Optimización de portafolio
Método de mínima varianza y máximo sharpe ratio

Pablo González B., Juan Esteban Cepeda

[1] Departamento de Matemáticas y Ciencias de la Computación
Universidad Nacional de Colombia – Sede Bogotá D.C

pgonzalezb@unal.edu.co, jecepedab@unal.edu.co

1. Introducción y formulación del problema

El problema de optimización de portafolios fue por primera vez planteado en el artículo titulado *Portfolio Selection* escrito en 1952 por el reconocido economista estadounidense Harry Markowitz, dando lugar al nacimiento de las finanzas modernas. Este modelo, supuso un antes y un después en la historia de la finanzas, ya que antes de esta fecha, las personas fundamentaban sus decisiones de Inversion únicamente en el concepto de rentabilidad, ignorando un factor crucial en el proceso de Inversion: el riesgo [1].

Así, el análisis de carteras de Markowitz es un procedimiento matemático para determinar las carteras óptimas en las que invertir. El objetivo principal del análisis de portafolios es encontrar un *conjunto de portafolios eficiente*. Los portafolios eficientes deben satisfacer uno de los dos siguientes criterios: 1) tener el mayor rendimiento esperado para un determinado nivel de riesgo o, por el contrario, 2) que ofrezca el riesgo más bajo para un nivel determinado de rendimiento esperado [2].

Supongamos que existe un portafolio con N activos y se desea invertir en cada uno de ellos. Una estrategia para realizar esta tarea consiste en invertir en cada activo $\frac{C}{N}$, donde C es el capital total. No obstante, puede que esta solución no sea óptima en términos de maximizar la rentabilidad y minimizar el riesgo, debido a que no todos los activos tienen las misma rentabilidad ni volatilidad (riesgo) esperados. De lo anterior, surgen las siguientes preguntas:

1. ¿Cómo se puede determinar la asignación óptima de capital en cada activo, con la finalidad de minimizar el riesgo dado un cierto objetivo de rentabilidad?
2. ¿Cómo se puede determinar la asignación óptima de capital en cada activo, con la finalidad de maximizar la rentabilidad y minimizar el riesgo?

Este proyecto, se concentra en estas dos preguntas planteando dos problemas de optimización distintos, utilizando las referencias [2], y [3].

2. Modelo de Markowitz

Antes de definir formalmente el problema de optimización que se resolverá, a continuación se introduce la notación matemática necesaria para poder entender completamente la formulación del problema de optimización del portafolio de Markowitz, a saber:

- N : número de activos en el portafolio.
- T : periodo de tiempo durante el cual se compran los activos del portafolio.

- $p_i = s_1^{(i)}/s_0^{(i)}$: cambio de precio relativo del activo i durante el periodo T, donde $s_0^{(i)}$ es el precio inicial y $s_1^{(i)}$ es el precio al final del periodo T.
- $w \in \mathbf{R}^n$: vector de pesos, donde cada componente es el porcentaje de capital que se le asigna a cada activo.
- $R_w = p^T w$: retorno total del portafolio.
- $\bar{R}_w$: valor esperado del retorno del portafolio, donde

$$\bar{R} = \mathbb{E}(R_w) = \mathbb{E}(p^T w) = (\mathbb{E}(p))^T w = \bar{p}^T w.$$

- $w^T \Sigma w$: varianza del portafolio (riesgo), donde

$$\begin{aligned}
\delta_r^2 &= \mathbb{E}\left((r-\bar{r})^2\right) \\
&= \mathbb{E}\left(\left(p^\top w - \bar{p}^\top w\right)^2\right) \\
&= \mathbb{E}([(p-\bar{p})^T w]^2) \\
&= \mathbb{E}([(p-\bar{p})^T w]^T [(p-\bar{p})^T w]) \\
&= \mathbb{E}(w^T (p-\bar{p})(p-\bar{p})^T w) \\
&= w^T \mathbb{E}((p-\bar{p})(p-\bar{p})^T) w \\
&= w^T \Sigma w.
\end{aligned}$$

- $\sigma_w = \sqrt{w^T \Sigma w}$: desviación estandar del portafolio.
- $SR(w)$: Ratio de Sharpe del portafolio (relación entre riesgo y retorno), donde

$$SR(w) = \frac{\overline{R}_w - r_f}{\sigma_w}.$$

  donde r_f es una tasa libre de riesgo, es decir, es la rentabilidad esperada por invertir en un activo que es considerado como libre de riesgo.
- r_{min} : tasa de retorno mínima aceptada.
- $\mathbf{1}^T w = 1$: los pesos del portafolio deben sumar 1.

