

# Portfolio

## Carlos Herreros-Lucas

1. Data Acquisition

2. Evaluation Of a Trading Strategy

3. Complementary applications

# 1. Data Acquisition

## 1.1 Web Scrapping of a Static Website

AIM: Check all the consultancy agencies in León (my hometown) and obtain their details (i. e. website url) in one dataframe.

Páginas Amarillas

The screenshot shows a search interface for 'asesoria' in 'leon'. The results list three consultancy agencies:

- Asesoria en Leon (León)**: 189 reviews. Services: Negocios llamados "Asesoria". Address: Paseo Condesa Sagasta, 46 1-A, 24001, León. Contact: Pedir cita, Te llamamos gratis, Ver teléfono.
- Asesoria Álvarez-Canal & Cia S.L.**: +20 años con experiencia. Services: Asesorías de empresas | Asesoria civil | Economista - Auditores - Abogados - Laboral - Concursal. Address: Paseo Condesa Sagasta, 46 1-A, 24001, León. Contact: Pedir cita, Te llamamos gratis, Ver teléfono.
- Bufete & Gestión**: +20 años con experiencia. Services: Abogados | Asesoria civil | Jurídico - Fiscal - Laboral - Herencias - Comunidades - Clausuras | Suelo - Reclamaciones Bancarias - Accidentes de Tráfico. Address: Calle Roa de la Vega,, 14 ENTRESUELO-DOHA, 24002, León. Contact: Pedir cita, Te llamamos gratis, Ver teléfono.

On the right side, there is a map of León city center with various landmarks labeled (e.g., MUSAC, LA PALOMERIA, Catedral de León, Plaza del Grano, Estación de Autobuses). A sidebar lists service types: asesoramiento fiscal (68), asesoria civil y mercantil (17), asesoria comunitaria (10), and asesoria contable (20).

Website

Out[8]:

Web
0 http://www.alvarez-canal.com?utm_campaign=paginasam...
1 http://www.asemorca.com?utm_campaign=paginasam...
2 http://www.asesoriabsasesores.com?utm_campaign=pag...
3 http://www.bufeteygestion.es?utm_campaign=paginas...
4 http://www.asesoriasantlagochamorro.es?utm_campa...
5 http://gestorialeigho.net?utm_campaign=paginasam...
6 http://www.etmasesores.com?utm_campaign=pagina...
7 http://www.alvarezreguera.com/?utm_campaign=pa...
8 http://www.condesa10.es?utm_campaign=paginasam...
9 http://www.casadoyalegresesores.es?utm_campa...
10 http://www.mendezdeconomistas.com?utm_campaign=...
11 http://www.abarroso.com?utm_campaign=paginasam...
12 http://www.adecocomunidades.com?utm_campaign=p...
13 http://www.asesoriaperezgestion.es?utm_campa...
14 http://www.amadeoasesores.com?utm_campaign=pag...
15 no tiene
16 no tiene
17 http://www.abogadaineslopezdelacalzada.com?utm...
18 http://www.gestierzo.es?utm_campaign=paginasam...
19 http://www.asesoriascisterna.com?utm_campaign=...
20 http://www.charollamera.com?utm_campaign=pagin...
21 http://www.isolinaarias.es?utm_campaign=pagina...
22 http://www.merinoyasociados.com?utm_campaign=p...
23 https://www.aranzazuguerrezoblanca.com
24 no tiene
25 http://www.bufeteprida.es?utm_campaign=paginas...
26 http://www.despachotruvium.com/?utm_campaign=p...
27 http://www.gestoriasenleonhermes.com?utm_campa...
28 http://www.manuellosadasaabogados.com?utm_campa...

Dataframe

# 1. Data Acquisition

## 1.2 Web Scrapping of Dynamic Website (rendering)

AIM: Get all the details of the main indexes (i. e. US30, DAX, ... ) from a website (inveting.com) in a dataframe

The screenshot shows the Investing.com homepage with a search bar at the top containing 'Indice euro'. Below the search bar, there's a navigation menu with links like 'Mercados', 'Mi cartera', 'Criptomonedas', 'Noticias', 'InvestingPro', 'Análisis', 'Gráficos', 'Técnico', 'Brokers', 'Herramientas', 'Academia', and 'Más'. A sidebar on the left lists various stock indices with their current values and changes. The main content area displays a news article about industrial air conditioning with a large blue arrow pointing right.

Website

Indice	Mes	Ultimo	Maximo	Minimo	Var.	% Var.	Hora
US 30	NaN	33.756,50	33.971,50	33.397,50	+41,1	+0,12%	11/11
US 500	NaN	3.991,90	4.001,60	3.944,50	+35,5	+0,90%	11/11
US Tech 100	NaN	11.802,20	11.841,20	11.539,40	+196,2	+1,69%	11/11
US 2000	NaN	1.880,60	1.899,30	1.863,20	+13,9	+0,75%	11/11
S&P 500 VIX	Dic 2022	24,05	25,21	23,83	-0,58	-2,35%	11/11
DAX	Dic 2022	14.328,00	14.351,00	14.167,00	+153,0	+1,08%	11/11
CAC 40	Dic 2022	6.633,00	6.642,50	6.573,00	+75,5	+1,15%	11/11
FTSE 100	Dic 2022	7.354,20	7.434,80	7.321,00	+48,8	+0,59%	11/11
Euro Stoxx 50	Dic 2022	3.889,00	3.895,00	3.844,00	+42	+1,05%	11/11
Italia 40	Dic 2022	24.490,00	24.595,00	24.285,00	+185,0	+0,76%	11/11
Swiza 20	Dic 2022	11.146,00	11.241,00	11.091,50	+43,0	+0,39%	11/11
IBEX 35	Dic 2022	8.133,00	8.204,00	8.072,00	+18,6	+0,23%	11/11
RTS	Dic 2022	114.395,00	114.845,00	109.040,00	+4.258	+3,50%	10/11
WIG20	Dic 2022	1.700,00	1.706,00	1.629,00	+42,0	+2,53%	10/11
AEX	Dic 2022	707,17	709,25	699,52	+71	+0,81%	11/11
Ibovespa	Dic 2022	113.815,00	114.265,00	110.180,00	+2.615	+2,36%	11/11
Nikkai 225	Dic 2022	28.090,00	28.328,00	27.972,50	+180,0	+0,64%	11/11
TOPIX	Dic 2022	1.969,00	1.982,50	1.961,00	+15,50	+0,85%	11/11
Hang Seng	Nov 2022	17.581,00	17.625,00	17.339,00	+201,0	+1,16%	11/11
China H-Shares	Nov 2022	5.942,00	5.976,00	5.877,00	+56,0	+0,95%	11/11
CSI 300	Nov 2022	3.696,80	3.712,80	3.671,60	+22,40	+0,60%	10/11
China A50	Nov 2022	12.359,00	12.418,00	12.337,00	-2,0	-0,02%	11/11
S&P/ASX 200	Dic 2022	7.202,50	7.209,50	7.123,50	+50,5	+0,71%	11/11
Singapur MSCI	Nov 2022	289,60	290,02	284,45	+11,80	+4,25%	11/11
Nifty 50	Nov 2022	18.500,25	18.502,75	18.332,00	+407,60	+2,25%	11/11
Bank NIFTY	Nov 2022	42.295,00	42.438,70	42.053,40	+563,95	+1,35%	11/11
KO SPT 200	Sep 2022	324,45	324,80	321,50	+11,50	+3,67%	11/11
SGX FTSE Taiwan F	Nov 2022	1.256,75	1.257,75	1.241,50	+13,50	+1,09%	11/11
South Africa 40	Dic 2022	67.379,00	67.410,00	65.902,00	+2.657	+4,17%	11/11
TecDAX	Dic 2022	3.098,50	3.104,50	2.889,25	+23,50	+0,76%	11/11

Dataframe

# 1. Data Acquisition

## 1.3 Web Scrapping by an application programming interface (API)

AIM: Get all the information of the ETH/EUR in a dataframe by making a connection to the Binance platform.



Website

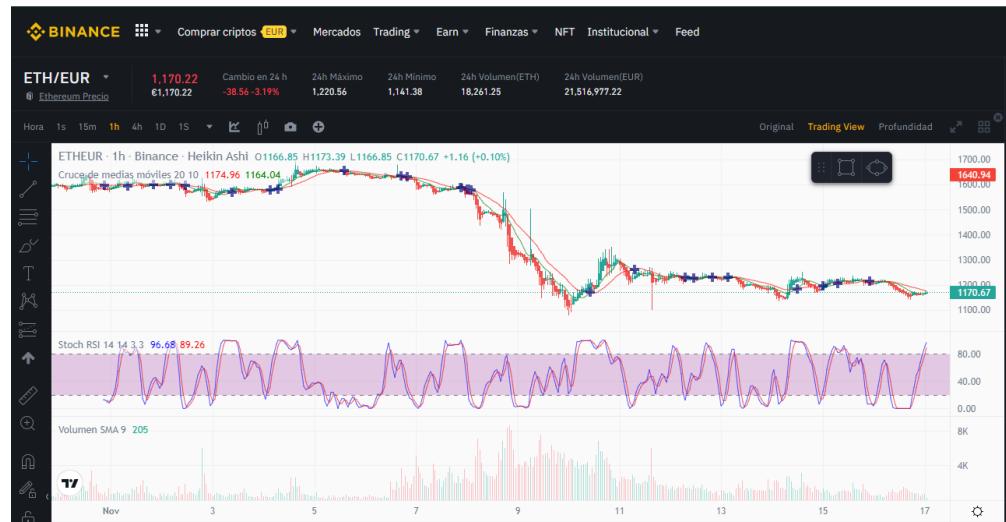


Plot of portion of the Dataframe

# 2. Evaluation of a Trading Strategy

## 2.1 Using the Stochastic indicator

AIM: To check if a strategy based just on the stochastic indicator is effective.



