Minería de Datos **Ibañez Guzman Osvaldo**
Grupo: 1 **Semestre 2023-1**
Dr. Guillermo Gilberto Molero Castillo **Práctica #3: Análisis exploratorio de datos (EDA)**

Objetivo

Hacer un análisis exploratorio de datos con base en información obtenida de Yahoo Finanzas sobre empresas mexicanas usando dicha plataforma ya que ofrece una amplia variedad de datos de mercado sobre acciones, bonos, divisas y criptomonedas. También proporciona informes de noticias con varios puntos de vista sobre diferentes mercados de todo el mundo, todos accesibles a través de la biblioteca yfinance.

Desarrollo

Primeramente, es necesario hacer la instalación de la biblioteca *yfinance* ya que esta nos permitirá obtener los datos de la bolsa de valores, además de las bibliotecas ya usadas en prácticas anteriores para la visualización, análisis y manejo de los datos obtenidos.

```
!pip install yfinance
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import yfinance as yf
```

La primer empresa que analizare será Bachoco una empresa productora avícola, usando la biblioteca de yfinance buscaremos la información referente a la empresa cuyo ticker es BACHOCOB.MX

```
DataBachoco = yf.Ticker('BACHOCOB.MX')
DataBachoco
```

```
yfinance.Ticker object <BACHOCOB.MX>
```

De los datos obtenidos será necesario buscar el datos del histórico empezando a buscar desde inicios del año 2019 hasta el 19 de septiembre de 2022, mostrando los datos diarios entre los datos que nos muestra la tabla están Open y Close que muestran los valores con lo que abre y cierra las acciones en ese día, High y Low el valor mas alto y bajo que tuvieron en ese día, volumen la cantidad de acciones comerciales, además de mostrar los dividendos.

```
BachocoHist = DataBachoco.history(start = '2019-1-1', end = '2022-9-19', interval = '1d')
BachocoHist
```

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2019-01-02	61.360897	62.802474	60.022284	62.493568	76323	0.0	0
2019-01-03	61.360893	62.044237	60.855402	61.201752	64983	0.0	0
2019-01-04	61.051979	64.318941	61.051979	63.354767	400430	0.0	0
2019-01-07	63.195636	65.217593	62.999057	63.298607	564164	0.0	0
2019-01-08	63.130117	64.552975	62.877368	63.420303	68198	0.0	0
...	...	...	...	...	...	...	...
2022-09-09	79.610001	79.889999	79.599998	79.690002	36495	0.0	0
2022-09-12	79.599998	80.000000	79.449997	79.519997	497214	0.0	0
2022-09-13	79.500000	80.050003	79.110001	79.379997	1022244	0.0	0
2022-09-14	79.709999	79.709999	78.400002	79.239998	2546708	0.0	0
2022-09-15	79.389999	80.120003	79.080002	79.919998	422159	0.0	0

936 rows × 7 columns

Entre las funciones para realizar el análisis son la función *head*, el cual el parámetro que reciba será el número de renglones que nos mostrara la tabla desde el primer dato, en este caso nos mostrara los primeros 10 registros.

```
BachocoHist.head(10)
```

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2019-01-02	61.360897	62.802474	60.022284	62.493568	76323	0.0	0
2019-01-03	61.360893	62.044237	60.855402	61.201752	64983	0.0	0
2019-01-04	61.051979	64.318941	61.051979	63.354767	400430	0.0	0
2019-01-07	63.195636	65.217593	62.999057	63.298607	564164	0.0	0
2019-01-08	63.130117	64.552975	62.877368	63.420303	68198	0.0	0
2019-01-09	63.663690	65.517150	63.298614	64.768280	596989	0.0	0
2019-01-10	64.590423	65.966481	64.178540	64.552979	242466	0.0	0
2019-01-11	64.169175	66.500045	64.169175	66.266022	229750	0.0	0
2019-01-14	65.994557	68.362871	65.910312	67.876106	244698	0.0	0
2019-01-15	68.259899	68.306707	66.032000	67.408058	317141	0.0	0

La función *shape* muestra las dimensiones de la matriz de datos, en este caso tendremos una matriz de siete columnas con 936 registros.

```
BachocoHist.shape
```

```
(936, 7)
```

También será necesario observar los tipos de datos con los que contamos, esto para poder realizar un análisis de los datos, se observa que todos nuestros datos son numéricos entre enteros y flotantes.

```
BachocoHist.dtypes
```

```
Open            float64
High            float64
Low             float64
Close           float64
Volume            int64
Dividends       float64
Stock Splits      int64
dtype: object
```