A continuación se plantean dos problemas de optimización que corresponden a las dos preguntas planteadas previamente, a saber: el primer problema busca minimizar el riesgo $w^T \Sigma w$ del portafolio exigiendo a la vez una tasa de retorno mínima r_{min}, esto es,

Problema 1. Portafolio de mímima varianza (MVP)

$$\begin{array}{rr}
\text{minimizar} & w^T \Sigma w \\
\text{sujeto a} & \bar{p}^T w \geq r_{\text{min}}, \\
& \mathbf{1}^T w = 1, \\
& w \geq 0
\end{array} \tag{1}$$

donde $w^T \Sigma w$ es el riesgo del portafolio y Σ es una matriz definida postiiva.

Por otro lado, el segundo problema de optimización busca maximizar el ratio de Sharpe $SR(w)$ del portafolio, es decir maximizar la relación entre el riesgo y el retorno del portafolio (lo que es equivalente a minimizar $-SR(w)$), esto es,

Problema 2. Portafolio de máximo Ratio de Sharpe (SRP)

$$\begin{array}{ll} \text{minimizar} & -SR(w) \\ \text{sujeto a} & 0 \leq w_i \leq 1, \ i = 1, 2, ..., n, \\ & \mathbf{1}^T w = 1. \end{array} \tag{2}$$

donde Σ es una matriz definida positiva.

Observe que la variable objetivo asociada a ambos problemas de optimización es el vector de pesos w, de tal manera que éste determina cual es la alocación óptima del capital C distribuido entre los N activos presentes en el mercado, con la finalidad de alcanzar los objetivos expuestos anteriormente.

Finalizamos esta sección con una pregunta natural, ¿por qué se consideran dos problemas de optimización diferentes? Es importante aclarar que con el primer problema si bien se obtiene un portafolio de mínima varianza, no se está maximizando el retorno esperado; en cambio, éste se ingresa al problema como una restricción adicional, esto es, $\bar{p}^T w \geq r_{\text{min}}$. De esta manera, la ventaja del segundo problema de optimización es que se maximiza el retorno esperado y se minimiza la varianza del portafolio simultáneamente, asegurando obtener una relación óptima entre el nivel de riesgo admitido por el inversionista por cada unidad monetaria obtenida.

3. Obtención de los datos

Para obtener los datos financieros de este proyecto, se utilizó el servicio Stock Screener de Yahoo Finance, donde se filtraron las empresas que pertenecían al sector *Tecnología* tales que tuviesen una capitalización bursátil 'grande', (más de US$10B). Para acceder a Stock Sreener se puede utilizar el siguiente enlace [4].

Ahora bien: para extraer los *tickers* de Stock Screener se utilizó una técnica de extracción de información de páginas web denominada *webscraping*. De esta manera, sólamente se incluyeron las empresas que tenían datos históricos del precio de su acción desde el año 2011-7-2 hasta 2021-7-2. Inicialmente en el aplicativo se encontraron 196 empresas, de las cuales quedaron 103 luego del proceso de filtrado. Para acceder a estos datos, el lector puede referirse al siguiente enlace [5].

4. Análisis exploratorio de los datos

Para una mejor visualización del análisis exploratorio del conjunto de datos que se utiliza en este problema de optimización, y ver con más facilidad como se llegó a dicho conjunto de datos, el lector puede referirse al siguiente enlace de un notebook en Jupyter [6].

