde:

L: representa la tendencia a largo plazo

P: representa la parte periódica o estacional

R: da cuenta del comportamiento aleatorio

¿Cómo proceder?

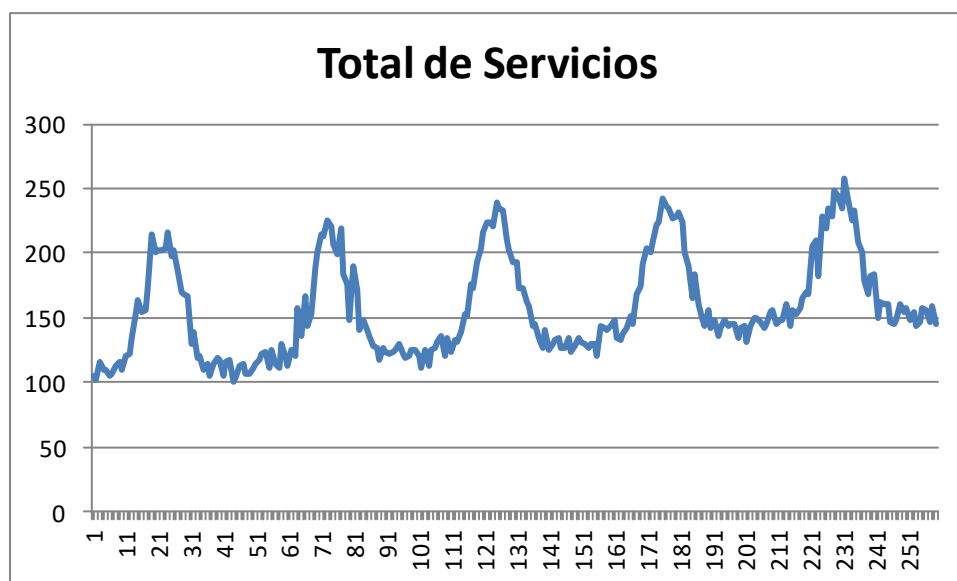
L(t) puede obtenerse con una media móvil si se elige adecuadamente la cantidad de elementos a promediar. Debiera ser el mínimo común múltiplo de todos los períodos involucrados.

P(t) puede estimarse a partir de una transformada de fourier. En el espectro de potencias aparecerán las frecuencias principales de la serie.

R(t) se obtiene luego por diferencia. Nos interesará saber qué tipo de distribución siguen los residuos. Para eso podremos recurrir, típicamente a tests del tipo de Chi cuadrado.

### Ejemplo

Cantidad de pedidos de médico a domicilio por semana:



Los datos se encuentran en la máquina virtual provista dentro de la base de datos "seriestemporales" en la tabla "pedidosmedicos"

### Clasificación cualitativa de las series temporales

#### Estacionarias:

- Media estable en el tiempo
- Amplitud constante en el tiempo
- Período constante en el tiempo

#### No Estacionarias:

No se cumple alguna de las condiciones necesarias para que la serie sea estacionaria.

### Diferenciación de la serie

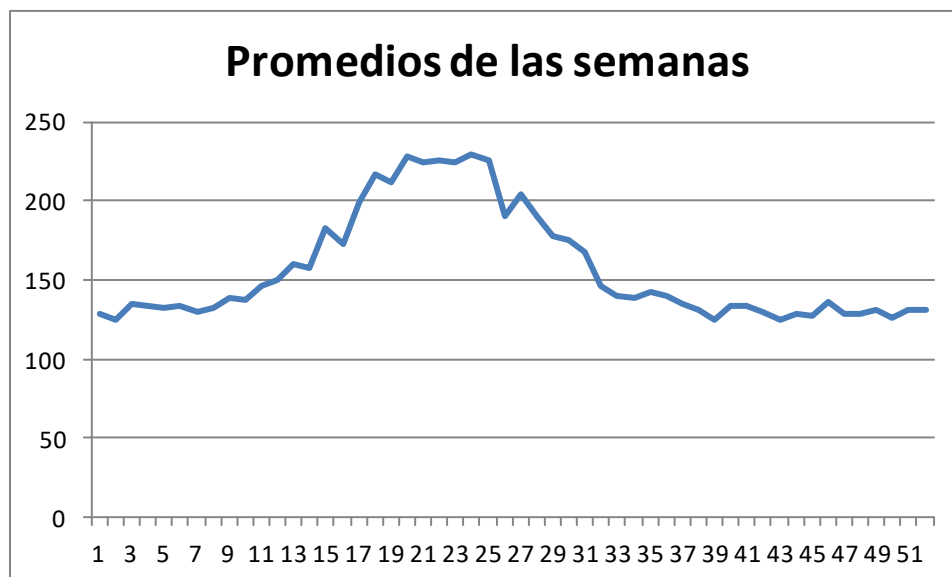
Una para eliminar la tendencia consiste en suponer que la que esta última evoluciona lentamente en el tiempo, de manera que en el instante  $t$  la tendencia debería estar muy próxima a la tendencia en el instante  $t - 1$ . Así, si restamos a cada valor de la serie el valor anterior, la serie resultante estará aproximadamente libre de tendencia. Este truco se llama diferenciación de la serie.

