



## Using Simulation Model For Nestlé and Telefónica

Presentado a:

PhD. MsC. Esp. Ing. Juan Francisco Mendoza Moreno

Presentado por:

Est. Ángel Manuel Correa Rivera

Est. Hugo Emmanuel Hernández Ramírez

Ing. Est. Luis Felipe Narváez Gómez

Universidad Santo Tomas Seccional Tunja

Facultad de Ingeniería de Sistemas

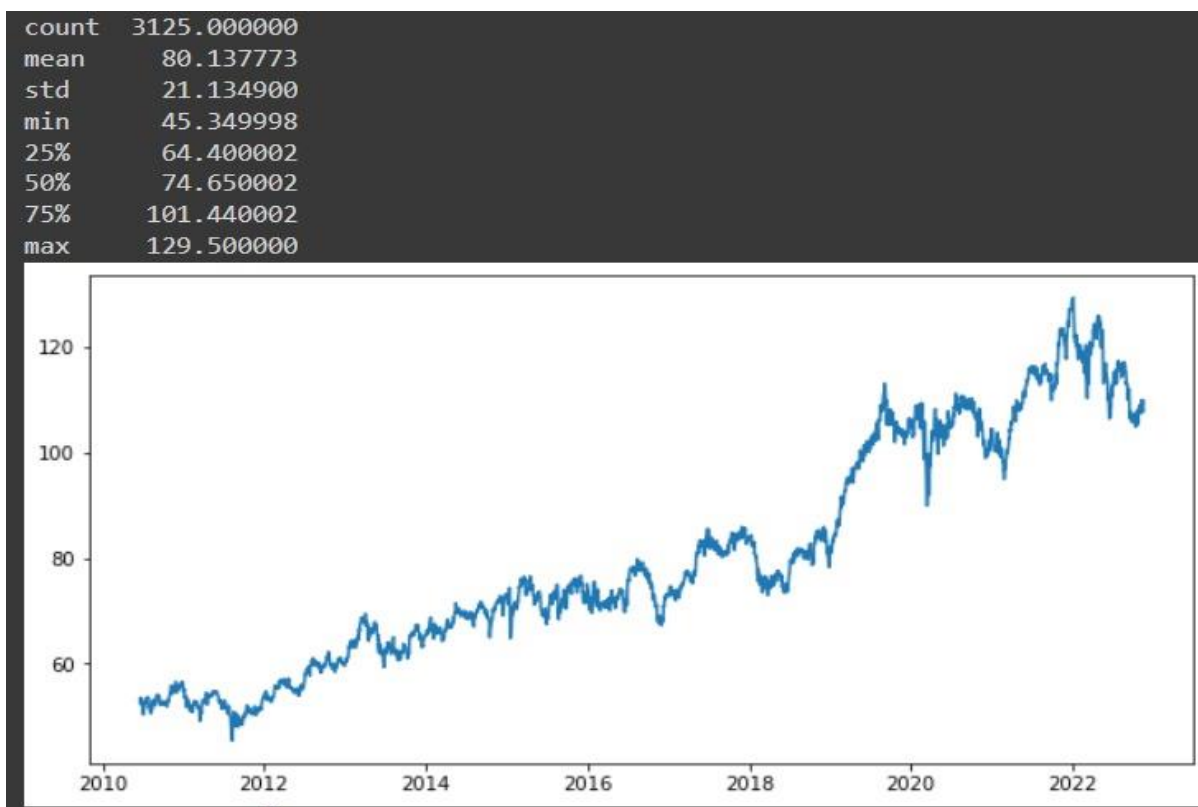
Asignatura de Simulación

2022-2

## 1. ACCIONES DE NESTLE

En este ejercicio se tomaron en cuenta 3125 datos de registro (Fecha y valor en bolsa), los cuales se muestran gráficamente en la siguiente gráfica:

Con este grafico se puede ver la tendencia a lo largo del tiempo de las acciones de Nestlé.



En el anterior grafico se puede ver las acciones que tiene la compañía Nestlé en la bolsa en un transcurso de 12 años (2010-2022).