5. Solución analítica de MVP y SRP

A continuación, se presentan las soluciones analíticas de MVP y SRP utilizando el método de multiplicadores de Lagrange. Adicionalmente, se presenta el método numérico de programación secuencial de mínimos cuadrados, el cual se emplea con frecuencia para solucionar ambos problemas.

5.1. Solución de MVP

A continuación, se presenta la solución analítica del problema MVP con base en [2], en el cual se busca minimizar la varianza del portafolio sujeta a un nivel de rendimiento mínimo. Para solucionar este problema de optimización se puede utilizar el **método de Lagrange**, por lo que se define a continuación el **lagrangiano** del problema, a saber,

$$L(w,\lambda,\nu) = w^T\Sigma w - \lambda(\overline{p}^T w - r_{\text{min}}) - \nu(\mathbf{1}^T w - 1)$$

Por lo tanto la **función dual de Lagrange** queda definida como

$$g(\lambda,\nu) = \inf_w L(w,\lambda,\nu)$$

Para obtener el ínfimo, se calcula el vector gradiente de $L(w,\lambda,\nu)$ y se iguala a cero, esto es

$$\frac{\partial L}{\partial w} = 2\Sigma w - \lambda\overline{p} - \nu\mathbf{1} = 0, \quad \text{lo que implica,}$$
$$\Sigma w = \frac{1}{2}(\lambda\overline{p} + \nu\mathbf{1}).$$

Como Σ es una matriz definida positiva (y por lo tanto existe su inversa), entonce se sigue que

$$w^* = \frac{1}{2}\Sigma^{-1}(\lambda\overline{p} + \nu\mathbf{1})$$

Ahora utilizando el hecho de que $w^{*T}\overline{p} \geq r_{\text{min}}$ (utilizaremos simplemente que $w^{*T}\overline{p} = r_{\text{min}}$) y que $w^{*T}\mathbf{1} = 1$, se obtiene el siguiente sistema de ecuaciones

$$\begin{cases} (\overline{p}^T\Sigma^{-1}\overline{p})\lambda + (\overline{p}^T\Sigma^{-1}\mathbf{1})\nu = 2\cdot r_{\text{min}}, \\ \\ (\mathbf{1}^T\Sigma^{-1}\overline{p})\lambda + (\mathbf{1}^T\Sigma^{-1}\mathbf{1})\nu = 2. \end{cases}$$

Sean $a = (\overline{p}^T\Sigma^{-1}\overline{p})$, $b = (\overline{p}^T\Sigma^{-1}\mathbf{1})$, $c = (\mathbf{1}^T\Sigma^{-1}\mathbf{1})$, entonces el sistema de ecuaciones se reduce a

$$\begin{cases} a\lambda + b\nu = 2\cdot r_{\text{min}}, \\ b\lambda + c\nu = 2. \end{cases}$$

De donde, resolviendo el sistema de ecuaciones, se puede concluir que

$$\begin{bmatrix}\lambda \\ \nu\end{bmatrix} = \frac{2}{ac-b^2}\begin{bmatrix} cr_{\text{min}} - b \\ a - br_{\text{min}}\end{bmatrix}.$$

Por lo tanto, reemplazando en la expresión que se tenía de w^*, el valor que minimiza la función del problema de optimización por el **método de Lagrange** es

$$w^* = \frac{1}{ac-b^2}\Sigma^{-1}\Big[(cr_{\text{min}} - b)\overline{p} + (a - br_{\text{min}})\mathbf{1}\Big]$$

De donde ya se conoce a, b y c.

5.2. Solución de SRP

Al igual que en la solución de MVP, a continuación se utiliza el método de Lagrange para resolver SRP. Observe que SRP se puede formular como sigue,

Problema 2. Portafolio de máximo Ratio de Sharpe (SRP)

$$\begin{array}{ll} \text{maximizar} & \bar{p}w - r_f/(w^T \Sigma w)^{\frac{1}{2}} \\ \text{sujeto a} & 0 \leq w_i \leq 1, \; i = 1, 2, ..., n, \\ & \mathbf{1}^T w = 1, \end{array} \tag{3}$$

donde Σ es una matriz definida positiva. Nóte que este problema si bien tienes restricciones afines, su objetivo no es convexo, no obstante, podemos seguir obteneniendo la solución analítica del problema por medio del método de Lagrange.