Generamos así una nueva serie cuya tendencia, por lo menos, quedará muy reducida. Aquí podemos con más facilidad aplicar el análisis de Fourier (ver tutorial aplicando Fourier)

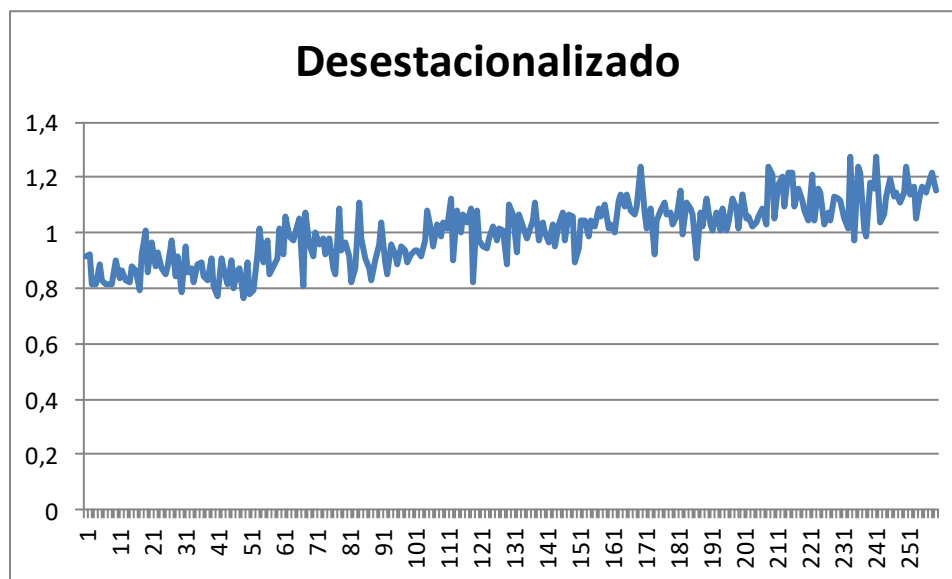
Para el ejemplo que veníamos tratando mostraría un pico alrededor de 52 dando cuenta del período anual que tiene nuestro problema. Con un poco más de esfuerzo, seríamos capaces de identificar otras ondas más cortas (armónicos superiores) y así identificar la función periódica que está oculta.

Una vez que hemos identificado el período más largo que tiene sentido en nuestro problema podemos proceder a desestacionalizar la serie.

Para eso tomamos a lo largo de toda la serie el promedio de todas las primeras semanas de cada 52 (un año), de todas las segundas, de todas las terceras, etc.



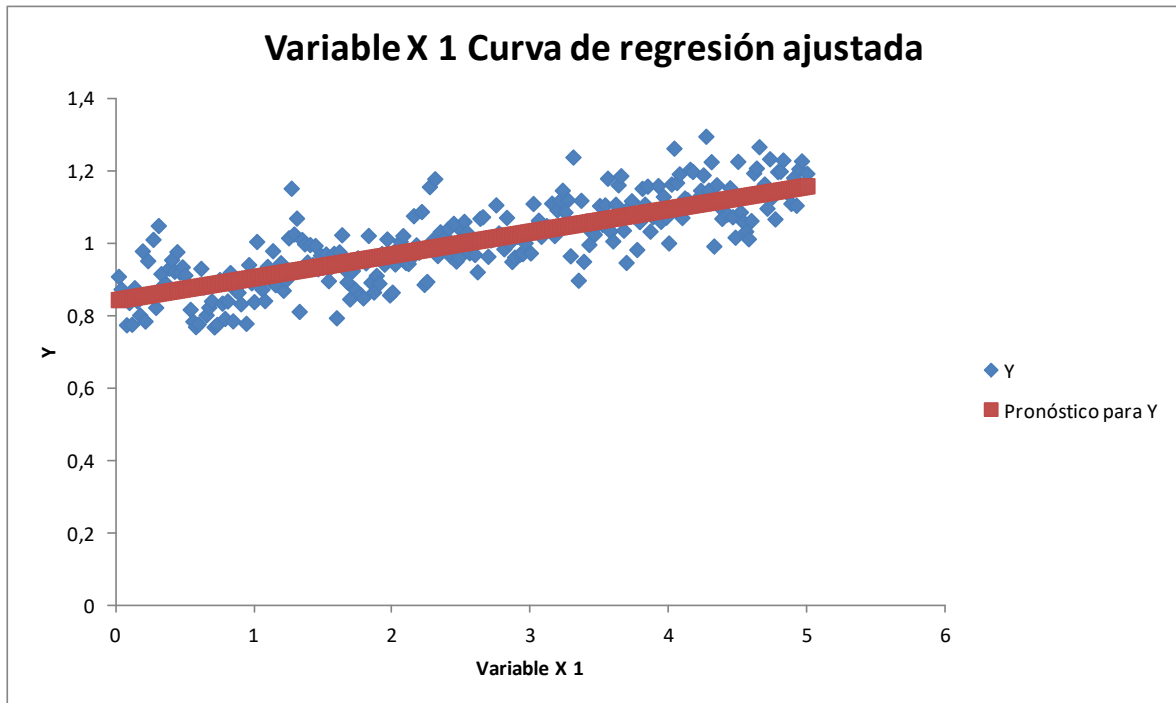
Con esto estamos en condiciones de desestacionalizar la serie total, simplemente dividiendo el valor que toma cada semana de la serie original por el promedio que acabamos de graficar:



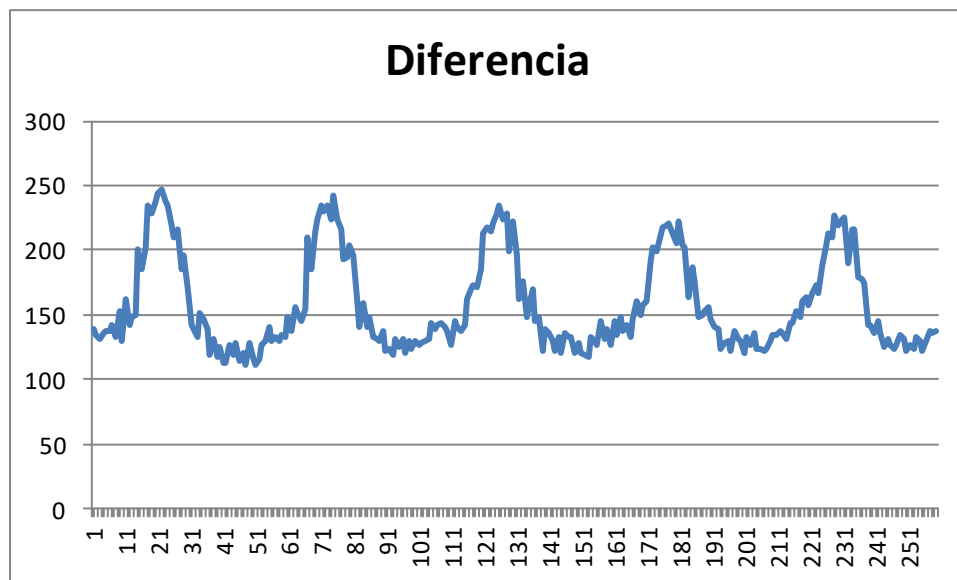
Ya tenemos un conocimiento más preciso de lo que pasa con nuestra serie original:  
Tenemos una oscilación de período anual (52 semanas)  
Tenemos armónicos superiores (26, 13 semanas, quizá más)  
Tenemos un aumento de apariencia lineal

La regresión se parece a:





Al dividir la serie original por el valor teórico de la regresión nos quedamos con la parte periódica separada completamente de la tendencia a largo plazo:



Con este trabajo hemos sido capaces de separar los componentes de la serie temporal y estamos en condiciones de intentar una interpretación:

- El crecimiento a largo plazo podría ser debido a un crecimiento en la cartera de clientes cubiertos por nuestra empresa.

- La oscilación anual se debe a la epidemia periódica de gripe que tiene un máximo cada invierno
- Los términos superiores dan cuenta de la forma del pico que no coincide con una senoidal pura.