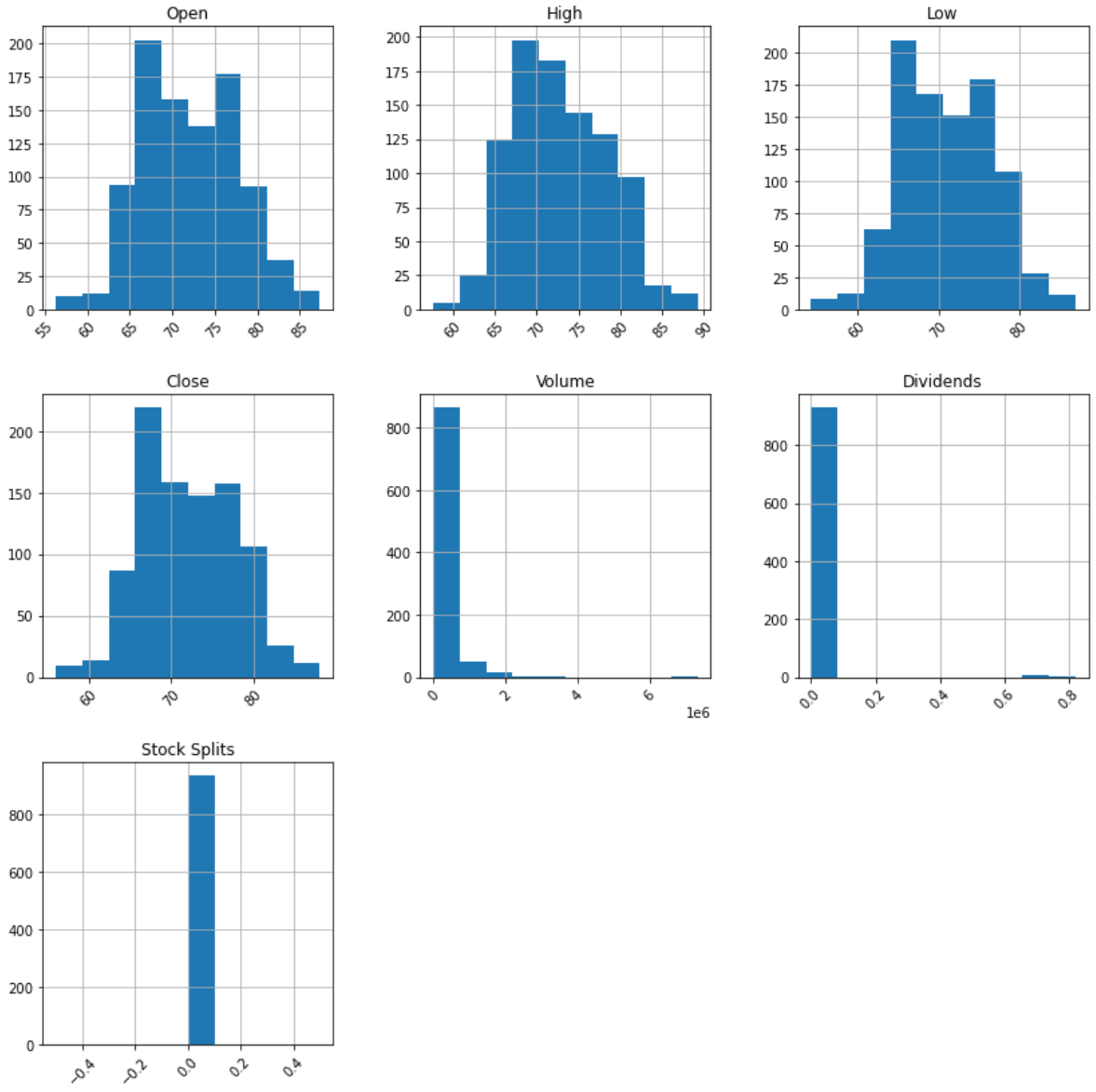
Otra función es *info* que nos muestra toda la información mostrada, como son el nombre de la columna, los datos no nulos y el tipo de datos, además poder ver las dimensiones de la matriz.

```
BachocoHist.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 936 entries, 2019-01-02 to 2022-09-15
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Open          936 non-null    float64
 1   High          936 non-null    float64
 2   Low           936 non-null    float64
 3   Close         936 non-null    float64
 4   Volume        936 non-null    int64
 5   Dividends     936 non-null    float64
 6   Stock Splits  936 non-null    int64
dtypes: float64(5), int64(2)
memory usage: 58.5 KB
```

Otra forma de analizarlo es de forma grafica para esto analizaremos los datos de las siete columnas, esto se hace para la búsqueda de datos atípicos, de forma general podemos decir que la grafica de Volumen se encuentra sesgada a la izquierda y que Dividends y Stock Splits la mayoría de los valores son ceros.

```
BachocoHist.hist(figsize=(14,14), xrot=45)
plt.show()
```



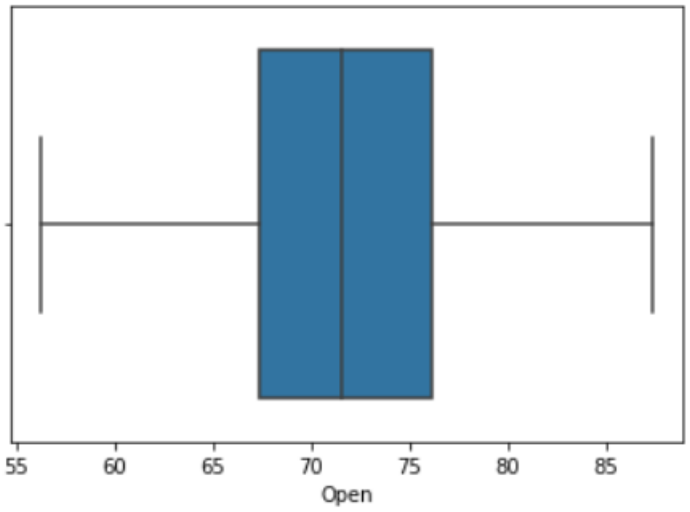
Seguiremos obteniendo un resumen de las variables numéricas, nos muestra el numero de registros por columna, el promedio de los datos, su desviación estándar, así como los valores mínimos y máximos, así como el valor de cada cuartil, aquí es posible observar que en la sección de Dividends existen valores diferentes de 0 que es 0.82, lo cual podría llegar a ser un valor atípico o algo normal, pero habría que investigarlo para tener datos confiables.

