



## ENGIN 604 INTRODUCCIÓN A PYTHON PARA LAS FINANZAS — OTOÑO 2021

### TAREA 4

Entrega: 11:59pm, Sábado, Abril 17

Enviar a: [engin604assignments@gmail.com](mailto:engin604assignments@gmail.com)

Límite máximo de páginas: 5 páginas

Ver políticas de tareas en <https://docenciaweb.fen.uchile.cl>



El archivo `ETFs.xlsx` contiene el OHCL (*Open, High, Close* y *Low*) de los siguientes *Exchange Traded Funds*<sup>1</sup> con frecuencia diaria:

- SPY** (S&P500 ETF), desde 2012-11-26 hasta 2020-12-30
- EFA** (a non-US equities ETF), desde 2012-12-31 hasta 2020-12-30
- IJS** (a small-cap value ETF), desde 2012-10-31 hasta 2020-12-30
- EEM** (an emerging-markets ETF), desde 2012-04-25 hasta 2020-12-30
- AGG** (A bond ETF), desde 2012-12-31 hasta 2020-12-30

Todos los datos fueron extraído desde *Yahoo Finance* utilizando la librería `YFinance`.

- Cargue el archivo `ETFs.xlsx` a su espacio de trabajo de manera que cada hoja sea almacenada en un diccionario.
- Genere un `DataFrame` que contenga el precio al cierre de cada ETF.
  - La columna (`columns`) debe tener el nombre del ETF y en el índice (`index`) la fecha.
  - Solo trabaje con los datos desde 2012-12-31 hasta 2020-12-30.Explique al menos dos maneras de realizar (2.b).
- Transforme el `DataFrame` creado en (2) a frecuencia mensual. Lo anterior se realiza asumiendo que el último precio disponible del mes corresponde al precio mensual.
- Utilizando una función `lambda` (anónima) genere el retorno logarítmico mensual para cada ETF. No olvide eliminar la fila de `NaNs`.

$$R_{j,t} = \ln\left(\frac{P_{j,t}}{P_{j,t-1}}\right) \quad j = 1, \dots, 5 \quad t = 1, \dots, 97$$

*Hint:* NumPy tiene la función `np.log()` para calcular el logaritmo natural.

---

<sup>1</sup>Son fondos de inversión colectiva cuya política de inversión consiste en reproducir un índice.

- Genere una función que permita calcular una breve estadística descriptiva que contenga: promedio aritmético, desviación estandar, *kurtosis*, *skewness* y ratio de Sharpe (asuma tasa libre de riesgo igual a 0). Para la función tener en consideración lo siguiente:

- El *input* debe ser un Pandas con estructura Series.
- El *output* debe ser un Pandas con estructura Series donde:
  - El atributo **name** contenga el nombre del *input*.
  - El atributo **index** contenga el nombre de cada estadística descriptiva realizada.

Aplique la función al DataFrame creado en (4) mediante una función **lambda** (anónima).

- Mediante un gráfico de línea grafique el retorno logarítmico acumulado de cada ETF.
- El portafolio tangente es aquel maximiza el ratio de Sharpe. El problema de optimización expresado es su forma matricial es:

$$\max_{\mathbf{t}} \frac{\mathbf{t}'\boldsymbol{\mu} - r_f}{(\mathbf{t}'\boldsymbol{\Sigma}\mathbf{t})^{\frac{1}{2}}} = \frac{\mu_{p,t} - r_f}{\sigma_{p,t}} \quad s.t. \quad \mathbf{t}'\mathbf{1} = 1$$

Cuya solución (también matricial) es:

$$\mathbf{t} = \frac{\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - r_f \cdot \mathbf{1})}{\mathbf{1}'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - r_f \cdot \mathbf{1})}$$

Donde  $\mathbf{t}$  es el vector de pesos óptimos del portafolio tangente,  $\mathbf{1}$  un vector de unos cuya dimensión es  $j \times 1$  siendo  $j$  el total de activos y  $\boldsymbol{\Sigma}^{-1}$  el inverso de la matriz de varianza-covarianza ( $j \times j$ ).

Utilizando la solución matricial para  $\mathbf{t}$ , genere una función cuyos *inputs* sean: el vector de retornos promedio, la matriz de varianza-covarianza de los activos (ETFs) que conformarían el portafolio tangente y la tasa libre de riesgo (escalar).

El *output* de la función debe ser los pesos  $\mathbf{t}$ , el retorno esperado y desviación estandar del portafolio tangente.

- Retorno esperado del portafolio tangente es:

$$R_{p,t} = \mathbf{t}' \cdot \boldsymbol{\mu}$$

Donde  $\boldsymbol{\mu}$  es el vector de retornos promedio de dimensión  $j \times 1$ .

- La desviación estandar del portafolio de mínima varianza es:

$$\sigma_{p,t} = \sqrt{\mathbf{t}'\boldsymbol{\Sigma}\mathbf{t}}$$

Pruebe la función utilizando los retornos promedios creados en (5) y genere la matriz de varianza-covarianza utilizando el DataFrame generado en (4). Explique línea por línea su función. Asuma una tasa libre de riesgo igual a 0.005.

*Hint:* Para transformar filas o columnas de un DataFrame a NumPy se utiliza luego de la selección (**loc** o **iloc**) el atributo **.to\_numpy()**.