Untitled2

November 27, 2021

[34]:
```
import requests
from bs4 import BeautifulSoup
import numpy as np
import pandas as pd
from selenium.webdriver import Firefox
from selenium.webdriver.firefox.options import Options
import matplotlib.pyplot as plt
import seaborn as sns

URL = "https://www.bolsamadrid.es/esp/aspx/Indices/Resumen.aspx"
page = requests.get(URL)
soup = BeautifulSoup(page.content, "html.parser")
table = soup.find('table', attrs={'id': 'ctl00_Contenido_tblÍndices'})
```

[35]:
```
matrix_row = []
matrix = []
columns_name = []
header_row = True
for row in table.find_all("tr"):
    if header_row:
        header_row = False
        for column in row.find_all('th'):
            columns_name.append(column.text)
    else:
        matrix_row = []
        for column in row.find_all('td'):
                if column.text.strip() == '-':
                    matrix_row.append('0')
                else:
                    matrix_row.append(column.text)
        matrix.append(matrix_row)
```

[36]:
```
df = pd.DataFrame(matrix, columns =columns_name)
df.columns
```

[36]:
```
Index(['Nombre', 'Anterior', 'Último', '% Dif.', 'Máximo', 'Mínimo', 'Fecha',
       'Hora', '% Dif.Año 2021'],
      dtype='object')
```

Modifiquemos los tipos de datos de la columna.

[37]:
```
df['Anterior'] = df['Anterior'].apply(lambda x: float(x.split()[0].replace('.',␣
 ↪'+').replace(',', '.').replace('+', '')))
df['Último'] = df['Último'].apply(lambda x: float(x.split()[0].replace('.',␣
 ↪'+').replace(',', '.').replace('+', '')))
df['Máximo'] = df['Máximo'].apply(lambda x: float(x.split()[0].replace('.',␣
 ↪'+').replace(',', '.').replace('+', '')))
df['Mínimo'] = df['Mínimo'].apply(lambda x: float(x.split()[0].replace('.',␣
 ↪'+').replace(',', '.').replace('+', '')))
df['% Dif.'] = df['% Dif.'].apply(lambda x: float(x.split()[0].replace('.',␣
 ↪'+').replace(',', '.').replace('+', '')))
df['% Dif.Año 2021'] = df['% Dif.Año 2021'].apply(lambda x: float(x.split()[0].
 ↪replace('.', '+').replace(',', '.').replace('+', '')))
df['Fecha'] = pd.to_datetime(df['Fecha'], format='%d/%m/%Y')
df.astype({"Nombre":str,'Hora':str})
```

[37]:
```
                    Nombre  Anterior   Último  % Dif.   Máximo   Mínimo  \
0                  IBEX 35®    8840.9   8438.5   -4.55   8592.3   8410.9
1   IBEX 35® con Dividendos   27187.9  25956.4   -4.53  26410.3  25871.5
2          IBEX MEDIUM CAP®   13411.6  13020.0   -2.92  13193.0  13011.6
3           IBEX SMALL CAP®    8256.1   7887.9   -4.46   8150.5   7887.9
4           IBEX 35® Bancos     469.8    440.3   -6.28    452.5    439.1
..                      …       …      …      …      …      …
73    Índice ITX Inverso X3     173.7    199.7   14.97    200.0    181.4
74    Índice TEF Inverso X5   10528.0  11905.5   13.08  12801.4  11480.5
75    Índice SAN Inverso X5    4040.0   5606.8   38.78   5725.8   4932.4
76   Índice BBVA Inverso X5    8734.0  11286.0   29.22  11398.7  10196.3
77    Índice ITX Inverso X5     860.2   1074.8   24.95   1077.6    923.8

        Fecha      Hora  % Dif.Año 2021
0  2021-11-26  16:42:59            4.52
1  2021-11-26  16:42:59            6.98
2  2021-11-26  16:42:38            2.39
3  2021-11-26  16:42:51           -2.60
4  2021-11-26  16:42:59           17.07
..        …       …               …
73 2021-11-26  16:42:46          -60.97
74 2021-11-26  16:42:53          917.17
75 2021-11-26  16:42:59          539.09
76 2021-11-26  16:42:57          563.64
77 2021-11-26  16:42:46          -84.77

[78 rows x 9 columns]
```

