

Requerimiento #4 - Información de Rentabilidad

En su cuarta semana de trabajo, su jefe le pide ayuda con uno de los reportes que se envían al Directorio, relacionado con la rentabilidad de las acciones de la empresa. El requerimiento es el siguiente:

Buenos días,

Desde GG me preguntan si puedes contribuir al Dashboard que se está preparando para el directorio. Específicamente, nos piden un código en Python que permita crear el cuadro resumen con la rentabilidad de las acciones de la empresa. Por si no lo has visto, el cuadro resumen de rentabilidad es un reporte con esta información:

Instrumento	Ult. Precio	Rent 1d	Rent 1m	Rent ytd*	Rent 1y*	Rent 5y*
A	100	1,4%	1,4%	-0,7%	1,5%	0,1%
B	200	0,0%	1,9%	1,9%	-0,3%	-0,4%
IPSA	4000	1,6%	1,8%	-0,8%	1,5%	-0,1%
Dólar	920	-0,8%	1,6%	0,6%	0,5%	-0,4%
Fecha de referencia: 29-10-2023						
(*) Rentabilidades anualizadas						

Obviamente, A y B son ejemplos. Debes usar los nemotécnicos asociados a la empresa. Además, notar que hay dos filas extras: una para la rentabilidad del IPSA, y otra para el dólar.

Te dejo una base SQL con los precios de acciones y el IPSA que se armó en otro proyecto. Como esta base no tiene dato de dólar, te pido por favor armar una pequeña consulta que se conecte al Banco Central y descargue esos datos directo de allí.

El código que armes debe ser capaz de tomar distintas fechas de referencia y nemotécnicos. Además del código, favor preparar un Excel con el cuadro de rentabilidades usando como fecha de referencia el 29 de septiembre de 2023.

El Cálculo de Rentabilidad

Uno de sus colegas le explica algunos conceptos antes de partir:

- La “fecha de referencia” se refiere a la fecha principal sobre la que se arma el cuadro.
- Así, el campo “Ult. Precio” corresponde al precio de la acción (o dólar) observado en la fecha de referencia dada.

Sobre las otras columnas del cuadro resumen, le explica:

- “Rent 1d” significa rentabilidad a 1 día. Vale decir, el cambio porcentual de precio entre un día (bursátil) anterior y la fecha de referencia.
- “Rent 1m” significa la rentabilidad a 1 mes (respecto a la fecha de referencia).
- “Rent ytd” significa rentabilidad entre el 1 de enero del año en curso, y la fecha de referencia. En este caso, se debe presentar anualizada.
- “Rent 1y” significa rentabilidad a un año (y la fecha de referencia). Por construcción, está anualizada.
- “Rent 5y” significa rentabilidad a cinco años (y la fecha de referencia). En este caso, se debe presentar anualizada.

Finalmente, le da algunos tips que podrían servir para hacer más robusto (y útil) el código para armar el cuadro de rentabilidades:

- Si en la “fecha de referencia” dada no hay información de precios (por ejemplo, porque se consultó una fecha en el futuro para la cual no hay datos), el cuadro se debe armar para la última fecha disponible en la base.
- Puede ocurrir que, al buscar información hacia el pasado para calcular las rentabilidades pasadas, no exista información de precios tan antigua (por ej, puede ser que la empresa no existía 5 años antes de alguna fecha de referencia).
 - Si ocurre lo anterior, el cuadro de rentabilidades debería mostrar el valor “NaN”.
- Puede ocurrir también que, al buscar información hacia el pasado, no exista información de precios en la base de datos, ya que ese día en particular fue feriado o fin de semana.
 - Si ocurre lo anterior, el código debería buscar el último día disponible, partiendo desde la fecha dada.

El Código

Para responder el requerimiento, le recomienda lo siguiente:

- El objetivo principal es crear una función en Python que tome una fecha de referencia y una lista de nemotécnicos, y le entregue un DataFrame con la información del cuadro de rentabilidades. Esta función debe entregar además un segundo output, correspondiente a la fecha de referencia utilizada para el cálculo (que podría ser distinta a la fecha de referencia ingresada por el usuario).
- Llame a la función anterior **“rentabilidades”**, y **guárdela en el archivo “auxiliares4.py”**

- Dicha función puede, internamente, llamar a otras funciones auxiliares. Por ejemplo, podrían haber funciones auxiliares para:
 - Determinar si existe información para la fecha de referencia; de lo contrario, cambiar la fecha de referencia a la última fecha disponible.
 - Determinar las fechas exactas a ser consultadas, dada una fecha de referencia (1 día atrás, 1 mes atrás, etc), teniendo en cuenta que en dichas fechas pasadas debe haber información de precios (**recordar tips entregados antes**).
 - Extraer información desde la base de datos SQL, dada una lista de fechas y nemotécnicos de interés.
 - Extraer información desde la API del Banco Central, dada una lista de fechas de interés.
- **Todas las funciones auxiliares que cree deben estar en el archivo “auxiliares4.py”.**
- **El archivo Excel con el cuadro de rentabilidades para el 29 de septiembre de 2023, debe llamarlo “req4.xlsx”.**

Tickers vs Nemotécnicos

Su colega le comenta que una pequeña dificultad del trabajo es poder identificar los “tickers” de interés. Un “ticker” es la codificación que usa Bloomberg para los instrumentos financieros, mientras que los “Nemotécnicos” es la codificación que usa la Bolsa de Comercio para los instrumentos financieros. Para la presentación final del cuadro de rentabilidades, en el caso de la(s) accion(es) se sugiere presentar el nemotécnico del instrumento; para las filas del IPSA o Dólar, se puede usar “IPSA” y “Dólar” respectivamente (ya que no tienen un “nemotécnico”).

La Entrega

Fecha de entrega: lunes 13 de noviembre (hasta las 23.59hrs).

Entregables:

- Archivo Py: “auxiliares4.py”
- Archivo Excel: “req4.xlsx”

Todos los entregables deben estar en una sub-carpeta de la carpeta compartida llamada “requerimiento4”.