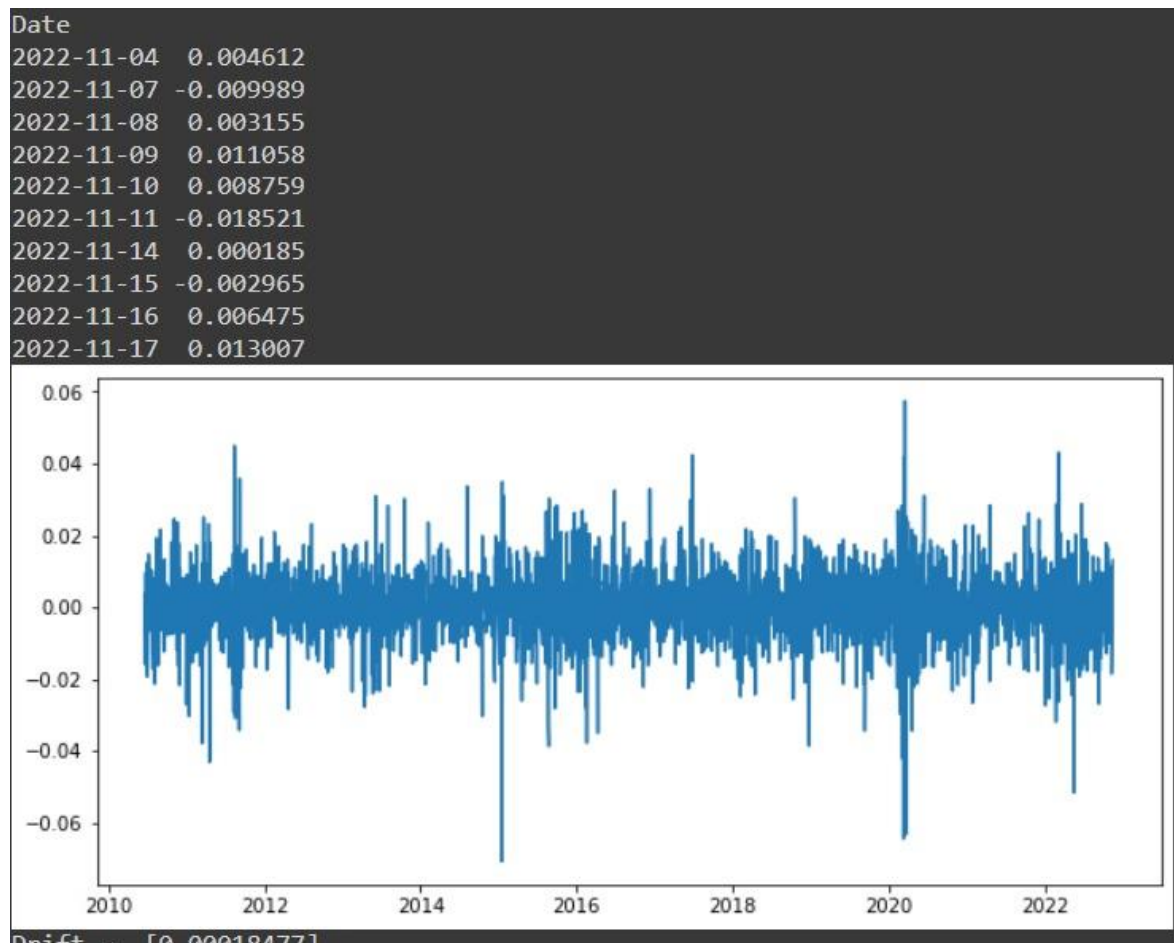
También se obtiene la información sobre la cantidad de datos que se utilizaron, se calcula la media y la desviación estándar. También el puntaje más bajo, el más alto y los porcentajes 25, 50 y 75 entre estos dos.

La media de estos datos es de 80,137773 puntos y su desviación estándar es de 21,134900.

De estos 3125 datos podemos rescatar que el valor mínimo que alcanzaron las acciones de Nestlé fue de 45,349998 puntos en el año 2011 y que su valor máximo fue de 129,5 puntos en el año 2021

## 2. VALORES LOGARÍTMICOS

El cambio porcentual es importante ya que devuelve la información sobre como los datos sufrieron un cambio sobre el tiempo, esto ayuda a controlar la cotización de la acción y sus índices en el mercado.



En esta grafica se obtiene de forma logarítmica la información de cambio sobre el tiempo de las acciones de Nestlé.

Se puede observar que cada día que pasa se obtiene un aumento o disminución en el mercado, y esto se puede observar solo trabajando a una escala logarítmica, ya que los valores devueltos, están más cercanos a la inicial.

### 3. DRIFT NESTLE

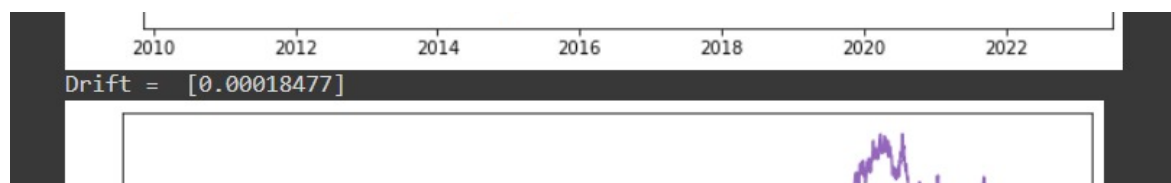
Anteriormente se calculó la media y la desviación estándar, para calcular el drift se calculará también la varianza de los datos.

```
MeanLogReturns = np.array(AtviLogReturns.mean())
VarLogReturns = np.array(AtviLogReturns.var())
StdevLogReturns = np.array(AtviLogReturns.std())
```

Y con la fórmula de drift:

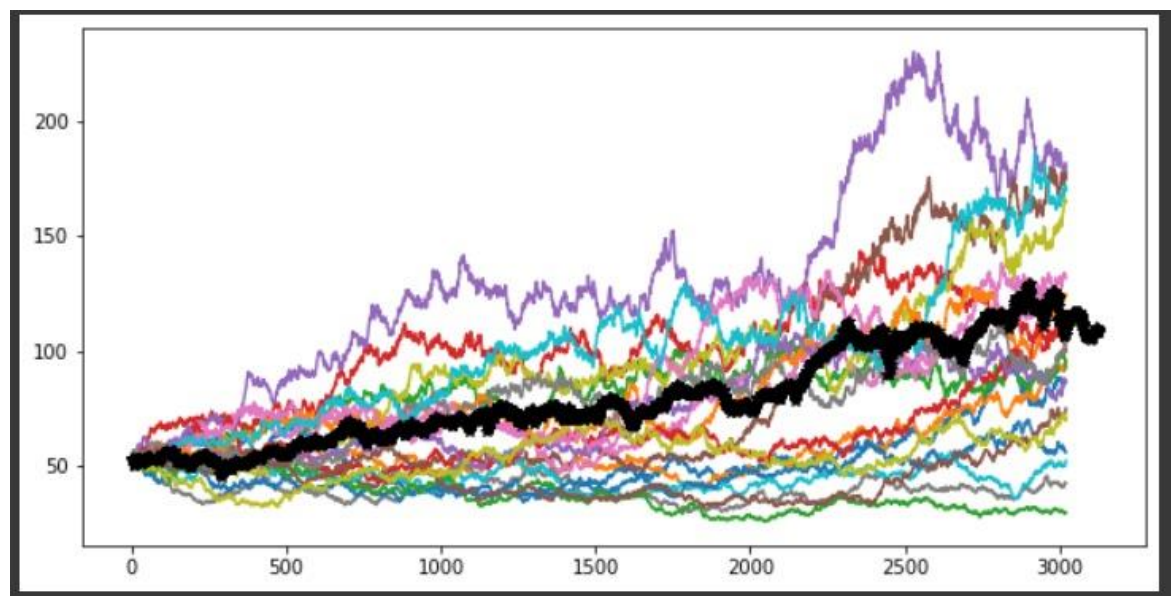
```
Drift = MeanLogReturns - (0.5 * VarLogReturns)
print("Drift = ", Drift)
```

Se calcula un drift de [0,00018477] puntos para la compañía de Nestlé. Este valor ayuda a compensar la asimetría en los resultados.