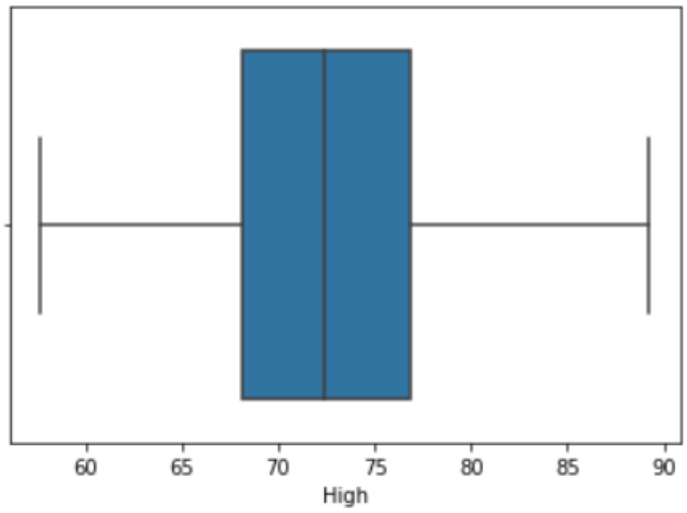
```
BachocoHist.describe()
```

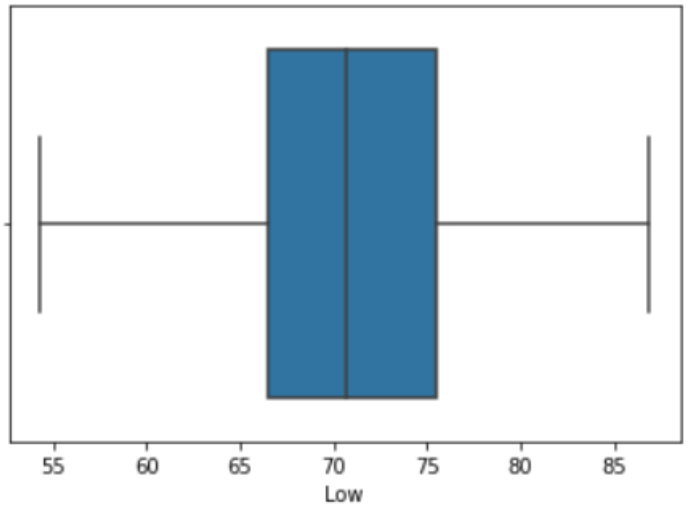
	Open	High	Low	Close	Volume	Dividends	Stock Splits
count	936.000000	936.000000	936.000000	936.000000	9.360000e+02	936.000000	936.0
mean	71.970276	72.766310	71.088340	71.944658	2.792215e+05	0.005299	0.0
std	5.667321	5.634104	5.687990	5.692017	4.317420e+05	0.061231	0.0
min	56.309686	57.624082	54.271406	55.966793	0.000000e+00	0.000000	0.0
25%	67.373903	68.172862	66.507212	67.276970	7.232875e+04	0.000000	0.0
50%	71.544575	72.368562	70.714461	71.550030	1.551165e+05	0.000000	0.0
75%	76.190202	76.819681	75.537580	76.191931	3.161140e+05	0.000000	0.0
max	87.360016	89.245892	86.883784	88.045792	7.330487e+06	0.820000	0.0

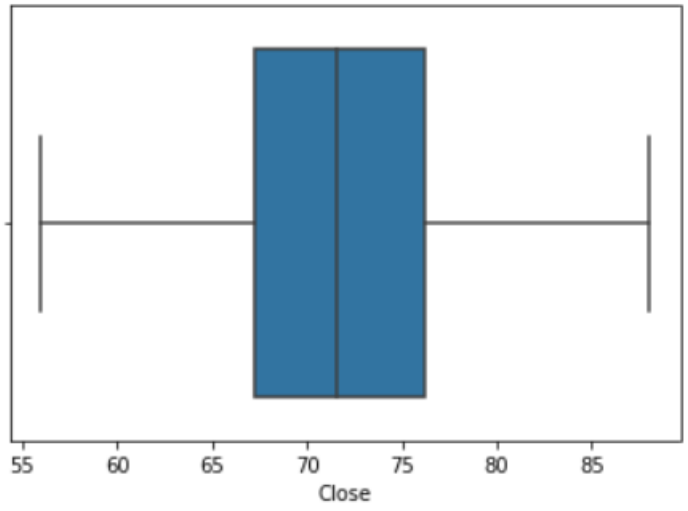
Otra forma grafica de visualizar los datos son con diagramas de caja, esta nos permitirá ver cuántos datos se encuentran fuera del rango normal de los registros, una vez que observamos los diagramas de caja todos los datos se encuentran en el rango normal, excepto por volumen ya que e encuentran datos fuera del promedio de dichos datos.

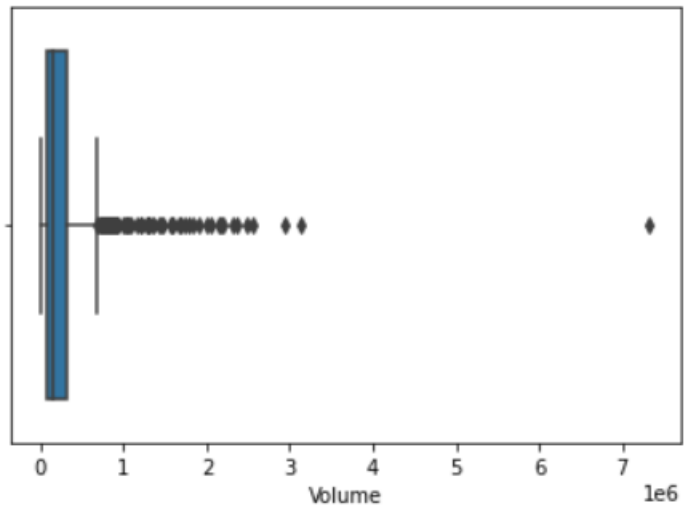
```
VariablesValoresAtipicos = ['Open', 'High', 'Low', 'Close', 'Volume']
for col in VariablesValoresAtipicos:
  sns.boxplot(col, data=BachocoHist)
  plt.show()
```