Una observación importante del problema anterior es que w^* se conoce como el portafolio tangente. Ahora bien, el lagrangiano de este problema está definido por la siguiente expresión:

$$L(w, \lambda) = (\bar{p}w - r_f)(w^T \Sigma t)^{-\frac{1}{2}} + \lambda(w\mathbf{1} - 1).$$

Ahora, utilizando la regla de la cadena, obtenemos las siguientes condiciones de primer orden:

$$\frac{\partial L(w, \lambda)}{\partial w} = \bar{p}(w^T \Sigma w)^{-\frac{1}{2}} - (w^T \bar{p} - r_f)(w^T \Sigma w)^{-\frac{3}{2}} \Sigma w + \lambda \mathbf{1} = 0.$$

$$\frac{\partial L(w, \lambda)}{\partial \lambda} = w^T \mathbf{1} - 1 = 0.$$

De estas ecuaciones se obtiene que:

$$w^* = \frac{\Sigma^{-1}(\bar{p} - r_f \cdot \mathbf{1})}{\mathbf{1}^T \Sigma^{-1}(\bar{p} - r_f \cdot \mathbf{1})}.$$

La ubicación de portafolio tangente, y el signo del ratio de Sharpe dependen de la relación entre la tasa libre de riesgo r_f y el retorno esperado de la variance mínima global del portafolio. Si $\bar{p} > r_f$ (lo cual es usual) entonces el portafolio tangente tiene un ratio de Sharpe positivo. En contraste, si $\bar{p} < r_f$ (lo cual puede ocurrir en economías en recesión), entonces el portafolio tangente puede tener una pendiente negativa, en cuyo caso los portafolios eficientes consisten en tomar posiciones en corto en el portafolio tangente e invertir en bonos del estado.

6. Solución con métodos numéricos

A continuación se presentan las soluciones de MVP y SRP utilizando métodos numéricos en el lenguaje de programación *Python*. Para resolver MVP se utilizó la librería *cvxpy*, mientras que para resolver SRP se empleó *scipy*. Es importante señalar que en ninguna de las dos librerías se requiere especificar el tipo de método numérico que se utilizará para calcular las soluciones, no obstante, ambas librerías soportan una gran cantidad de algoritmos para solucionar instancias de problemas de programación cuadrática. De esta manera, en caso en que no se indique el algoritmo a utilizar, la librería procede a identificar el método numérico más apropiado de acuerdo a las características del objetivo y las restricciones del problema. Para calcular las soluciones de ambos problemas con base en los datos datos presentados en la sección 3 *(obtención de los datos)* se utilizaron dos métodos numéricos distintos, a saber: OSQP (o por sus siglas en inglés, *Operator Splitting Quadratic Program*) para MVP y SLSQP (o por sus siglas en inglés, *Sequential Least Squares Programming*) para SRP, los cuales, a *grosso modo*, consisten en definir secuencialmente programas cuadráticos a partir de aproximaciones cuadráticas del lagrangiano del problema original, de tal manera que en un número finito de pasos permitan calcular una solución aproximada al óptimo w^* (de lo anterior, es claro que el objetivo no necesariamente debe ser cuadrático).

6.1. Solución de MVP utilizando cvxpy

Para este caso recordamos el problema de optimización a resolver:

$$\begin{array}{rr} \text{minimizar} & w^T \Sigma w \\ \text{sujeto a} & \bar{p}^T w \geq r_{\min}, \\ & \mathbf{1}^T w = 1, \\ & w \geq 0. \end{array} \tag{4}$$

El conjunto de datos que se va a utilizar se llama `returns`, entonces definimos el valor esperado de los retornos del portafolio $\bar{p}$ y la matriz de varianzas y covarianzas de nuestro conjunto de datos Σ como:

```
pBar = np.asarray(np.mean(returns.to_numpy().T, axis=1))
Sigma = np.asmatrix(np.cov(returns.to_numpy().T))
```