I have evaluate the strategy in ETH/EUR from 2022-10-27 20:00:00 to 2022-11-17 15:00:00  
~ 20 days 19:00:00  
I invested:900  
I gained:774.43  
so the benefit is :-125.57

So far it doesn't seem to be an effective strategy in 1H

# 3. Complementary applications

## 3.1 Screenshots

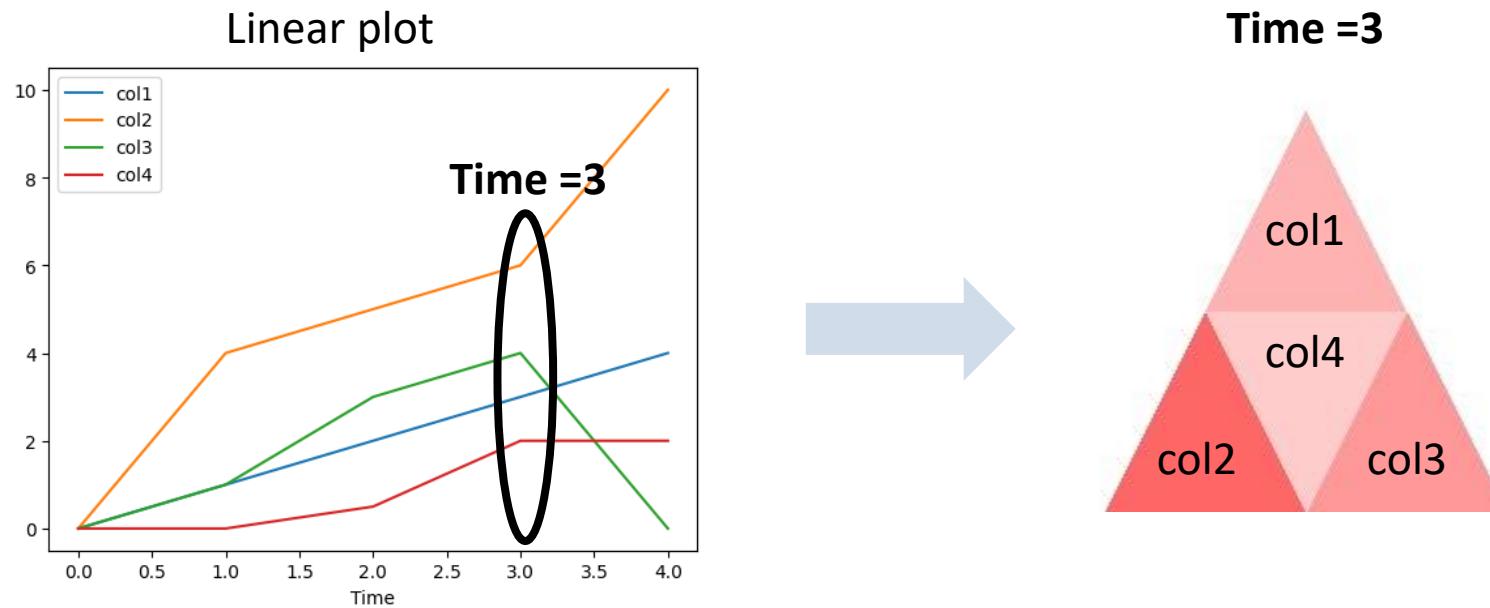
AIM: Take Screenshots during the trading session to facilitate a subsequent evaluation



# 3. Complementary applications

## 3.2 Custom data visualization

AIM: Represent the differences between samples (i. e. col1, col2,...) at a given time. The higher the value the higher the intensity.

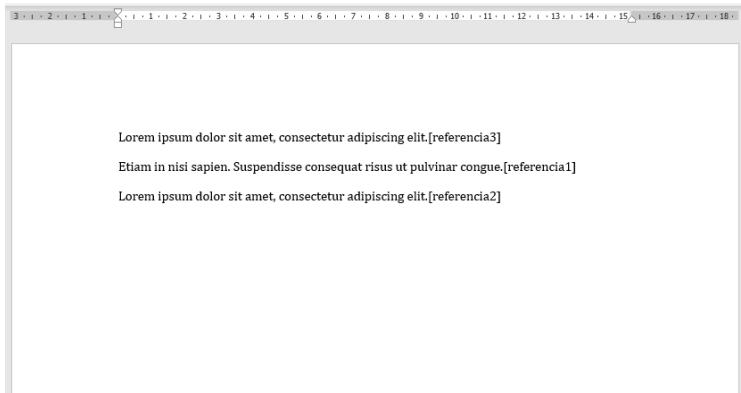


# 3. Complementary applications

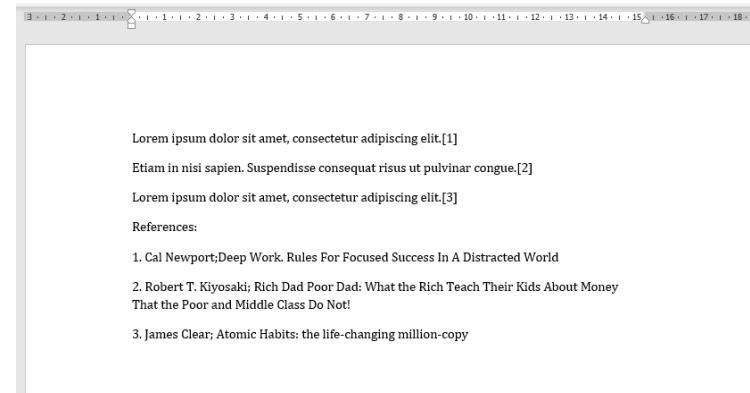
## 3.3 Word manipulation

AIM: Create a reference list in a word document.

Word before



Word after



# Portafolio

## 1 Data acquisition

### 1.1 Web Scrapping of Static webpages

PáginasAmarillas

The screenshot shows a search results page for 'asesoria' in León on the Páginas Amarillas website. The search bar at the top has 'asesoria' and 'leon' entered. Below the search bar, there's a banner with a woman holding a laptop and the text '¿Quieres que tus clientes te encuentren en este listado?'. The main content area displays three service cards:

- Asesoria en Leon (León) (189)**: A service with a yellow star rating, 20 years of experience, and a 'Pedir cita' button.
- Bufete & Gestión**: A service with a yellow star rating, 20 years of experience, and a 'Pedir cita' button.
- Asesoria Bs & Asesores**: A service with a yellow star rating, 15 years of experience, and a 'Pedir cita' button.

On the right side of the page, there are sidebar sections for 'Tipos de asesoría' (listing fiscal, civil, mercantil, etc.) and 'Búsquedas relacionadas con asesoria en León' (listing abogado asesoria derecho).

In [1]: `#Request and BeautifulSoup to extract the information from the url`

```
# pandas to create the df
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

In [2]: `url= "https://www.paginasamarillas.es/search/asesoria/all-ma/all-pr/all-is/leon/all-ba/all-pu/all-nc/1?wha`

```
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36'}
r = requests.get(url, headers=headers)
soup = BeautifulSoup(r.text, features = "xml")
```

In [3]: `#Check that I have the information of the website in the "soup". For example, I checked the title of the s`

```
print(soup.title)
```

```
<title>asesoria en Leon | Páginas Amarillas</title>
```

In [4]: `#after looking at the html code of the website, I identified that the desired information is under class "s`

```
box = soup.find_all("div", class_="servicios")
for i in range(0,3):
    print(box[i])
    print(" ")
```

```
<div class="servicios"><div class="web-row"><i class="fa icon-link"/><a class="web" data-omniclick="website" href="http://www.alvarez-canal.com?utm_campaign=paginasamarillas=paginasamarillas=referral" itemprop="url" rel="noopener noreferrer" target="_blank" title="Se abre en una nueva ventana"><span>Web</span></a></div></div>

<div class="servicios"><div class="web-row"><i class="fa icon-link"/><a class="web" data-omniclick="website" href="http://www.asemorca.com?utm_campaign=paginasamarillas=paginasamarillas=referral" itemprop="url" rel="noopener noreferrer" target="_blank" title="Se abre en una nueva ventana"><span>Web</span></a></div></div>

<div class="servicios"><div class="web-row"><i class="fa icon-link"/><a class="web" data-omniclick="website" href="http://www.bufeteygestion.es?utm_campaign=paginasamarillas=paginasamarillas=referral" itemprop="url" rel="noopener noreferrer" target="_blank" title="Se abre en una nueva ventana"><span>Web</span></a></div></div>
```

In [5]: *#The information is saved in a list to create later on the dataframe*

```
webs=[]
for i in range(0,len(box)):
    try:
        webs.append(box[i].find("a").attrs["href"])
    except:
        webs.append("no tiene")

print("The first three examples out of {}:".format(str(len(box))))
webs[0:3]
```

The first three examples out of 30:

Out[5]: ['http://www.alvarez-canal.com?utm\_campaign=paginasamarillas=paginasamarillas=referral',  
 'http://www.asemorca.com?utm\_campaign=paginasamarillas=paginasamarillas=referral',  
 'http://www.bufeteygestion.es?utm\_campaign=paginasamarillas=paginasamarillas=referral']

In [6]: *#Creation of the df*

