

Proyecto Fin de Grado  
Grado en Ingeniería de las Tecnologías  
Industriales

Implementación en lenguaje Python de la Teoría de  
la cartera media-varianza de Harry Markowitz

Autor: Jesús Parejo Palop

Tutor: Emilio Freire Macías

Tutor: Manuel Ordóñez Sánchez

Sevilla, 2022





Proyecto Fin de Grado  
Grado en Ingeniería de las Tecnologías Industriales

# **Implementación en lenguaje Python de la Teoría de la cartera media-varianza de Harry Markowitz**

Autor:

Jesús Parejo Palop

Tutores:

Emilio Freire Macías

Profesor titular

Manuel Ordóñez Sánchez

Profesor titular

Dpto. de Matemática Aplicada II  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2022



Proyecto Fin de Carrera: Implementación en lenguaje Python de la Teoría de la cartera media-varianza de Harry Markowitz

Autor: Jesús Parejo Palop

Tutor: Emilio Freire Macías

Tutor: Manuel Ordóñez Sánchez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2022

El Secretario del Tribunal



*A mi familia*

*A mis amigos*

*A mis hermanos de comunidad*

*A mis compañeros*

*A mis maestros*





# Agradecimientos

---

Con este Proyecto se puede dar por concluido mi etapa en el Grado de Ingeniería de las Tecnologías Industriales por la Universidad de Sevilla. Al finalizar esta fase de mi vida no sería justo no acordarme de todas las personas que han contribuido a que esto sea posible.

En primer lugar, a mi familia, por seguir queriéndome aunque mis esfuerzos no fueran a parar a ella.

A mis amigos que conocí aquí, porque sin ellos este tiempo hubiera sido totalmente imposible de llevar

A mis amigos de siempre, por proporcionarme apoyo incondicional y grandes momentos.

A todo el personal docente que ha facilitado la consecución de los objetivos en sus correspondientes materias.

A Ana por estar ahí y ser quien me ha motivado para poder terminar este ciclo.

Por último, agradecer a mis compañeros de Altran por hacer este último periodo de grado y mi primer contacto con el mundo laboral agradable, divertido y fácil.

*Jesús Parejo Palop*

*Grado en Ingeniería de las Tecnologías Industriales*

*Sevilla, 2022*



# Resumen

---

En este proyecto se hace una introducción a los conceptos básicos de las inversiones, principalmente, en el mercado de valores. Para ello, en primer lugar, se encuentra la definición de los términos y principios que se emplean a lo largo del documento como son los retornos esperados de una inversión y el riesgo. Además, se define lo que representa un activo dentro de las carteras de valores.

En segundo lugar, se realiza un análisis sobre un modelo matemático de Harry Markowitz que se emplea para la optimización de estas carteras. Este modelo le permitió ganar un premio Nobel en 1990.

A lo largo de los capítulos 3 y 4 se establece tanto el objetivo del modelo de la Teoría de la cartera media-varianza como las condiciones que deben cumplir las carteras. También se incluye un procedimiento para generar carteras optimizadas haciendo uso del modelo de optimización. A partir de este modelo, surgen nuevos conceptos que se explican posteriormente y que son conceptos que se siguen empleando en los entornos dedicados al mundo de las inversiones en el mercado bursátil.

A continuación, se hace una breve introducción al lenguaje de programación Python. Un lenguaje que se emplea más adelante para conseguir una implementación del modelo de Markowitz. Adicionalmente, se incluye una descripción de esta implementación.

En último lugar, se llevan a cabo distintas simulaciones, con el fin de poder analizar los resultados obtenidos y sacar conclusiones que ayudan a comprender mejor cómo conseguir la cartera óptima para cada inversor. Estas conclusiones quedan recogidas al final del proyecto.



# Abstract

---

This project provides an introduction to the basic concepts of investments, mainly in the stock market. For this, firstly, there is the definition of the terms and principles that are used throughout the document, such as the expected returns of an investment and the risk. In addition, what an asset represents within the securities portfolios is defined.

Secondly, an analysis is carried out on a mathematical model of Harry Markowitz that is used for the optimization of these portfolios. This model served him to receive a Nobel Prize in 1990. Throughout a chapter, both the objective of the model and the conditions that portfolios must meet are established. A procedure to generate optimized portfolios using the optimization model is also included. From this model, new concepts arise that are explained later and that are concepts that are still used in environments dedicated to the world of investments in the stock market.

Here is a brief introduction to the Python programming language. A language that is used later to achieve an implementation of the Markowitz model. Additionally, a description of this implementation is included.

Finally, different simulations are carried out, in order to be able to analyze the results obtained and draw conclusions that help to better understand how to achieve the optimal portfolio for each investor. These conclusions are collected at the end of the project.



# Índice

---

<b>Agradecimientos</b>	<b>ixx</b>
<b>Resumen</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Índice</b>	<b>xixv</b>
<b>Índice de Tablas</b>	<b>xvii</b>
<b>Índice de Figuras</b>	<b>xix</b>
<b>1 Objetivo</b>	<b>1</b>
<b>2 Ámbito de aplicación de la teoría de la cartera media-varianza</b>	<b>3</b>
<b>3 Introducción a la teoría de la media-varianza</b>	<b>5</b>
3.1 <i>Tasas de retorno de un activo</i>	5
3.2 <i>Introducción a las carteras</i>	6
<b>4 Bases de la teoría de la cartera media-varianza</b>	<b>9</b>
4.1 <i>La función objetivo del modelo de Markowitz</i>	9
4.2 <i>Restricciones del modelo de Markowitz</i>	10
4.3 <i>Condiciones de Karush-Kuhn-Tucker</i>	11
4.4 <i>Procedimiento de resolución del modelo de Markowitz</i>	13
4.4.1 <i>Revisar la admisibilidad del modelo</i>	13
4.4.2 <i>Revisar la solución de mínima varianza</i>	13
4.4.3 <i>Calcular una segunda cartera que sea admisible</i>	13
<b>5 Conclusiones de la teoría de la cartera media-varianza</b>	<b>15</b>
5.1 <i>La función objetivo del modelo de Markowitz</i>	15
5.2 <i>Restricciones del modelo de Markowitz</i>	17
5.3 <i>Condiciones de Karush-Kuhn-Tucker</i>	98
5.4 <i>Procedimiento de resolución del modelo de Markowitz</i>	20
<b>6 Introducción al lenguaje de programación Python</b>	<b>25</b>
6.1 <i>Variables en Python</i>	27
6.1.1 <i>Variables de texto</i>	27
6.1.2 <i>Variables numéricas</i>	28
6.1.3 <i>Variables de tipo secuencia</i>	28
6.1.4 <i>Variables de tipo mapeo</i>	29
6.2 <i>Funciones en Python</i>	29

6.3	<i>Clases en Python</i>	30
6.4	<i>Librerías en Python</i>	30
<b>7</b>	<b>Implementación en Python de la Teoría de la cartera media-varianza</b>	<b>33</b>
7.1	<i>Introducción</i>	33
7.2	<i>Implementación en Python</i>	34
7.2.1	<i>Librerías y constantes</i>	34
7.2.2	<i>Clase Asset</i>	35
7.2.3	<i>Clase Portfolio</i>	36
7.2.4	<i>Función auxiliar yfinance</i>	38
7.2.5	<i>Representación gráfica</i>	39
7.2.6	<i>Obtención de los pesos de cada cartera optimizada</i>	40
<b>8</b>	<b>Aplicación del modelo de Markowitz en lenguaje de programación Python</b>	<b>43</b>
8.1	<i>Modificaciones del código original</i>	43
8.2	<i>Simulaciones</i>	43
8.2.1	<i>Simulación de ejemplo</i>	44
8.2.2	<i>Simulación en el sector de las empresas energéticas y petrolíferas en España</i>	46
8.2.3	<i>Simulación en el sector de las industrias en España</i>	48
8.2.4	<i>Simulación en el sector de las tecnológicas en España</i>	50
8.2.5	<i>Simulación en el sector de los bienes de consumo en España</i>	52
8.2.6	<i>Simulación en el sector de los servicios financieros en España</i>	54
8.2.7	<i>Simulación en el sector servicios en España</i>	56
<b>9</b>	<b>Conclusiones</b>	<b>59</b>
	<b>Referencias</b>	<b>61</b>



# ÍNDICE DE TABLAS

---

Tabla 8–1: Relación de pesos de las carteras optimizadas con activos de ejemplo	46
Tabla 8–2: Relación de pesos de las carteras optimizadas con activos del sector de las energéticas y petrolíferas	48
Tabla 8–3: Relación de pesos de las carteras optimizadas con activos del sector de las industrias	50
Tabla 8–4: Relación de pesos de las carteras optimizadas con activos del sector de las tecnológicas	52
Tabla 8–5: Relación de pesos de las carteras optimizadas con activos del sector de los bienes de consumo	54
Tabla 8–6: Relación de pesos de las carteras optimizadas con activos del sector de los servicios financieros	56
Tabla 8–7: Relación de pesos de las carteras optimizadas con activos del sector servicios	58



# ÍNDICE DE FIGURAS

---

Ilustración 1	16
Ilustración 2	17
Ilustración 3	21
Ilustración 4	22
Ilustración 5	25
Ilustración 6	27
Ilustración 7	28
Ilustración 8	28
Ilustración 9	29
Ilustración 10	30
Ilustración 11	30
Ilustración 12	31
Ilustración 13	40
Ilustración 14	41
Ilustración 15	45
Ilustración 16	47
Ilustración 17	49
Ilustración 18	51
Ilustración 19	53
Ilustración 20	55
Ilustración 21	57



# 1 OBJETIVO

---

Las inversiones en el mercado de valores son una realidad que se da en numerosos ámbitos, desde el profesional hasta el particular mediante la utilización de brókers. En este, se trata de obtener siempre el mayor beneficio en la operación de compra y posterior venta de un bien. Actualmente, los brókers utilizan un modelo matemático para conseguir las mejores operaciones. Este modelo fue desarrollado por Harry Markowitz en 1952 publicado en un artículo del Journal of Finance titulado Portfolio Selection.

Por otra parte, el lenguaje de programación Python se califica como uno de los más populares. Su éxito se basa en dos pilares: es un software con licencia de código abierto y es muy versátil, porque permite las programaciones: orientada a objetos, imperativa, funcional y secuencial. Python se puede usar mediante la interacción directa con el intérprete de Python en 'Command Prompt' de WINDOWS o mediante programas como IDLEs (entorno de desarrollo integrado para Python).