60
65
70
75
80
85
90
High

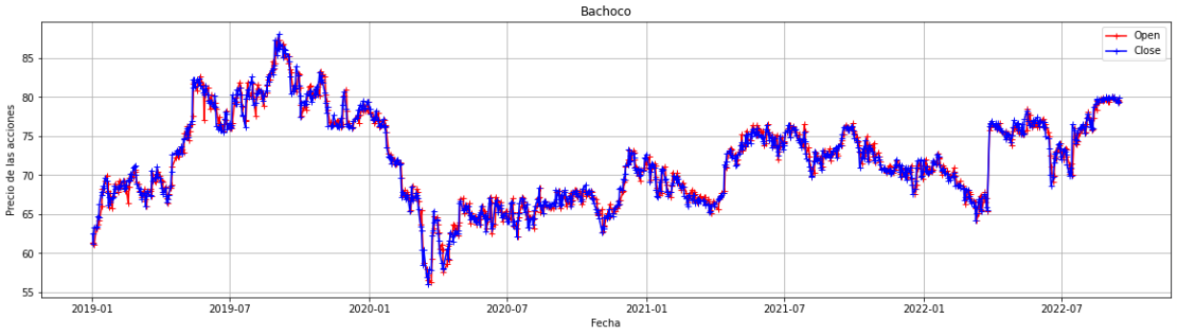

55
60
65
70
75
80
85
Low


55
60
65
70
75
80
85
Close



Continuaremos realizando una comparativa pero ahora entre los valores con los que se abrió el día y con los datos que finalizo el día, al hacer la gráfica colocaremos de diferentes colores los dos datos para la visualización sea un poco mas sencilla, donde de color rojo mostraremos el valor con los que abrió y con azul los valores con los que cerro, es posible ver que los valores prácticamente son los mismos, ya que apenas se notan ligeras variaciones entre unos datos y otros.

```
plt.figure(figsize=(20, 5))
plt.plot(BachocoHist['Open'], color='red', marker='+', label='Open')
plt.plot(BachocoHist['Close'], color='blue', marker='+', label='Close')
plt.xlabel('Fecha')
plt.ylabel('Precio de las acciones')
plt.title('Bachoco')
plt.grid(True)
plt.legend()
plt.show()
```



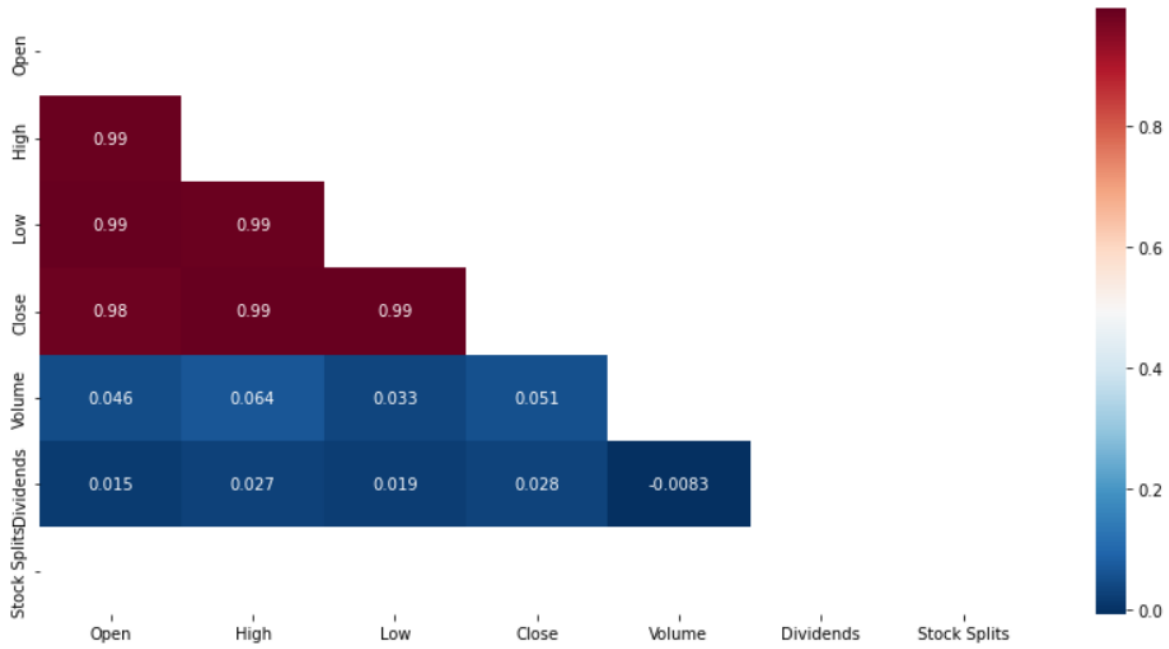
Procederemos a realizar una matriz de correlación para ver que tan fuerte o débil son las relaciones entre los datos, además de mostrar su mapa de calor para poderlo visualizar mas fácilmente, podemos observar que existen relaciones muy fuertes en la mayoría de nuestros datos, esto debido a que las acciones abrieron el día con un valor alto, probablemente el valor de cierre también sea alto, y el también el valor mas alto y bajo serán mayores que un día anterior, es por eso que contamos con muchas relaciones fuertes.

