



Trabajo Práctico N° 2:

Análisis bursátil

Introducción

En este TP trabajarán en la creación y utilización de una librería de funciones para el análisis bursátil. Deberán crear funciones para cargar los datos y analizarlos, así como también generar reportes en texto y visualizaciones con gráficos. Por un lado, el objetivo es que demuestren su capacidad de leer y crear archivos, analizar datos y realizar gráficos. Por otra parte, profundizaremos en modularizar sus programas para que estos no repitan código y sean legibles.

Trabajarán con un archivo llamado **bolsa.csv**. Este contiene los precios de cierre, día por día, de 4 acciones del mercado durante el período de octubre 2021 hasta septiembre 2022.

Al finalizar este TP habrán creado un programa, con varias funciones, que emite archivos .txt, .csv y .png (imágenes de gráficos). Estos archivos representan distintas observaciones y análisis sobre acciones del mercado de capitales.

El análisis del mercado de capitales es una tarea realizada frecuentemente por inversores y economistas. Estos, por lo general, buscan estudiar las tendencias históricas del mercado, para así entender las opciones que ofrece el mercado en la actualidad y con eso maximizar la ganancia de sus inversiones.

Motivación

En este TP se les va pedir que trabajen con funciones de lectura/escritura a archivo. Algunos puntos de interés a aprender o reforzar en el transcurso de este TP son los siguientes:

- **Divide and conquer.** Las funciones son un concepto poderoso que permiten modularizar el código en bloques semi-independientes más pequeños. Pueden ahorrar mucho trabajo si son definidas adecuadamente.
- **Extensión de funciones.** Si uno necesita crear una nueva función que haría algo parecido a otra función ¿Cómo reconcilian las dos funciones? ¿Copian y pegan el código? O ven cómo modularizar el programa y descomponer la función existente en más funciones.
- **Memoria persistente.** Cómo bien ya saben, los archivos son la memoria persistente de una computadora. No se borran una vez terminado el programa. Es muy importante tener un buen manejo de escritura y lectura a archivo para poder aprovecharlo.
- **Gráficos.** La habilidad de presentar datos en una forma clara y convincente es invaluable. Su utilidad va más allá del mundo de negocios y forma una parte central en la argumentación. Recomendamos enfáticamente que utilicen este trabajo como ejercicio para aprender a hacerlo.

Datos de entrada

La cátedra les entregará un archivo .csv con el precio de cotización de las acciones de distintas empresas. Cada fila del archivo representa una fecha y cada columna representa el precio de la acción en esa fecha.

Date	SATL	MELI	MHVYF	RTX
2021-10-04	9.93	1650.90	26.59	88
2021-10-05	9.92	1603.72	26.59	89.48
2021-10-06	9.90	1577.18	26.59	89.40
2021-10-07	9.91	1580.65	26.59	90.62
2021-10-08	9.93	1571.45	26.59	90.83

Advertencia: Recuerde que el contenido de **bolsa.csv** es demostrativo! Sus programas deben poder ejecutarse con un archivo similar **que contengan acciones con otros nombres**

Advertencia: Todas las funciones deben estar definidas al comienzo del archivo .py

Análisis de datos

1. Haga una función **read_file** que tome como entrada la ruta hacia un archivo CSV, lo lea y devuelva un **diccionario** definido a continuación: las llaves deben ser los nombres de las columnas del archivo, mientras que los valores de cada llave deben ser una secuencia con los datos de cada columna.

```
diccionario = read_file("bolsa.csv")
```

Opcional: utilice arrays de Numpy como secuencia para almacenar los datos.

2. Haga una función **monthly_average** que reciba el nombre de una acción junto con el diccionario que entrega la función anterior, y calcule el precio promedio de la acción mes a mes. La función debe devolver dos secuencias. La primera debe contener las fechas del primer día de cada mes, y la segunda debe contener los precios promedios. Las secuencias deberían ser de la misma longitud.

```
fechas, promedios_mes = monthly_average("SATL", diccionario)
```

3. Genere un archivo en formato csv con el resultado generado al aplicar **monthly_average** a la acción "SATL". El nombre del archivo debe ser "monthly_average_SATL.csv". Si ya existe el archivo se debería sobrescribir.
4. Mirando el historial de precios, es fácil determinar cuándo hubiera sido el día ideal, en el pasado, para comprar una acción y generar la mayor ganancia. Para ello se define la *ganancia de una inversión* como:
$$\text{ganancia} = (\text{precio de venta} - \text{precio de compra}) / \text{precio de compra}$$

Escriba una función **max_gain** que reciba el nombre de una acción, el diccionario de datos y una fecha de venta. La función debe buscar la fecha de compra (en el pasado) que hubiera generado la mayor ganancia. La función debe devolver esta fecha y el retorno de la inversión.