### 4. PREDICCIONES NESTLE

En la siguiente grafica se pueden evidenciar las diferentes predicciones que se hicieron en el simulador comparadas con la tendencia de los datos originales.



Como se puede observar, la línea negra es la tendencia de los datos originales, y algunas de las simulaciones tuvieron una tendencia similar.

Esto demuestra que con implementando el método de Monte Carlo, se puede llegar a predecir la subida y bajada de una acción en la bolsa de comercio.

También el grafico denota bajo la línea con mayor altura que, si se desea tener una gran ganancia o la máxima, se debería hacer una inversión muy agresiva, a lo contrario con la línea verde se da a entender que de ser invertido muy poco habría muchas perdidas.

## 5. ACCIONES DE TELEFÓNICA

Para las acciones de Telefónica nos encontramos con distintos datasets que utilizar, estos siendo de cada una de las diferentes plazas que la empresa mantiene activas en sus sucursales, cada una de un país diferente y con valores de moneda distintos.

Los datasets de Telefonica recogidos para este ejercicio son 5 que corresponden a las diferentes finanzas dadas en los países donde la empresa cotiza.

1. O2D.DE.csv => Telefónica Deutschland Holding AG (O2D.DE) XETRA - XETRA Delayed Price. Currency in EUR
2. TEF.csv => Telefónica, S.A. (TEF) NYSE - Nasdaq Real Time Price. Currency in USD
3. TEF.MC.csv => Telefónica, S.A. (TEF.MC) MCE - MCE Delayed Price. Currency in EUR
4. VIV.BA.csv => Telefónica Brasil S.A. (VIV.BA) Buenos Aires - Buenos Aires Delayed Price. Currency in ARS
5. VIV.csv => Telefónica Brasil S.A. (VIV) NYSE - Nasdaq Real Time Price. Currency in USD

El valor de la moneda en cada uno de los datasets es la siguiente:

1. O2D.DE.csv => Currency in EUR
2. TEF.csv => Currency in USD
3. TEF.MC.csv => Currency in EUR
4. VIV.BA.csv => Currency in ARS
5. VIV.csv => Currency in USD

Antes de seleccionar alguno de los datasets con el cual trabajar podemos echar un vistazo de su comportamiento de acciones y luego un vistazo de su histórico. Para esto creamos un código de iPython a manera de Notebook, este lo vemos a continuación:

1. Las librerías que utilizamos para el análisis de los datasets son las siguientes:

```
import pandas as pd
import matplotlib.pyplot as plt
```

2. Los métodos que utilizamos son los siguientes:

```
def informacion(dataframe):  
    print("\n informacion general del dataset: ", dataframe.info())  
    print("\n Numero de datos (filas): ", dataframe.shape[0])  
    print("\n Numero de columnas: ", dataframe.shape[1])  
    print("\n Tipos de datos de este dataset: \n", dataframe.dtypes)  
    print("\n Nombre de columnas: ", dataframe.keys())  
    print("\n Cuantos nulos hay en este dataset: \n",  
dataframe.isnull().sum())  
    print("\n descripcion general del dataset: ", dataframe.describe())  
  
def head_data (dataframe):  
    return dataframe.head(3)  
  
def graph_mode(dataframe):  
    x = dataframe['Date']  
    y = dataframe['Close']  
  
    fig, ax = plt.subplots(figsize=(20,5))  
    ax.fill_between(x, y, alpha=0.5, color='g')  
  
    plt.show()
```

El primer método le llega por parámetro un dataframe, este corresponde a la lectura del dataset “n” que se esté analizando al momento de ser utilizado. Trae la información general de este dataframe como lo puede ser:

- Información general.
- Numero de filas o cantidad total de datos.
- Numero de columnas.
- Tipos de datos que contiene el dataframe
- El nombre de las columnas de la información que yace en el dataframe.
- Si hay datos nulos en cada una de las columnas y cuantos son.
- Descripción general del dataframe.

El segundo método trae una vista de “n” filas del dataframe con el que se este trabajando a manera de tabla de datos.

El tercer método elabora un gráfico 2D donde e evidencie el cambio de las acciones con respecto al tiempo o el histórico del mismo dataset, esto se logra al comparar los datos de tiempo para el eje x de la grafica y el valor de cierre de la acción en el eje y.