El objeto de este proyecto es implementar, usando el lenguaje Python, el modelo de Markowitz por el cual se obtiene las cantidades optimizadas a invertir en cada uno de los activos seleccionados que se consideren de interés.

En esta memoria se tratará de explicar la teoría de Harry Markowitz y se recogerán los resultados de aplicar ésta a una cartera de valores. También se presentará el software por el que se consigue una implementación del modelo.



## 2 ÁMBITO DE APLICACIÓN DE LA TEORÍA DE LA CARTERA MEDIA-VARIANZA

---

La teoría de Markowitz [1] se aplica fundamentalmente sobre el mercado de valores pero realmente se puede aplicar a cualquier tipo de inversiones. Esta teoría puede usarse siempre que se pueda estimar una rentabilidad y las medidas de dispersión correspondientes.

La rentabilidad esperada de un activo se puede obtener de distintas formas. La manera más utilizada es calculando la esperanza a partir del histórico de los valores de un activo. Esta es una de las maneras más sencillas. También se podría usar otros modelos de predicción como medias móviles o ajuste exponencial.

Una de las medidas de dispersión que hace falta conocer para aplicar el modelo de Markowitz es la varianza del valor de los activos. Toda variable (el valor de nuestro activo, en este contexto) presenta la posibilidad de desviarse de una referencia. En la estadística este valor de referencia es la esperanza de la variable aleatoria. Si esta variable tiene probabilidades elevadas de desviarse seriamente, su varianza será mayor que si, por el contrario, tiene escasas posibilidades de alejarse de esta referencia.

Otra medida de dispersión necesaria para poder utilizar la teoría de Markowitz es la covarianza. La covarianza no es más que una medida de la relación que tienen dos variables aleatorias distintas respecto a sus referencias. La covarianza puede tomar valores positivos, negativos y nulo. Cuando la covarianza de dos variables toma valor nulo significa que no guardan relación alguna entre ellas, si una aumenta, la otra puede hacer cualquier cosa. Si esta medida es mayor que cero implica que las variables están relacionadas positivamente y si una aumenta, la otra también lo hará. Por último, si la covarianza resulta negativa, indica que las variables guardan una relación negativa y cuando una aumenta la otra disminuye.

Estas medidas de dispersión componen lo que se conoce como riesgo. El riesgo se entiende como la suma de variables aleatorias tipo amplitud, normalmente siguiendo una distribución de Poisson con parámetros distintos. Estas desviaciones pueden ser tanto a positivas como negativas. A mayores amplitudes, mayor riesgo. Un ejemplo de una mala repercusión en el valor de los activos de una empresa sería si se destapara un escándalo que envuelve a directivos de esta. Entonces el valor de los activos de dicha empresa probablemente se viera menguado.





# 3 INTRODUCCIÓN A LA TEORÍA DE LA CARTERA MEDIA-VARIANZA

---

*A good portfolio is more than a long list of good stocks and bonds. It is a balanced whole, providing the investor with protections and opportunities with respect to a wide range of contingencies*

*- Harry Markowitz -*

**E**n este capítulo se tratará de hacer comprender los principales términos que se utilizan en la teoría descrita por Harry Markowitz [1]. En esta teoría se busca conseguir un rendimiento financiero a partir de inversiones en el mercado. Ahora se procederá a explicar los conceptos sobre los que versa esta teoría.

## 3.1 Tasas de retorno de un activo

Un activo es un bien y/o un derecho que se puede comprar y vender y que tiene un valor monetario que no es fijo. Éstos se pueden adquirir por una cantidad  $x_0$ . Este valor puede ser desde céntimos hasta miles de euros y cualquier otra cantidad. Este mismo activo, se compra en una fecha determinada y en el momento de la venta su valor es  $x_1$ , y su valor puede ser distinto o el mismo que en el instante de compra  $x_0$ .

Si se divide el valor de la venta entre el de compra, se obtiene el retorno de la inversión. Este valor puede ser mayor que 1 en caso de que se consiga una venta por un valor mayor al de la compra. Cuando el cociente resulta mayor que la unidad entonces se sabe que la operación ha reportado beneficios. Si el resultado de esta operación es menor que 1 significa que el valor de venta ha sido menor que el valor de la compra del activo en el que se ha invertido. Se obtiene este tipo de resultados cuando la operación nos ha causado pérdidas. También podría ser que el cociente resultara igual a 1. En este caso, la operación no ha provocado ningún cambio al montante de nuestra inversión.

$$R = \frac{x_1}{x_0}$$

Donde:

- R: retorno del activo
- $x_i$ : valor del activo en el instante  $i$

Este retorno sobre el activo también se le conoce como la rentabilidad del activo o de la inversión. Este número es el gran objetivo de todos los inversores ya que todos ellos buscan conseguir el mayor beneficio posible y si es posible invirtiendo la menor cantidad de dinero. La rentabilidad se calcula siempre en base a las series temporales de históricos de aumentos o descensos del valor del activo. El periodo a tener en cuenta para calcular la rentabilidad de un activo no es fijo, se puede fijar según se estime oportuno.

Otro índice a estudiar para saber lo rentable que es una inversión será la tasa de retorno relativa de una transacción. Esta cifra nos indica el porcentaje de la inversión inicial que se gana o pierde. Si la relación de retorno es positiva, indica que la diferencia entre el valor de venta y el valor de compra de un activo es positiva también. En cambio, si es negativa es porque un activo se ha vendido a un precio inferior al que se compró. Es otro indicador distinto de lo rentable que ha sido una inversión, pero también es una señal que se entiende más fácilmente que el retorno del activo.

$$r = \frac{x_1 - x_0}{x_0}$$

Donde:

- $r$ : tasa de retorno relativa del activo
- $x_i$ : valor del activo en el instante  $i$

### 3.2 Introducción a las carteras

Las carteras de valores o portfolios son un conjunto de activos. Un activo ya hemos visto que puede ser bienes o derechos con valor monetario que puede ser objeto de compra y/o venta. Pues bien, cuando un individuo posee un único activo, la cartera de su propiedad incluye a este único activo a la que puede incluir otros nuevos.

Todas las propiedades de sus activos las recoge una cartera. Se supone que se construye una cartera con  $n$  activos. Esta cartera se consigue con un presupuesto de  $x_0$  que se reparte entre todos los activos que la compongan. Se considera que cada componente  $i$  de la cartera tiene un montante que verifica esta igualdad:  $x_{0i} = \omega_i \cdot x_0$ . Cada activo dentro de la cartera tendrá peso  $\omega_i$ . Estos pesos deben cumplir la condición de que la suma de todos ellos sea igual a 1:  $\sum_{i=1}^n \omega_i = 1$ . De esta forma, se verifica que:

$$\text{Valor cartera} = \text{Suma valores de los activos} = \sum_{i=1}^n w_i \cdot x_{0i} = x_0 \cdot \sum_{i=1}^n w_i = x_0$$

Donde:

- $w_i$ : peso del activo  $i$
- $x_{0i}$ : valor de compra del activo  $i$

Continuando con las propiedades de los activos que se recogen conjuntamente en una cartera, se aclara cómo la cartera refleja el retorno de todo el grupo de inversiones. Si  $R_i$  denota el retorno de un activo  $i$ , entonces se cumple que el valor de una cartera en un instante será la suma ponderada por los tres elementos: el retorno, el valor de compra y el peso de cada activo. Es decir, se cumplen las siguientes expresiones:

$$x_1 = \sum_{i=1}^n R_i \cdot w_i \cdot x_{0i} = x_0 \cdot \sum_{i=1}^n R_i \cdot w_i$$

Donde:

- $R_i$ : retorno del activo  $i$
- $w_i$ : peso del activo  $i$
- $x_{0i}$ : valor de compra del activo  $i$
- $x_1$ : valor de la cartera en el instante 1
- $x_0$ : valor de compra de la cartera

A partir de la ecuación anterior se puede deducir que el retorno de toda la cartera se obtiene con la siguiente fórmula:

$$R_{cartera} = R = \sum_{i=1}^n R_i \cdot w_i$$

Una vez que se ha definido el retorno de una cartera, queda por concretar cómo se refleja en una cartera las tasas de retorno de cada uno de sus activos. Suponiendo que cada activo  $i$  tiene una tasa  $r_i$  que, se puede demostrar, donde cumple la igualdad  $r_i = R_i - 1$ . En las carteras se cumple esta identidad. En base a esta premisa, se puede determinar la siguiente ecuación:

$$r_{cartera} = r = R - 1 = \left( \sum_{i=1}^n R_i \cdot w_i \right) - \left( \sum_{i=1}^n w_i \right) = \sum_{i=1}^n (R_i - 1) \cdot w_i = \sum_{i=1}^n r_i \cdot w_i$$

Donde:

- $R_i$ : retorno del activo  $i$
- $w_i$ : peso del activo  $i$
- $R$ : retorno de la cartera
- $r_i$ : tasa de retorno del activo  $i$

Es importante terminar con las tasas de retorno de una cartera porque es un dato fundamental para poder aplicar el modelo matemático de optimización en el que se basa Markowitz para formular su teoría. Las bases de este modelo matemático se tratarán en el siguiente capítulo.



# 4 BASES DE LA TEORÍA DE LA CARTERA MEDIA-VARIANZA

---

*The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.*

Claude Shannon, 1948

En la teoría de Markowitz [1] se modelan las tasas de retorno de los activos como variables aleatorias. El objetivo del modelo es conseguir una relación de pesos óptimas. Markowitz establece que la relación óptima es aquella que consigue una referencia de tasas de retorno esperada aceptable minimizando la volatilidad. Es decir, conseguir una proporción de cada activo que nos proporcione una rentabilidad aceptable minimizando el riesgo. La volatilidad es una medida de la frecuencia y la intensidad de los cambios de precio de un activo. Aquí la varianza ocupa el papel de la volatilidad.

En este capítulo se introducirá a los modelos matemáticos de optimización. También se plantearán desarrollos matemáticos para demostrar la configuración del modelo específico de Markowitz

## 4.1 La función objetivo del modelo

En todo modelo de optimización hay una función objetivo. El objetivo de esta función es evaluar la optimalidad de un dato o conjunto de ellos. Este dato es llamado solución cuando cumple los requisitos que impone el problema que resuelve el modelo. Estos requisitos se llaman restricciones. El tema de las restricciones se tratará más adelante.