[39]:
```
df.shape
```

[39]: (78, 9)

[12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 78 entries, 0 to 77
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Nombre          78 non-null     object
 1   Anterior        78 non-null     float64
 2   Último          78 non-null     float64
 3   % Dif.          78 non-null     float64
 4   Máximo          78 non-null     float64
 5   Mínimo          78 non-null     float64
 6   Fecha           78 non-null     datetime64[ns]
 7   Hora            78 non-null     object
 8   % Dif.Año 2021  78 non-null     float64
dtypes: datetime64[ns](1), float64(6), object(2)
memory usage: 5.6+ KB
```

Tenemos 9 columnas: * Nombre: Nombre del índice * Anterior: Valor anterior * Último: Último valor * % Dif.: Diferencia * Máximo: Valor máximo * Mínimo: Valor mínimo * Fecha: Fecha que la medición * Hora: Hora de la medición * % Dif.Año 2021: Diferencia respecto al 2021

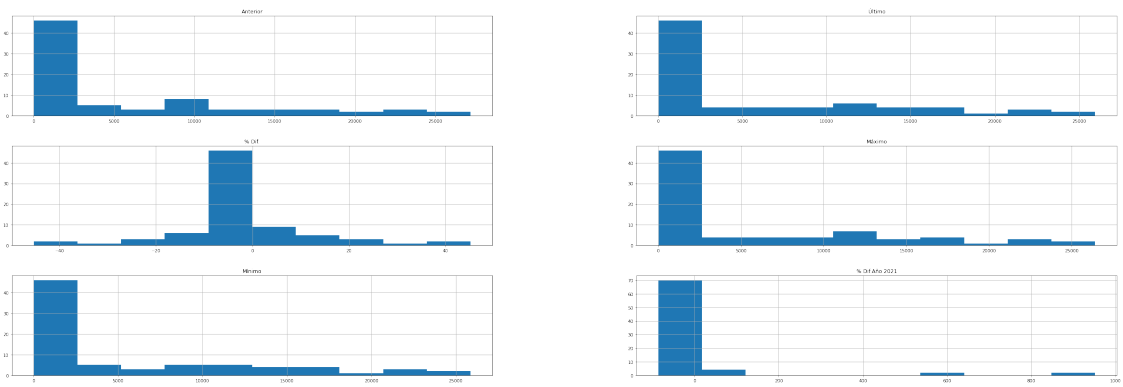
[40]:
```
df.describe()
```

[40]:

	Anterior	Último	% Dif.	Máximo	Mínimo	% Dif.Año 2021
count	78.000000	78.000000	78.000000	78.000000	78.000000	78.000000
mean	5739.233846	5588.093846	-2.523462	5709.689487	5531.529615	37.312179
std	7405.558437	7137.282387	13.940385	7266.097577	7096.767921	172.852636