Luego de definir estos dos valores, definimos el problema de optimización:

```
n = len(returns.columns) # Numero de activos en el portafolio

rMin = 0.001 # Debe ser menor a np.max(pBar)
w = cp.Variable(n)
risk = cp.quad_form(w, Sigma)

constraints = [pBar.T @ w >= rMin, cp.sum(w) == 1, w >= 0]
prob = cp.Problem(cp.Minimize(risk),
                  constraints)
```

En el código anterior definimos `n` como el número de activos que tenemos en nuestro portafolio, posteriormente definimos nuestra tasa de retorno mínima aceptada `rMin`, es importante tener en cuenta que esta tasa de retorno es **diaria**, por eso es pequeña. Finalmente definimos la varianza `risk = cp.quad_form(w, Sigma)` de nuestro portafolio y definimos el problema a resolver, ahora lo resolvemos con la siguiente línea de código:

```
prob.solve(solver = cp.OSQP)
```

Al momento de escribir este documento, dicha linea de código retorna el valor `0.00010334198391038593`, junto a `prob.status = 'optimal'` indicando que ha encontrado el óptimo del problema de optimización.

Posteriormente, habiendo hallado el vector de pesos óptimo `w`, supongamos que se cuenta con un capital disponible de **$100,000** y se desea invertir en el portafolio con los pesos óptimos que se obtuvieron, entonces en el siguiente fragmento de código se imprime cada activo del portafolio y la respectiva Inversion que le corresponde teniendo en cuenta el vector de pesos `w` que se obtuvo. Es importante tener en cuenta que hay activos que les correspondió un peso cercano a 0, por lo cual en el fragmento de código se filtran los activos que les corresponde una Inversion mayor a **$1**:

```
i = 0
for column in returns.columns:
    if np.round(w.value[i]*100000) > 1:
        print("Inversion para {}: ${}".format(column,
            np.round(w.value[i]*100000)))
    i += 1
```

el fragmento de código anterior muestra el siguiente resultado:

```
Inversion para TOELF: $15901.0
Inversion para SNPS: $5040.0
Inversion para CNSWF: $28392.0
Inversion para CAPMF: $1221.0
Inversion para AMADF: $3870.0
Inversion para GRMN: $37.0
Inversion para LOGI: $3866.0
Inversion para NTDTY: $1203.0
Inversion para BR: $12136.0
Inversion para TYL: $16227.0
Inversion para MPWR: $1335.0
Inversion para NICE: $1469.0
Inversion para BYDIF: $3253.0
Inversion para JKHY: $905.0
Inversion para MNBEY: $5145.0
```

y así concluimos este problema de optimización.

6.2. Solución de SRP utilizando scipy

Ahora para este caso recordamos nuevamente el problema de optimización a resolver:

$$\begin{array}{rr} \text{minimizar} & -SR(w) \\ \text{sujeto a} & 0 \leq w_i \leq 1,\ i = 1, 2, ..., n, \\ & \mathbf{1}^T w = 1. \end{array} \tag{5}$$

Para este caso, la libreria que se va a utilizar es `scipy`, pues este problema no es convexo ya que la función objetivo no es convexa. Como en el caso anterior, el conjunto de datos que se va a utilizar se llama `returns`, entonces definimos el valor esperado de los retornos del portafolio $\bar{p}$ y la matriz de varianzas y covarianzas de nuestro conjunto de datos Σ como:

```
pBar = returns.mean()
Sigma = returns.cov()
```

Ahora, se definen las funciones del sharpe ratio y la restricción de igualdad, respectivamente así:

```
def sr(w):
    ret = np.sum( pBar * w )
    risk = np.sqrt( np.dot( w.T, np.dot(Sigma, w) ) )
    return -( ret / risk )

def checkSum(w):
    return np.sum(w)-1
```