Para evaluar 2 soluciones distintas, se introducen en la función objetivo y ésta (teóricamente) nos dará 2 valores distintos. A priori, no se puede saber cuál es mejor que la otra. Para decidir qué solución es más próxima a la solución óptima, antes se ha definido qué se busca en la función objetivo, si un máximo o un mínimo. Si se busca un máximo, la solución que obtenga mayor resultado al aplicar la función objetivo será mejor solución que la otra. Por ende, si se busca un mínimo, será la solución que menor resultado obtenga.

En consecuencia, si una solución, entre todas las demás, consigue el máximo valor de la función objetivo (o menor en caso de que el modelo busque minimizar), esta solución se denominará óptimo o solución óptima.

Con todo esto, ya se puede explicar cómo se define la función objetivo del modelo de Markowitz. En su teoría, Markowitz establece que la solución óptima sea la que presente menor volatilidad. Como variables que nos ofrecerán la posibilidad de evaluar las soluciones serán las tasas de retornos de los activos  $r_i$ . Para recoger las tasas de cada activo  $i$ , para  $i = 1, 2, \dots, n$ , se define el siguiente vector, que será un vector de variables aleatorias:

$$z = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}$$

Para medir la volatilidad de la solución, se establece una matriz de covarianzas de todos los activos de la cartera y que son variables aleatorias. Esta matriz se define como  $\Sigma = \text{cov}(z)$ . Se establece el vector  $\omega = (\omega_1, \omega_2, \dots, \omega_n)^T$  como vector que recoge los pesos que corresponden a cada uno de los activos que componen la cartera.

Con todo esto se una función cuyo objetivo es minimizar la volatilidad de la variable aleatoria  $z$ . Como medida de la volatilidad se utiliza la covarianza de esta variable, que recoge  $n$  tasas de retorno aleatorias. Sin embargo, para transformar la matriz  $\Sigma$  en un número, se multiplica por un medio y 2 veces por el vector  $\omega$  de los pesos de cada activo dentro de la cartera. Por lo que la función objetivo quedaría así:

$$\text{minimizar } \frac{1}{2} \cdot \omega^T \cdot \Sigma \cdot \omega$$

De esta manera, esta función es capaz de evaluar la volatilidad de una proporción de pesos de unos activos dada. Por lo que evaluando distintos resultados que ofrecen las distintas relaciones de pesos se puede discernir cuáles ofrece mejores o peores resultados.

## 4.2 Restricciones del modelo de Markowitz

Como anteriormente se ha afirmado, las soluciones de los problemas de optimización se encuentran sujetos a unos requisitos que deben cumplir. Estos requisitos se llaman restricciones. Existen 2 tipos de restricciones: las restricciones de igualdad y de desigualdad. Las restricciones de igualdad son de la forma:

$$g_i(x) = c_i \quad \forall i = 1, 2, \dots$$

Por el contrario, las restricciones de desigualdad presentan el siguiente aspecto:

$$h_j(x) \geq d_j \quad \forall j = 1, 2, \dots$$

Todas las restricciones se pueden representar en el espacio. Todas las restricciones deben formar una región cerrada para que el problema tenga una solución. El conjunto de soluciones que forman esta región comprendida entre las restricciones se llama región admisible y recoge todas las soluciones admisibles que cumplen todas las restricciones del problema. Cuando una solución no se encuentra en esta región o no cumple al menos 1 restricción se le llama solución no admisible o también se dice que no es solución del problema. Nunca aplicará evaluar la función objetivo de una solución no admisible del problema.

Con todo esto, se pueden definir las restricciones del modelo. Como se ha mencionado anteriormente, todas las soluciones válidas deben cumplir 2 requisitos que están representados en sendas restricciones. La primera de ellas se menciona en el capítulo 3 cuando se introducen los pesos de los activos que componen una cartera. Se estableció que la suma de todos los pesos de los activos debía ser igual a 1. Esto se traduce en la siguiente restricción de igualdad:

$$e^T \cdot \omega = 1$$

Donde:

- $e$ : vector de unos
- $\omega$ : vector que contiene los pesos de todos los activos que componen una cartera

La segunda restricción viene de traducir el requisito que se describe en la introducción del presente capítulo 4. El requisito implica que el modelo de Markowitz busca conseguir una proporción de cada activo que nos proporcione, al menos, una rentabilidad aceptable minimizando el riesgo. Esta rentabilidad mínima aceptable es un dato que puede variar en función de la naturaleza del inversor y las posibilidades del mercado. Por otra parte, la rentabilidad esperada de una cartera se puede estimar aplicando la esperanza matemática sobre las el histórico de tasas de retorno de los activos de la cartera. De esta forma se obtiene una estimación sobre la magnitud de la rentabilidad que podemos esperar de cada activo. Con todo esto se puede establecer la siguiente

restricción del modelo:

$$m^T \cdot \omega \geq \mu_b$$

Donde:

- $m = (\mu_1, \mu_2, \dots, \mu_n)^T$
- $\mu_i = E(r_i)$
- $\mu_b$ : rentabilidad mínima esperada de la cartera

A continuación, se detallan todas las variables y datos que componen el modelo junto con la función objetivo y las restricciones que permitirán la optimización de las variables:

$$\begin{aligned} \text{minimizar} \quad & \frac{1}{2} \cdot \omega^T \cdot \Sigma \cdot \omega \\ \text{sujeto a} \quad & e^T \cdot \omega = 1 \\ & m^T \cdot \omega \geq \mu_b \end{aligned}$$

Donde:

- $\omega = (\omega_1, \omega_2, \dots, \omega_n)^T$
- $z = (r_1, r_2, \dots, r_n)^T$
- $\Sigma = \text{cov}(z)$
- $e = (1, 1, \dots, 1)^T$
- $\mu_i = E(r_i)$
- $m = (\mu_1, \mu_2, \dots, \mu_n)^T$
- $\mu_b = \text{valor numérico}$

En este capítulo describimos los puntos que pueden incluirse con el formato propuesto. En primer lugar, la longitud de un libro, en general, justifica su separación en partes. Una posibilidad es que un libro esté dividido en Partes y esta a su vez en Capítulos. Y, por último, a veces existen Apéndices que se incorporan cuando han acabado los capítulos. En nuestro caso sólo hemos considerado la posibilidad de dividir el libro en capítulos. Además, existen un conjunto de elementos como dedicatoria, prefacio, agradecimientos, cubierta, etc, que también son partes que se han tenido en cuenta.

En un nivel de descripción diferente, podríamos considerar que un libro se encuentra dividido en cubierta, página de título y trasera de la página de título, elementos antes del cuerpo del libro, tales como agradecimientos, prefacio, índices, etc, el cuerpo del libro en sí, dividido en capítulos y esto a su vez en secciones, subsecciones, subsubsecciones, subcapítulos, apéndices y, por último, la parte del libro después del cuerpo, que agruparía elementos tales como la lista de figuras del libro, la bibliografía, el índice, etc.

### 4.3 Condiciones de Karush-Kuhn y Tucker

El modelo que se ha presentado anteriormente presenta un problema para su resolución. Los problemas con una función objetivo no lineal no se pueden resolver directamente. Existen distintas formas de resolver los problemas de optimización como por ejemplo los multiplicadores de Lagrange. Este método consiste en eliminar cada restricción añadiéndola, junto a su correspondiente multiplicador, a la función objetivo junto con un multiplicador. De esta forma se consigue una combinación lineal de la función objetivo y las funciones implicadas en las restricciones.

Otra manera de evaluar la optimalidad de una solución admisible del problema es utilizando las condiciones de Karush-Kuhn-Tucker [1] [2] (condiciones KKT de aquí en adelante). Una solución, para ser óptima al menos localmente, obligatoriamente tiene que satisfacer las condiciones KKT. La forma de afrontar los problemas de optimización es una generalización del método de los multiplicadores de Lagrange, que solo admite restricciones de igualdad. Gracias a las condiciones de KKT se resuelve la limitación principal del método de los multiplicadores porque desde entonces es posible afrontar los problemas con restricciones de desigualdad.

La primera de las condiciones de KKT es llamada la condición estacionaria y se formula como la igualdad del gradiente del *Lagrangiano* a 0. El *Lagrangiano* es una función que está formada por la resta a la función objetivo de la suma de los productos de las nuevas variables conocidas como *multiplicadores de Lagrange* y una restricción del problema. Cada restricción tiene asociada un único multiplicador. Para calcular esta función hay que tomar la desigualdad con un 0 en uno de los miembros de la desigualdad. Se puede llegar a la siguiente expresión del *Lagrangiano*:

$$L(\omega, \lambda, \gamma) = \text{función objetivo}(\omega) - \sum_i \lambda_i \cdot \text{restricción}_i(\omega)$$

$$L(\omega, \lambda, \gamma) = \frac{1}{2} \cdot \omega^T \cdot \Sigma \cdot \omega - \lambda \cdot (m^T \cdot \omega - \mu_b) - \gamma \cdot (m^T \cdot \omega - 1)$$

Siendo  $\lambda$  y  $\gamma$  los *multiplicadores de Lagrange* del problema. Cuando se calcula el gradiente de esta función, en este caso, como el problema solo tiene una variable, se calcula la derivada parcial de la función respecto a la variable del problema, se obtiene la siguiente ecuación que es la primera condición de KKT:

$$\nabla L = \frac{\partial L}{\partial \omega} = \Sigma \cdot \omega - \lambda \cdot m - \gamma \cdot e = 0$$

La segunda condición es la condición de admisibilidad por la que se mantienen las restricciones del problema. De esta manera se asegura que toda solución que cumpla las condiciones de KKT sea admisible y por tanto una solución viable del problema.

$$e^T \cdot \omega = 1$$

$$m^T \cdot \omega \geq \mu_b$$

La tercera condición de KKT, conocida como condición de holgura, procede cuando el problema tiene restricciones de desigualdad no estrictas. Estas restricciones en primer lugar se tienen que transformar para tener la forma:

$$f(x) \leq 0$$

A continuación, se procede convirtiendo la desigualdad en igualdad añadiendo una nueva variable de holgura que corresponde con uno de los *multiplicadores de Lagrange*. Este multiplicador, operando, se coloca multiplicando a la función obtenida en el primer paso a seguir. Por lo que finalmente debe quedar nuestra condición de KKT de la siguiente manera:

$$\lambda^T \cdot (m^T \cdot \omega - \mu_b) = 0$$

Por último, la cuarta condición de KKT, también llamada condición de signo. En este caso, se impone que los multiplicadores de las restricciones de desigualdad no sean positivos para el problema que estamos tratando. En nuestro problema solo tenemos una restricción de desigualdad. Esta restricción, aparece multiplicada en el *Lagrangiano* por el multiplicador  $\lambda$  por lo que la cuarta condición de KKT tiene la siguiente formulación:

$$\lambda \geq 0$$

Con todo esto, ya conocemos las condiciones de KKT. Estas son condiciones necesarias que debe cumplir todo punto cercano a los puntos críticos de una función objetivo. Además, cuando una solución cumple todas las condiciones se puede asegurar que no se trata de ningún punto crítico degenerado. Y finalmente, las condiciones de KKT del modelo son las siguientes:

$$\begin{aligned} \Sigma \cdot \omega - \lambda \cdot m - \gamma \cdot e &= 0 \\ \mu_b - m^T \cdot \omega \leq 0, \quad e^T \cdot \omega - 1 &= 0 \\ \lambda^T \cdot (m^T \cdot \omega - \mu_b) &= 0 \\ \lambda \geq 0, \quad \gamma &\geq 0 \end{aligned}$$

Para algún  $\lambda, \gamma \in \mathbb{R}$ . Nuestra variable, la matriz covarianza, siempre va a ser simétrica y definida positiva.