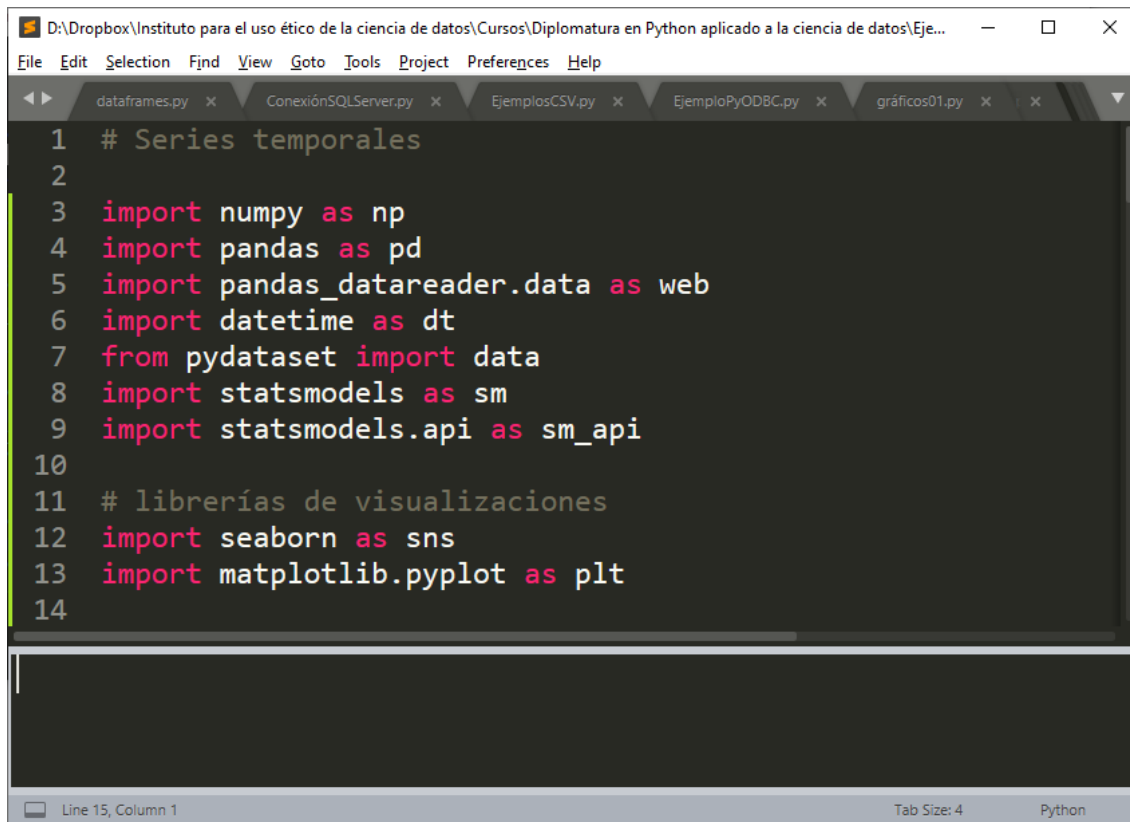
## Implementación en Python

### Manipulación de series temporales con Pandas

La librería Pandas nos provee una serie de herramientas para trabajar con series temporales:

- Creación de una serie
- Consulta de valores
- Agregado de valores al cierre de un período
- Desplazamientos temporales
- Visualización
- Medias móviles
- Separación en componentes
- Pronósticos con ARIMA

Carga de las librerías que usaremos en el ejemplo:

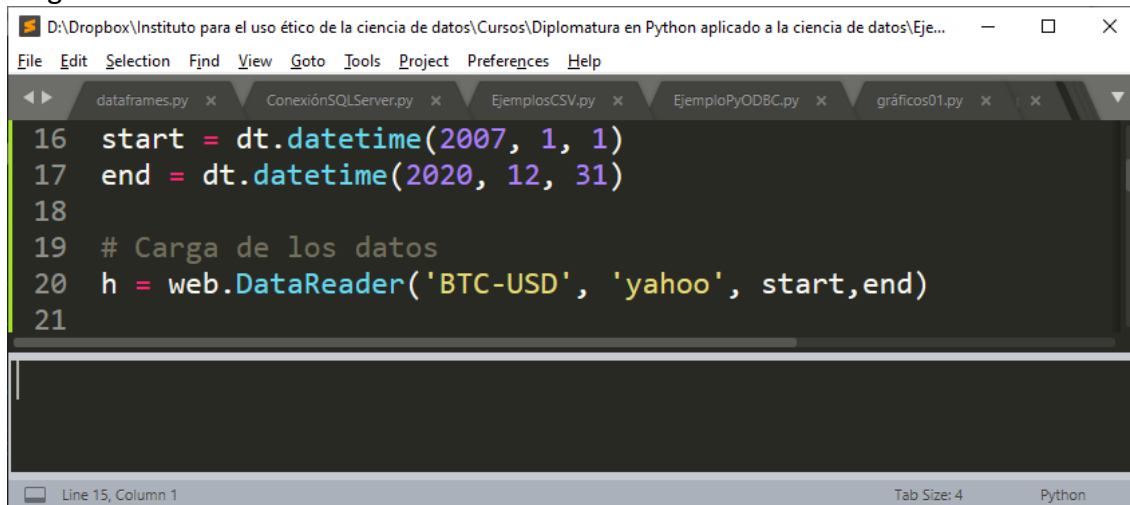


A screenshot of a Python IDE window. The title bar shows the file path: D:\Dropbox\Instituto para el uso ético de la ciencia de datos\Cursos\Diplomatura en Python aplicado a la ciencia de datos\Eje... The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. The tab bar shows several open files: dataframes.py, ConexiónSQLServer.py, EjemplosCSV.py, EjemploPyODBC.py, and gráficos01.py. The main editor area displays the following Python code:

```
1 # Series temporales
2
3 import numpy as np
4 import pandas as pd
5 import pandas_datareader.data as web
6 import datetime as dt
7 from pydataset import data
8 import statsmodels as sm
9 import statsmodels.api as sm_api
10
11 # librerías de visualizaciones
12 import seaborn as sns
13 import matplotlib.pyplot as plt
14
```

The status bar at the bottom indicates 'Line 15, Column 1', 'Tab Size: 4', and 'Python'.

Carga de los datos de los valores del Bitcoin:



A screenshot of a Python IDE window, similar to the one above. The title bar shows the same file path. The menu bar and tab bar are identical. The main editor area displays the following Python code:

```
16 start = dt.datetime(2007, 1, 1)
17 end = dt.datetime(2020, 12, 31)
18
19 # Carga de los datos
20 h = web.DataReader('BTC-USD', 'yahoo', start, end)
21
```