Y posteriormente se define el vector de pesos inicial `w0` desde el cual el programa va a empezar a iterar para encontrar el valor óptimo junto con las cotas del problema y la restricción de igualdad así:

```
w0 = [1/n]*n
bounds = tuple([(0, 1) for i in range(n)])
constraints = ({'type':'eq', 'fun':checkSum})
```

Finalmente resolvemos nuestro problema utilizando el método `SLSQP` de la librería `scipy` de la siguiente manera:

```
w_opt = minimize(sr, w0, method='SLSQP',
                    bounds=bounds, constraints=constraints)
```

Luego de obtener el mensaje `'Optimization terminated successfully'`, volvemos a suponer que contamos con un capital de **$100,000** para invertir en el portafolio, para finalmente imprimir el capital que le corresponde a cada activo del portafolio utilizando el vector de pesos óptimo `w_opt` que retornó el problema, filtrando solamente aquellos en los que se invierte un capital mayor que **$1**:

```
i = 0
for column in returns.columns:
    if np.round(w_opt.x[i]*100000) > 1:
        print("Inversion para {}: ${}".format(column,
                np.round(w_opt.x[i]*100000)))
    i += 1
```

y obtenemos este resultado:

```
Inversion para TOELF: $15859.0
Inversion para SNPS: $5106.0
Inversion para CNSWF: $27710.0
Inversion para CAPMF: $1493.0
Inversion para AMADF: $4069.0
Inversion para GRMN: $323.0
Inversion para LOGI: $3745.0
Inversion para NTDTY: $1324.0
Inversion para BR: $11987.0
Inversion para TYL: $15610.0
Inversion para MPWR: $726.0
Inversion para NICE: $1663.0
Inversion para BYDIF: $3194.0
Inversion para JKHY: $1948.0
Inversion para MNBEY: $5242.0
```

y así finalizamos con la solución de nuestro segundo problema de optimización.

7. Comparación entre soluciones analíticas

En esta sección se realiza una comparación entre las soluciones analíticas y las soluciones computacionales de cada uno de los dos problemas, es decir del problema de mínima varianza (MVP) y máximo Ratio de Sharpe (SRP).
En primera instancia vamos a calcular el vector de pesos óptimo $w*$ que nos dió la solución analítica por el método de Lagrange, lo hacemos de la siguiente manera:

```
def ourSolver_mv(rMin, Sigma, pBar):
    N = len(Sigma)
    o = np.ones(N)
    SigmaInv = np.asarray(np.linalg.inv(Sigma))
    a = np.dot(pBar.T, np.dot( SigmaInv, pBar ))
    b = np.dot(pBar.T, np.dot( SigmaInv, o ))
    c = np.dot(o.T, np.dot( SigmaInv, o ))
    return ((1/(a*c - b**2)) * np.dot(SigmaInv,
                        ((c*rMin - b)*pBar + (a-b*rMin)*o)))
```

