

Ciclo 1 Fundamentos de Programación

Reto 5: Inversiones Sector Energético - Petróleo



Inversionistas del sector energético colombiano, necesitan determinar el momento adecuado para negociar. Los consultores de apoyo, han recolectado el valor del dólar o Tasa Representativa del Mercado (TRM – Peso colombiano por dólar de los Estados Unidos), el valor de las acciones de Ecopetrol, y el número de contagios de COVID19 del último trimestre, como base para el análisis. Los inversores necesitan observar el comportamiento de las acciones de Ecopetrol los días en que los contagios son inferiores a la media de casos, desde el primero de marzo de 2021 hasta los primeros días de junio de 2021, y observar si se presenta alguna correlación.

El indicador debe construirse obteniendo el valor promedio de variación absoluta de las acciones de Ecopetrol, la cual se calcula de la siguiente manera:

$$\text{Variación} = \frac{|\text{Precio anterior} - \text{Precio actual}|}{\text{Precio anterior}} * 100$$

Los inversionistas necesitan conocer el promedio de variación de las acciones, junto con los registros (días) donde el número de contagios es igual o menor a la media, para analizar el impacto que los casos de COVID-19 pueden tener en el sector energético/petrolero.

Para lo anterior, se requiere una función que procese una base de datos en formato CSV (del inglés, *comma-separated values*), que contiene fecha, la TRM, el número de casos presentados el día registrado, y el valor de la acción de Ecopetrol. A continuación se especifica el tratamiento que se debe realizar de la información:

- 1) Validar que el origen de los datos tenga formato (extensión) CSV antes de realizar cualquier cómputo, retornando el mensaje “Extensión inválida” en caso de que no cumpla.
- 2) Intentar realizar la carga del archivo CSV a un dataframe de pandas con separación de comas “,”. En caso de que falle el intento, retornar el mensaje “Error al leer el archivo de datos”.



- 3) Cambiar el campo que contiene la fecha en formato string a tipo fecha de pandas, y convertirla en el índice del dataframe, dado que este especifica el día de cada registro, pero no se realizarán cálculos con esta información.
- 4) Calcular el valor promedio de la variación absoluta de las acciones de Ecopetrol.
- 5) Obtener un dataframe que contenga únicamente los registros donde el número de contagios es menor o igual a la media de casos de COVID-19.

A continuación se especifican los detalles técnicos del requerimiento. El esqueleto de la función, indicado en la Figura 1, recibe la ruta del archivo (que podría ser también una dirección de Internet), y retorna un diccionario con el promedio de variación, y el dataframe resultante de la manipulación indicada en los numerales anteriores.

```
def indicadoresPetroleo(rutaArchivo: str)->dict:  
    """Desarrollar aquí el requerimiento"""  
    pass
```

Figura 1. Prototipo de la función

Especificación de la entrada

Entrada	Tipo de Dato	Descripción
rutaArchivo	string	Ruta del archivo, la cual podría ser local (mismo directorio) o podría ser una dirección web. Tener esto en cuenta para la validación de formato.

El archivo o base de datos que se requiere procesar, se encuentra en el siguiente enlace:
<https://github.com/luismescobarf/clasesCiclo1/blob/master/BaseDeDatosReto5.csv>

Se recomienda para la realización de las pruebas y observación de la base de datos, descargar el archivo del enlace anterior, en la misma carpeta donde se desarrollará la función del requerimiento.

Estructura de la Salida

Tipo de Retorno	Descripción
dict	Diccionario con el promedio de variación de las acciones de Ecopetrol (float) y dataframe tipo pandas con los registros que son iguales o inferiores al promedio de casos.

Ejemplo Salida:



{'Promedio Variación': 1.3354164974888902, 'Registros Menores':

TRM Casos

Ecopetrol

Fecha

2021-05-03 3740.14 11599 2219.00

2021-04-07 3645.14 11381 2337.00

.

.

.

2021-03-02 3622.36 4339 2245.00

2021-03-01 3624.39 3570 2246.00}

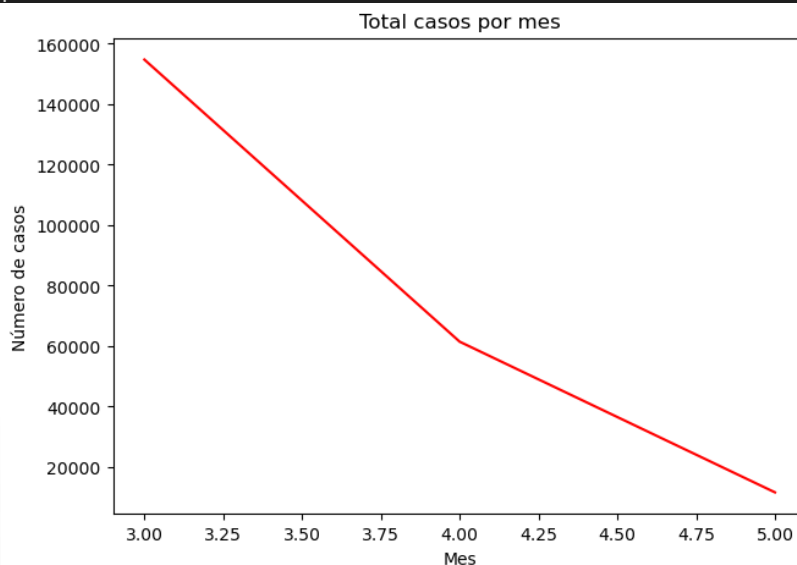
Visualizaciones Complementarias:

A continuación se presentan las gráficas que deben obtenerse si el tratamiento de los datos fue el adecuado. Es importante resaltar que estas gráficas son opcionales y **no deben ser incluidas en la función que se subirá a la plataforma iMaster**. Suponiendo que en el dataframe denominado *data*, se ha almacenado el dataframe denominado en el diccionario de retorno con la llave 'Registros Menores', las siguientes instrucciones de graficado deben producir la visualización correspondiente:

Visualización 1

```
#Graficado opcional de los casos acumulados en cada mes después de la  
manipulación
```

```
import matplotlib.pyplot as plt  
dataCasosMes = data["Casos"].groupby(data.index.month).sum()  
dataCasosMes.plot(color="red")  
plt.title("Total casos por mes")  
plt.xlabel("Mes")  
plt.ylabel("Número de casos")  
plt.show()
```

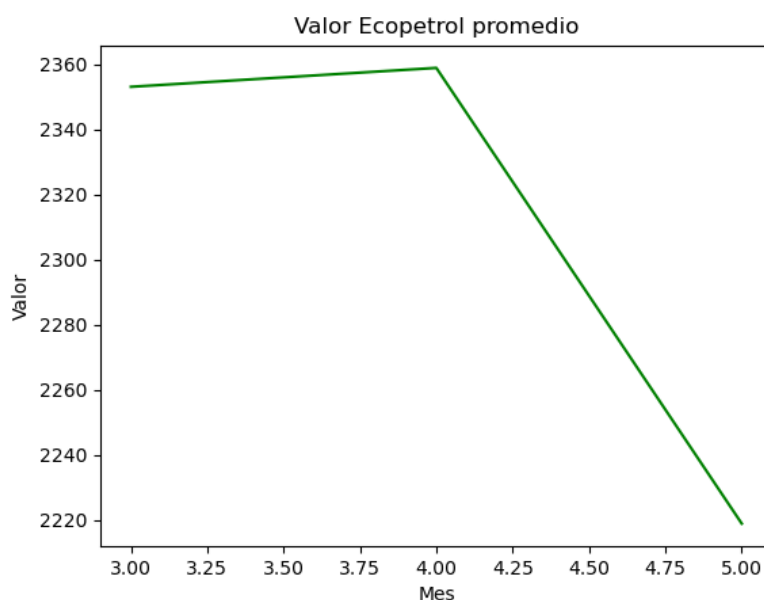




Visualización 2

```
#Graficado opcional del valor medio de la acción de Ecopetrol de cada mes después  
de la manipulación
```

```
import matplotlib.pyplot as plt  
dataMediaPrecioMes = data['Ecopetrol'].groupby(data.index.month).mean()  
dataMediaPrecioMes.plot(color = "green")  
plt.title("Valor Ecopetrol promedio")  
plt.xlabel("Mes")  
plt.ylabel("Valor")  
plt.show()
```



Partiendo de lo anterior, los casos de prueba son los siguientes:

Caso de Prueba 1:

Llamado:

```
print(indicadoresPetroleo('BaseDeDatosReto5.zip'))
```

Resultado esperado:

```
Extensión inválida
```

Caso de Prueba 2:

Llamado:

```
print(indicadoresPetroleo('BaseDeDatos.csv'))
```



Resultado esperado:

Error al leer el archivo de datos

Caso de Prueba 3:

Llamado:

```
print(indicadoresPetroleo('https://raw.githubusercontent.com/luismescobarf/clases  
Ciclo1/master/BaseDeDatosReto5.csv'))
```

Resultado esperado:

```
{'Promedio Variación': 1.33541649748889, 'Registros Menores':  
Casos Ecopetrol TRM  
Fecha  
2021-05-03 3740.14 11599 2219.00  
2021-04-07 3645.14 11381 2337.00  
2021-04-05 3678.62 10190 2325.00  
2021-04-04 3678.62 9022 2372.77  
2021-04-03 3678.62 9149 2371.07  
2021-04-02 3678.62 10222 2371.56  
2021-04-01 3678.62 11449 2375.67  
2021-03-31 3736.91 8646 2366.00  
2021-03-30 3705.85 7952 2373.00  
2021-03-29 3665.41 7049 2388.00  
2021-03-28 3665.41 7139 2329.75  
2021-03-27 3665.41 8254 2320.00  
2021-03-26 3658.22 7395 2351.00  
2021-03-25 3635.12 6732 2315.00  
2021-03-24 3589.82 5986 2333.00  
2021-03-23 3553.34 4946 2281.00  
2021-03-22 3553.34 5128 2402.50  
2021-03-21 3553.34 5963 2412.08  
2021-03-20 3553.34 6761 2423.40  
2021-03-19 3569.45 5133 2370.00  
2021-03-18 3578.02 5139 2350.00  
2021-03-17 3553.51 4554 2457.00  
2021-03-16 3575.63 3716 2460.00  
2021-03-15 3575.30 2740 2480.00  
2021-03-14 3575.30 4062 2363.80  
2021-03-13 3575.30 4465 2361.00  
2021-03-12 3534.62 4078 2378.00  
2021-03-11 3561.91 4579 2368.00  
2021-03-10 3598.77 3588 2373.00
```



2021-03-09	3623.61	3511	2339.00
2021-03-08	3640.20	2205	2347.00
2021-03-07	3640.20	3411	2292.80
2021-03-06	3640.20	3663	2285.00
2021-03-05	3647.99	3371	2400.00
2021-03-04	3676.94	3565	2279.00
2021-03-03	3646.61	3047	2255.00
2021-03-02	3622.36	4339	2245.00
2021-03-01	3624.39	3570	2246.00

Recordar: En la plataforma debe subirse una función con **el mismo nombre, la misma cantidad de argumentos, y el retorno debe tener exactamente la estructura de la salida que se presenta en este documento.**