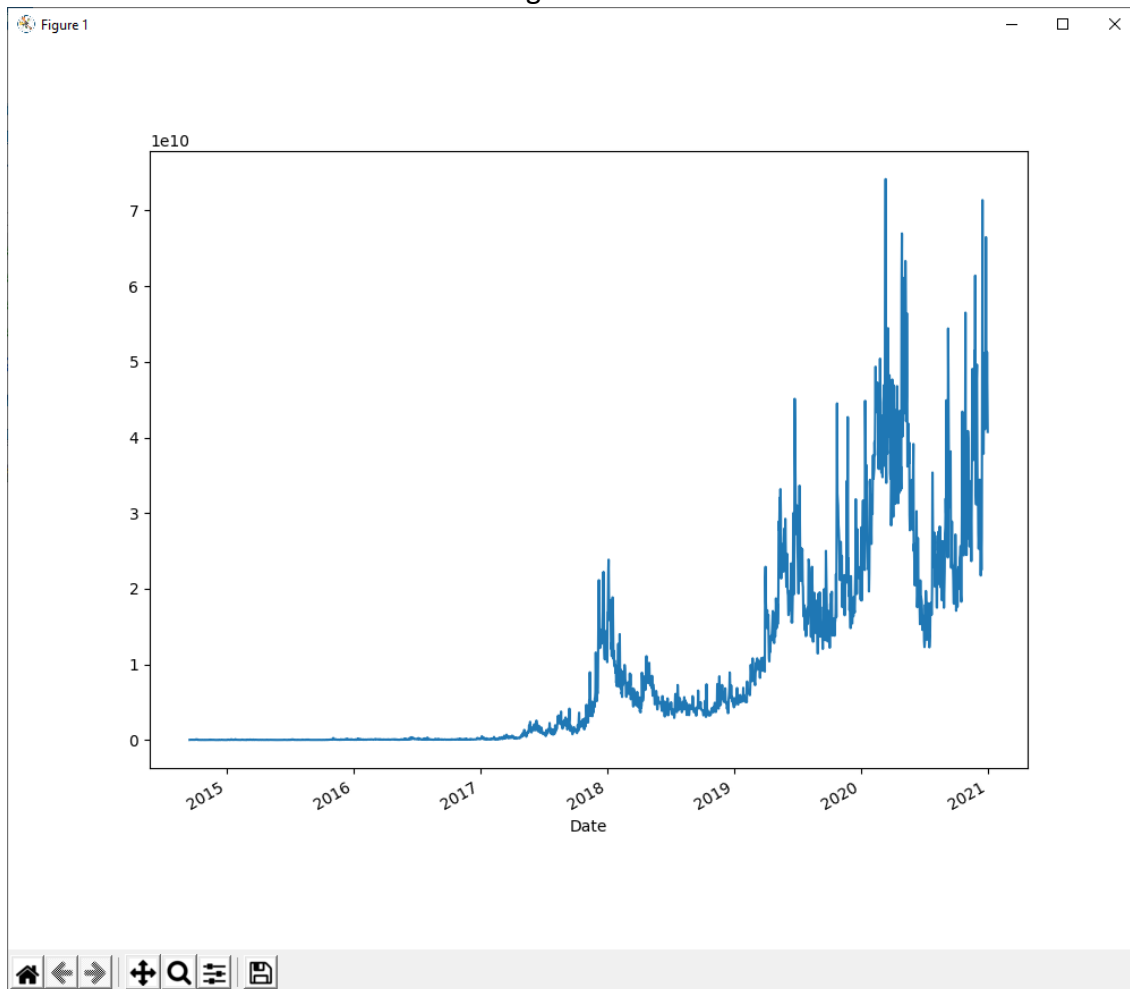
The status bar at the bottom indicates 'Line 15, Column 1', 'Tab Size: 4', and 'Python'.

Visualizamos los datos que acabamos de cargar:

```
D:\Dropbox\Instituto para el uso ético de la ciencia de datos\Cursos\Diplomatura en Python aplicado a la ciencia de datos\Eje...
File Edit Selection Find View Goto Tools Project Preferences Help
dataframes.py x ConexiónSQLServer.py x EjemplosCSV.py x EjemploPyODBC.py x gráficos01.py x x
25 # visualización de los datos
26
27 serie = h["Volume"].squeeze()
28
29 plot = serie.plot(figsize=(10,8))
30 plt.show()
```