```
df = pd.DataFrame({"Web":webs})
df
```

Out[6]:

**Web**

- 0 http://www.alvarez-canal.com?utm\_campaign=pagi...
- 1 http://www.asemorca.com?utm\_campaign=paginasam...
- 2 http://www.bufeteygestion.es?utm\_campaign=pagi...
- 3 http://www.asesoriabsasesores.com?utm\_campaign...
- 4 http://www.asesoriasantiagochamorro.es?utm\_camp...
- 5 http://www.etmasesores.com?utm\_campaign=pagina...
- 6 http://gestorialegio.net?utm\_campaign=paginas...
- 7 http://www.alvarezreguera.com/?utm\_campaign=pa...
- 8 http://www.casadoyalegreasesores.es?utm\_campa...
- 9 http://www.asesoriaperezgestion.es?utm\_campaig...
- 10 http://www.mendezeconomistas.com?utm\_campaign=...
- 11 http://www.condesa10.es?utm\_campaign=paginasam...
- 12 http://www.adecocomunidades.com?utm\_campaign=p...
- 13 http://www.abarroso.com?utm\_campaign=paginasam...
- 14 no tiene
- 15 http://www.abogadaineslopezdelacalzada.com?utm...
- 16 http://www.amadeoasesores.com?utm\_campaign=pag...
- 17 no tiene
- 18 http://www.gesbierzo.es?utm\_campaign=paginasam...
- 19 http://www.asesorascistierna.com?utm\_campaign...
- 20 http://www.isolinaarias.es?utm\_campaign=pagina...
- 21 http://www.charollamera.com?utm\_campaign=pagin...
- 22 http://www.merinoyasociados.com?utm\_campaign=p...
- 23 http://www.bufeteprida.es?utm\_campaign=paginas...
- 24 no tiene
- 25 http://www.gestoriasenleonhermes.com?utm\_campa...
- 26 https://www.aranzazugutierrezblanca.com
- 27 http://www.asesorianocedo.es?utm\_campaign=pagi...
- 28 http://www.despachotrivium.com/?utm\_campaign=p...
- 29 http://www.manuellosadaabogados.com?utm\_campai...

## 1.2 Web Scrapping of dynamic webpages (rendering)

Investing.com Buscar en esta web... Iniciar sesión / Registrarse gratis

Mercados ▾ Mi cartera Criptomonedas Noticias InvestingPro Análisis Gráficos Técnico Brokers Herramientas Academia Más ▾

Índices IBEX 35 Futuros de índices Principales índices Índices mundiales Índices CFDs I.G. Bolsa Madrid Índices españoles

Previa Ejecución Término Descargar datos

Índice: Mes: Último Máximo Mínimo Var.: % Var.: Hora:

US 30	33.756,50	33.971,50	33.397,50	+41,1	+0,12%	11/11
US 500	3.991,90	4.001,60	3.944,50	+35,5	+0,90%	11/11
US Tech 100	11.802,20	11.841,20	11.539,40	+156,2	+1,69%	11/11
US 2000	1.880,60	1.899,30	1.863,20	+13,9	+0,75%	11/11
\$&P 500 VIX	Nov 2022	24,05	25,21	-2,83	-0,58 -2,35%	11/11
DAX	Dic 2022	14.326,00	14.351,00	14.167,00	+153,0 +1,08%	11/11
CAC 40	Nov 2022	6.633,00	6.642,50	6.573,00	+75,5 +1,15%	11/11
FTSE 100	Dic 2022	7.354,20	7.434,80	7.321,00	+80,8 -0,55%	11/11
Euro Stoxx 50	Dic 2022	3.889,00	3.895,00	3.844,00	+42 +1,09%	11/11
Italia 40	Dic 2022	24.490,00	24.595,00	24.285,00	+185,00 +0,76%	11/11
Suiza 20	Dic 2022	11.146,00	11.241,00	11.091,50	+43,0 +0,35%	11/11
IBEX 35	Nov 2022	8.133,00	8.204,00	8.072,00	+18,8 -0,23%	11/11
RTS		114.395,00	114.845,00	109.040,00	+4.255 +3,90%	11/11
WIG20	Dic 2022	1.700,00	1.706,00	1.629,00	+42,00 +2,53%	10/11
AEX	Nov 2022	707,17	709,25	699,52	+5,71 +0,81%	11/11
IBovespa		113.615,00	114.265,00	110.180,00	+2.615 +2,38%	11/11
Nikkel 225	Dic 2022	26.090,00	26.328,00	27.972,50	-180,0 -0,64%	11/11
TOPIX	Dic 2022	1.969,00	1.982,50	1.961,00	+13,50 -0,68%	11/11
Hang Seng	Nov 2022	17.561,00	17.825,00	17.339,00	+201,0 +1,16%	11/11
China H-Shares	Nov 2022	5.942,00	5.978,00	5.877,00	+56,0 +0,95%	11/11
CSI 300		3.696,80	3.712,80	3.671,60	+22,40 -0,50%	10/11
China A50	Nov 2022	12.359,00	12.418,00	12.337,00	-2,0 -0,02%	11/11
S&P/A SX 200	Dic 2022	7.202,50	7.209,50	7.123,50	+50,5 +0,71%	11/11
Singapur MSCI	Nov 2022	289,60	290,02	284,45	+11,80 +4,25%	11/11
Nifty 50	Nov 2022	18.500,25	18.502,75	18.332,00	+407,90 +2,25%	11/11
Bank NIFTY	Nov 2022	42.295,00	42.438,70	42.053,40	+563,95 +1,35%	11/11
KOSPI 200	Sep 2022	324,45	324,80	321,50	+11,50 +3,67%	11/11
SGX FTSE Taiwan F	Nov 2022	1.256,75	1.257,75	1.241,50	+13,50 +1,09%	11/11
South Africa 40	Dic 2022	67.379,00	67.410,00	65.902,00	+2.857 +4,17%	11/11
TecDAX	Dic 2022	3.098,50	3.104,50	2.889,25	+23,50 +0,76%	11/11



#### EMPIECE A OPERAR

Avísate ¡Están a la venta las cuentas de los pequeños inversores que pierden dinero cuando negocian con operaciones con alavancamiento con este corredor!

```
In [7]: # Selenium to extract information dynamically from the website
# time to allow the website to load completely before working on it
# pandas to create the df
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
import time
import pandas as pd

# Open the website using Chrome
options = webdriver.ChromeOptions()
options.add_argument('--start-maximized')
options.add_argument('--disable-extensions')

driver_path = r'C:\...\...\...\...\...\Chromedriver'

driver = webdriver.Chrome(driver_path, options=options)

time.sleep(1)
# Inicializamos el navegador
driver.get('https://es.investing.com/indices/indices-futures')
```

```
In [8]: #Remove pop up by clicking using the XPATH
WebDriverWait(driver, 5)\n    .until(EC.element_to_be_clickable((By.XPATH,\n                                         "/html/body/div[2]/div[2]/div/div/div[2]/div/div/button")))\n    .click()
```

```
In [9]: #Save the data of the table
texto_columnas = driver.find_element_by_xpath('/html/body/div[1]/div/div/div[2]/main/div[3]/div/table')
texto_columnas = texto_columnas.text
```

```
In [10]: #Check how the data is saved:
```

```
datos=texto_columnas.split("\n")
datos[0:10]
```

```
Out[10]: ['Índice',
 'Mes',
 'Último',
 'Máximo',
 'Mínimo',
 'Var.',
 '% Var.',
 'Hora',
 'US 30',
 '33.593,50 33.706,80 33.445,80 +0,6 0,00% 16/11']
```

```
In [11]: #Cleaning the data I:
    #Extract columns name
tcol=[]
for i in range(0,8):
    tcol.append(datos[i])

    #Extract index names
indices=[]
for i in range(8,len(datos),2):
    indices.append(datos[i])

    #Extract the remaining information
data=[]
for i in range(9,len(datos),2):
    data.append(datos[i])
print(data)
```