3. Ahora se tiene el código de la lectura de cada uno de los 5 datasets que hallamos, con la librería de pandas podemos leer los archivos descargables en formato CSV

separado por comas y grabar cada archivo como un dataframe individual con el cual trabajar.

```
# Rutas de los datasets
route_1 = './datasets/telefonica/O2D.DE.csv'
route_2 = './datasets/telefonica/TEF.csv'
route_3 = './datasets/telefonica/TEF.MC.csv'
route_4 = './datasets/telefonica/VIV.BA.csv'
route_5 = './datasets/telefonica/VIV.csv'

# Leer los datasets
df_eins = pd.read_csv(route_1)
df_zwei = pd.read_csv(route_2)
df_drei = pd.read_csv(route_3)
df_vier = pd.read_csv(route_4)
df_fünf = pd.read_csv(route_5)
```

- Ahora vamos a ver la información obtenida de cada uno de los dataframes que corresponden a los 5 datasets. Para esto se utiliza la invocación de los métodos que ya creamos previamente, esto de la siguiente manera:

```
head_data(escribe aqui tu dataframe)
graph_mode(escribe aqui tu dataframe)
informacion(escribe aqui tu dataframe)
```

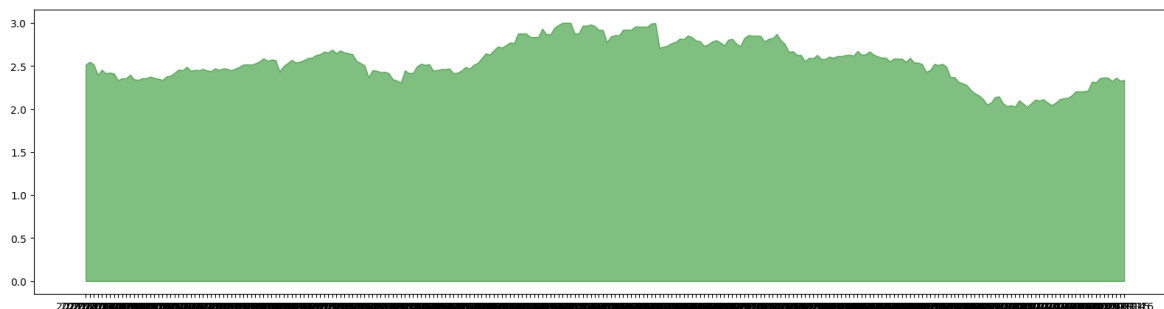
Cada una de las anteriores líneas de código pueden ser utilizadas en cualquier orden pues no están alterando la integridad de los datos de los dataframes, solo está expresando información de ellos.

Para el Dataset 1 tenemos:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2021-11-16	2.524	2.543	2.515	2.516	2.360851	5740347
1	2021-11-17	2.520	2.549	2.513	2.549	2.391816	4492715
2	2021-11-18	2.537	2.550	2.501	2.519	2.363666	3678836
3	2021-11-19	2.521	2.528	2.391	2.397	2.249189	7262573
4	2021-11-22	2.401	2.470	2.401	2.456	2.304551	6509221
...	...	...	...	...	...	...	...
253	2022-11-10	2.363	2.372	2.303	2.364	2.364000	4851726
254	2022-11-11	2.379	2.379	2.288	2.325	2.325000	6608821
255	2022-11-14	2.330	2.376	2.324	2.365	2.365000	3529417
256	2022-11-15	2.358	2.358	2.308	2.329	2.329000	3625765
257	2022-11-16	2.322	2.349	2.308	2.339	2.339000	3460418

258 rows x 7 columns



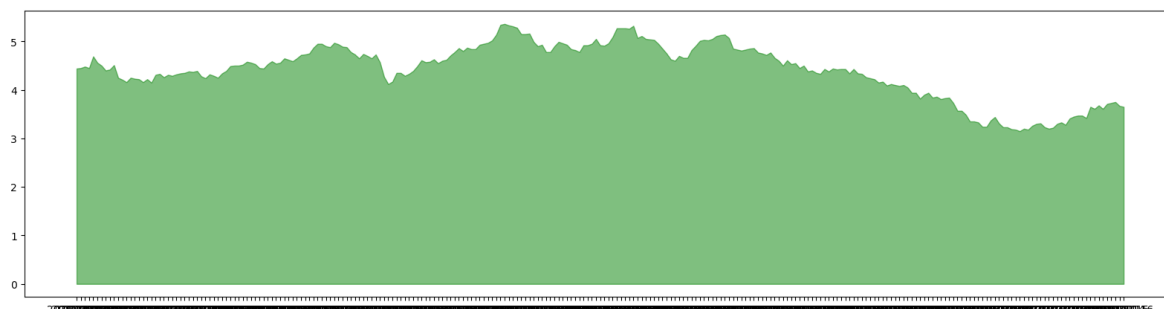


Para el Dataset 2 tenemos:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2021-11-16	4.53	4.54	4.42	4.44	4.149565	1114800
1	2021-11-17	4.40	4.45	4.40	4.45	4.158911	1401900
2	2021-11-18	4.50	4.51	4.46	4.48	4.186949	2106200
3	2021-11-19	4.49	4.51	4.42	4.45	4.158911	1783500
4	2021-11-22	4.64	4.72	4.63	4.69	4.383212	2725300
...	...	...	...	...	...	...	...
248	2022-11-10	3.73	3.75	3.69	3.71	3.710000	1086700
249	2022-11-11	3.72	3.74	3.69	3.73	3.730000	764900
250	2022-11-14	3.75	3.79	3.75	3.75	3.750000	1080100
251	2022-11-15	3.72	3.72	3.67	3.67	3.670000	1580200
252	2022-11-16	3.63	3.66	3.62	3.65	3.650000	612513

253 rows × 7 columns

```
RangeIndex: 253 entries, 0 to 252
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Date        253 non-null   object
1   Open        253 non-null   float64
2   High        253 non-null   float64
3   Low         253 non-null   float64
4   Close       253 non-null   float64
5   Adj Close   253 non-null   float64
6   Volume      253 non-null   int64
dtypes: float64(5), int64(1), object(1)
```