Line 30, Column 1 Tab Size: 4 Python

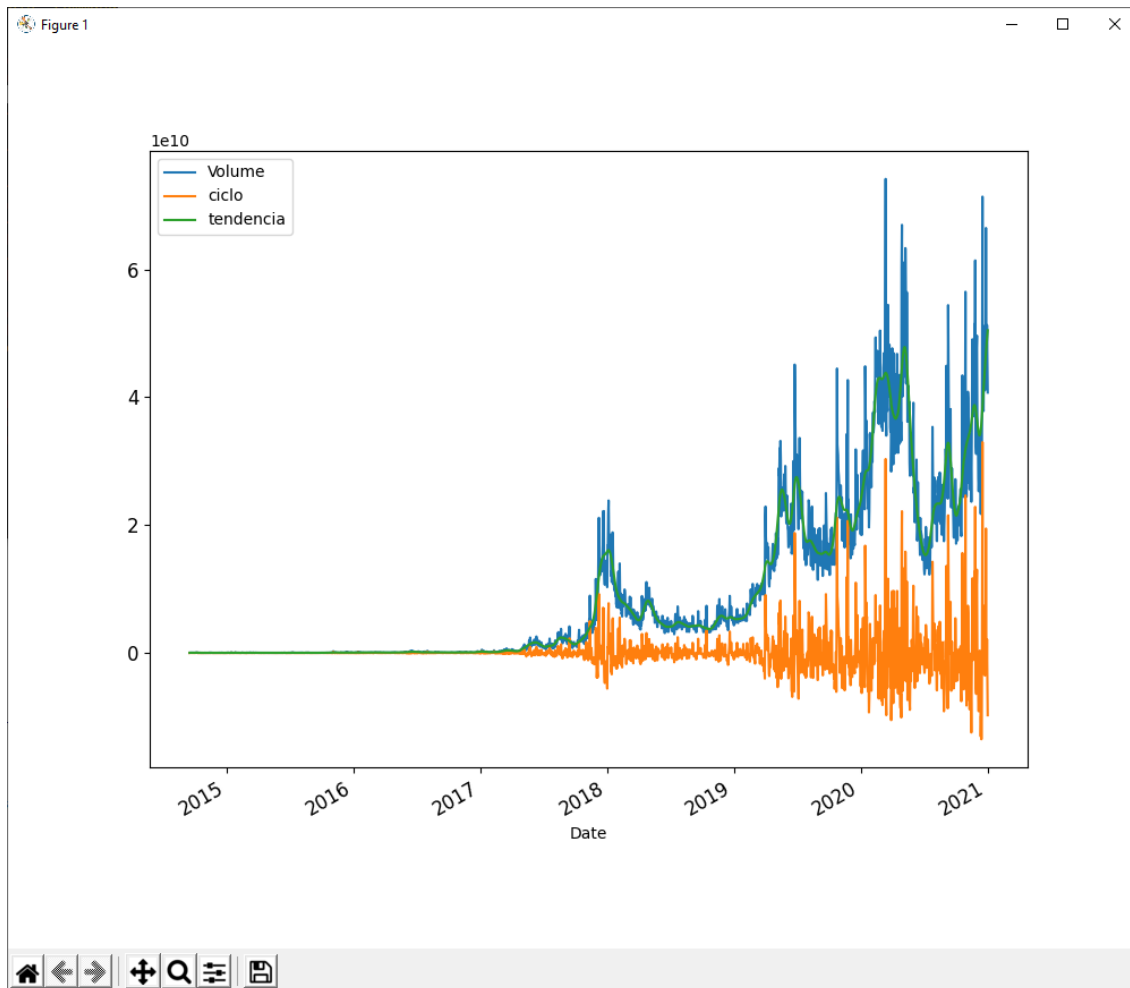
Y así vemos la evolución del volumen negociado:



Separamos la parte cíclica de la tendencia para tratar de cumplir con las condiciones de estacionariedad:

```
D:\Dropbox\Instituto para el uso ético de la ciencia de datos\Cursos\Diplomatura en Python aplicado a la ciencia de datos\Ejemplos de clase\SerieT...
File Edit Selection Find View Goto Tools Project Preferences Help
dataframes.py x ConexiónSQLServer.py x EjemplosCSV.py x EjemploPyODBC.py x gráficos01.py x AlgoritmosGenéticos.py x
31
32 # Filtro de Hodrick-Prescott para separar en tendencia y componente c
33
34
35 ciclo, tendencia = sm_api.tsa.filters.hpfilter(serie)
36
37 h["ciclo"] = ciclo
38 h["tendencia"] = tendencia
39
40 h[["Volume", "ciclo", "tendencia"]].plot(figsize=(10,8), fontsize=12)
41 legend = plt.legend()
42 legend.prop.set_size(14)
43 plt.show()
```

Y obtenemos:

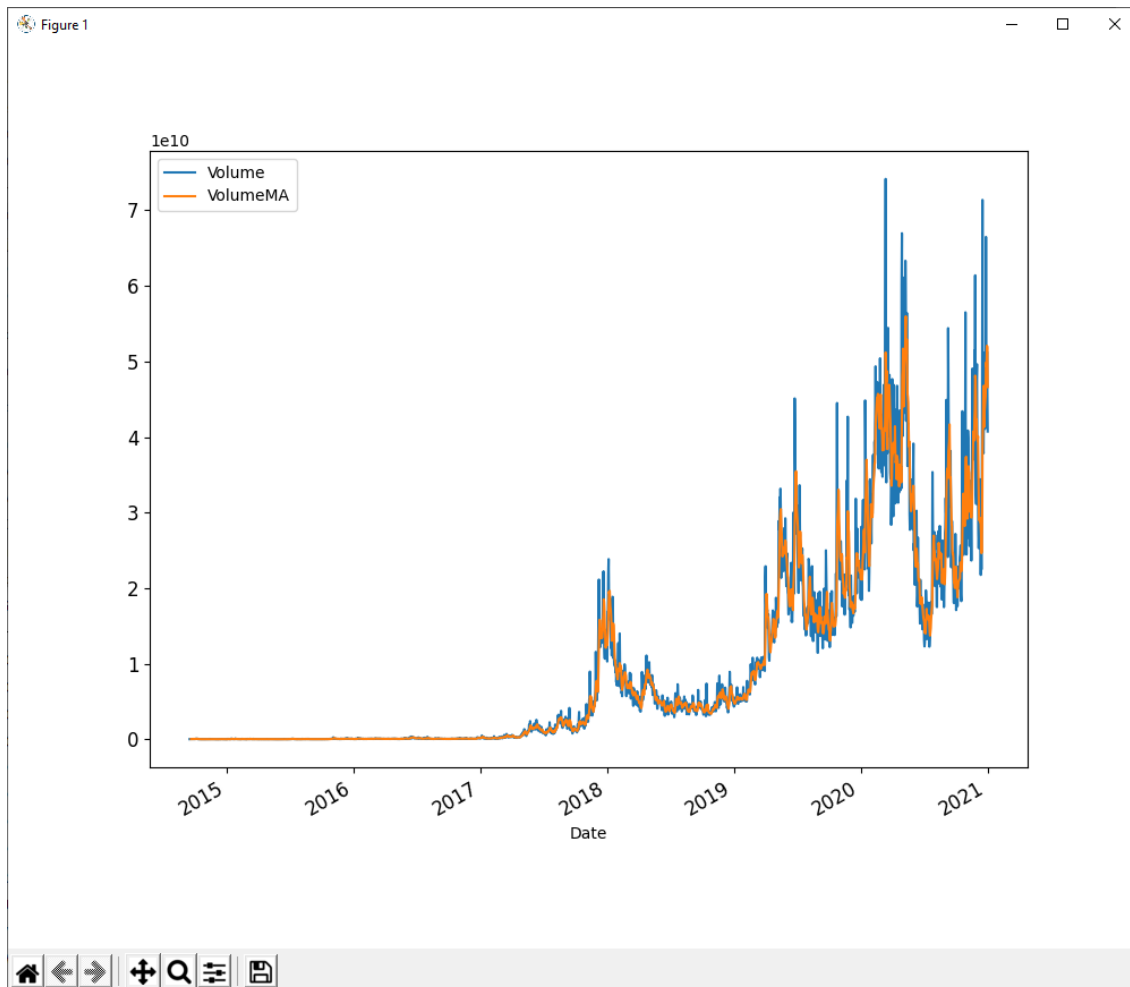


Para calcular la media móvil recurrimos a:

```
D:\Dropbox\Instituto para el uso ético de la ciencia de datos\Cursos\Diplomatura en Python aplicado a la ciencia de datos\Ejemplos de clase\SerieT...
File Edit Selection Find View Goto Tools Project Preferences Help
dataframes.py x ConexiónSQLServer.py x EjemplosCSV.py x EjemploPyODBC.py x gráficos01.py x AlgoritmosGenéticos.py x
44
45 VolumeMA = serie.rolling(window=5).mean()
46
47 h["VolumeMA"] = VolumeMA
48
49 plot = h[["Volume", "VolumeMA"]].plot(figsize=(10,8), fontsize = 12)
50 plt.show()
51
```