Tanto para el Volume como Dividends su relación es prácticamente cero comparada con los otros valores por lo que se podría decir que son datos independientes, en el caso de Stock Splits sus valores al ser completamente ceros no presentan ningún tipo de relación.

```
BachocoHist.corr()
```

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Open	1.000000	0.987856	0.990425	0.984848	0.046145	0.014918	NaN
High	0.987856	1.000000	0.985749	0.992923	0.064323	0.026821	NaN
Low	0.990425	0.985749	1.000000	0.991688	0.033322	0.018516	NaN
Close	0.984848	0.992923	0.991688	1.000000	0.051169	0.028433	NaN
Volume	0.046145	0.064323	0.033322	0.051169	1.000000	-0.008350	NaN
Dividends	0.014918	0.026821	0.018516	0.028433	-0.008350	1.000000	NaN
Stock Splits	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
plt.figure(figsize=(14,7))
MatrizInf = np.triu(BachocoHist.corr())
sns.heatmap(BachocoHist.corr(), cmap='RdBu_r', annot=True, mask=MatrizInf)
plt.show()
```



Una vez realizado este análisis deberemos realizar el mismo proceso con otras tres empresas mexicanas, la primera a analizar es Bimbo, cuyo ticker es BIMBOA.MX, del cual obtendremos sus registros en las mismas fechas que Bachoco.

```
DataBimbo = yf.Ticker('BIMBOA.MX')
```

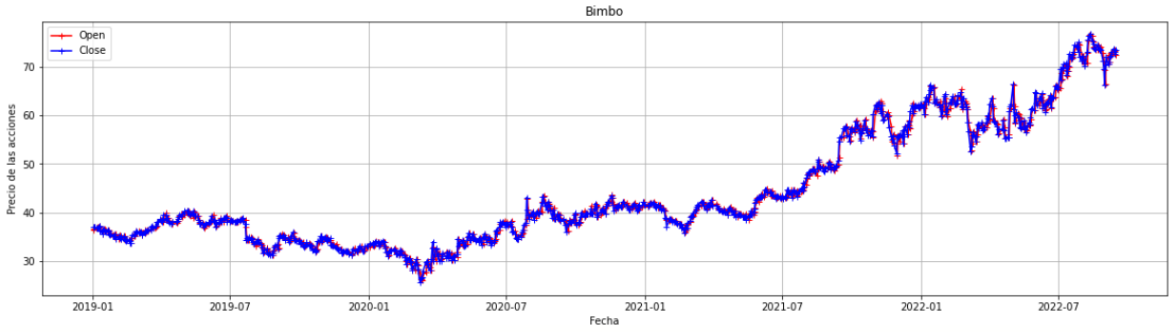
```
BimboHist = DataBimbo.history(start = '2019-1-1', end = '2022-9-19', interval='1d')
BimboHist
```

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2019-01-02	36.489381	37.147865	36.291837	36.987946	925915	0.0	0
2019-01-03	36.724553	37.279561	36.301242	36.987946	1574973	0.0	0
2019-01-04	37.016163	37.345406	36.479969	37.053791	1882812	0.0	0
2019-01-07	36.865655	37.241932	36.498786	36.790401	1567381	0.0	0
2019-01-08	37.157274	37.157274	36.592862	36.931507	1776183	0.0	0
...	...	...	...	...	...	...	...
2022-09-09	72.440002	73.360001	71.400002	73.029999	1144724	0.0	0
2022-09-12	73.540001	74.239998	73.150002	73.699997	869317	0.0	0
2022-09-13	73.410004	74.050003	72.300003	73.029999	1762960	0.0	0
2022-09-14	73.269997	75.000000	71.709999	72.650002	2429018	0.0	0
2022-09-15	72.519997	73.779999	72.209999	73.449997	10694764	0.0	0

936 rows × 7 columns

Posteriormente compararemos sus valores Open y Close, pata ver los cambios que pudieron haberse generado aunque de igual forma parecen muy similares, solo en algunos casos se llega a ver una ligera varianza.

```
plt.figure(figsize=(20, 5))
plt.plot(BimboHist['Open'], color='red', marker='+', label='Open')
plt.plot(BimboHist['Close'], color='blue', marker='+', label='Close')
plt.xlabel('Fecha')
plt.ylabel('Precio de las acciones')
plt.title('Bimbo')
plt.grid(True)
plt.legend()
plt.show()
```



La siguiente empresa a analizar será Cementos de México o CEMEX, con el ticker CEMEXCPO.MX, realizare el mismo proceso que con Bimbo, obtener sus datos históricos en el intervalo de fechas

usado anteriormente y posteriormente comparar sus valores de Open y Close, este proceso se hará de igual manera con Aeromexico cuyo ticker es AEROMEX.MX

```
DataCemex = yf.Ticker('CEMEXCPO.MX')
```

```
CemexHist = DataCemex.history(start = '2019-1-1', end = '2022-9-19', interval='1d')
CemexHist
```

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2019-01-02	9.211399	9.542815	9.201651	9.523320	11805436	0.0	0
2019-01-03	9.464836	9.523322	9.250391	9.338119	19429709	0.0	0
2019-01-04	9.396603	9.542816	9.260138	9.367360	18881858	0.0	0
2019-01-07	9.386855	9.679280	9.240641	9.640290	20506020	0.0	0
2019-01-08	9.698775	9.883979	9.650038	9.854735	35977574	0.0	0
...	...	...	...	...	...	...	...
2022-09-09	7.710000	7.920000	7.710000	7.880000	10737320	0.0	0
2022-09-12	7.910000	8.050000	7.900000	8.000000	11344674	0.0	0
2022-09-13	7.880000	7.930000	7.680000	7.720000	11104198	0.0	0
2022-09-14	7.770000	7.770000	7.530000	7.560000	28073140	0.0	0
2022-09-15	7.590000	7.740000	7.210000	7.260000	88414424	0.0	0

936 rows × 7 columns