Esta función calcula directamente el vector de pesos óptimo que obtuvimos en la solución analítica del problema de mínima varianza, utilizando la inversa de la matriz de covarianzas, la tasa de retorno mínima y el vector de los retornos esperados. El resultado que obtuvimos es el siguiente:

```
Inversion para FIS: $672.0
Inversion para INFY: $1054.0
Inversion para TOELY: $1634.0
Inversion para DASTY: $2385.0
Inversion para TOELF: $8654.0
```

```
Inversion para ADI: $8145.0
Inversion para DASTF: $8395.0
Inversion para NXPI: $471.0
Inversion para HNHPF: $1218.0
Inversion para IFNNF: $2607.0
Inversion para MRAAY: $855.0
Inversion para KLAC: $865.0
Inversion para TEL: $2445.0
Inversion para SNPS: $7842.0
Inversion para ERIXF: $3386.0
Inversion para CDNS: $1093.0
Inversion para WIT: $241.0
Inversion para FJTSY: $4735.0
Inversion para MSI: $5914.0
Inversion para XLNX: $1340.0
Inversion para CNSWF: $9595.0
Inversion para AMADY: $890.0
Inversion para CAPMF: $4005.0
Inversion para AMADF: $4499.0
Inversion para NOKBF: $574.0
Inversion para GRMN: $4686.0
Inversion para MXIM: $4364.0
Inversion para PCRFF: $3016.0
Inversion para VRSN: $1069.0
Inversion para ZTCOF: $139.0
Inversion para CAJFF: $10301.0
Inversion para GIB: $3217.0
Inversion para IT: $1507.0
Inversion para FLT: $1196.0
Inversion para APH: $6015.0
Inversion para LOGI: $4148.0
Inversion para NTDTY: $1800.0
Inversion para STX: $3695.0
Inversion para UMC: $3587.0
Inversion para BR: $9395.0
Inversion para TYL: $5592.0
Inversion para RNECF: $1363.0
Inversion para MPWR: $2181.0
Inversion para CHKP: $4275.0
Inversion para OMRNY: $730.0
Inversion para ENTG: $880.0
Inversion para NLOK: $2647.0
Inversion para NICE: $5603.0
Inversion para TDY: $2188.0
Inversion para NUAN: $5.0
Inversion para LDOS: $323.0
Inversion para OTEX: $1169.0
Inversion para BYDIF: $1327.0
Inversion para LNVGF: $2539.0
```

```
Inversion para JKHY: $7530.0
Inversion para MNBEY: $3407.0
Inversion para SGPYY: $1361.0
Inversion para DOX: $10030.0
Inversion para DLB: $1937.0
```

Podemos ver con claridad que el resultado que obtuvimos en este caso agrupa muchas mas empresas dentro del portafolio. Ahora para calcular directamente el vector de pesos óptimo que obtuvimos con la solución analítica del problema de máximo sharpe ratio, utilizamos la siguiente función:

```
def ourSolver_sr(Sigma, pBar, risk_free_rate = 0.00011):
    N = len(Sigma)
    o = np.ones(N)
    SigmaInv = np.asarray(np.linalg.inv(Sigma))
    return ((np.dot( SigmaInv, (pBar - np.dot(risk_free_rate,
                        o)))) / (np.dot(o.T, np.dot(SigmaInv,
                        (pBar - np.dot(risk_free_rate, o))))))
```

En este caso no utilizamos la tasa de retorno mínima, pues inicialmente no la incluimos en este problema de optimización. Incluimos la tasa libre de riesgo que, en caso de no ser especificada, es igual a 0.011 %, es una cantidad bastante pequeña pero recordemos que esto es diario, luego anual sería igual a 3 %. En este caso obtenemos el siguiente resultado:

```
Inversion para LRCX: $10209.0
Inversion para MU: $6261.0
Inversion para INFY: $7542.0
Inversion para FISV: $4928.0
Inversion para TOELY: $8802.0
Inversion para DASTY: $11454.0
Inversion para TOELF: $17790.0
Inversion para ADI: $8988.0
Inversion para DASTF: $15864.0
Inversion para NXPI: $9203.0
Inversion para IFNNF: $2092.0
Inversion para MRAAY: $4512.0
Inversion para KLAC: $5872.0
Inversion para TEL: $2452.0
Inversion para SNPS: $12419.0
Inversion para FTNT: $6392.0
Inversion para CDNS: $13957.0
Inversion para FJTSY: $10368.0
Inversion para MSI: $18642.0
Inversion para CNSWF: $31856.0
Inversion para AMADY: $1441.0
Inversion para SWKS: $3260.0
```

```
Inversion para CAPMF: $3150.0
Inversion para AMADF: $13830.0
Inversion para NOKBF: $1794.0
Inversion para ZBRA: $6928.0
Inversion para GRMN: $11855.0
Inversion para PCRFF: $397.0
Inversion para VRSN: $2734.0
Inversion para IT: $5509.0
Inversion para FLT: $7019.0
Inversion para APH: $24090.0
Inversion para LOGI: $17211.0
Inversion para NTDTY: $1178.0
Inversion para STX: $19147.0
Inversion para UMC: $11592.0
Inversion para BR: $29213.0
Inversion para TYL: $22960.0
Inversion para MPWR: $25008.0
Inversion para ATEYY: $1313.0
Inversion para PTC: $1729.0
Inversion para OMRNY: $7261.0
Inversion para ENTG: $6699.0
Inversion para NLOK: $2130.0
Inversion para NICE: $12513.0
Inversion para TDY: $18947.0
Inversion para CGNX: $1320.0
Inversion para FICO: $4115.0
Inversion para BYDIF: $5018.0
Inversion para LNVGF: $6944.0
Inversion para JKHY: $6333.0
Inversion para MNBEY: $9134.0
Inversion para DXC: $110.0
Inversion para DOX: $6483.0
```