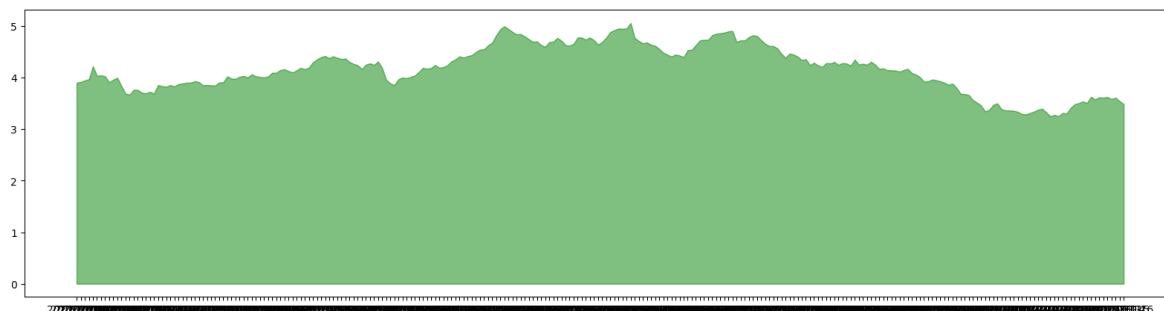


Para el Dataset 3 tenemos:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2021-11-16	3.9085	3.9840	3.8990	3.9000	3.523742	43630864
1	2021-11-17	3.9000	3.9230	3.8750	3.9160	3.538198	9450597
2	2021-11-18	3.9055	3.9875	3.9000	3.9455	3.564852	30913830
3	2021-11-19	3.9840	4.0080	3.9085	3.9675	3.584730	32014850
4	2021-11-22	4.0265	4.2400	4.0240	4.2210	3.813773	61007552
...	...	...	...	...	...	...	...
253	2022-11-10	3.6140	3.6800	3.5960	3.6240	3.624000	15750363
254	2022-11-11	3.6200	3.6260	3.5510	3.5830	3.583000	13069345
255	2022-11-14	3.6000	3.6500	3.5930	3.6110	3.611000	9987446
256	2022-11-15	3.6050	3.6220	3.5260	3.5450	3.545000	16486168
257	2022-11-16	3.5390	3.5430	3.4630	3.4870	3.487000	15511990

258 rows × 7 columns

```
RangeIndex: 258 entries, 0 to 257
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Date        258 non-null   object
1   Open        258 non-null   float64
2   High        258 non-null   float64
3   Low         258 non-null   float64
4   Close       258 non-null   float64
5   Adj Close   258 non-null   float64
6   Volume      258 non-null   int64
dtypes: float64(5), int64(1), object(1)
```

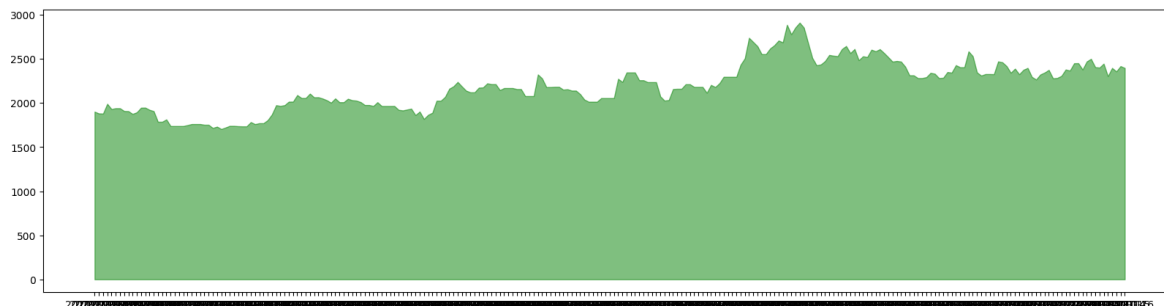


Para el Dataset 4 tenemos:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2021-11-16	1901.0	1901.0	1901.0	1901.0	1900.389404	0
1	2021-11-17	1881.0	1881.0	1881.0	1881.0	1880.395752	41
2	2021-11-18	1915.5	1915.5	1879.0	1879.0	1878.396362	6
3	2021-11-19	1960.0	2009.5	1960.0	1990.0	1989.360718	4984
4	2021-11-23	1930.0	1948.0	1920.0	1930.5	1929.879883	2279
...	...	...	...	...	...	...	...
240	2022-11-10	2315.0	2344.0	2302.0	2303.5	2303.500000	236
241	2022-11-11	2401.5	2409.0	2389.0	2396.5	2396.500000	397
242	2022-11-14	2369.0	2373.5	2357.0	2357.0	2357.000000	43
243	2022-11-15	2346.0	2417.0	2346.0	2417.0	2417.000000	346
244	2022-11-16	2416.0	2416.0	2395.0	2396.0	2396.000000	6

245 rows × 7 columns

```
RangeIndex: 245 entries, 0 to 244
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date        245 non-null   object
1   Open        245 non-null   float64
2   High        245 non-null   float64
3   Low         245 non-null   float64
4   Close       245 non-null   float64
5   Adj Close   245 non-null   float64
6   Volume      245 non-null   int64
dtypes: float64(5), int64(1), object(1)
```