Gracias a esto, se puede afirmar que cuando  $(\omega, \lambda, \gamma)$  satisfacen las condiciones de KKT entonces el vector de los pesos de los activos de la cartera  $\omega$  es inequívocamente una solución al modelo de Markowitz. De esto, es fácilmente deducible que el modelo es admisible y, por tanto, siempre habrá una solución y también se puede encontrar un conjunto de  $(\omega, \lambda, \gamma)$  que satisfagan las condiciones de KKT.

Con todo esto, se puede establecer una guía para el procedimiento de resolución del problema de optimización que plantea el modelo de la teoría de la cartera media-varianza de Markowitz.

## 4.4 Procedimiento de resolución del modelo de Markowitz

Como ya se ha explicado, el problema que estamos tratando es un problema con función objetivo no lineal y restricciones de igualdad y desigualdad. Por esto haremos uso de las condiciones de KKT para verificar si un punto es solución cercana al óptimo. Con todo esto, se puede establecer una guía para el procedimiento de resolución del problema de optimización que plantea el modelo de la teoría de la cartera media-varianza de Markowitz.

### 4.4.1 Revisar admisibilidad del modelo

En primer lugar, es necesario comprobar que el modelo es admisible. Para asegurar que el modelo es admisible hay que revisar que el vector  $m$  es paralelo al vector de unos  $e$ . Si son vectores paralelos, se puede establecer que:

$$m = \tau \cdot e$$

para algún valor de  $\tau \in \mathbb{R}$ . Una vez verificado el paralelismo de ambos vectores habría que calcular la inversa de la matriz de covarianza  $\Sigma^{-1}$  para posteriormente hallar los pesos de mínima varianza. Esta relación de pesos es una solución del problema.

$$\omega_{\text{mínima varianza}} = \omega_{\text{mín var}} = \frac{\Sigma^{-1} \cdot e}{e^T \cdot \Sigma^{-1} \cdot e}$$

Sin embargo, para que  $\omega_{\text{mín var}}$  sea solución admisible del problema se tiene que verificar la siguiente condición:

$$\mu_b \leq \tau$$

### 4.4.2 Revisar la solución de mínima varianza

Para comprobar si la solución de mínima varianza es una solución cercana al óptimo, tenemos que revisar las condiciones de KKT anteriormente explicadas. En concreto, habría que comprobar que se cumple que:

$$m^T \cdot \omega_{\text{mín var}} \geq \mu_b$$

Aunque se haya comprobado en el apartado 4.4.1 que existe una solución admisible al problema siempre que los vectores  $m$  y  $e$  sean paralelos, si no cumple la condición anterior,  $\omega_{\text{mín var}}$  no es una solución que se pueda considerar óptima y habría que buscar otra solución.

### 4.4.3 Calcular una segunda cartera que sea admisible

Una vez verificado que el problema es admisible y después de comprobar que la solución de mínima varianza no es una solución óptima, necesitamos hallar una solución que sí sea óptima. Para ello, se calcula la solución de los pesos del mercado:

$$\omega_{\text{market}} = \omega_{mk} = \frac{\Sigma^{-1} \cdot m}{e^T \cdot \Sigma^{-1} \cdot m}$$

Posteriormente, se construye un vector con la diferencia de los valores de los pesos de los vectores  $\omega_{mk}$  y

$\omega_{\min var}$ . A este vector lo llamaremos  $v$  y se define a continuación:

$$v = \omega_{mk} - \omega_{\min var}$$

Este vector se utilizará para explorar las soluciones que son combinación lineal de estas 2 soluciones que ya sabemos que sí son solución del problema. El conjunto de las carteras que son combinación lineal de estas dos soluciones cumple la siguiente ecuación:

$$\omega = \omega_{\min var} + \alpha \cdot (\omega_{mk} - \omega_{\min var}) = \omega_{\min var} + \alpha \cdot v$$

A partir de esta ecuación y obligando a este conjunto de soluciones a cumplir la condición de KKT imponiendo una igualdad para conseguir la identidad:

$$m^T \cdot \omega_{\min var} = \mu_b$$

Y ahora podemos hallar el valor de  $\alpha$  que verifica la condición de optimalidad y sabemos que es óptima porque es combinación lineal de 2 soluciones que lo son. Esto se puede afirmar gracias a que la matriz de covarianza es definida positiva y por tanto la región de admisibilidad que forman todas las soluciones admisibles es convexa.

$$\alpha = \frac{\mu_b - m^T \cdot \omega_{\min var}}{m^T \cdot v}$$

De esta manera se puede conseguir una solución que sea admisible además de óptima porque cumplen las condiciones de KKT del modelo.

# 5 CONCLUSIONES DE LA TEORÍA DE LA CARTERA MEDIA-VARIANZA

---

A raíz de la teoría de la cartera de la media-varianza de Markowitz [1], se desarrollan varios conceptos útiles para comprender mejor el ambiente del mercado bursátil. En este capítulo se explicarán varias ideas que surgen a partir de la de Markowitz. Conceptos como la frontera eficiente, el teorema de los dos fondos, el modelo de valoración de activos financieros CAPM (Capital Asset Pricing Model en inglés) y la razón de Sharpe entre otros.

## 5.1 Frontera eficiente

Como se menciona en la introducción al capítulo, este concepto nace a partir del modelo que definió Harry Markowitz en 1952. En este concepto se asume que todas las carteras se caracterizan por su rentabilidad esperada y por su desviación típica.

La rentabilidad esperada de una cartera se obtiene a partir de la media ponderada de las rentabilidades esperadas de los activos que la componen. Por lo que la ecuación que define la rentabilidad esperada de una cartera se formula de la siguiente manera:

$$\text{rentabilidad cartera} = r_s = \frac{\sum_{i=0}^n r_i \cdot \omega_i}{\sum_{i=0}^n \omega_i}$$

En cambio, la desviación típica no responde de igual manera. En las carteras, la desviación típica ( $\sigma_i$ ) se utiliza como medida del riesgo que conllevan los activos que forman parte de la cartera. La dispersión de una cartera no depende únicamente de las propiedades individuales de cada activo, también dependen de la relación que tengan todos los pares de activos entre sí. Para que el riesgo de la cartera se vea influenciado por este efecto se utiliza la covarianza ( $\sigma_{ij}$ ). Sin embargo, trabajar con desviaciones típicas directamente puede ser más complicado y por eso se utiliza la varianza ( $\sigma_i^2$ ). Estas son las ecuaciones que cumplen estas propiedades:

$$\text{Desviación típica de la cartera} = \sqrt{\sigma_s^2} = \sigma_s$$

$$\text{Varianza de la cartera} = \sigma_s^2 = \sum_{i=1}^n \sum_{j=1}^n x_i \cdot x_j \cdot \sigma_{ij}$$

Con todo esto, se pueden determinar todo un conjunto de carteras que es infinito ya que pueden tener cualquier combinación de rentabilidad esperada y desviación típica. Por ejemplo, se utilizan 3 activos arbitrarios y se crea una muestra con las rentabilidades que poseen durante 12 periodos distintos. A partir de esta muestra, se obtienen la rentabilidad esperada y la desviación típica muestral de cada uno de los activos. Para calcular la covarianza, se emplea la herramienta Análisis de datos de Microsoft Excel, introduciendo como entrada la muestra de las rentabilidades de cada activo en cada periodo. Acto seguido, se genera un número de arbitrario (en nuestro ejemplo usaremos 2500) de vectores aleatorios que contienen proporciones de pesos de 2500

carteras. Una vez que obtenidas la rentabilidad esperada de cada activo, la desviación típica de cada activo, la matriz de covarianzas y combinaciones de pesos aleatorios, se calculan la rentabilidad esperada y la desviación típica de cada una de las carteras. Se genera una gráfica cuyo eje de abscisas será la desviación típica de la cartera y su eje de ordenada sea la rentabilidad esperada de la cartera. Si en estos ejes se reprecetan los puntos que ocupan cada una de las carteras, se consigue una gráfica como la siguiente:

Si se observa la gráfica, se forma una curva y otros puntos interiores a ésta. La frontera eficiente la forman todos los puntos comprendidos desde el vértice de la curva hasta el punto más elevado de la gráfica. La frontera eficiente está conformada por el conjunto de todas las carteras eficientes generadas a partir de los datos que se han calculado a modo de ejemplo. Una cartera eficiente es toda aquella cartera que permite maximizar la rentabilidad para cada nivel de riesgo. Para cada nivel de riesgo que pueden ofrecer las distintas carteras, le corresponde un valor máximo de rentabilidad. Ese valor máximo de rentabilidad con cada nivel de riesgo es lo que define a una cartera eficiente.