Probar la función con la fecha del 6 de junio de 2022.

```
fecha_compra, ganancia = max_gain("RTX", diccionario, fecha_venta)
```

5. Generar una función **report_max_gains** que reciba el diccionario de datos y una fecha de venta, y genere un archivo de resumen en formato texto que informe la mayor ganancia de todas las acciones.

Para cada acción, el archivo debe tener una línea de texto con la oración:

*"{XXX} genera una ganancia de {ganancia*100}% habiendo comprando en {fecha_compra} y vendiendose en {fecha_venta}"*

Donde {XXX} es el nombre de la acción. Si la ganancia generada por una acción es negativa (es decir, se pierde dinero), debería añadirse la aclaración: *"La acción {XXX} solo genera pérdidas"*.

El archivo debe llamarse "resumen_mejor_compra.txt" y, si ya existe, se debería sobrescribir.

Nota: Utilizar la función **max_gain** del ítem anterior.

Opcional: Agregar una [tabla en markdown](#) con los 4 datos al final del resumen. En cuyo caso el archivo debe llamarse "resumen_mejor_compra.md"

Visualización de datos

6. Haga una función **plot_price** que reciba el nombre de una acción y genere un gráfico de líneas del precio, fecha por fecha, de dicha acción. Guardar el gráfico con el nombre "price_{XXX}.png" donde {XXX} es el nombre de la acción graficada.

```
plot_price("MELI", diccionario)
```

La función también debe poder recibir los parámetros opcionales *start* y *end* representando las fechas que se quiere graficar.

```
plot_price("MELI", diccionario, start="2022-01-01",  
          end="2022-06-01")
```

7. Haga una función **monthly_average_bar_plot** que reciba el nombre de una acción y el diccionario de datos, y genere un gráfico de barras con el precio promedio de la acción, mes a mes. El gráfico se debe guardar con el nombre "monthly_average_bar_plot_XXX.png".

```
monthly_average_bar_plot("SATL", diccionario)
```

Crear una función **plot_max_gains** que reciba el diccionario de datos y una fecha de venta, y genere un gráfico de barras donde cada barra representa la ganancia de la mejor inversión de cada acción. Debajo de cada barra debe estar el nombre de la acción. Guardar el gráfico con el nombre "max_gains.png".

```
plot_max_gains(diccionario, fecha_venta)
```

Nota: Utilizar la función **max_gain** para no repetir código.

8. **Opcional:** Escriba una función **save_report** que cree un reporte financiero en formato markdown. El informe se debe llamar "report_DD-MM-YYYY.md". donde DD-MM-YYYY es la fecha del día en que se generó el reporte.

```
save_report()
```

El reporte deberá tener las siguientes secciones:

- a. Resumen (3 líneas de texto):
 - i. Información sobre los datos y gráficos que se muestran en el informe.
 - ii. Fecha inicial y fecha final del análisis bursátil según la columna de fecha del archivo `bolsa.csv`.
 - iii. Nombre de las acciones que se estudian en el informe
- b. Máximos rendimientos:
 - i. El máximo rendimiento posible de cada acción, considerando la última fecha del archivo como la fecha de venta. Estos rendimientos deben ser representados en formato de tabla o lista junto con las fechas de compra y venta. También, se debe señalar en **negrita** la acción con mayor retorno

- ii. Un gráfico con el máximo rendimiento posible de todas las acciones. También utilizando la última fecha del archivo como fecha de venta. El gráfico debe estar incluido dentro del reporte markdown
- c. Mejor inversión:
 - i. Un gráfico de línea del precio de la acción que obtuvo el mayor retorno. El gráfico debe tener el título “XXX fue la acción de mayor retorno”

Entrega del TP

Deberán subir los siguientes archivos al campus:

- El código (.py)
- El informe (.pdf)
- Un archivo .zip con:
 - Los archivos de texto generados (.txt, .csv, .md, ...)
 - Las figuras generadas (.png)