```
plt.figure(figsize=(20, 5))
plt.plot(CemexHist['Open'], color='red', marker='+', label='Open')
plt.plot(CemexHist['Close'], color='blue', marker='+', label='Close')
plt.xlabel('Fecha')
plt.ylabel('Precio de las acciones')
plt.title('CEMEX')
plt.grid(True)
plt.legend()
plt.show()
```

Al hacer el análisis con Aeromexico, podemos ver que existe un repunte en sus acciones, esto puede deberse al momento en que los vuelos comerciales volvieron a aparecer después de la pandemia por COVID-19 y por ende regresaron las acciones de esta aerolínea.

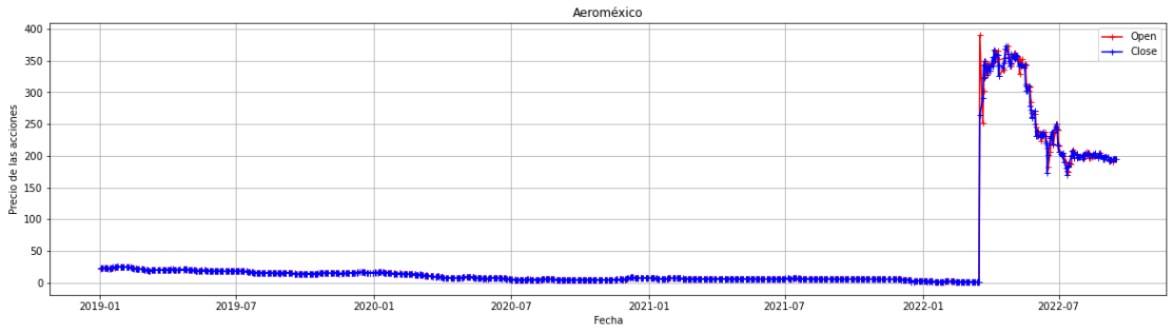
```
DataAeromex = yf.Ticker('AEROMEX.MX')
```

```
AeromexHist = DataAeromex.history(start = '2019-1-1', end = '2022-9-19', interval='1d')
AeromexHist
```

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2019-01-02	22.000000	23.080000	22.000000	22.530001	54890	0	0
2019-01-03	22.500000	23.049999	22.500000	22.770000	88098	0	0
2019-01-04	23.430000	23.840000	22.780001	23.160000	69569	0	0
2019-01-07	23.180000	24.500000	23.100000	23.270000	160120	0	0
2019-01-08	23.219999	23.870001	22.820000	23.230000	190421	0	0
...	...	...	...	...	...	...	...
2022-09-09	193.029999	193.029999	193.029999	193.029999	0	0	0
2022-09-12	190.630005	195.500000	190.309998	195.500000	1681	0	0
2022-09-13	195.500000	195.500000	195.500000	195.500000	0	0	0
2022-09-14	195.500000	195.500000	195.500000	195.500000	0	0	0
2022-09-15	195.500000	195.500000	195.500000	195.500000	0	0	0

936 rows × 7 columns

```
plt.figure(figsize=(20, 5))
plt.plot(AeromexHist['Open'], color='red', marker='+', label='Open')
plt.plot(AeromexHist['Close'], color='blue', marker='+', label='Close')
plt.xlabel('Fecha')
plt.ylabel('Precio de las acciones')
plt.title('Aeroméxico')
plt.grid(True)
plt.legend()
plt.show()
```



Para hacer un análisis de todos las empresas únicamente usaremos la columna de valores Close, por lo que al usar la función *drop* y pasarle el nombre de las columnas, esto hará que dichas columnas se quiten del registro, además cambiaremos el nombre de la columna por el nombre de la empresa, ya que esto nos ayudará a identificarlos de mejor forma.

```
BachocoClose = BachocoHist.drop(columns = ['Open', 'High', 'Low', 'Volume',
                                           'Dividends', 'Stock Splits'])
BachocoClose.rename(columns = {'Close': 'Bachoco'}, inplace = True)
BachocoClose
```

	Bachoco
Date	
2019-01-02	62.493568
2019-01-03	61.201752
2019-01-04	63.354767
2019-01-07	63.298607
2019-01-08	63.420303
...	...
2022-09-09	79.690002
2022-09-12	79.519997
2022-09-13	79.379997
2022-09-14	79.239998
2022-09-15	79.919998

936 rows × 1 columns

El proceso anterior se realzará con Bimbo, Cemex y Aeromexico, esto para crear una comparativa entre las cuatro empresas que se quieren analizar.

```
BimboClose = BimboHist.drop(columns = ['Open', 'High', 'Low', 'Volume',
                                       'Dividends', 'Stock Splits'])
BimboClose.rename(columns = {'Close': 'Bimbo'}, inplace = True)
BimboClose
```

Date	Bimbo
2019-01-02	36.987946
2019-01-03	36.987946
2019-01-04	37.053791
2019-01-07	36.790401
2019-01-08	36.931507
...	...
2022-09-09	73.029999
2022-09-12	73.699997
2022-09-13	73.029999
2022-09-14	72.650002
2022-09-15	73.449997

936 rows × 1 columns

```
CemexClose = CemexHist.drop(columns = ['Open', 'High', 'Low', 'Volume',
                                       'Dividends', 'Stock Splits'])
CemexClose.rename(columns = {'Close': 'Cemex'}, inplace = True)
CemexClose
```

Date	Cemex
2019-01-02	9.523320
2019-01-03	9.338119
2019-01-04	9.367360
2019-01-07	9.640290
2019-01-08	9.854735
...	...
2022-09-09	7.880000
2022-09-12	8.000000
2022-09-13	7.720000
2022-09-14	7.560000
2022-09-15	7.260000

936 rows × 1 columns

```
AeromexClose = AeromexHist.drop(columns = ['Open', 'High', 'Low', 'Volume',
                                           'Dividends', 'Stock Splits'])
AeromexClose.rename(columns = {'Close': 'Aeromexico'}, inplace = True)
AeromexClose
```

Date	Aeromexico
2019-01-02	22.530001
2019-01-03	22.770000
2019-01-04	23.160000
2019-01-07	23.270000
2019-01-08	23.230000
...	...
2022-09-09	193.029999
2022-09-12	195.500000
2022-09-13	195.500000
2022-09-14	195.500000
2022-09-15	195.500000

936 rows × 1 columns

Con las tablas ya modificadas, deberemos concatenar las cuatro tablas obtenidas y las uniremos con el valor *inner* ya que esto permite que todos los datos se unan según la columna que comparte, en este caso el día por lo que nos permitirá tener las cuatro tablas con los datos de cierre en el mismo día.