Si en lugar de emplear 3 activos distintos se utilizaran únicamente 2, entonces solo se formará una curva sin puntos interiores. En este caso, la frontera eficiente se puede observar de una forma mucho menos intuitiva, pero gracias a ello se introduce la premisa del teorema de los dos fondos. La curva que formarían las carteras formadas por 2 activos cualesquiera sería de la forma que se representa en la siguiente gráfica:

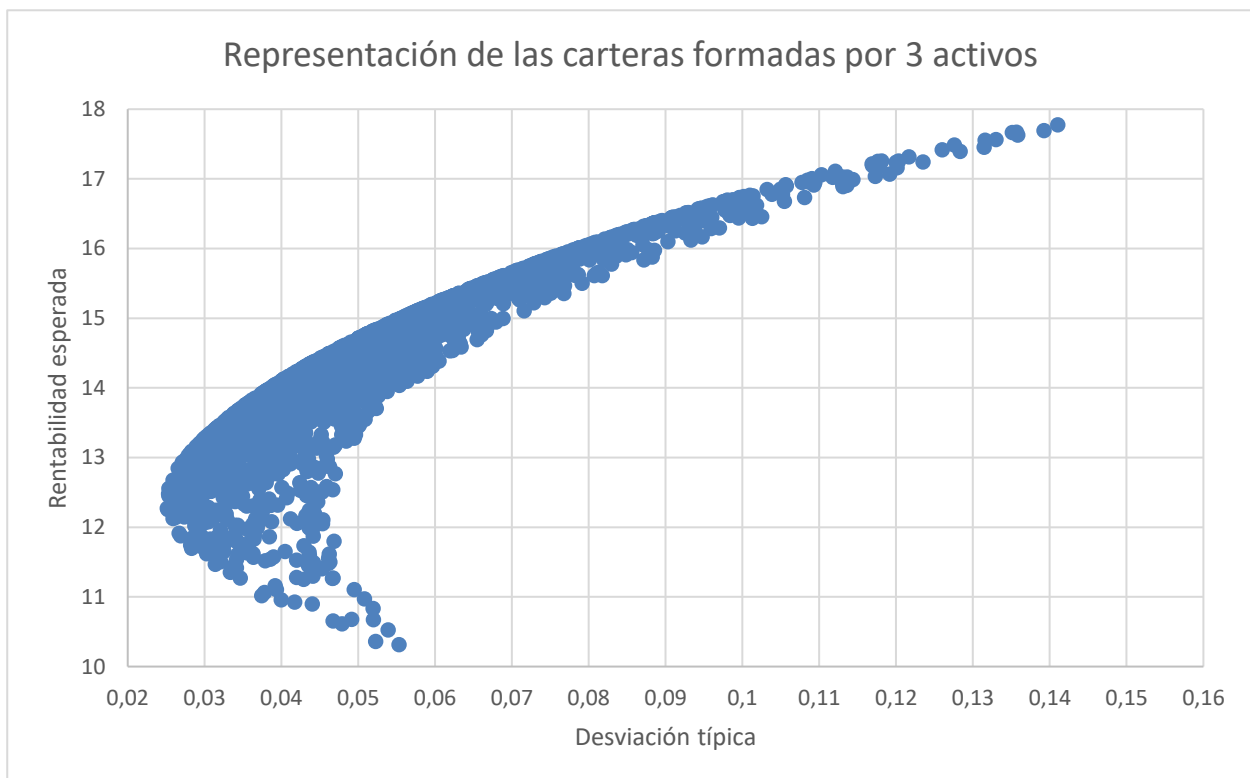


Ilustración 1

Si se observa la gráfica, se forma una curva y otros puntos interiores a ésta. La frontera eficiente la forman todos los puntos comprendidos desde el vértice de la curva hasta el punto más elevado de la gráfica. La frontera eficiente está conformada por el conjunto de todas las carteras eficientes generadas a partir de los datos que se han calculado a modo de ejemplo. Una cartera eficiente es toda aquella cartera que permite maximizar la rentabilidad para cada nivel de riesgo. Para cada nivel de riesgo que pueden ofrecer las distintas carteras, le corresponde un valor máximo de rentabilidad. Ese valor máximo de rentabilidad con cada nivel de riesgo es lo que define a una cartera eficiente.

Si en lugar de emplear 3 activos distintos se utilizaran únicamente 2, entonces solo se formará una curva sin puntos interiores. En este caso, la frontera eficiente se puede observar de una forma mucho menos intuitiva,

pero gracias a ello se introduce la premisa del teorema de los dos fondos. La curva que formarían las carteras formadas por 2 activos cualesquiera sería de la forma que se representa en la siguiente gráfica:

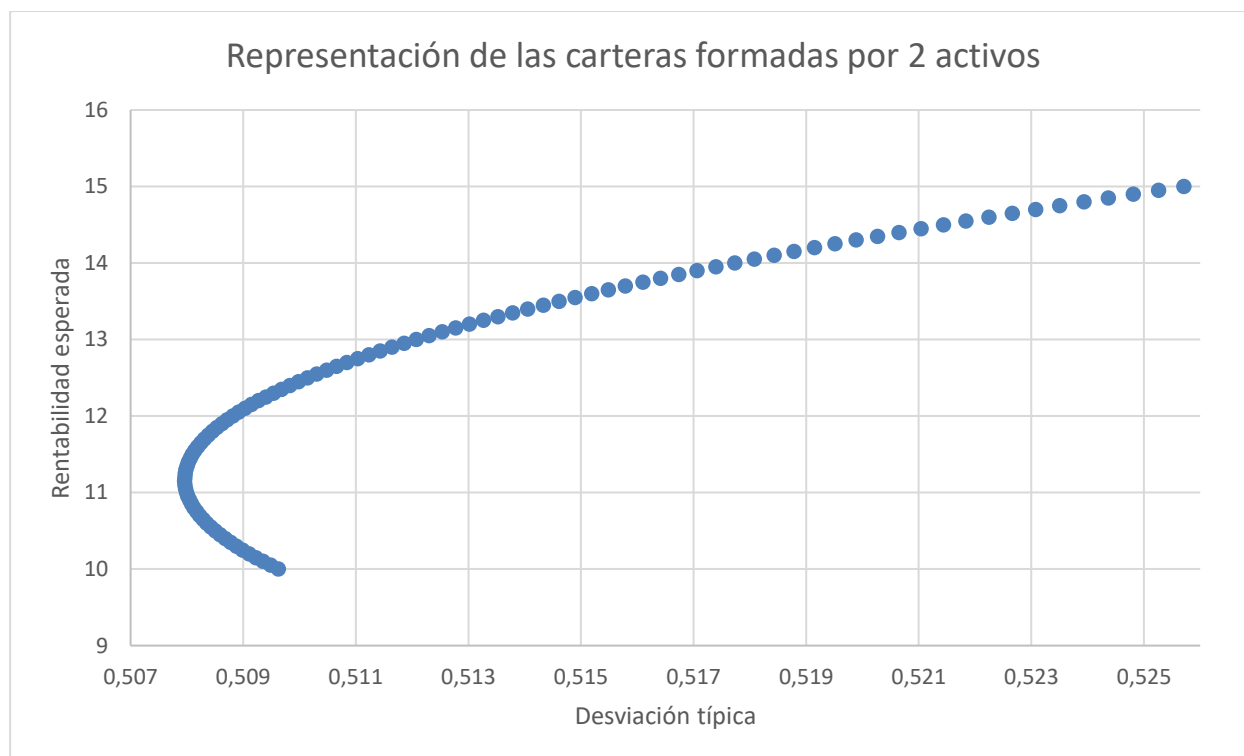


Ilustración 2

Esta idea junto con este gráfico serán la base para introducir el teorema de los dos fondos en el siguiente subcapítulo.

El motivo por el cual a mayores rentabilidades esperadas mayores riesgos es muy simple. El valor de un activo es una cierta cantidad. Cuando un inversor adquiere dicho activo, esperará que dicho valor aumente para sacar rentabilidad de la operación. Pues bien, cuando el valor del activo cambia, hay una diferencia entre el nuevo valor y un valor medio que es la referencia. Esta diferencia y su frecuencia es lo que se entiende por riesgo. Por lo que para conseguir que el valor de un activo aumente mucho, hay que aceptar que aumente también el riesgo.

Por otro lado, cabe destacar de qué factores depende la frontera eficiente para trazar una curva semejante pero cambiando su posición o curvatura. La posición dependerá de donde se sitúen los activos que van a componer la cartera. Estos puntos pueden estar muy distantes si los activos tienen rentabilidad esperada y desviación típica muy dispares. La curvatura dependerá del coeficiente de correlación que guarden los dos activos. Este coeficiente de correlación es una medida de la dispersión de las muestras de los dos activos de la cartera. Este coeficiente puede tomar valores entre -1 y 1. Si toma de valor 1 la curva del gráfico sería una recta que une los puntos que representan las carteras formadas únicamente por cada activo de los dos que pueden componer la cartera. En cambio, cuando toma un valor de -1 se forman dos segmentos que se unen en un punto del eje de ordenadas.

## 5.2 Teorema de los dos fondos

La frontera eficiente es la parte de la curva que se encuentra desde la mínima desviación típica hasta el último punto de la curva siguiendo un sentido ascendente. Por definición, las carteras que componen la frontera eficiente son carteras eficientes por lo que se podría asegurar que son carteras cercanas un valor óptimo. A partir de aquí, se puede considerar que estos dos extremos de la frontera eficiente son dos carteras distintas. En el caso de las carteras formadas por 2 activos distintos, cada extremo de la curva que se ha representado serán las carteras formadas íntegramente por cada uno de los activos. Como se explica en el subcapítulo 5.1, la

forma de la curva que da lugar a la cartera eficiente depende de la posición que ocupan las 2 carteras y del coeficiente de correlación que generan ambos activos.

Este teorema establece que, dadas dos soluciones diferentes del modelo pertenecientes a la frontera eficiente, se pueden generar un conjunto de soluciones que se encuentran dentro de la misma frontera eficiente.

Este teorema se puede demostrar de forma sencilla. Se suponen las rentabilidades esperadas de dos activos mayores que la rentabilidad esperada de la solución de mínima varianza. Bajo esta premisa, la condición de admisibilidad se cumple para ambas soluciones puesto que son mayores que la de la solución de mínima varianza que ya se ha asumido que es mayor que la rentabilidad mínima esperada. Adicionalmente, también se cumple la condición de holgura de KKT. Por lo que las condiciones de KKT que deben cumplir todas las soluciones eficientes son:

$$\Sigma \cdot \omega - \lambda \cdot m - \gamma \cdot e = 0$$

$$e^T \cdot \omega = 1$$

$$m^T \cdot \omega = \mu$$

Estas condiciones resultantes son 3 ecuaciones lineales que componen un sistema compatible para todo valor de  $\mu > \mu_{\min var}$ . Por lo que, teniendo 2 activos con rentabilidades mayores que la de la solución de mínima varianza tal que  $\mu_1 < \mu_2$ , se puede recorrer la frontera eficiente a partir de todas las rentabilidades comprendidas entre estos dos activos utilizando la siguiente expresión:

$$\mu_\alpha = \alpha \cdot \mu_1 + (1 - \alpha) \cdot \mu_2 \quad \forall \alpha \in \mathbb{R}$$

Pues si a cada una de estas rentabilidades le corresponde una solución perteneciente a la frontera eficiente, todas las soluciones asociadas a estas rentabilidades se pueden obtener con la ecuación:

$$(\omega_\alpha, \lambda_\alpha, \gamma_\alpha) = \alpha \cdot (\omega_1, \lambda_1, \gamma_1) + (1 - \alpha) \cdot (\omega_2, \lambda_2, \gamma_2)$$

Entonces la cartera asociada viene dada por:

$$r_\alpha = \alpha \cdot r_1 + (1 - \alpha) \cdot r_2 \quad \forall \alpha \in \left[ \frac{\mu_{\min var} - \mu_1}{\mu_1 - \mu_2}, \infty \right)$$

Conforme varía el parámetro  $\alpha$ , las parejas de desviación típica y de rentabilidad esperada de la cartera, van trazando una curva que forma parte de la frontera eficiente y es la curva de desviación media estándar entre las dos carteras  $r_1$  y  $r_2$ .