```
['33.593,50 33.706,80 33.445,80 +0,6 0,00% 16/11', '3.965,30 4.007,40 3.954,30 -26,4 -0,66% 16/11', '11.7
31,60 11.923,50 11.670,40 -139,5 -1,18% 16/11', '1.852,80 1.896,30 1.847,00 -35,5 -1,88% 16/11', 'Dic 202
2 26,92 27,51 26,85 -0,37 -1,35% 16/11', 'Dic 2022 14.267,40 14.402,00 14.200,50 -109,6 -0,76% 16/11', 'D
ic 2022 6.596,20 6.655,50 6.585,20 -43,8 -0,66% 16/11', 'Dic 2022 7.337,20 7.396,80 7.305,50 -25,8 -0,35%
16/11', 'Dic 2022 3.878,00 3.919,00 3.858,00 -27 -0,69% 16/11', 'Dic 2022 24.410,00 24.645,00 24.390,00 -
179,00 -0,73% 16/11', 'Dic 2022 10.912,00 11.031,00 10.908,50 -128,0 -1,16% 16/11', 'Dic 2022 8.086,00 8.
265,00 8.015,00 -77,0 -0,94% 16/11', '115.105,00 115.625,00 112.785,00 +1.325 +1,16% 16/11', 'Dic 2022 1.
752,00 1.775,00 1.741,00 -25,00 -1,41% 16/11', 'Dic 2022 707,42 717,75 707,17 -8,88 -1,24% 16/11', '110.5
00,00 114.315,00 110.365,00 -5.000 -4,33% 16/11', 'Dic 2022 27.943,00 28.108,00 27.732,50 -87,0 -0,31% 1
6/11', 'Dic 2022 1.959,50 1.969,00 1.957,00 -6,00 -0,31% 16/11', 'Nov 2022 18.063,00 18.319,00 18.014,00
-261,0 -1,42% 16/11', 'Nov 2022 6.153,00 6.252,00 6.127,00 -112,0 -1,79% 16/11', '3.866,40 3.879,80 3.77
9,80 +68,80 +1,81% 15/11', 'Nov 2022 12.403,00 12.472,00 12.373,00 -69,0 -0,55% 16/11', 'Dic 2022 7.116,5
0 7.158,50 7.108,00 -36,5 -0,51% 16/11', 'Nov 2022 299,20 302,10 292,45 +7,05 +2,41% 16/11', 'Nov 2022 1
8.413,75 18.499,50 18.385,80 -48,30 -0,26% 16/11', 'Nov 2022 42.563,85 42.666,00 42.421,00 +80,15 +0,19%
16/11', 'Sep 2022 322,70 323,60 319,10 +0,10 +0,03% 16/11', 'Nov 2022 1.277,50 1.289,25 1.273,75 -5,25 -
0,41% 16/11', 'Dic 2022 66.626,00 67.420,00 66.507,00 -330 -0,49% 16/11', 'Dic 2022 3.079,00 3.132,75 3.0
75,00 -51,00 -1,63% 16/11']
```

```
In [ ]: #Cleaning data II: working with the "remaining information"
mes=[]
ultimo=[]
maximo=[]
minimo=[]
var=[]
pvar=[]
hora=[]

for i in range(0,len(data)):
    hora.append(data[i].split(" ")[-1])
    pvar.append(data[i].split(" ")[-2])
    var.append(data[i].split(" ")[-3])
    minimo.append(data[i].split(" ")[-4])
    maximo.append(data[i].split(" ")[-5])
    ultimo.append(data[i].split(" ")[-6])
    try:
        mes.append(data[i].split(" ")[-7])
    except:
        mes.append("NaN")
```

```
In [13]: #Save all the data in a dataframe
df=pd.DataFrame({"Índice":indices,"Mes":mes,"Último":ultimo,"Máximo":maximo,"Mínimo":minimo,"Var":var,"% \\\\"
```

```
df=df.set_index("Indice")
df
```

Out[13]:

	<b>Mes</b>	<b>Ultimo</b>	<b>Maximo</b>	<b>Minimo</b>	<b>Var</b>	<b>% Var</b>	<b>Hora</b>
<b>Indice</b>							
<b>US 30</b>	NaN	33.593,50	33.706,80	33.445,80	+0,6	0,00%	16/11
<b>US 500</b>	NaN	3.965,30	4.007,40	3.954,30	-26,4	-0,66%	16/11
<b>US Tech 100</b>	NaN	11.731,60	11.923,50	11.670,40	-139,5	-1,18%	16/11
<b>US 2000</b>	NaN	1.852,80	1.896,30	1.847,00	-35,5	-1,88%	16/11
<b>S&amp;P 500 VIX</b>	2022	26,92	27,51	26,85	-0,37	-1,35%	16/11
<b>DAX</b>	2022	14.267,40	14.402,00	14.200,50	-109,6	-0,76%	16/11
<b>CAC 40</b>	2022	6.596,20	6.655,50	6.585,20	-43,8	-0,66%	16/11
<b>FTSE 100</b>	2022	7.337,20	7.396,80	7.305,50	-25,8	-0,35%	16/11
<b>Euro Stoxx 50</b>	2022	3.878,00	3.919,00	3.858,00	-27	-0,69%	16/11
<b>Italia 40</b>	2022	24.410,00	24.645,00	24.390,00	-179,00	-0,73%	16/11
<b>Suiza 20</b>	2022	10.912,00	11.031,00	10.908,50	-128,0	-1,16%	16/11
<b>IBEX 35</b>	2022	8.086,00	8.265,00	8.015,00	-77,0	-0,94%	16/11
<b>RTS</b>	NaN	115.105,00	115.625,00	112.785,00	+1.325	+1,16%	16/11
<b>WIG20</b>	2022	1.752,00	1.775,00	1.741,00	-25,00	-1,41%	16/11
<b>AEX</b>	2022	707,42	717,75	707,17	-8,88	-1,24%	16/11
<b>iBovespa</b>	NaN	110.500,00	114.315,00	110.365,00	-5.000	-4,33%	16/11
<b>Nikkei 225</b>	2022	27.943,00	28.108,00	27.732,50	-87,0	-0,31%	16/11
<b>TOPIX</b>	2022	1.959,50	1.969,00	1.957,00	-6,00	-0,31%	16/11
<b>Hang Seng</b>	2022	18.063,00	18.319,00	18.014,00	-261,0	-1,42%	16/11
<b>China H-Shares</b>	2022	6.153,00	6.252,00	6.127,00	-112,0	-1,79%	16/11
<b>CSI 300</b>	NaN	3.866,40	3.879,80	3.779,80	+68,80	+1,81%	15/11
<b>China A50</b>	2022	12.403,00	12.472,00	12.373,00	-69,0	-0,55%	16/11
<b>S&amp;P/ASX 200</b>	2022	7.116,50	7.158,50	7.108,00	-36,5	-0,51%	16/11
<b>Singapur MSCI</b>	2022	299,20	302,10	292,45	+7,05	+2,41%	16/11
<b>Nifty 50</b>	2022	18.413,75	18.499,50	18.385,80	-48,30	-0,26%	16/11
<b>Bank NIFTY</b>	2022	42.563,85	42.666,00	42.421,00	+80,15	+0,19%	16/11
<b>KOSPI 200</b>	2022	322,70	323,60	319,10	+0,10	+0,03%	16/11
<b>SGX FTSE Taiwan F</b>	2022	1.277,50	1.289,25	1.273,75	-5,25	-0,41%	16/11
<b>South Africa 40</b>	2022	66.626,00	67.420,00	66.507,00	-330	-0,49%	16/11
<b>TecDAX</b>	2022	3.079,00	3.132,75	3.075,00	-51,00	-1,63%	16/11

## 1.3 API



```
In [2]: api_key = '.....'
api_secret = '.....'
```

```
In [3]: from binance.client import Client
client = Client(api_key, api_secret)

#Create a function to transform the raw data from the API to a dataframe
def listToDF(candles):
    import pandas as pd

    quote=[]
    base=[]
    time=[]
    trades=[]
    quotevol=[]
    vol=[]
    close=[]
    low=[]
    high=[]
    openn=[]

    #columns= ["time", "open", "high", "Low", "close", "volumen", "qouteVolume", "trades", "base", "quote"]
    #0,1,2,3,4,5,7,8,9,10

    for i in range(0,len(candles)):
        time.append(candles[i][0]/1000)
        openn.append(candles[i][1])
        high.append(candles[i][2])
        low.append(candles[i][3])
        close.append(candles[i][4])
        vol.append(candles[i][5])
        quotevol.append(candles[i][7])
        trades.append(candles[i][8])
        base.append(candles[i][9])
        quote.append(candles[i][10])

    df=pd.DataFrame({"open":openn,"high":high,"low":low,"close":close,"volume":vol,"quoteVolume":quotevol,
    df.index.name="time"

    df["open"] = df["open"].astype("float")
    df["high"] = df["high"].astype("float")
    df["low"] = df["low"].astype("float")
    df["close"] = df["close"].astype("float")
    df["volume"] = df["volume"].astype("float")
    df["quoteVolume"] = df["quoteVolume"].astype("float")
    df["base"] = df["base"].astype("float")
```

```

df["quote"] = df["quote"].astype("float")

#convertir timestamp to datetime
    #copy index to a new column and then make the transformation

df["time"] = df.index
df["time"] = pd.to_datetime(df["time"], unit="s")

    #o cambio el index directamente
df.index = pd.to_datetime(df.index, unit="s")