```
Acciones = pd.concat([BachocoClose, BimboClose, CemexClose, AeromexClose],
                     axis = 'columns', join = 'inner')
Acciones
```

Date	Bachoco	Bimbo	Cemex	Aeromexico
2019-01-02	62.493568	36.987946	9.523320	22.530001
2019-01-03	61.201752	36.987946	9.338119	22.770000
2019-01-04	63.354767	37.053791	9.367360	23.160000
2019-01-07	63.298607	36.790401	9.640290	23.270000
2019-01-08	63.420303	36.931507	9.854735	23.230000
...	...	...	...	...
2022-09-09	79.690002	73.029999	7.880000	193.029999
2022-09-12	79.519997	73.699997	8.000000	195.500000
2022-09-13	79.379997	73.029999	7.720000	195.500000
2022-09-14	79.239998	72.650002	7.560000	195.500000
2022-09-15	79.919998	73.449997	7.260000	195.500000

936 rows × 4 columns

Para tener un mejor análisis usaremos la función *dropna* la cual eliminará todos los registros nulos, esto hará que tengamos los datos completos para poder analizarlos de mejor manera, en este caso no se eliminó ningún registro ya que no contamos con valores nulos.

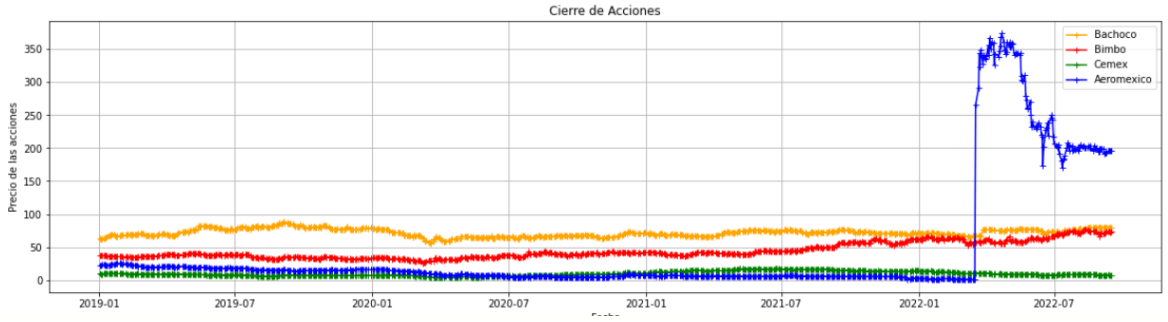
```
AccionesNN = Acciones.dropna()
AccionesNN
```

Date	Bachoco	Bimbo	Cemex	Aeromexico
2019-01-02	62.493568	36.987946	9.523320	22.530001
2019-01-03	61.201752	36.987946	9.338119	22.770000
2019-01-04	63.354767	37.053791	9.367360	23.160000
2019-01-07	63.298607	36.790401	9.640290	23.270000
2019-01-08	63.420303	36.931507	9.854735	23.230000
...	...	...	...	...
2022-09-09	79.690002	73.029999	7.880000	193.029999
2022-09-12	79.519997	73.699997	8.000000	195.500000
2022-09-13	79.379997	73.029999	7.720000	195.500000
2022-09-14	79.239998	72.650002	7.560000	195.500000
2022-09-15	79.919998	73.449997	7.260000	195.500000

936 rows × 4 columns

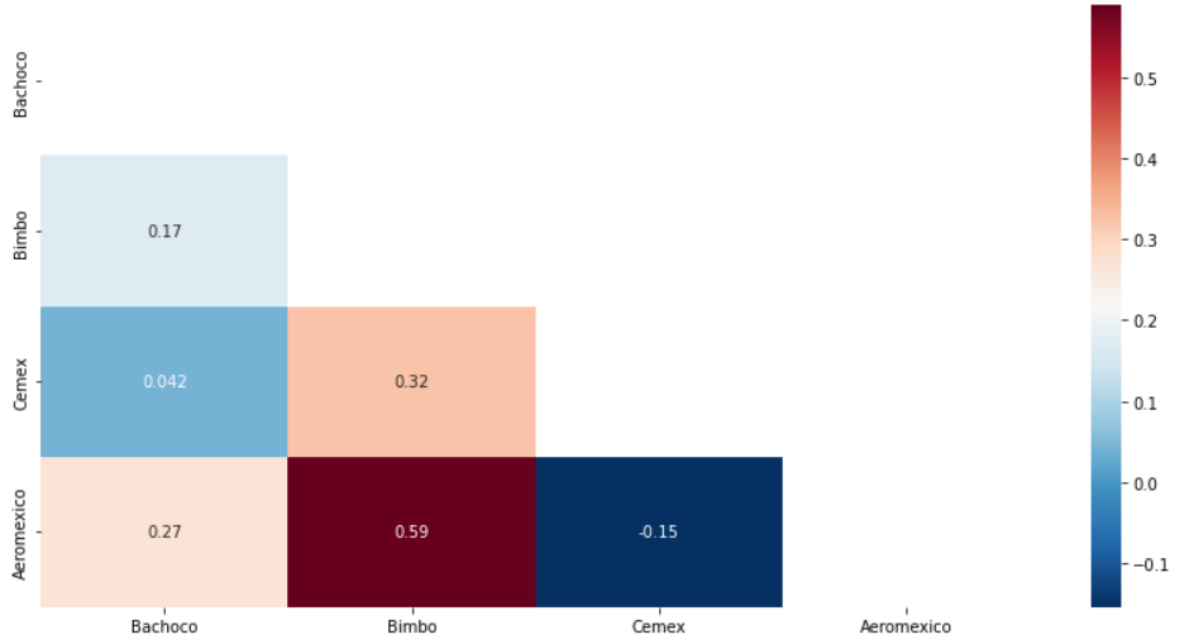
Usaremos la forma gráfica para analizarlo más fácilmente haciendo una comparativa entre las fechas y el valor de cierre, de esta forma es posible observar el repunte que tuvo Aeromexico en comparación con las otras empresas, algunas como Bimbo hubo fechas donde comenzaron a subir y otras como Bachoco y Cemex que se mantuvieron prácticamente constantes en todo el tiempo del análisis.