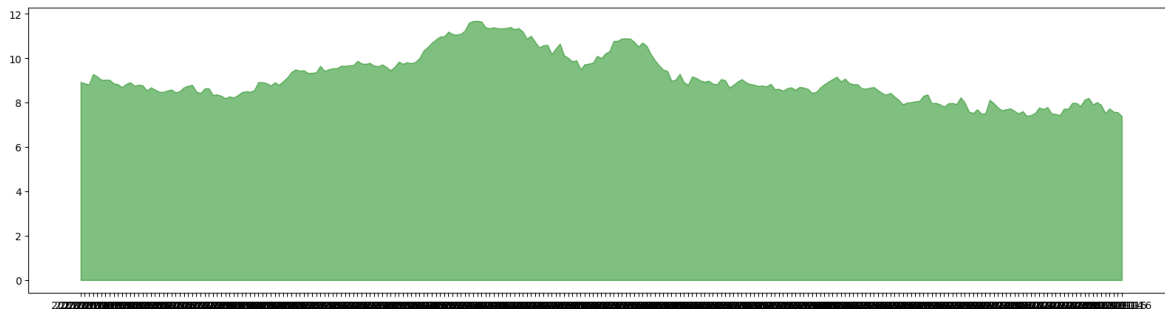


Para el Dataset 5 tenemos:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2021-11-16	8.89	8.95	8.83	8.93	8.422136	821300
1	2021-11-17	8.86	8.91	8.81	8.87	8.365548	880200
2	2021-11-18	8.87	8.94	8.82	8.82	8.318392	970900
3	2021-11-19	9.28	9.42	9.18	9.29	8.761662	2514400
4	2021-11-22	9.30	9.41	9.16	9.18	8.657919	2576400
...	...	...	...	...	...	...	...
248	2022-11-10	7.60	7.67	7.43	7.53	7.530000	2448600
249	2022-11-11	7.77	7.83	7.67	7.74	7.740000	2734000
250	2022-11-14	7.71	7.73	7.56	7.59	7.590000	1925100
251	2022-11-15	7.75	7.76	7.51	7.57	7.570000	1952300
252	2022-11-16	7.52	7.54	7.36	7.39	7.390000	694321

253 rows × 7 columns

```
RangeIndex: 253 entries, 0 to 252
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Date        253 non-null   object
1   Open        253 non-null   float64
2   High        253 non-null   float64
3   Low         253 non-null   float64
4   Close       253 non-null   float64
5   Adj Close   253 non-null   float64
6   Volume      253 non-null   int64
dtypes: float64(5), int64(1), object(1)
```



Cabe mencionar que lo expresado en el presente documento en la imagen correspondiente a la información retornada al ejecutar el método para adquirir información está incompleta, sin embargo, la imagen ejemplifica de forma correcta el tipo de datos que pueden extraerse de un dataframe.

Cada una de las plazas de telefónica posee un histórico de rango distinto esto debido a que cada una de ellas empieza a operar en tiempos distintos. Esto sin mencionar que para el ejercicio se debe tener una data que recomendablemente no tenga valores en nulos o 0 en la columna del valor de la acción en el corte, así como un histórico amplio para poder “predecir” la posible inversión.

En yahoo finance existe un historico maximo de:

- |   |                                |
|---|--------------------------------|
| 1. Telefónica Deutschland Holding AG (02D.DE) | => Dec 12, 2013 - Nov 20, 2022 |
| 2. Telefônica Brasil S.A. (VIV)               | => Nov 15, 1998 - Nov 20, 2022 |
| 3. Telefónica, S.A. (TEF)                     | => Jun 11, 1987 - Nov 20, 2022 |
| 4. Telefónica, S.A. (TEF.MC)                  | => Jan 02, 2000 - Nov 20, 2022 |
| 5. Telefónica Chile S.A. (CTC.SN)             | => Mar 12, 2018 - Nov 20, 2022 |

Como podemos observar en el registro historico, el que mas mantiene dada es:

- |                           |                                |
|---------------------------|--------------------------------|
| 3. Telefónica, S.A. (TEF) | => Jun 11, 1987 - Nov 20, 2022 |
|---------------------------|--------------------------------|

Así pues, con el dataset escogido tenemos:

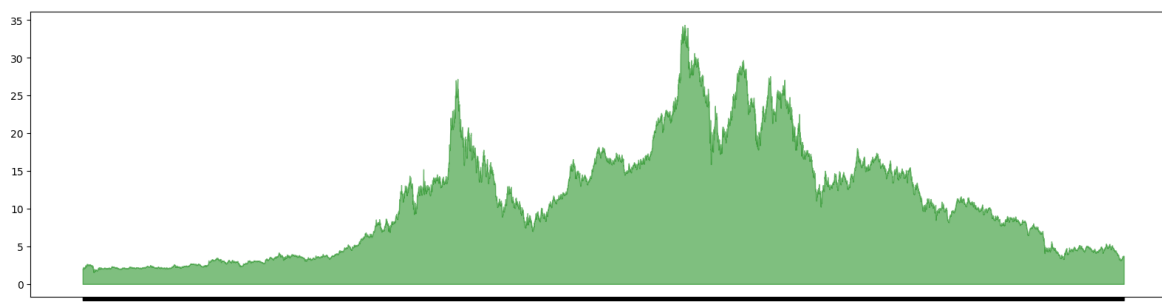
```
route = './datasets/TEF.csv'
df_telefonica = pd.read_csv(route)
```

```
head_data(df_telefonica)
df_telefonica.head(df_telefonica.shape[0])
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	1987-06-12	2.006067	2.085854	1.960475	2.051660	0.460372	50527023
1	1987-06-15	2.108650	2.120048	2.051660	2.063058	0.462930	17437104
2	1987-06-16	2.063058	2.063058	1.971873	1.983271	0.445026	14805088
3	1987-06-17	2.006067	2.006067	1.971873	1.994669	0.447584	4271542
4	1987-06-18	1.994669	2.085854	1.994669	2.085854	0.468045	8514571
...	...	...	...	...	...	...	...
8930	2022-11-16	3.630000	3.660000	3.620000	3.630000	3.630000	1409000
8931	2022-11-17	3.600000	3.700000	3.590000	3.680000	3.680000	1718600
8932	2022-11-18	3.670000	3.690000	3.660000	3.680000	3.680000	680200
8933	2022-11-21	3.690000	3.710000	3.680000	3.680000	3.680000	904000
8934	2022-11-22	3.690000	3.740000	3.690000	3.740000	3.740000	840100

8935 rows × 7 columns