return df

```

In [4]: `#Get the data of ETHEUR in the interval time of 1H  
rawdata = client.get_klines(symbol='ETHEUR', interval=Client.KLINE_INTERVAL_1HOUR)  
rawdata[0:2]`

Out[4]: [[1666900800000,  
'1567.93000000',  
'1568.77000000',  
'1514.00000000',  
'1533.90000000',  
'2867.18830000',  
1666904399999,  
'4403521.37063300',  
7670,  
'1338.48190000',  
'2055782.19582800',  
'0'],  
[1666904400000,  
'1533.59000000',  
'1535.48000000',  
'1521.18000000',  
'1530.17000000',  
'570.43720000',  
1666907999999,  
'870646.10488800',  
1458,  
'252.98240000',  
'385898.34447400',  
'0']]

In [5]: `#Transform the raw data to a df  
df=listToDF(rawdata)  
df`

Out[5]:

	open	high	low	close	volume	quoteVolume	trades	base	quote	time
time										
2022-10-27 20:00:00	1567.93	1568.77	1514.00	1533.90	2867.1883	4.403521e+06	7670	1338.4819	2.055782e+06	2022-10-27 20:00:00
2022-10-27 21:00:00	1533.59	1535.48	1521.18	1530.17	570.4372	8.706461e+05	1458	252.9824	3.858983e+05	2022-10-27 21:00:00
2022-10-27 22:00:00	1529.75	1534.01	1515.46	1522.82	872.8997	1.331909e+06	2316	383.4239	5.848822e+05	2022-10-27 22:00:00
2022-10-27 23:00:00	1522.70	1524.32	1507.88	1519.47	728.8405	1.106116e+06	1932	366.1535	5.557927e+05	2022-10-27 23:00:00
2022-10-28 00:00:00	1519.91	1529.32	1506.93	1524.86	985.2546	1.496763e+06	2645	456.1557	6.929404e+05	2022-10-28 00:00:00
...	...	...	...	...	...	...	...	...	...	...
2022-11-17 11:00:00	1152.47	1164.16	1152.20	1163.00	849.9868	9.854283e+05	2335	460.4949	5.336514e+05	2022-11-17 11:00:00
2022-11-17 12:00:00	1163.16	1166.50	1157.29	1165.46	759.3608	8.812381e+05	2127	436.0648	5.060576e+05	2022-11-17 12:00:00
2022-11-17 13:00:00	1165.52	1165.52	1148.00	1154.66	1443.9228	1.668009e+06	3333	725.1911	8.376562e+05	2022-11-17 13:00:00
2022-11-17 14:00:00	1154.68	1158.41	1149.51	1153.88	930.2531	1.074135e+06	2339	473.7928	5.471035e+05	2022-11-17 14:00:00
2022-11-17 15:00:00	1153.89	1170.07	1150.34	1169.29	325.9937	3.773809e+05	822	155.7790	1.803746e+05	2022-11-17 15:00:00

500 rows × 10 columns

In [7]:

```
#plot a region of the data to further verify the data
# import modules
import matplotlib.pyplot as plt
from mpl_finance import candlestick_ohlc
import matplotlib.dates as mpl_dates

#transform time format
df["date"] = df["time"].apply(mpl_dates.date2num)

#create the region
df2 = df.loc["2022-10-29 16:00:00":"2022-10-30 16:00:00"]

#make the plot
fig, ax = plt.subplots()
candlestick_ohlc(ax, df2[["date", "open", "high", "low", "close"]].values, width= 0.01, colorup = 'green', colordown = 'red')
ax.grid(True)
date_format=mpl_dates.DateFormatter("%d-%b-%Y %H:%M")
ax.xaxis.set_major_formatter(date_format)
fig.autofmt_xdate()
fig.tight_layout()
plt.show();
```



## 2 Evaluation of a Trading Strategy

```
In [8]: #### Import data

rawdata = client.get_klines(symbol='ETHEUR', interval=Client.KLINE_INTERVAL_1HOUR)
df1H=listToDF(rawdata)

df1H[ "date" ]=df1H[ "time" ].apply(mpl_dates.date2num)

### make the plot
import matplotlib.pyplot as plt

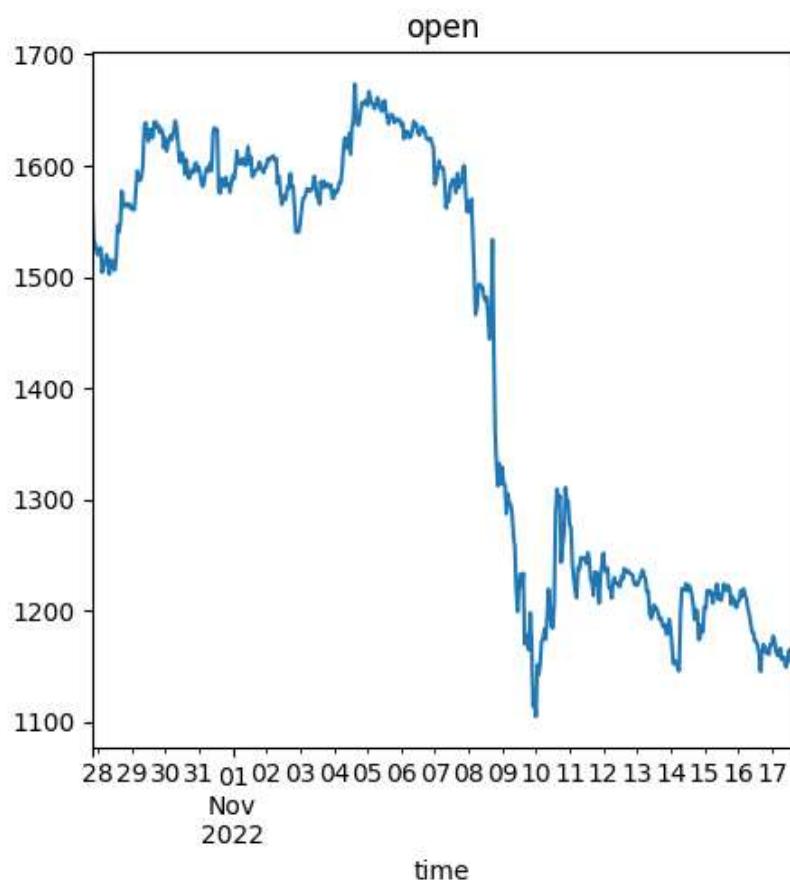
plt.figure(figsize=(5,5))
df1H[ "open" ].plot()
plt.title("open")