Line 50, Column 1 Tab Size: 4 Python

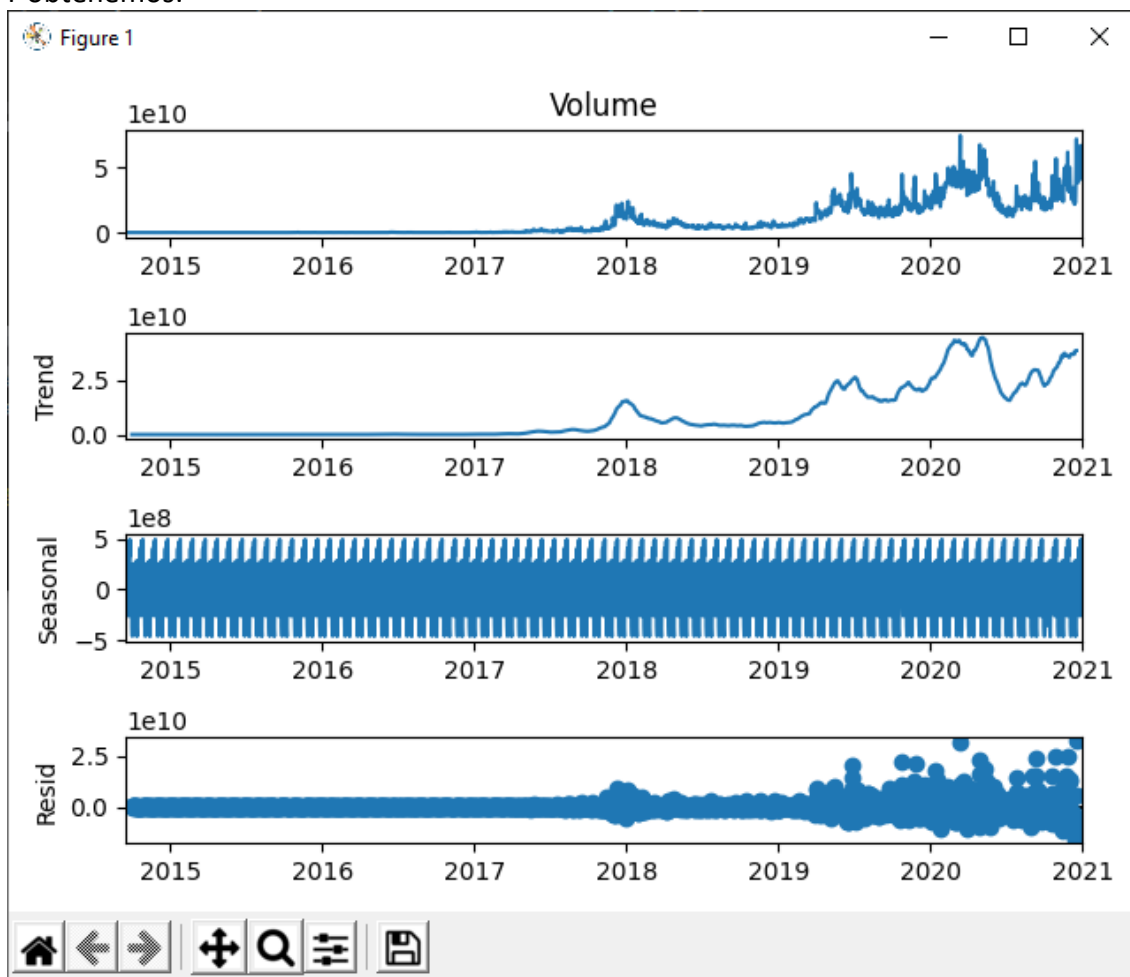
Y obtenemos



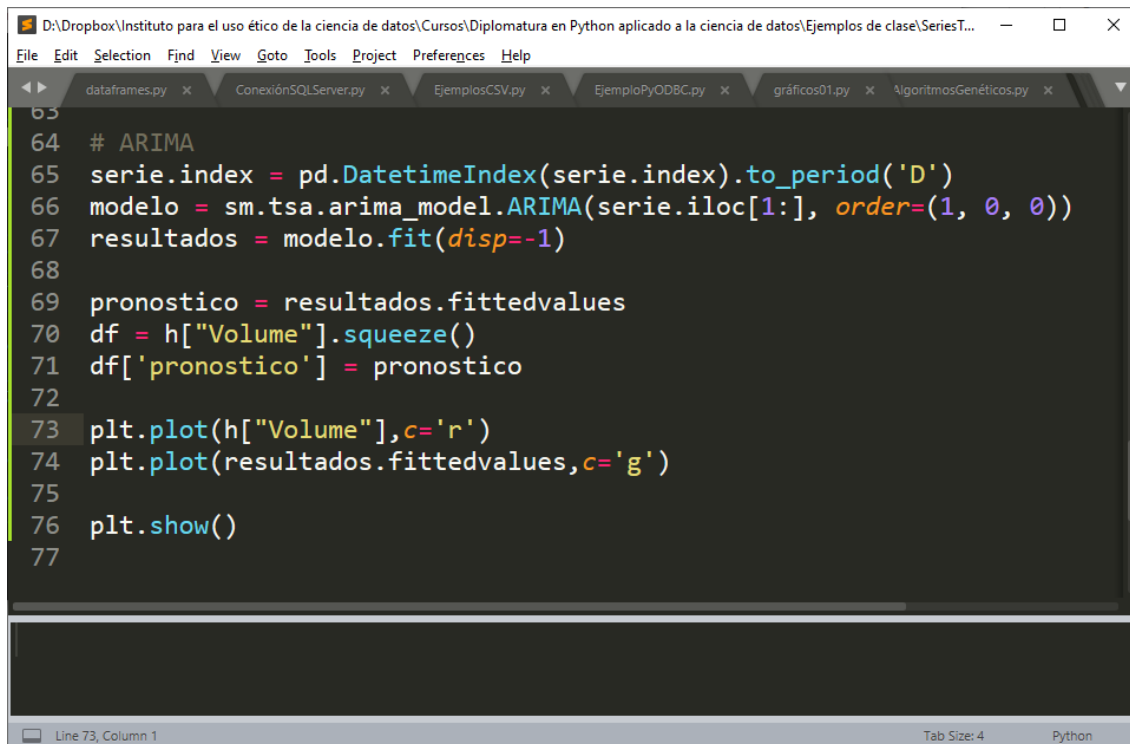
La descomposición de nuestra serie temporal la conseguimos con:

```
D:\Dropbox\Instituto para el uso ético de la ciencia de datos\Cursos\Diplomatura en Python aplicado a la ciencia de datos\Ejemplos de clase\SerieT...
File Edit Selection Find View Goto Tools Project Preferences Help
dataframes.py x ConexiónSQLServer.py x EjemplosCSV.py x EjemploPyODBC.py x gráficos01.py x AlgoritmosGenéticos.py x
51
52 descomposicion = sm_api.tsa.seasonal_decompose(serie, model='additive
53
54 fig = descomposicion.plot()
55 plt.show()
```

Y obtenemos:



Tratamos, por fin, de construir un modelo con ARIMA:



```
63
64 # ARIMA
65 serie.index = pd.DatetimeIndex(serie.index).to_period('D')
66 modelo = sm.tsa.arima_model.ARIMA(serie.iloc[1:], order=(1, 0, 0))
67 resultados = modelo.fit(dispatch=-1)
68
69 pronostico = resultados.fittedvalues
70 df = h["Volume"].squeeze()
71 df['pronostico'] = pronostico
72
73 plt.plot(h["Volume"], c='r')
74 plt.plot(resultados.fittedvalues, c='g')
75
76 plt.show()
77
```

The screenshot shows a Python IDE window with the title 'D:\Dropbox\Instituto para el uso ético de la ciencia de datos\Cursos\Diplomatura en Python aplicado a la ciencia de datos\Ejemplos de clase\SerieT...'. The code in the editor implements an ARIMA model for time series analysis. It converts the index to a daily period, fits an ARIMA(1,0,0) model, and plots the original data in red and the fitted values in green. The status bar at the bottom indicates 'Line 73, Column 1', 'Tab Size: 4', and 'Python'.

Y obtenemos:



