UNIVERSIDAD TORCUATO DI TELLA

Métodos Analíticos Aplicados a los Negocios II
Segundo Cuatrimestre 2022
TP3

Recomendaciones generales

Esta guía contiene un Trabajo Práctico de MAAN II. El mismo debe ser resuelto íntegramente en `Python`. La resolución no es simple y se espera que resolverlo tome cierto tiempo por fuera de las clases.

Recuerden que pueden consultarlos sin ningún tipo de problema y que este trabajo y los siguientes son parte de la evaluación que tendrán de la materia. Se sugiere que empiecen a resolver el TP con tiempo de modo de tener tiempo para hacer consultas y no estar apurados a último momento.

El Trabajo Práctico debe resolverse de a grupos de **2 o 3 alumnos, no más y no menos**. Las consultas podrán realizarse por mail y en los momentos asignados en clase. Tener en cuenta que si la consulta es realizada a último momento puede no ser factible responderla. **No está permitida la interacción entre grupos.**

implementar

Scraping Yahoo! Finance

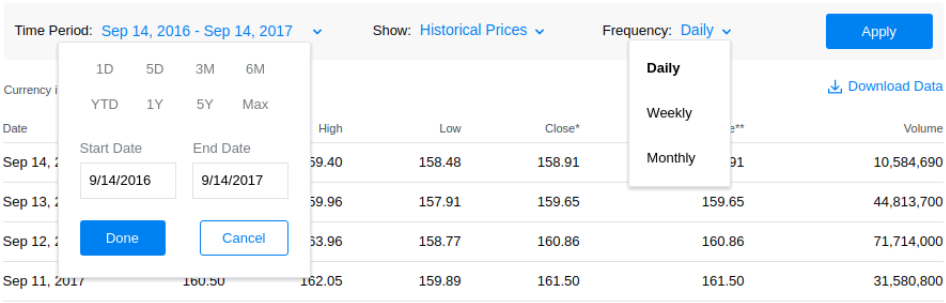
Existen distintas formas de evaluar un activo o una acción, por ejemplo en función de su retorno esperado y/o su riesgo. En general estas métricas se calculan en base a datos históricos, comúnmente en formatos de series de tiempo que consideran a los precios. Muchos sitios web y páginas de finanzas ofrecen diversas métricas sobre los distintos activos del mercado de forma resumida. Sin embargo, si uno quisiera desarrollar su propia metodología el acceso a la información detallada es, para el usuario común, engorroso y difícil de llevar adelante. En este ejercicio nos concentraremos en obtener datos históricos sobre acciones a partir de Yahoo! Finance de forma simple y fácil usando *Python*.

Para obtener los datos, Yahoo! Finance provee una página web donde se pueden consultar desde un navegador. Si bien es posible bajar la página y analizar su código HTML , Yahoo! Finance provee, escondido dentro de su sitio, una *Application Programming Interface* (API) que nos permite obtener la misma información que a través de su página web pero en un formato mucho más cómodo para trabajar desde nuestros códigos. Para poder accederlo, necesitamos saber dos cosas:

1. dónde está ubicado, y cómo acceder a la información específica de una acción;
2. una vez que localizamos la información, en qué formato viene y cómo podemos trabajar con ella.

Para el primer punto, sólo es cuestión de generar una dirección URL de forma conveniente y Yahoo! Finance nos devolverá el resultado. La dirección está compuesta por una *dirección base* sumado a algunos *parámetros* que nos permiten determinar los parámetros de la búsqueda.

Esto mismo puede hacerse desde la página web, donde podemos elegir cuál es la acción elegida, el rango de fechas que buscamos y el intervalo de tiempo para el que nos retorna los precios. Puede verse un ejemplo en la siguiente imagen.



Al acceder mediante la API podemos especificarle lo mismo. Supongamos que queremos acciones de Apple, cuyo código es AAPL en un rango determinado de fechas a intervalo de una semana. La dirección es la que sigue a continuación, que luego explicaremos que significa cada parte.

https://query2.finance.yahoo.com/v8/finance/chart/AAPL?period1=1472785200&period2=1504321200&interval=1wk&events=history

- Aquello marcado en color azul corresponde a la dirección base, o son tecnicismos relacionados el el formato de dirección URL. Son cosas que **NO** vamos a modificar.
- Las partes marcadas en verde son los nombres de los parámetros que refinan nuestra búsqueda. Éntre ellos:
  - period1: fecha de inicio de la consulta.
  - period2: fecha de fin de la consulta.
  - interval: cada cuanto queremos que nos retorne el precio.
- Las partes marcadas en violeta son los valores de los parámetros que refinan nuestra búsqueda. Éstos son:
  - código de la acción: en nuestro caso, AAPL.
  - fecha de inicio de la consulta, expresada en cantidad de segundos desde 1/1/1970.
  - fecha de fin de la consulta, expresada en cantidad de segundos desde 1/1/1970.
  - granularidad de los datos: "1wk"significa una semana. Puede ser cualquier de los siguientes: "1d","5d","1wk","1mo","3mo","6mo","1y","2y","5y","10y","ytd","max".

A modo de ejemplos, si quisiésemos realizar exactamente la misma consulta pero para Google (GOOG) en lugar de Apple usaríamos (con la diferencia respecto de la anterior marcada en rojo):

https://query2.finance.yahoo.com/v8/finance/chart/GOOG?period1=1472785200&period2=1504321200&interval=1wk&events=history

Si además quisiéramos los datos cada intervalos de un mes, agregamos la siguiente modificación:

https://query2.finance.yahoo.com/v8/finance/chart/GOOG?period1=1472785200&period2=1504321200&interval=1mo&events=history

Una vez generada la consulta, Yahoo! Finance responde con los datos usando el formato de intercambio muy utilizado en la práctica, JavaScript Object Notation (JSON), visto en clase. El mismo no es otra cosa que una combinación entre listas, strings y diccionarios que con un formato definido por el emisor del mensaje permite transmitir la información deseada.

El objetivo del trabajo es desarrollar la primera herramienta completa (sin necesidad de modificar el código para poder ejecutarla con distintas opciones) de la materia, enfocada en acceder a información de acciones a través de Yahoo! Finance y procese la información. El alcance del trabajo llegará a realizar algunas visualizaciones respecto de las acciones, pero sienta las bases para que ustedes puedan extenderla.:

El trabajo consiste en los siguientes puntos:

1. **(1.50 puntos) Lectura de parámetros de ejecución.** El programa debe leer la información relacionada a las acciones y la configuración de la ejecución de un archivo de texto ubicado en el mismo directorio de ejecución del programa, y con el nombre `input.cfg`. El archivo tendrá un parámetro por línea, en el siguiente orden: `fecha de inicio` (formato `yyyy-mm-dd`), `fecha de fin` (formato `yyyy-mm-dd`), `intervalo` (como se especifica respecto a la granularidad), `lista de acciones a buscar`, una por línea. Esta última no tiene límite y puede ser variable. Se muestra a continuación un ejemplo con 4 acciones:

   ```
   2014-08-01
   2017-08-31
   1mo
   GOOG
   AAPL
   AMZN
   MELI
   ```

   Sugerencia: en caso de tener problemas con el caracter `'\n'`, considerar la documentación de la función `strip` para `string`.

2. **(1.00 puntos) Consulta a API Yahoo! Finance y utilización JSON.** Siguiendo los lineamientos descriptos en este enunciado y las librerías vistas en clase, implementar una función que tome la fecha de inicio, la de fin, el intervalo y una acción, acceda a la API de Yahoo! Finance y obtenga el correspondiente JSON con la información requerida para ser procesada.

3. **(2.00 puntos) Cálculo de métricas para todas las acciones.** Para cada acción, extraer del correspondiente JSON la estructura que contiene las fechas, la de los precios y calcular una métrica a fin de poder visualizarla. Puede ser el precio, o el rendimiento diario, o una media móvil, o alguna otra opción que el grupo considere relevante. Respecto al rendimiento, llamemos p_i al precio de la acción del día i. Luego, si compramos la acción en un día i y la vendemos en el día j, con $j > i$, el rendimiento obtenido se calcula como

$$r = \frac{p_j - p_i}{p_i} = \frac{p_j}{p_i} - 1.$$

4. **(1.50 puntos) Visualizaciones de resultados.** Utilizar la librería `matplotlib` para visualizar la evolución de las series (precios o rendimientos) en el período elegido. Si bien la librería ofrece múltiples opciones, inicialmente la utilizaremos en su versión más básica. En este contexto, uno debe incluir el paquete correspondiente. Luego, utilizando `plot` agrega una a una las series y, finalmente, el método `show` muestra el gráfico correspondiente generado hasta el momento. Se muestra a continuación un ejemplo:.

```
import matplotlib.pyplot as plt

# y = x**2 y z = x**3 son dos series distintas.
x = list(range(10))
y = [0]*10
z = [0]*10

for i in range(10):
    y[i] = x[i]**2
    z[i] = x[i]**3

# Agregamos la serie (x,y)
# El tercer parametro es el formato ('o' es punto, "-" linea)..
plt.plot(x,y,'o-')
# Agregamos la serie (x,z)
# El tercer parametro es el formato ('x' es cruz, "-" linea)..
plt.plot(x,z,'x-')

# Agregamos la leyenda.
plt.legend(['x','y'])

# Mostramos el grafico
plt.show()
```

Si uno quisiera generar un segundo gráfico, se puede a continuación agregar nuevas series y finalmente hacer un nuevo `show`.

5. **(2.50 puntos) Cálculos adicionales.** Queremos calcular algunas métricas adicionales. Estamos interesados en evaluar, para cada acción, las siguientes métricas:

   - Período con la máxima cantidad de días consecutivos con rendimiento positivo. En caso de haber más de uno, devolver cualquiera.
   - Período con la máxima cantidad de días consecutivos con rendimiento negativo. En caso de haber más de uno, devolver cualquiera.
   - El máximo rendimiento obtenible en el período (independientemente de la cantidad de días), junto las fechas correspondientes de compra / venta, y la cantidad de días transcurridos.
   - Análogo al anterior, el mínimo rendimiento obtenible en el período (eventualmente, negativo), junto con las fechas correspondientes de compra / venta, y la cantidad de días transcurridos.

Se pide entonces agregar el cálculo de estas métricas al código. Para ello, se recomienda investigar cómo operar con el módulo `datetime`. Adicionalmente, es requisito implementar casos de test unitario adecuados para cada funcionalidad utilizando `unittest`. Este punto también contemplará en su corrección la modularización, funciones auxiliares y comentarios de todo el código así como también los casos de test unitario diseñados para este ítem en particular.

Los valores obtenidos pueden ser reportados por pantalla o en algún archivo adicional (formato a elección, JSON o texto plano son dos opciones).
Aclaración: las métricas deben calcularse asumiendo que solamente se puede invertir en long. No se posible hacer un short.

6. **(1.50 puntos) Utilización de la herramienta.** Buscar utilizando la herramienta un conjunto de acciones (al menos 3) y un período que resulte interesante desde el análisis de los precios y/o rendimientos durante el período. Comentar ventajas y desventajas respecto a la utilización, y posibles mejoras a incorporar.

7. Se pide agregar un informe, de a lo sumo 8 páginas, explicando brevemente la visualizaciones generadas, la estrategia para resolver el ejercicio 5 y la explicación del caso propuesto en el ejercicio 6.

Modalidad de entrega

Además del código con las implementaciones es posible entregar un mini informe de no más de tres carillas en donde detallen, en caso de considerarlo necesario, decisiones que tomaron para resolver uno o más ejercicios y que ayuden a entender la estrategia de resolución.

Fechas de entrega

Formato Electrónico: **lunes 7 de noviembre, hasta las 23:59 hs.**, enviando el trabajo (informe + código) a la dirección `maan2utdt@gmail.com`. El subject del email debe comenzar con el texto [TP3 MAAN II] seguido de la lista de apellidos de los integrantes del grupo. Todos los integrantes del grupo deben estar copiados en el email.

Importante: El horario es estricto. Los correos recibidos después de la hora indicada serán considerados re-entrega.