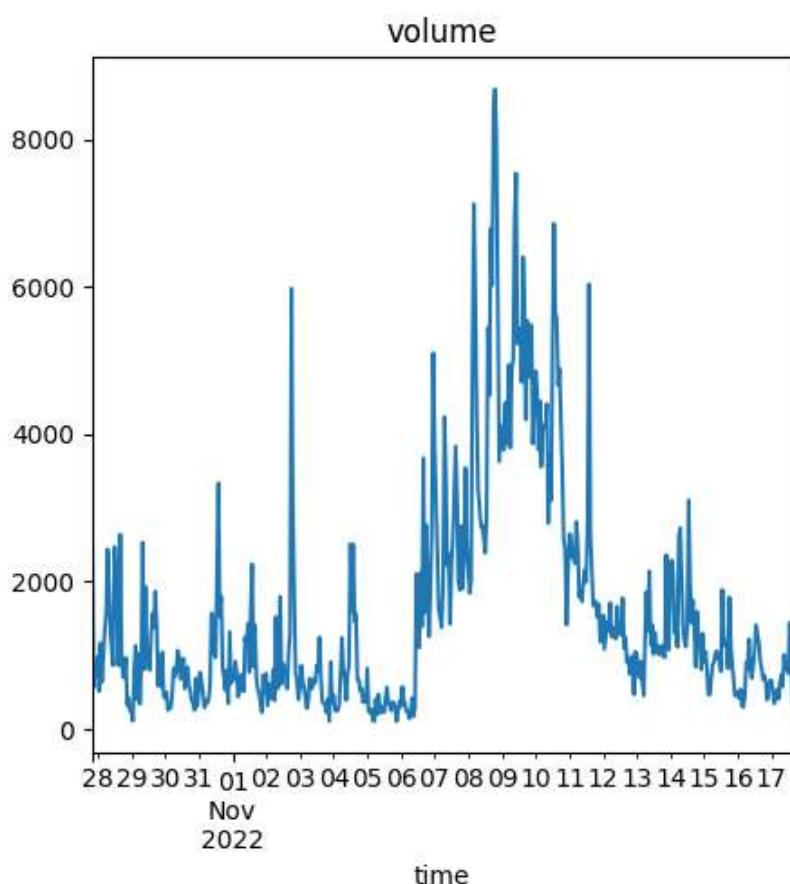
plt.figure(figsize=(5,5))
df1H[ "volume" ].plot()
plt.title("volume");
df1H
```

Out[8]:

	open	high	low	close	volume	quoteVolume	trades	base	quote	time	da
time											
2022-10-27 20:00:00	1567.93	1568.77	1514.00	1533.90	2867.1883	4.403521e+06	7670	1338.4819	2.055782e+06	2022-10-27 20:00:00	19292.8333
2022-10-27 21:00:00	1533.59	1535.48	1521.18	1530.17	570.4372	8.706461e+05	1458	252.9824	3.858983e+05	2022-10-27 21:00:00	19292.8750
2022-10-27 22:00:00	1529.75	1534.01	1515.46	1522.82	872.8997	1.331909e+06	2316	383.4239	5.848822e+05	2022-10-27 22:00:00	19292.9166
2022-10-27 23:00:00	1522.70	1524.32	1507.88	1519.47	728.8405	1.106116e+06	1932	366.1535	5.557927e+05	2022-10-27 23:00:00	19292.9583
2022-10-28 00:00:00	1519.91	1529.32	1506.93	1524.86	985.2546	1.496763e+06	2645	456.1557	6.929404e+05	2022-10-28 00:00:00	19293.0000
...	...	...	...	...	...	...	...	...	...	...	...
2022-11-17 11:00:00	1152.47	1164.16	1152.20	1163.00	849.9868	9.854283e+05	2335	460.4949	5.336514e+05	2022-11-17 11:00:00	19313.4583
2022-11-17 12:00:00	1163.16	1166.50	1157.29	1165.46	759.3608	8.812381e+05	2127	436.0648	5.060576e+05	2022-11-17 12:00:00	19313.5000
2022-11-17 13:00:00	1165.52	1165.52	1148.00	1154.66	1443.9228	1.668009e+06	3333	725.1911	8.376562e+05	2022-11-17 13:00:00	19313.5416
2022-11-17 14:00:00	1154.68	1158.41	1149.51	1153.88	930.2531	1.074135e+06	2339	473.7928	5.471035e+05	2022-11-17 14:00:00	19313.5833
2022-11-17 15:00:00	1153.89	1172.47	1150.34	1172.47	335.8638	3.889416e+05	858	164.7747	1.909128e+05	2022-11-17 15:00:00	19313.6250

500 rows × 11 columns

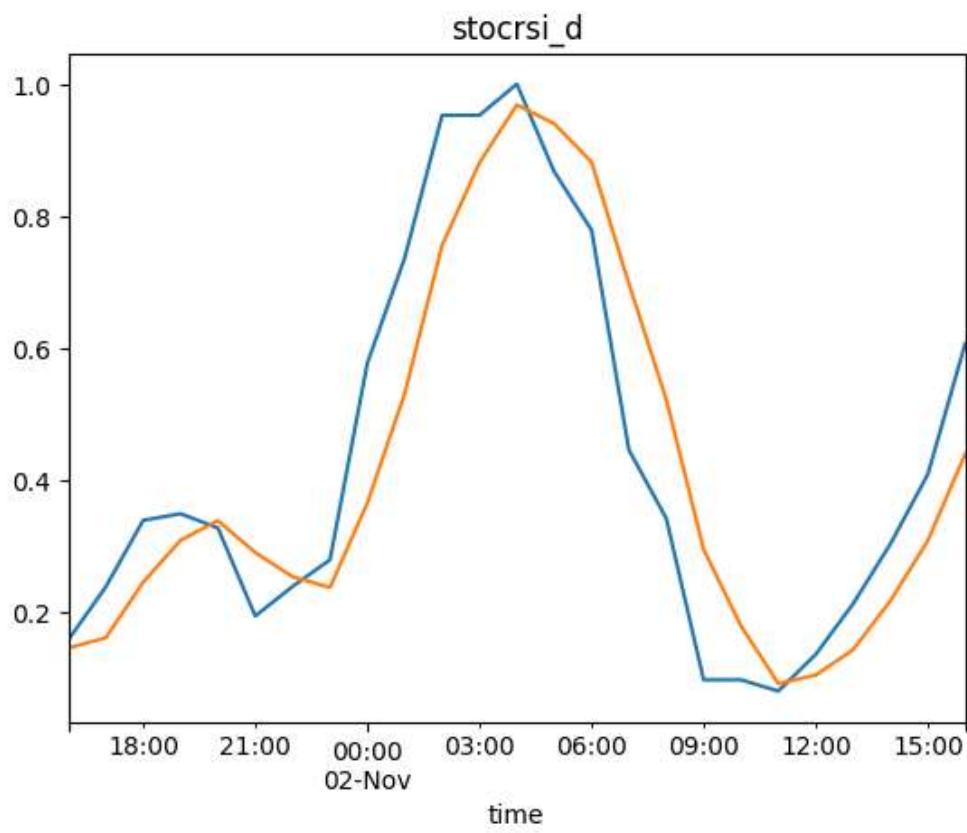
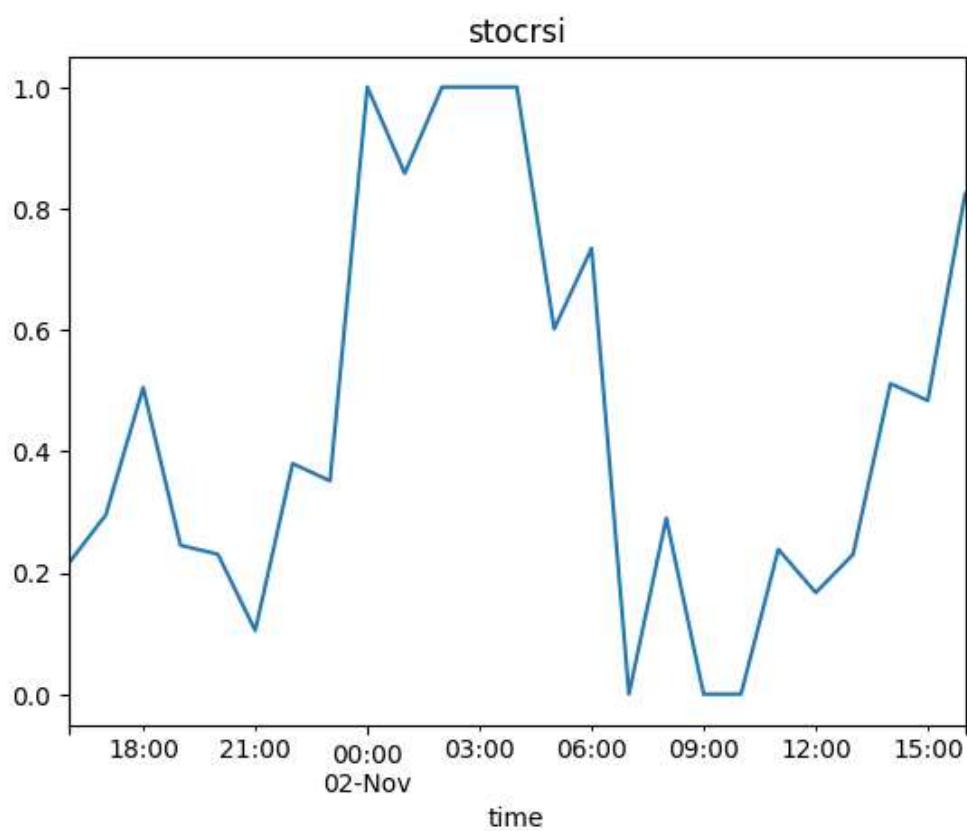




```
In [9]: ### Calculate Stochastic RSI using thecnical analysis Library
import ta
#Creo las variables
df1H["stocrsi"] = ta.momentum.StochRSIIndicator(df1H["close"], window=14).stochrsi()
##The %D Line is the "slow" stochastic or "signal" Line and is a moving average of %K.
df1H["stocrsi_d"] = ta.momentum.StochRSIIndicator(df1H["close"], window=14).stochrsi_d()
##The %K Line is the "fast" stochastic and refers to the number of RSI periods used in the stochastic calculation
df1H["stocrsi_k"] = ta.momentum.StochRSIIndicator(df1H["close"], window=14).stochrsi_k()

#For facilitate the representation, I plot just a region of the whole df
df2=df1H.loc["2022-11-01 16:00:00":"2022-11-02 16:00:00"]
plt.title("stocrsi")
df2["stocrsi"].plot()

plt.figure()
plt.title("stocrsi_d")
df2["stocrsi_k"].plot()
df2["stocrsi_d"].plot();
```



## Strategy

The Stochastic indicator is a momentum indicator that shows how strong or weak the current trend is by identifying overbought and oversold market conditions within a trend.

The strategy that we are going to evaluate would be based on buying when the market is oversold (i. e. below 0.3) and selling when the market is overbought (i. e. higher than 0.7)

```
In [11]: #Create a new variable to represent this Strategy. It would be 1 when buying and -1 when selling.
```

```
df1H["strategy1"] = 0
```

```
#buying: df["entrada"] = 1
```

```

##conditions:
#K > D
cnd1=(df1H["stocrsi_k"].shift(-1)<df1H["stocrsi_d"].shift(-1))&(df1H["stocrsi_k"] > df1H["stocrsi_d"])
#RSI <0.3
cnd2=df1H["stocrsi_k"]<0.3

df1H.loc[(cnd1)&(cnd2),"strategy1"]=1

#how many times do I buy?:
entradas=len(df1H[df1H["strategy1"]==1])
print("hago {} entradas".format(str(entradas)))

#selling df["entrada"]=-1
#K < D
cnd3=(df1H["stocrsi_k"].shift(-1)>df1H["stocrsi_d"].shift(-1))&(df1H["stocrsi_k"] < df1H["stocrsi_d"])
#RSI >0.7
cnd4=df1H["stocrsi_k"]>0.7
df1H.loc[(cnd3)&(cnd4),"strategy1"]=-1

salidas=len(df1H[df1H["strategy1"]== -1])
print("hago {} salidas".format(str(salidas)))

operaciones = df1H[df1H["strategy1"]!=0][["open","strategy1"]]