Por lo tanto, gracias a la teoría de la media-varianza de Markowitz, solo se necesitan dos fondos para que cualquier inversor obtenga una cartera que le proporcione el equilibrio deseado entre retorno y riesgo.

### 5.3 Teorema de los fondos mutuos

Anteriormente, se ha asumido que todos los activos que componen la cartera tienen riesgo y la matriz de covarianza de los retornos de estos activos no es singular y por tanto invertible. No obstante, existen activos, como las letras del tesoro, que tienen un nivel de riesgo muy bajo y se modelan como si fueran activos *libres de riesgo*.

Introduciendo el concepto de activo libre de riesgo dentro de la teoría de Markowitz, se denota este activo libre de riesgo como  $f$ , con retorno libre de riesgo  $r_f$ . Como consecuencia de utilizar la varianza de las tasas de retorno como medida del riesgo, la varianza de los activos libres de riesgo será 0. Tomando un vector aleatorio con los retornos de los activos con riesgo  $x$ , se conforma el vector  $\hat{x}$  que se forma incluyendo el retorno del activo libre de riesgo.

$$\hat{x} = \begin{pmatrix} r_f \\ x \end{pmatrix} = \begin{pmatrix} r_f \\ r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}$$

La matriz de covarianza que surge de este vector será de la forma:

$$\hat{\Sigma} = \begin{bmatrix} 0 & 0 \\ 0 & \Sigma \end{bmatrix}$$

Siendo  $\Sigma$  la matriz de covarianza de  $x$ . Implementando estos cambios al modelo de Markowitz, el nuevo modelo estaría compuesto de la siguiente manera:

$$\begin{aligned} & \text{minimizar} && \frac{1}{2} \cdot \omega^T \cdot \Sigma \cdot \omega \\ & \text{sujeto a:} && \omega_0 + e^T \cdot \omega = 1 \\ & && r_f \cdot \omega_0 + m^T \cdot \omega \geq \mu_b \end{aligned}$$

Donde  $\omega_0$  es el peso correspondiente al activo libre de riesgo y  $\mu_b, m, \omega$  y  $e$  tienen la misma interpretación que en el capítulo 4. Si una cartera incluye un activo libre de riesgo, la rentabilidad mínima exigible  $\mu_b$  será mayor que la rentabilidad  $r_f$  de este activo. También se considera que la matriz de covarianza mantiene su condición de no singularidad y que se trata de una matriz invertible. Con todo esto, como se explica en el subcapítulo 4.3, se deducen las siguientes condiciones de KKT:

$$\begin{aligned} (1) \quad & \Sigma \cdot \omega - \lambda \cdot m - \gamma \cdot e = 0 \\ (2) \quad & \lambda \cdot r_f + \gamma = 0 \\ (3) \quad & \lambda^T \cdot (m^T \cdot \omega - \mu_b) = 0 \\ (4) \quad & \omega_0 + e^T \cdot \omega = 1 \\ (5) \quad & r_f \cdot \omega_0 + m^T \cdot \omega \geq \mu_b \\ (6) \quad & \lambda \geq 0 \end{aligned}$$

A continuación, se plantean 2 posibles escenarios según sea la restricción  $r_f \cdot \omega_0 + m^T \cdot \omega \geq \mu_b$  inactiva o activa. Una restricción de desigualdad no estricta es activa cuando ambos miembros de la restricción son iguales. Si es inactiva, entonces obligatoriamente el multiplicador de *Lagrange*  $\lambda$  tomará el valor 0. Consecuentemente, el valor del multiplicador  $\gamma$  tomará el mismo valor, y esto hará que el vector de pesos  $\omega$  asociado a los activos con riesgo tome el valor de 0. Entonces, el óptimo vendrá dado por  $\omega_0 = 1$ , una cartera optimizada compuesta íntegramente por el activo sin riesgo.

En cambio, si la restricción resultase activa, el procedimiento para obtener la solución sería el siguiente:

1. Multiplicar la ecuación (1) por  $\Sigma^{-1}$  y aplicando la ecuación (2) se obtendría la siguiente igualdad:

$$(7) \quad \omega = \lambda \cdot \Sigma^{-1} \cdot (m - r_f \cdot e)$$

2. Utilizar la igualdad de la restricción activa y la ecuación (4) se obtienen unas identidades que componen el siguiente sistema de ecuaciones:

$$\begin{bmatrix} 1 & e^T \cdot \Sigma^{-1} \cdot (m - r_f \cdot e) \\ r_f & m^T \cdot \Sigma^{-1} \cdot (m - r_f \cdot e) \end{bmatrix} \cdot \begin{pmatrix} \omega_0 \\ \lambda \end{pmatrix} = \begin{pmatrix} 1 \\ r_f \end{pmatrix}$$

3. Una vez se tiene el sistema de ecuaciones, se puede resolver aplicando la eliminación Gaussiana y se obtiene como solución:

$$\begin{pmatrix} \omega_0 \\ \lambda \end{pmatrix} = \begin{pmatrix} 1 - (\mu_b - r_f) \cdot \frac{e^T \cdot \Sigma^{-1} \cdot (m - r_f \cdot e)}{(m - r_f \cdot e)^T \cdot \Sigma^{-1} \cdot (m - r_f \cdot e)} \\ \frac{\mu_b - r_f}{(m - r_f \cdot e)^T \cdot \Sigma^{-1} \cdot (m - r_f \cdot e)} \end{pmatrix}$$

4. Cuando se ha conseguido esta solución del sistema, sabemos que  $m \neq r_f \cdot e$ , porque con lo que se ha detallado anteriormente se sabe que, si no es así,  $\omega$  y  $\omega_0$  valdrían 0 y 1 respectivamente. Por lo que sustituyendo en la solución del sistema con la ecuación (7) resulta:

$$\begin{pmatrix} \omega_0 \\ \omega \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha \cdot \left[ \begin{pmatrix} 1 - e^T \cdot \Sigma^{-1} \cdot (m - r_f \cdot e) \\ \Sigma^{-1} \cdot (m - r_f \cdot e) \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right]$$

Donde

$$\alpha = \frac{\mu_b - r_f}{(m - r_f \cdot e)^T \cdot \Sigma^{-1} \cdot (m - r_f \cdot e)}$$

5. Gracias a este proceso, se puede afirmar que los pesos de la cartera óptima será una combinación lineal de los vectores:

$$\omega_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \omega = \begin{pmatrix} 1 - e^T \cdot \Sigma^{-1} \cdot (m - r_f \cdot e) \\ \Sigma^{-1} \cdot (m - r_f \cdot e) \end{pmatrix}$$

A partir de este procedimiento, se enuncia el teorema de los fondos mutuos (*The One Fund Theorem* en inglés) que dice:

*Si la cartera seleccionada incluye un activo libre de riesgo, entonces existe una única cartera P (o fondo) tal que toda cartera eficiente puede construirse como combinación lineal del activo libre de riesgo y la cartera P.*

## 5.4 Modelo de valorización de activos financieros (CAPM)

Después de conocer el teorema de los fondos mutuos, sobre el que se basa William Sharpe para desarrollar este modelo, queda demostrado que se puede elegir un activo libre de riesgo como parte de una cartera. Este hecho provoca que la frontera eficiente del modelo de Markowitz resulte una recta. En primer lugar, se define una cartera que contiene los activos del mercado, que cumple la siguiente fórmula:

$$r_M = \begin{pmatrix} r_f \\ r \end{pmatrix}^T \cdot \begin{pmatrix} 1 - e^T \cdot \Sigma^{-1} \cdot (m - r_f \cdot e) \\ \Sigma^{-1} \cdot (m - r_f \cdot e) \end{pmatrix}$$

Siendo  $r$  cualquier cartera eficiente. Esta cartera, tiene una proporción de pesos  $\omega$  y  $\omega_0$  que son solución para el sistema de ecuaciones que se planteó en la deducción del *Teorema de los fondos mutuos*. Esta cartera viene definida por:

$$r = r_f \cdot \omega_0 + r^T \cdot \omega$$

Como se define en el subcapítulo 4.2, la rentabilidad esperada de un activo o cartera es el resultado de la esperanza matemática de la cartera. Por lo que la rentabilidad esperada de la cartera  $r$  tiene que verificar esta igualdad:

$$\mu_r = r_f + \frac{\mu_r - r_f}{\sigma_M} \cdot \sigma_r$$

Donde:

- $\mu_r = E(r)$
- $\sigma_r^2 = var(r)$

Con todo lo expuesto anteriormente, se pueden deducir las siguientes expresiones:

$$\begin{aligned} \mu_M &= E(r_M) = r_f + (m - r_f \cdot e)^T \cdot \Sigma^{-1} \cdot (m - r_f \cdot e) \\ \sigma_M^2 &= var(r_M) = (m - r_f \cdot e)^T \cdot \Sigma^{-1} \cdot (m - r_f \cdot e) \end{aligned}$$

Estas ecuaciones generan una recta que en este caso será la frontera eficiente.

Ahora, se puede considerar que esta cartera del mercado engloba todos los activos posibles de todo el mercado de valores. La recta que contiene la frontera eficiente comienza en los activos libres de riesgo en el punto con rentabilidad  $r_f$  y desviación típica igual a 0. Esta recta recibe el nombre de *línea del mercado de valores*. La pendiente de la recta se denomina el *precio del riesgo* y se calcula utilizando esta fórmula:



$$\text{precio del riesgo} = \frac{\mu_M - r_f}{\sigma_M}$$

Considerando una cartera compuesta por la cartera de mercado y cualquier activo  $i$ . Si se representa la curva de la media-varianza, esta debe encontrarse bajo la línea del mercado de valores, incluyendo un punto que coincide con la cartera  $r_M$ . Resumiendo, la línea del mercado de valores tiene que ser tangente en la cartera  $r_M$  a una curva descrita por la representación de la media-varianza de una cartera formada por  $r_M$  y un activo cualquiera  $i$ .