```
graph_mode(df_telefonica)
```



```
informacion(df_telefonica)
```

		RangeIndex: 8935 entries, 0 to 8934					
		Data columns (total 7 columns):					
Date	object	#	Column	Non-Null Count	Dtype	Date	0
Open	float64	---	-----	-----	-----	Open	0
High	float64	0	Date	8935 non-null	object	High	0
Low	float64	1	Open	8935 non-null	float64	Low	0
Close	float64	2	High	8935 non-null	float64	Close	0
Adj Close	float64	3	Low	8935 non-null	float64	Adj Close	0
Volume	int64	4	Close	8935 non-null	float64	Volume	0
dtype: object		5	Adj Close	8935 non-null	float64	dtype: int64	
		6	Volume	8935 non-null	int64		

Numero de datos (filas): 8935

Numero de columnas: 7

De esta manera podemos darnos cuenta fácilmente que la data esta libre de datos nulos, quizá no de ceros, pero si de nulos. Que su histórico es desde 1987 hasta el año actual 2022, que los datos de las columnas numéricos son computables al tratarse de tipos floats o int y que la cantidad de datos totales es de 8935. Además de esto ya podemos ver un comportamiento de las acciones en todo su histórico con la grafica y hacernos una idea general antes de proseguir con el código.

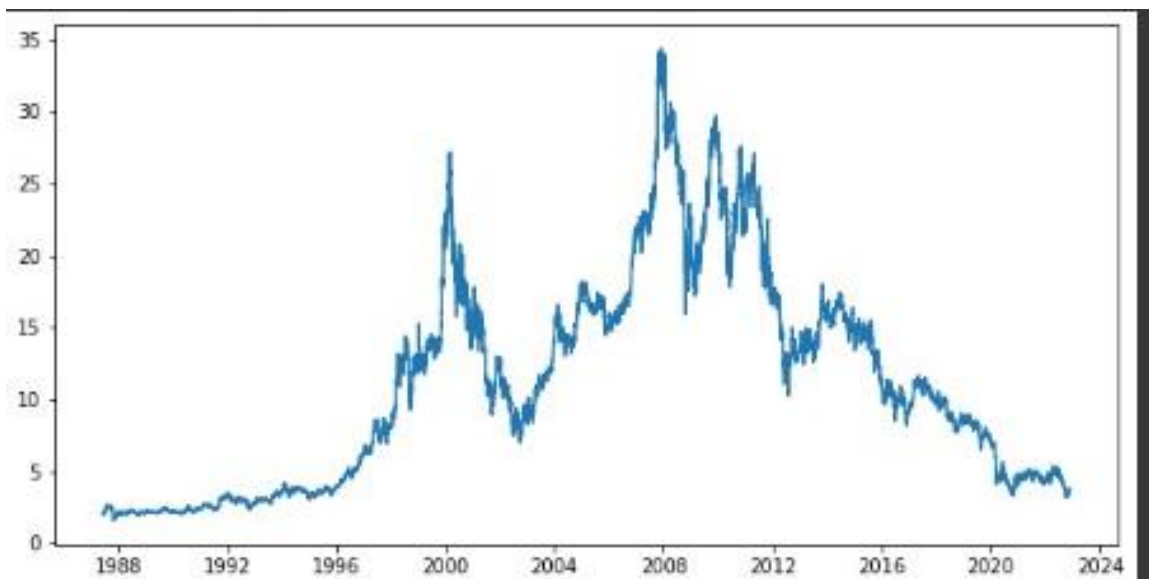
En este ejercicio se tomaron en cuenta 8935 datos de registro (Fecha y valor en bolsa), los cuales se muestran gráficamente en la siguiente gráfica:

Con este grafico se puede ver la tendencia a lo largo del tiempo de las acciones de Telefónica.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 8935 entries, 1987-06-12 to 2022-11-22
Data columns (total 1 columns):
#   Column  Non-Null Count  Dtype
---  ---
0    Close    8935 non-null       float64
dtypes: float64(1)
memory usage: 139.6 KB
None
```

Date	Close
1987-06-12	2.051660
1987-06-15	2.063058
1987-06-16	1.983271
1987-06-17	1.994669
1987-06-18	2.085854
2022-11-16	3.63
2022-11-17	3.68
2022-11-18	3.68
2022-11-21	3.68
2022-11-22	3.74

```
count    8935.000000
mean      11.046073
std       7.400561
min       1.572939
25%       3.899077
50%      10.235502
75%      15.810938
max      34.369999
```



En el anterior grafico se puede ver las acciones que tiene la compañía Telefónica en la bolsa en un transcurso de 14 años (2010-2024).

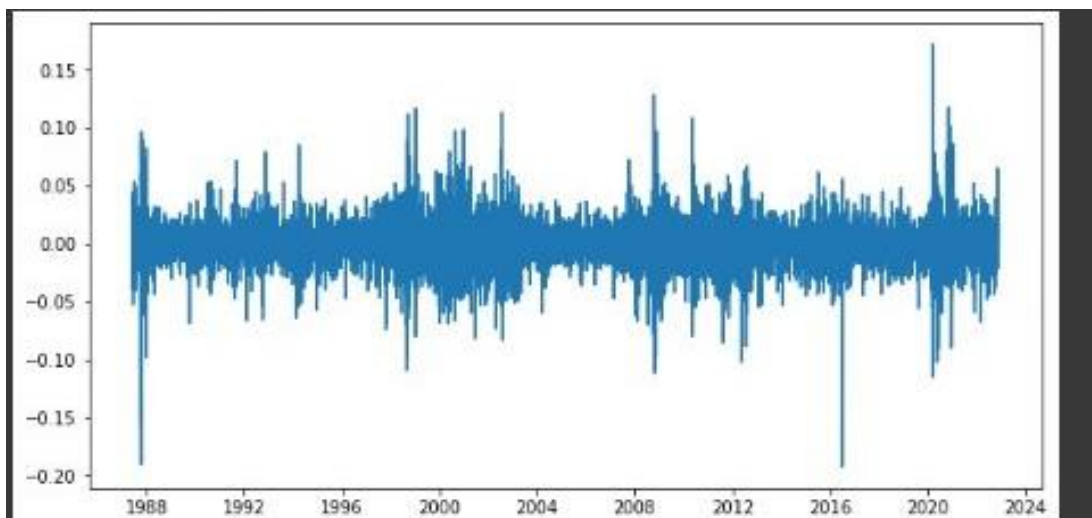
También se obtiene la información sobre la cantidad de datos que se utilizaron, se calcula la media y la desviación estándar. También el puntaje más bajo, el más alto y los porcentajes 25, 50 y 75 entre estos dos.

La media de estos datos es de 11.04 puntos y su desviación estándar es de 7,4.

De estos 8935 datos podemos rescatar que el valor mínimo que alcanzaron las acciones de Telefónica fue de 1,5 puntos en el año 2011 y que su valor máximo esperado es de 34,36 puntos para el año 2024.

## 6. VALORES LOGARÍTMICO

Date	Close
2022-11-09	-0.019205
2022-11-10	0.027324
2022-11-11	0.005376
2022-11-14	0.005348
2022-11-15	-0.021564
2022-11-16	-0.010959
2022-11-17	0.013680
2022-11-18	0.000000
2022-11-21	0.000000
2022-11-22	0.016173



En esta grafica se obtiene de forma logarítmica la información de cambio sobre el tiempo de las acciones de Telefónica.

Se puede observar que cada día que pasa se obtiene un aumento o disminución en el mercado, y esto se puede observar solo trabajando a una escala logarítmica, ya que los valores devueltos, están más cercanos a la inicial.

## 7. DRIFT TELEFÓNICA

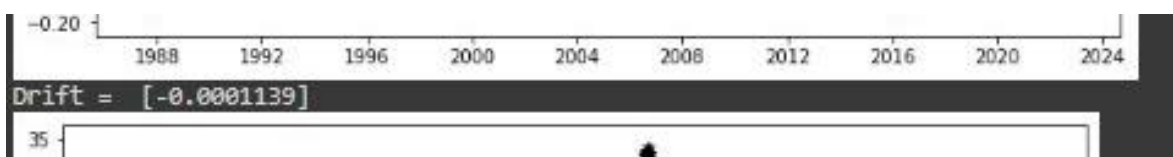
Anteriormente se calculó la media y la desviación estándar, para calcular el drift se calculará también la varianza de los datos.

```
MeanLogReturns = np.array(AtviLogReturns.mean())
VarLogReturns = np.array(AtviLogReturns.var())
StdevLogReturns = np.array(AtviLogReturns.std())
```

Y con la fórmula de drift:

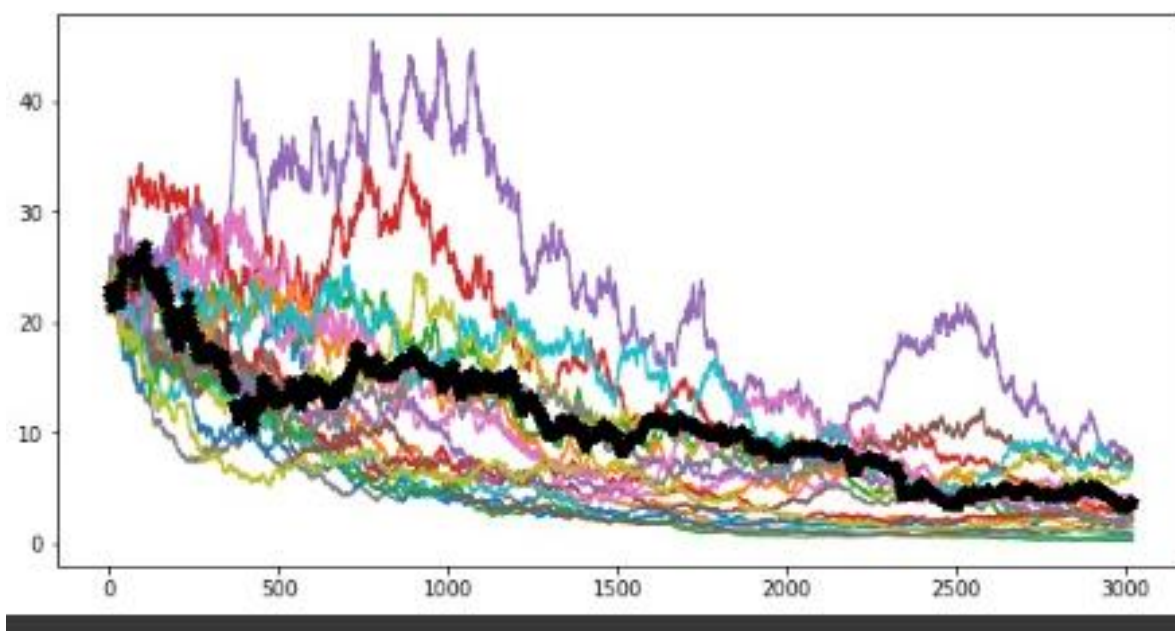
```
Drift = MeanLogReturns - (0.5 * VarLogReturns)
print("Drift = ", Drift)
```

Se calcula un drift de [-0,0001139] puntos para la compañía de Telefónica. Este valor ayuda a compensar la asimetría en los resultados.





## 8. PREDICCIONES TELEFÓNICA



Como se puede observar, la línea negra es la tendencia de los datos originales, y algunas de las simulaciones tuvieron una tendencia similar.

Esto demuestra que con implementando el método de Monte Carlo, se puede llegar a predecir la subida y bajada de una acción en la bolsa de comercio.

Con telefónica ocurre un caso especial, y es que, por medio de Monte Carlo, se predice que durante 2022-2023 puede llegar a subir las acciones a su punto mas alto, sin embargo, para el año 2024, estas acciones van a empezar a desplomarse.