min	7.050000	7.430000	-45.350000	7.430000	7.430000	-86.080000
25%	331.625000	255.700000	-5.347500	269.275000	249.800000	0.822500
50%	1538.750000	1477.015000	-3.840000	1497.865000	1475.715000	5.125000
75%	9492.450000	10362.030000	-0.425000	10544.865000	10155.712500	10.347500
max	27187.900000	25956.400000	45.210000	26410.300000	25871.500000	951.610000

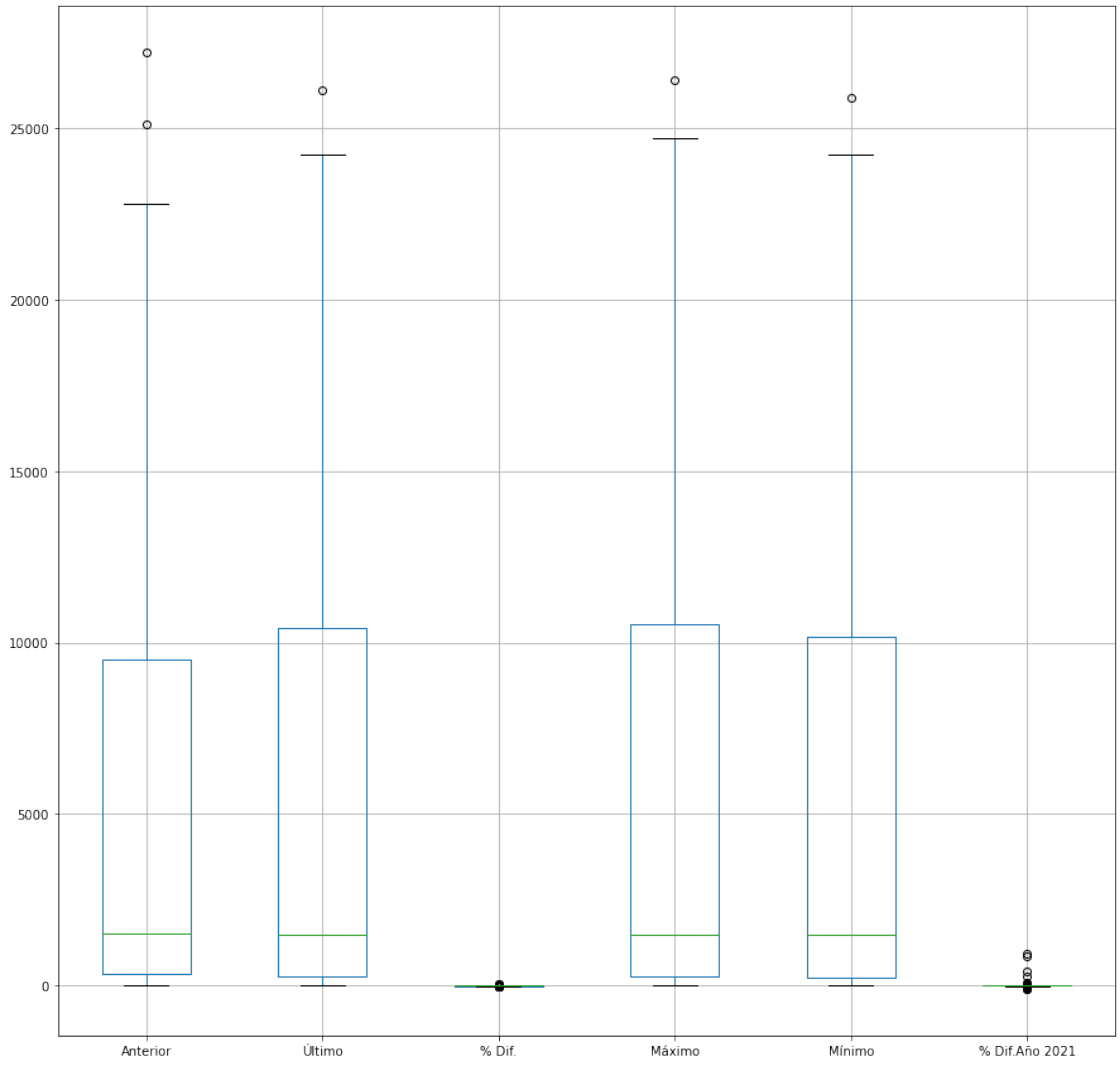
La diferencia entre la media y el promedio en algunas columnas como en % Dif.Año 2021 podemos decir que hay algunos outliers, lo confirmaremos con una gráfica boxplot.

[41]:
```
df.hist(figsize=(45,15));
```



Podemos ver que ninguna columna tiene una distribución normal, la única que se asemaja un poco es % Dif.

[25]:
```
df.boxplot(figsize=(15,15));
```



Aqui podemos confirmar la presencia de outliers.

El cojunto de datos describe los valores de lo índices que cotizan en la Bolsa de Madrid en un momento dado, específicamente a las 16:42.

Cuenta con 78 filas y 9 columnas.

Con estos datos pdríamos realizar una predicción de los valores de los índices. Para el mismo tendríamos que hacer un Web Scraping un poco más elaborado para obtener los valores cada cierto período de tiempo tiempo, e ir guardandolos.