Seguimos viendo que al calcular directamente el vector de pesos óptimo obtenido en la solución analítica obtenemos un portafolio con muchas mas empresas. Pero hay algo muy importante a notar acá, la comparación no debe ser directamente entre los vectores de pesos, pues para realmente hacer una comparación objetiva debemos comparar el riesgo que obtenemos con cada uno de estos vectores de pesos calculados.

Ahora, vamos a calcular el riesgo del portafolio con cada uno de estos vectores de pesos óptimos, primero vamos a comparar el que obtuvimos utilizando la libreria `cvxpy` junto con el que calculamos directamente con la función `ourSolver_mv`, y finalmente vamos a comparar el que obtuvimos utilizando la libreria `scipy` junto con el que calculamos directamente con la función `ourSolver_sr`, es decir básicamente vamos a comparar entre las soluciones analíticas y computacionales de cada problema de optimización.

El riesgo obtenido con la solución computacional y analitica del problema de mínima varianza son `6.828845095050163e-05` y `4.255370263278294e-05` respectivamente.

Y el riesgo obtenido con la solución computacional y analitica del problema de maximo Ratio de Sharpe son `4.02886423764766e-05` y `3.671333599535184e-05`. respectivamente.

En cada caso vemos que, si bien los vectores de pesos son distintos entre las soluciones analíticas y las soluciones computacionales, el riesgo obtenido es similar, por lo tanto no es objetivo comparar entre los vectores de pesos ya que el resultado final que necesitamos es minimizar el riesgo del portafolio y maximizar la rentabilidad, además distintos portafolios de inversión pueden tener tasas de riesgo iguales o muy parecidas.

8. Conclusiones

Es importante a observa que el resultado del segundo problema de optimización es muy parecido al resultado que se obtuvo en el primer problema, esto se debe a que si bien en el primer problema se calculó la mínima varianza del portafolio, la tasa de retorno mínima que se pidió era muy cercana a la máxima que podiamos pedir, que en este caso no podía ser mayor al valor máximo del vector `pBar`, por lo cual si en el primer problema de optimización se elige una tasa de retorno menor, los pesos del vector `w` van a alejarse del resultado que obtuvimos en este segundo problema y ya no serán tan parecidos.

En este caso no hay resultado mejor ni peor, pues no siempre se busca la mejor relación entre riesgo y retorno (es decir el máximo sharpe ratio) sino simplemente el portafolio con el menor riesgo posible que no necesariamente garantiza un retorno ideal, esto es algo que entra a depender del perfil de riesgo del inversor.

9. Bibliografía

1. **[Ref 1.]** Brealey, Myers y Allen (2015). Principios de Finanzas Corporativas. Estados Unidos: McGrawHill
2. **[Ref 2.]** Clark, J y Kim, D. (2013). Modern Portfolio Theory. Estados Unidos: John Wiley and Sons.
3. **[Ref 3.]** Chapter 1. Portfolio Theory with Matrix Algebra (2013). Recuperado de: https://faculty.washington.edu/ezivot/econ424/portfolioTheoryMatrix.pdf
4. **[Ref 4.]** Yahoo Finance. (sf). Stock Screener. Recuperado de: StockScreenerYF
5. **[Ref 5.]** Pablo González (2021). Datos proyecto optimización. Enlace: Datos proyecto.
6. **[Ref 6.]** Pablo González (2021). Análisis exploratorio de los datos del proyecto de optimización. Enlace: Jupyter Notebook.