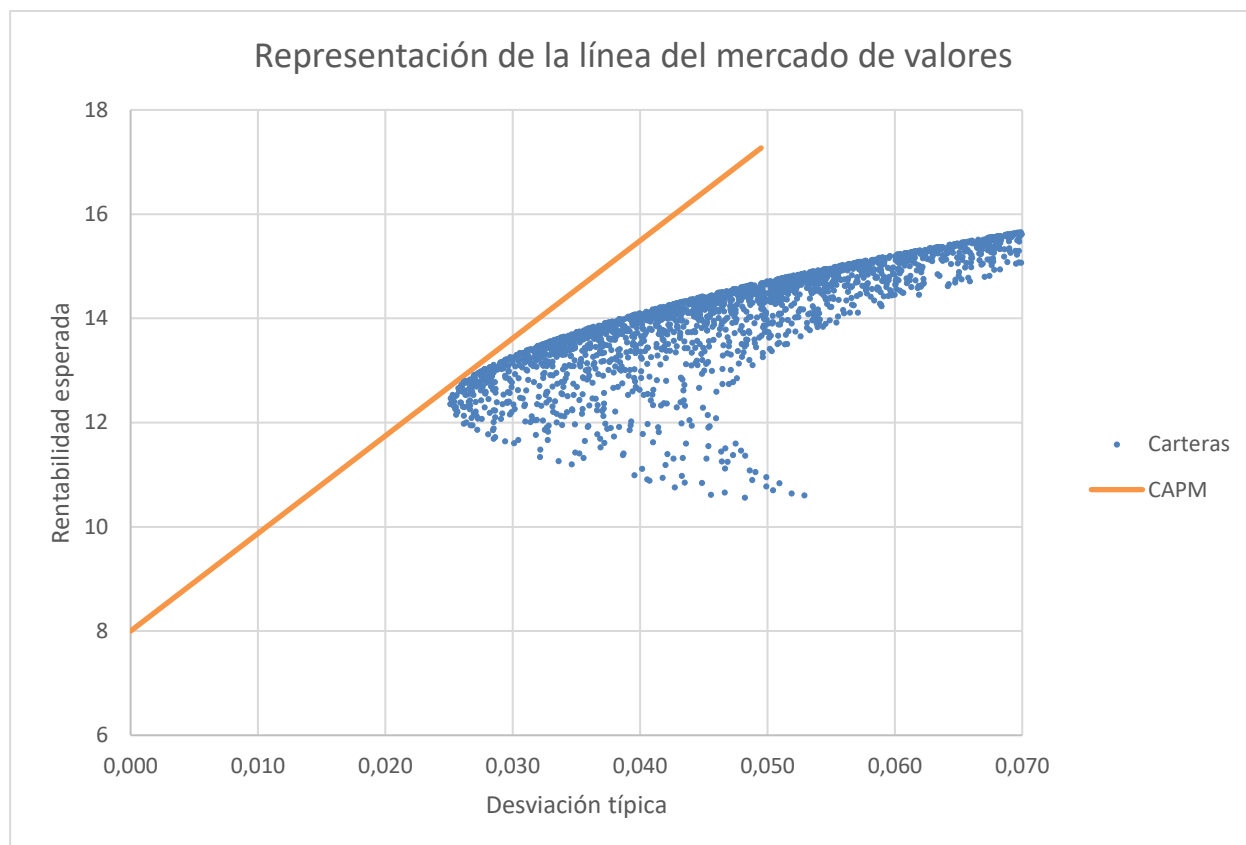


Ilustración 3

En el modelo de valoración de activos financieros CAPM se define un nuevo concepto, la *beta* de un activo. La beta de un activo indica cómo reacciona este activo ante una fluctuación en el mercado. Por lo que se puede estimar la rentabilidad esperada de un activo mediante la siguiente identidad:

$$\mu_M - r_f = \beta_i \cdot (\mu_M - r_f)$$

Donde  $\beta_i$  se define como:

$$\beta_i = \frac{\sigma_{iM}}{\sigma_M^2}$$

Donde  $\sigma_{iM}$  es la covarianza de los retornos del activo  $i$  y de la cartera mercado  $r_M$ . Esta *beta* es un indicador del riesgo de mercado de una acción. Si este indicador es positivo significa que este activo oscila en el mismo sentido que el mercado. Cuando es negativo, oscila al contrario. Mientras mayor sea el valor absoluto de la beta, mayor será la oscilación que sufra el retorno de la acción ante una fluctuación del mercado.

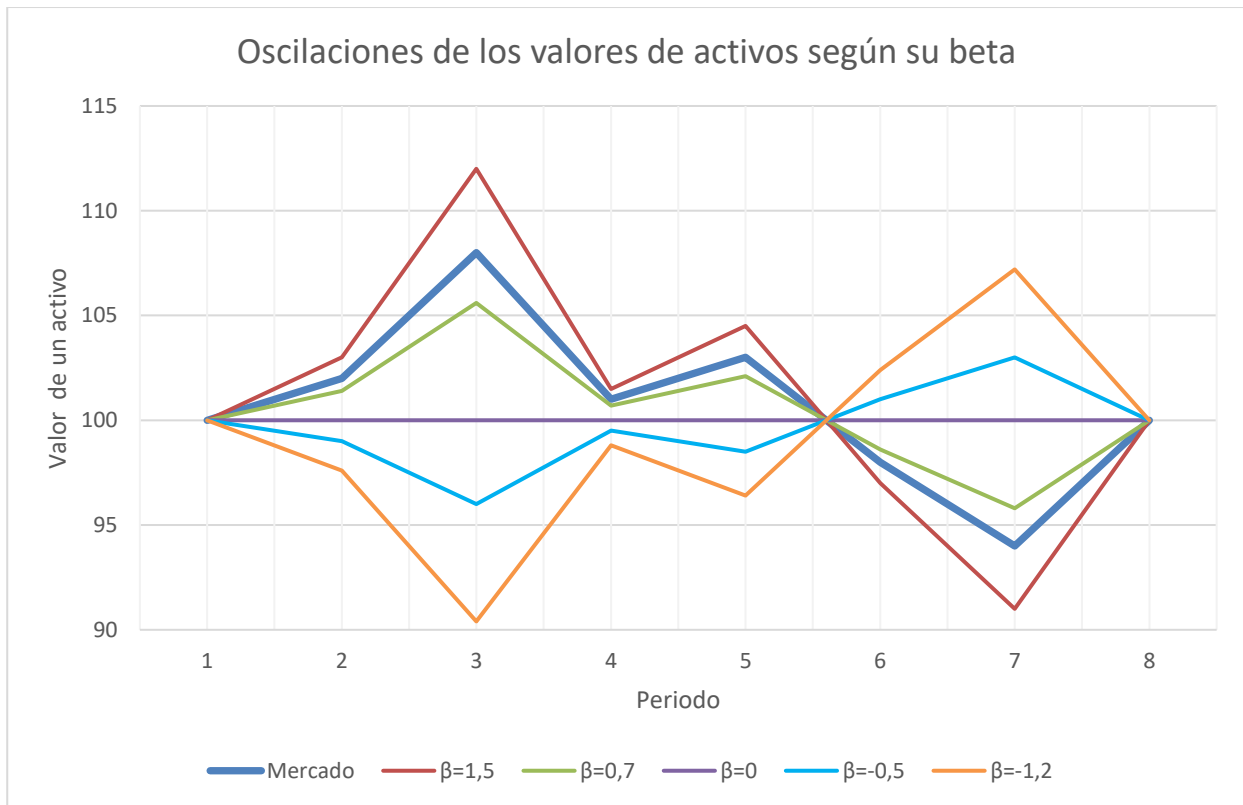


Ilustración 4

Este modelo CAPM se puede utilizar para poner precio a un activo. Se puede hacer usando el valor de compra  $P$  del activo y la variable aleatoria que modela el precio de venta  $Q$  del activo. Al ser una variable aleatoria se puede estimar como la media  $\mu_Q$ . La tasa de retorno de este activo se puede obtener mediante la siguiente identidad:

$$r = \frac{Q - P}{P}$$

De igual manera, se puede estimar la tasa de retorno esperada con la expresión:

$$\mu_r = \frac{\mu_Q - P}{P}$$

Haciendo uso de esta fórmula y la ecuación resultante del CAPM, se consigue la siguiente identidad que es la considerada *fórmula de fijación de precios de CAPM*:

$$\frac{\mu_Q - P}{P} = \mu_M = r_f + \beta_i \cdot (\mu_M - r_f) \Rightarrow P = \frac{\mu_Q}{1 + r_f + \beta_i \cdot (\mu_M - r_f)}$$

Donde  $\beta_i$  es la beta del activo  $i$  que por definición queda demostrado que se puede hallar de mediante:

$$\beta_i = \frac{\sigma_{iM}}{\sigma_M^2} = \frac{cov(r, r_M)}{\sigma_M^2} = \frac{cov\left(\frac{Q - P}{P}, r_M\right)}{\sigma_M^2} = \frac{cov(Q, r_M)}{P \cdot \sigma_M^2}$$

Sustituyendo esta forma de calcular la beta de un activo en la fórmula de fijación de precios se obtiene la *fórmula del precio equivalente de certeza*. Esta fórmula nos proporciona la cantidad de dinero garantizada que se estima de igual valor que un activo influenciado por el riesgo. Esta fórmula tiene la siguiente forma:

$$P = \frac{1}{1 + r_f} \cdot \left( \mu_Q - \frac{cov(Q, r_M) \cdot (\mu_M - r_f)}{\sigma_M^2} \right)$$

El segundo factor del producto de la ecuación recibe el nombre de equivalente cierto. El valor de este equivalente cierto es la cantidad de dinero garantizada que se estima de igual valor que un activo influenciado por el riesgo. Por lo que, dependiendo de la percepción del riesgo que tenga el inversor decidirá invertir en ese activo o no.

Esta ecuación demuestra que el valor de compra de un activo  $P$  depende linealmente de el valor de venta  $Q$ . Por lo que al ser el valor de venta una variable, el valor de compra  $P$  es una función de dicha variable.