#me quedo sólo con una salida (la primera)
#entre en todos los 1 pero sólo salgo en el -1
#primero borro las líneas con -1 que anteriormente tengan -1 (pondré -2)
cnd5=(operaciones["strategy1"].shift(-1)==-1)&(operaciones["strategy1"]== -1)
operaciones.loc[(cnd5),"strategy1"]=-2
operaciones[operaciones["strategy1"]!= -2]

operaciones = operaciones[operaciones["strategy1"]!= -2]
operaciones

```

hago 9 entradas  
hago 7 salidas

Out[11]: **open strategy1**

	time	
2022-10-30 16:00:00	1593.48	1
2022-10-31 08:00:00	1602.72	-1
2022-10-31 20:00:00	1582.76	1
2022-11-03 07:00:00	1576.81	-1
2022-11-05 12:00:00	1658.51	1
2022-11-05 17:00:00	1646.19	1
2022-11-05 22:00:00	1641.44	1
2022-11-07 18:00:00	1583.67	-1
2022-11-09 01:00:00	1314.25	1
2022-11-09 08:00:00	1267.39	1
2022-11-10 12:00:00	1184.49	-1
2022-11-13 19:00:00	1185.44	1
2022-11-15 07:00:00	1214.71	-1
2022-11-16 14:00:00	1167.92	1

In [12]: #Check visually if I really should be buying on the 2022-10-30 16:00:00  
df2=df1H.loc["2022-10-29 06:00:00":"2022-10-31 16:00:00"]

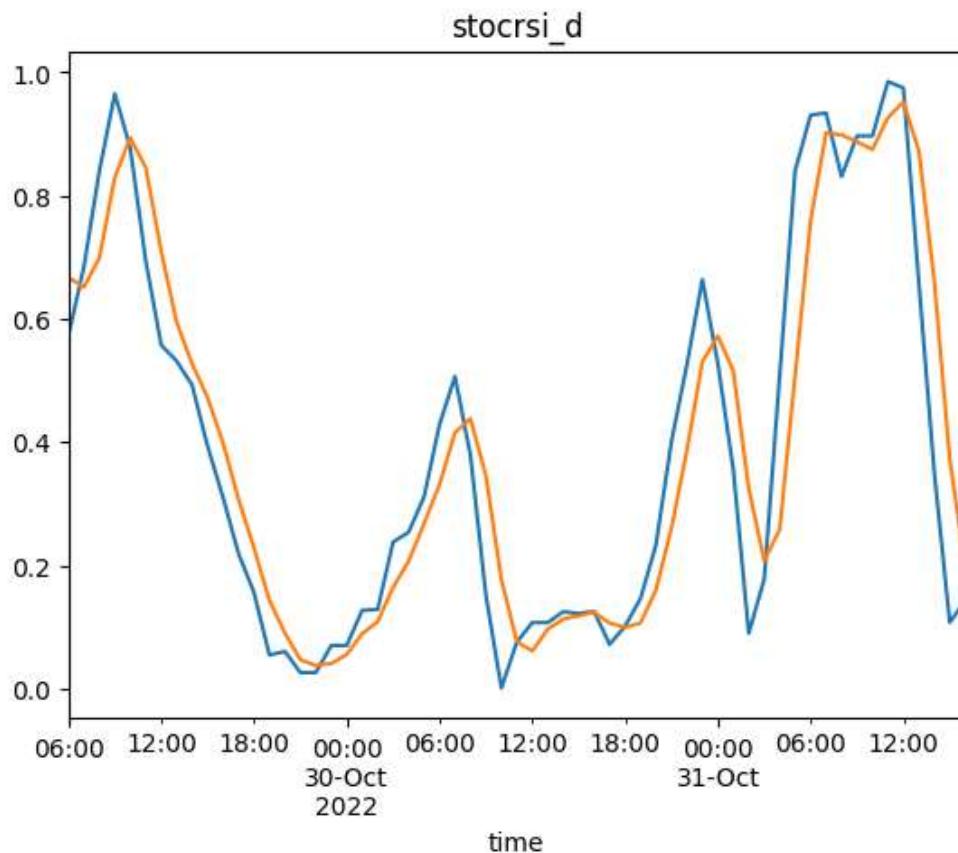
plt.title("stocrsi\_d")
df2["stocrsi\_k"].plot()
df2["stocrsi\_d"].plot()

fig,ax = plt.subplots()

```

candlestick_ohlc(ax,df2[["date","open","high","low","close"]].values,width= 0.01,colorup = 'green', colordown='red',ax.grid(True)
date_format=mpl_dates.DateFormatter ("%d_%H:%M")
ax.xaxis.set_major_formatter(date_format)
fig.autofmt_xdate()
fig.tight_layout()
plt.show();

```



In [16]: `operaciones["dinero"] = 0`

```

#To validate the strategy, I will say that everytime I buy 100EUR
operaciones.loc[operaciones["strategy1"]==1,"dinero"]+=100

```

```

#ETH that I bought at that time

```

```

ETHinvertido=[]
#ALL the ETH that I have in my wallet at that time
ETHacumulado=[]
#EUR that I obtained when I sell all the ETH at that time
Eretorno=[]
#create a variable to save all the ETH
ETHwallet=0

for i in range (0,len(operaciones.strategy1)):
    #
    if operaciones.strategy1[i] ==1:
        inversion=operaciones.dinero[i]/operaciones.open[i]
        ETHinvertido.append(inversion)
        ETHwallet+=inversion
        ETHacumulado.append(ETHwallet)
        Eretorno.append(0)
    else:
        retorno=ETHwallet*operaciones.open[i]
        Eretorno.append(retorno)
        inversion=0
        ETHinvertido.append(inversion)
        ETHwallet=inversion
        ETHacumulado.append(ETHwallet)

operaciones["ETHinvertido"] = ETHinvertido
operaciones["ETHacumulado"] = ETHacumulado
operaciones["Eretorno"] = Eretorno

operaciones

```

Out[16]:

	open	strategy1	dinero	ETHinvertido	ETHacumulado	Eretorno
time						
2022-10-30 16:00:00	1593.48	1	100	0.062756	0.062756	0.000000
2022-10-31 08:00:00	1602.72	-1	0	0.000000	0.000000	100.579863
2022-10-31 20:00:00	1582.76	1	100	0.063181	0.063181	0.000000
2022-11-03 07:00:00	1576.81	-1	0	0.000000	0.000000	99.624074
2022-11-05 12:00:00	1658.51	1	100	0.060295	0.060295	0.000000
2022-11-05 17:00:00	1646.19	1	100	0.060746	0.121041	0.000000
2022-11-05 22:00:00	1641.44	1	100	0.060922	0.181964	0.000000
2022-11-07 18:00:00	1583.67	-1	0	0.000000	0.000000	288.170185
2022-11-09 01:00:00	1314.25	1	100	0.076089	0.076089	0.000000
2022-11-09 08:00:00	1267.39	1	100	0.078902	0.154991	0.000000
2022-11-10 12:00:00	1184.49	-1	0	0.000000	0.000000	183.585687
2022-11-13 19:00:00	1185.44	1	100	0.084357	0.084357	0.000000
2022-11-15 07:00:00	1214.71	-1	0	0.000000	0.000000	102.469125
2022-11-16 14:00:00	1167.92	1	100	0.085622	0.085622	0.000000

In [21]:

```

#Summary:
print("I have evaluate the strategy in ETH/EUR from {} to {}".format(df1H.index[0],df1H.index[-1]))
print("~ {}".format(df1H.index[-1]-df1H.index[0]))
print("")

        #how much monecuanto he invertido en este perioro de tiempo
print("I invested:" + str(operaciones.dinero.sum()))
#cuanto dinero he tenido de retorno
print("I gained:" + str(round(operaciones.Eretorno.sum(),2)))
#beneficio?
beneficio=operaciones.Eretorno.sum()-operaciones.dinero.sum()
print("so the benefit is :" + str(round(beneficio,2)))

```

```
print("")  
print("So far it doesn't seem to be an effective strategy in 1H")
```

I have evaluate the strategy in ETH/EUR from 2022-10-27 20:00:00 to 2022-11-17 15:00:00  
~ 20 days 19:00:00

I invested:900  
I gained:774.43  
so the benefit is :-125.57

So far it doesn't seem to be an effective strategy in 1H

## 3 Other things

### 3.1 Screenshots

In [32]:

```
import pandas  
import os  
import pyautogui  
import time  
import datetime  
  
%config IPCompleter.greedy=True  
  
capturas =[]  
  
for i in range(1,300):  
    screen_shot = pyautogui.screenshot()  
    screen_shot.save("the_path...")  
    capturas.append("Screenshot_"+str(i)+".png")  
    time.sleep(30)
```

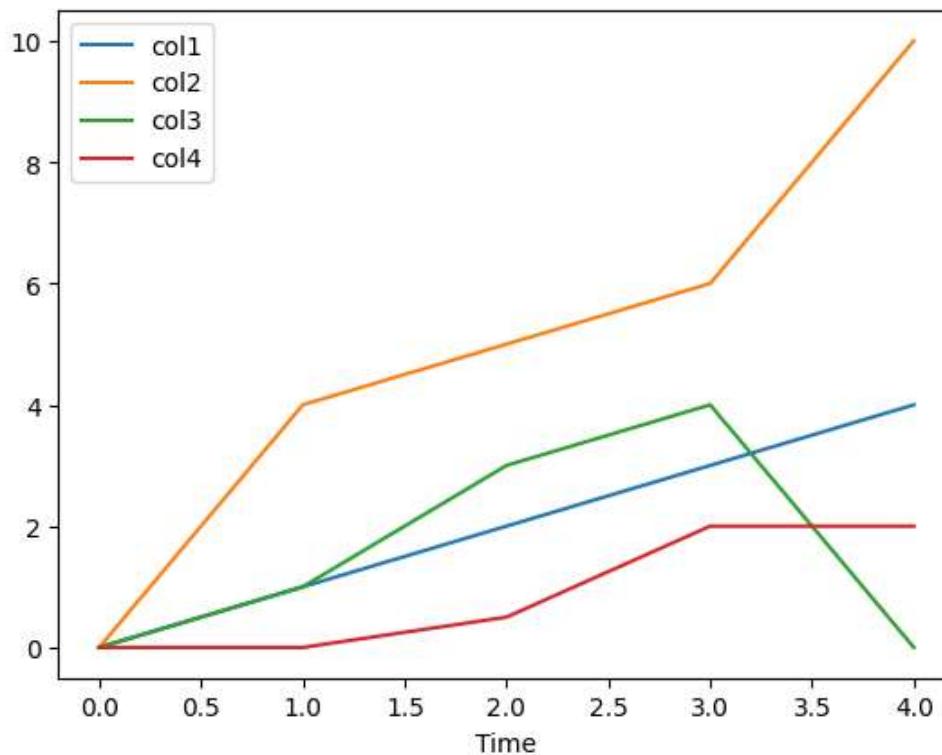
EXAMPLE:



### 3.2 Drawing

In [51]:

```
#create a df to make the draw  
import pandas as pd  
  
df=pd.DataFrame({"col1":[0,1,2,3,4],"col2":[0,4,5,6,10],"col3":[0,1,3,4,0],"col4":[0,0,0.5,2,2]})  
df.plot();  
plt.xlabel("Time");
```



```
In [81]: #find the min of the dataframe
def encontrarMin(df):
    min=df[df.columns[0]].min()
    for i in range (1,len(df)-1):
        if min>df[df.columns[i]].min():
            min=df[df.columns[i]].min()
    return min
#find the max of the dataframe
def encontrarMax(df):
    max=df[df.columns[0]].max()
    for i in range (1,len(df)-1):
        if max<df[df.columns[i]].max():
            max=df[df.columns[i]].max()
    return max
#Normalize each value with respect to the min and the max of the whole DATAFRAME
def CalcularTonalidades(df):
    max=encontrarMax(df)
    min=encontrarMin(df)
    for i in range(0,len(df.columns)):
        df[df.columns[i]+"%MAX"]=(df[df.columns[i]]-min)/(max-min)
    return
CalcularTonalidades(df)

#Create function to make the draw

import cv2
import numpy as np

def dibujar4Triangulos(puntos,ton_t1,ton_t2,ton_t3,ton_t4):
    ##Puntos is related to the number of columns
    ##ton_x is related to the percentage value

    #creo 3 array de 300x300 de unos*255
    image = np.ones((300,300,3),np.uint8)*255
    #image
    ##Triangulo
    #coordenadas del triángulo 1 (superior) A
    pt11 = (150,100)
    pt12 = (100,200)
    pt13 = (200,200)
    #coordenadas del triángulo 2 (abajo-izquierda) B
    pt21 = (100,200)
    pt22 = (50,300)
    pt23 = (150,300)
    #coordenadas del triángulo 3 (abajo-derecha) C
```

```

pt31 = (200,200)
pt32 = (150,300)
pt33 = (250,300)
#coordenadas del triángulo 4 (centro) D
pt41 = (100,200)
pt42 = (200,200)
pt43 = (150,300)

#Create the triangles
triangle_cnt1 = np.array([pt11,pt12,pt13])
triangle_cnt2 = np.array([pt21,pt22,pt23])
triangle_cnt3 = np.array([pt31,pt32,pt33])
triangle_cnt4 = np.array([pt41,pt42,pt43])

for i in range(0,puntos,1):
    #Dibujo
        #arriba
    cv2.drawContours(image,[triangle_cnt1],0,(255-255*ton_t1[i],255-255*ton_t1[i],255),-1)
        #abajo-izquierda
    cv2.drawContours(image,[triangle_cnt2],0,(255-255*ton_t2[i],255-255*ton_t2[i],255),-1)
        #abajo-derecha
    cv2.drawContours(image,[triangle_cnt3],0,(255-255*ton_t3[i],255-255*ton_t3[i],255),-1)
        #centro
    cv2.drawContours(image,[triangle_cnt4],0,(255-255*ton_t4[i],255-255*ton_t4[i],255),-1)
    cv2.imshow("image_punto_"+str(i),image)
    cv2.waitKey()
    #Filename (save each image in the folder)
    filename = "savedImage_punto"+str(i)+".jpg"
    cv2.imwrite(filename, image)

#closing all open windows
cv2.destroyAllWindows()

```

In [ ]: `#Draw the triangles  
#arriba = col1, #abajo-izquierda = col2, #abajo-derecha = col3, #centro = col4  
dibujar4Triangulos(len(df.index),df[df.columns[-4]].values,df[df.columns[-3]].values,df[df.columns[-2]].va`

Time = 0

All the columns have the same value (similar tonality) and is 0% (white)

Time = 1

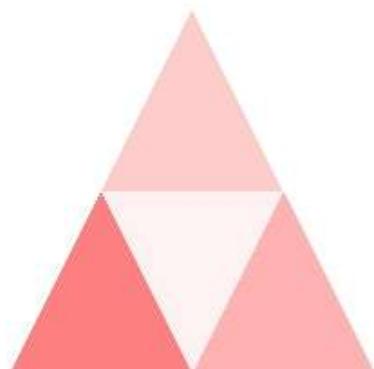
col2 (down-left) has a highest value of all of them so it is the more intense



col1 (up) and col3(down-right) has similar value

Time = 2

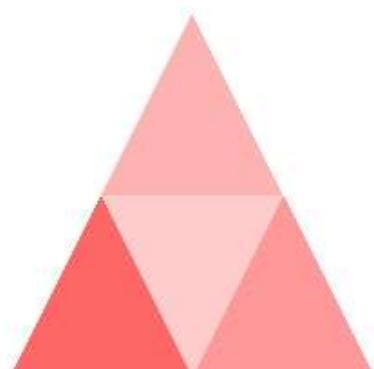
col2 (down-left) has a highest value of all of them so it is the more intense



col1 (up) has a slightly lowwer value than col3(down-right)

Time = 3

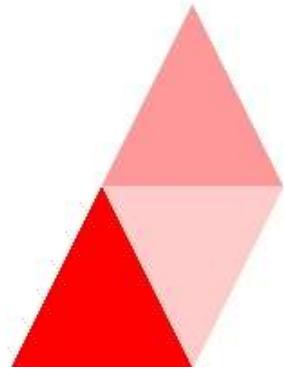
col2 (down-left) has a highest value of all of them so it is the more intense



col1 (up) has a slightly lowwer value than col3(down-right)

Time = 4

col2 (down-left) has a highest value of all of them so it is the more intense



col3 (down-right) has the smallest value of all and is 0

### 3.3 Word manipulation

In [67]:

```
#Create a documento
import docx
doc = docx.Document()
doc.save("prueba.docx")

#open the documento
doc = docx.Document("prueba.docx")

#Write in the document three paragraphs
doc.add_paragraph("Lorem ipsum dolor sit amet, consectetur adipiscing elit.[referencia3]")
doc.add_paragraph("Etiam in nisi sapien. Suspendisse consequat risus ut pulvinar congue.[referencia1]")
doc.add_paragraph("Lorem ipsum dolor sit amet, consectetur adipiscing elit.[referencia2]")

#Check the content of the documentver el contenido de todo el documento
par = doc.paragraphs
for parrafo in range(0,len(par)):
    print(doc.paragraphs[parrafo].text)

doc.save("prueba.docx")
```

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit.[referencia3]
Etiam in nisi sapien. Suspendisse consequat risus ut pulvinar congue.[referencia1]
Lorem ipsum dolor sit amet, consectetur adipiscing elit.[referencia2]
```

Word Before:

Lorem ipsum dolor sit amet, consectetur adipiscing elit.[referencia3]  
 Etiam in nisi sapien. Suspendisse consequat risus ut pulvinar congue.[referencia1]  
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.[referencia2]

In [70]: #Create a List of the references that appear on the text and another List with the details of each referer

```

#go through the text and extract the references:
references=[]

for i in range(0,len(par)):
    parrafo=doc.paragraphs[i].text
    if "[" in parrafo:
        references.append(parrafo.split("[")[1].split("]")[0])

#go through the files with the references and extract the information
content=[]

for files in range(0,len(references)):
    file =open(references[files]+".txt","r")
    content.append(file.read())
    file.close()

#check the information of each reference
for i in range(0, len(references)):
    print(references[i]+" : "+content[i])

```

referencia3 : Cal Newport;Deep Work. Rules For Focused Success In A Distracted World  
 referencia1 : Robert T. Kiyosaki; Rich Dad Poor Dad: What the Rich Teach Their Kids About Money That the Poor and Middle Class Do Not!  
 referencia2 : James Clear; Atomic Habits: the life-changing million-copy

In [63]: #Create the new content using the information of the document

```

#get the text from the document
oldparrafos=[]

for i in range(0,len(par)):
    oldparrafos.append(doc.paragraphs[i].text)

print(oldparrafos)
print("")

#replace the name of the references by its number
newparrafos=[]

for i in range(0,len(oldparrafos)):
    if references[i] in oldparrafos[i]:
        newparrafos.append(oldparrafos[i].replace(references[i],str(i+1)))
print(newparrafos)

```

```
#Create a reference List at the end of the document
```

```
newReferencias=[]

for i in content:
    newReferencias.append((str(i) +". "+ i))

['Lorem ipsum dolor sit amet, consectetur adipiscing elit.[referencia3]', 'Etiam in nisi sapien. Suspendisse consequat risus ut pulvinar congue.[referencia1]', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit.[referencia2]']

['Lorem ipsum dolor sit amet, consectetur adipiscing elit.[1]', 'Etiam in nisi sapien. Suspendisse consequat risus ut pulvinar congue.[2]', 'Lorem ipsum dolor sit amet, consectetur adipiscing elit.[3]']
```

In [71]:

```
#Create a document
doc = docx.Document()
doc.save("pruebaConReferencias.docx")

#open the document
doc = docx.Document("pruebaConReferencias.docx")

#Write in the document
#the new three paragraphs
for i in newparrafos:
    doc.add_paragraph(i)
#and the references List
doc.add_paragraph("References:")
number=1
for i in content:
    doc.add_paragraph((str(number) +". "+ i))
    number+=1

#Save the document
doc.save("pruebaConReferencias.docx")
```

Word after