```
plt.figure(figsize=(20, 5))
plt.plot(Acciones['Bachoco'], color='orange', marker='+', label='Bachoco')
plt.plot(Acciones['Bimbo'], color='red', marker='+', label='Bimbo')
plt.plot(Acciones['Cemex'], color='green', marker='+', label='Cemex')
plt.plot(Acciones['Aeromexico'], color='blue', marker='+', label='Aeromexico')
plt.xlabel('Fecha')
plt.ylabel('Precio de las acciones')
plt.title('Cierre de Acciones')
plt.grid(True)
plt.legend()
plt.show()
```



Usaremos la correlación entre las diferentes empresas para ver su comportamiento, esto podría permitirnos ver cómo es que se movió el mercado en sectores como alimentarios o de una industria especifica, en este caso no se cuenta con relaciones fuertes por lo que prácticamente todas son variables independientes.

```
plt.figure(figsize=(14,7))
MatrizInf = np.triu(Acciones.corr())
sns.heatmap(Acciones.corr(), cmap='RdBu_r', annot=True, mask=MatrizInf)
plt.show()
```




```
Acciones.corr()
```

	Bachoco	Bimbo	Cemex	Aeromexico
Bachoco	1.000000	0.168919	0.042199	0.269485
Bimbo	0.168919	1.000000	0.324019	0.588815
Cemex	0.042199	0.324019	1.000000	-0.153905
Aeromexico	0.269485	0.588815	-0.153905	1.000000

Otra forma de poder ver la comparativa es usando la función *download* donde el primer parámetro serán los tickers de todas las empresas que queremos comparar, después el intervalo de días que analizaremos, esto lo que nos permitirá es obtener una tabla comparativa de todos los datos de las empresas separados según su columna, es por lo que al normalmente tener 6 columnas ahora nuestra tabla contendrá 24 ya que tenemos cuatro empresas.

```
DataAcciones = yf.download(['BACHOCOB.MX', 'BIMBOA.MX', 'CEMEXCPO.MX', 'AEROMEX.MX'], start='2019-1-1', end='2022-9-19',
                           interval='1d')
DataAcciones
```

```
[**********100%**********]  4 of 4 completed
```

	Adj Close				Close				High	
	AEROMEX.MX	BACHOCOB.MX	BIMBOA.MX	CEMEXCPO.MX	AEROMEX.MX	BACHOCOB.MX	BIMBOA.MX	CEMEXCPO.MX	AEROMEX.MX	BACHOCOB.M
Date										
2019-01-02	22.530001	62.493568	36.987942	9.523321	22.530001	66.760002	39.320000	9.77	23.080000	67.0899
2019-01-03	22.770000	61.201752	36.987942	9.338118	22.770000	65.379997	39.320000	9.58	23.049999	66.2799
2019-01-04	23.160000	63.354767	37.053795	9.367360	23.160000	67.680000	39.389999	9.61	23.840000	68.7099
2019-01-07	23.270000	63.298607	36.790401	9.640291	23.270000	67.620003	39.110001	9.89	24.500000	69.6699
2019-01-08	23.230000	63.420296	36.931499	9.854735	23.230000	67.750000	39.259998	10.11	23.870001	68.9599
...	...	...	...	...	...	...	...	...	...	
2022-09-09	193.029999	79.690002	73.029999	7.880000	193.029999	79.690002	73.029999	7.88	193.029999	79.8899
2022-09-12	195.500000	79.519997	73.699997	8.000000	195.500000	79.519997	73.699997	8.00	195.500000	80.0000
2022-09-13	195.500000	79.379997	73.029999	7.720000	195.500000	79.379997	73.029999	7.72	195.500000	80.0500
2022-09-14	195.500000	79.239998	72.650002	7.560000	195.500000	79.239998	72.650002	7.56	195.500000	79.7099
2022-09-15	195.500000	79.919998	73.449997	7.260000	195.500000	79.919998	73.449997	7.26	195.500000	80.1200

936 rows × 24 columns

Conclusión

Es interesante el saber cómo se usan librerías específicas, pero ciertos sectores como en este caso para la parte económica, además de que aprendí el cómo es que se usan las diferentes herramientas que nos proporciona. En general nos da una visión de cómo funcionan las acciones en la bolsa, si bien no podríamos dar un juicio por no ser expertos en el área si pudiera tener una vaga idea de cómo funcionan, y una persona especialista en el tema podría sacarle aún más provecho a este tipo de investigaciones.

Liga GitHub Practica 3

Cuaderno de Python de la practica 3

https://github.com/OsvaldoIG/MineriaDatos/tree/main/P3