# 6 INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN PYTHON

Para llevar a cabo la implementación de este modelo de Markowitz, se hará uso del lenguaje Python de programación. El lenguaje Python es un lenguaje de programación de propósito general muy poderoso y flexible, a la vez que sencillo y fácil de aprender. Es un lenguaje de alto nivel interpretado, dinámico y multiplataforma. Este lenguaje permite procesar cualquier tipo de dato fácilmente, lo cual aporta al usuario mayor versatilidad. Además, se trata de un software de código abierto, por lo que su uso es libre y gratuito. Se puede obtener a través de la referencia [4] que es en la página oficial de Python. Todas estas características hacen de Python un lenguaje muy utilizado entre los programadores, de aquí se explica su gran popularidad. [3]

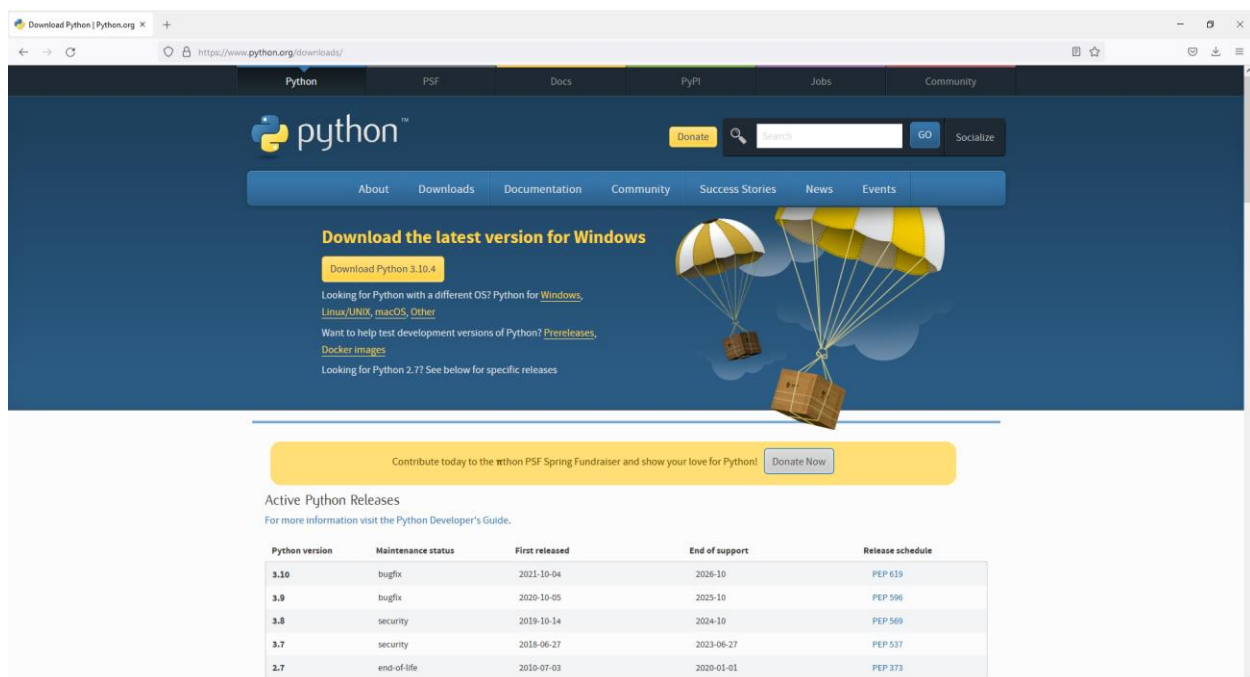


Ilustración 5

Retomando las características de Python, se trata de un lenguaje interpretado. Esto es que hace uso de un intérprete, lo cual genera una ventaja porque los compiladores no pueden traducir y ejecutar a medida, sino que los segundos tienen que compilar el código completo. Otra ventaja sería que el intérprete sí es capaz de guardar datos en memoria mientras que el compilador, una vez se ha ejecutado el programa, elimina los datos de la memoria una vez se termina la ejecución. Por otra parte, es un lenguaje con un fuerte tipado dinámico. Esto significa que las variables pueden contener distintos tipos de datos y que este tipo no cambia de forma repentina, sino que el usuario tiene que indicar al intérprete que quiere cambiar de un tipo a otro.

También se trata de un lenguaje multiparadigma, ya que permite distintos paradigmas de programación como por ejemplo la programación orientada a objeto, estructurada, imperativa y funcional, esta última en menor medida. Gracias a esto se considera que Python es un lenguaje muy versátil y por eso es de los más utilizados en la actualidad.

Estas características son consecuencia de la una serie de principios recogidos en *The Zen of Python* (*El Zen de Python* en español), los cuales se pueden encontrar a través de la web oficial de Python en la referencia [5]. Estos principios son:

- Beautiful is better than ugly (Lo bonito es mejor que lo feo).
- Explicit is better than implicit (Explícito es mejor que implícito).
- Simple is better than complex (Simple es mejor que complejo).
- Complex is better than complicated (Complejo es mejor que complicado).
- Flat is better than nested (Plano es mejor que anidado).
- Sparse is better than dense (Disperso es mejor que denso).
- Readability counts (La legibilidad cuenta).
- Special cases aren't special enough to break the rules (Los casos especiales no lo son lo suficiente como para saltarse las normas).
- Although practicality beats purity (Aunque la practicidad le gana a la pureza).
- Errors should never pass silently (Los errores no deben pasar nunca silenciosamente).
- Unless explicitly silenced (A menos que sean silenciado explícitamente).
- In the face of ambiguity, refuse the temptation to guess (Ante la ambigüedad, rehusa la tentación de adivinar).
- There should be one-- and preferably only one --obvious way to do it. (Debe de haber una – y preferiblemente sólo una única – forma obvia de hacerlo)
- Although that way may not be obvious at first unless you're Dutch (Aunque la forma no será tan obvia al principio si no eres holandés).
- Now is better than never (Ahora es mejor que nunca).
- Although never is often better than \*right\* now (Aunque nunca es con frecuencia mejor que ahora mismo).
- If the implementation is hard to explain, it's a bad idea (Si la implementación es difícil de explicar, es una mala idea).
- If the implementation is easy to explain, it may be a good idea (Si la implementación es fácil de explicar, puede ser buena idea).
- Namespaces are one honking great idea -- let's do more of those! (Los espacios de nombre son una gran idea -- ¡hagamos más cosas de esas!)

El lenguaje de programación Python, como se ha mencionado anteriormente, se trata de un lenguaje interpretado. Esto significa que las líneas que se escriban se ejecutan directamente, no requiere de una compilación previa para pasar las instrucciones del programa a instrucciones en lenguaje para que la máquina las entienda. Estas instrucciones se pueden pasar al intérprete de distintas formas. La primera de ellas es a través de la ventana de comandos o *Command Prompt* que presenta este aspecto:



Ilustración 6

Una vez se ha iniciado Python se puede escribir directamente código desde la consola. Una forma de introducir varias líneas de instrucciones es escribiendo en la ventana de comandos Python seguido por un espacio y el nombre de un archivo con extensión `.py` a continuación. Estos archivos se pueden crear a partir de un archivo de extensión `.txt`, seleccionando la herramienta “Guardar como” y añadiendo “.py” al final del nombre del documento. De esta forma se pueden ejecutar muchas líneas de código de una sola vez.

Sin embargo, esta forma de trabajar puede resultar difícil. La mayoría de los programadores eligen utilizar editores de texto o Entornos de Desarrollo Integrado (*Integrated Development Environment* o *IDE*). Estos presentan una interfaz más cómoda para el usuario. Pueden incluir ciertas funcionalidades como la predicción de la escritura, cierre de paréntesis o llaves automático, resalto de la escritura mediante colores, personalización de la interfaz, etc. Muchos de estos IDEs y editores de texto se pueden obtener de manera gratuita.

Estos programas permiten generar archivos con extensión `.py`, que se pueden utilizar desde el IDE o desde la ventana de comandos. Los editores de texto no tienen integrado ninguna versión del intérprete de Python. Al igual que Python, estas herramientas suelen ser multiplataforma. La mayoría son compatibles con Windows, Linux y Mac OS entre otros.

## 6.1 Variables en Python

En primer lugar, la principal diferencia entre Python y otros muchos lenguajes de programación es que Python no requiere que el usuario declare la variable. Para declarar variables en otros lenguajes como C, es necesario escribir un comando con el tipo de la variable seguido del nombre que se le quiera dar a ésta. De esta manera, el sistema reserva espacio en la memoria para guardar el valor que se le quiera asignar a esa variable. Usando Python, no es necesario declarar las variables. La variable se crea directamente en el momento en el que se le asigna un valor. La variable creada tendrá el tipo de dato que más se ajuste a el valor que se asigne en el momento de su creación. También permite Python una forma de especificar el tipo de variable que se quiere crear. Esto se hace escribiendo el nombre de la variable, seguido de el signo de igual junto a la abreviación del tipo con el valor que se desea guardar en la variable entre paréntesis.

Los valores que se asignan a las variables pueden ser numéricos o una cadena de caracteres. En las variables se pueden guardar uno o varios valores. En función de qué valor se guarde y de cuántos, Python presenta distintos tipos de variables. A continuación, se describen los principales tipos que se encuentran en las referencias [6] y [7] indicando a su vez algunos ejemplos utilizando la herramienta que se indica en la referencia [8].

### 6.1.1 Variables de texto

Las variables que guardan texto son del tipo *string* (*str*). En este tipo de variable se puede guardar una cadena de caracteres. Las cadenas de caracteres pueden incluir números, espacios y otros signos de puntuación. Para

indicar al intérprete que se quiere crear una variable de tipo *string* se colocan comillas a ambos lados del valor que se quiere guardar en la variable.

```
# nombre_variable = tipo_variable ( valor )
# nombre_variable = valor

x = str("Hola Mundo")
y = "Hello World"
```

Ilustración 7

En este caso, se han utilizado las dos formas de crear una variable *string*.

### 6.1.2 Variables numéricas

El otro tipo de variable que tiene Python según su valor es el numérico. Dentro de las variables numéricas, existen 3 tipos distintos. El tipo entero (*int*), flotante (*float*), complejo (*complex*) y booleano (*bool*). En las variables de tipo entero, se puede almacenar cualquier número entero. En las variables tipo *float* se pueden guardar números que incluyan decimales. En Python se utiliza el punto para separar la parte entera de la parte decimal. Además, Python también permite crear variables donde se guardan números complejos. Para ello, solo hay que incluir la letra *j* a continuación de la parte que se quiera designar como imaginaria del número complejo que se quiera asignar a la variable. Por último, Python soporta las variables booleanas, que son aquellas que solo pueden tomar valor verdadero, cuando es distinto de 0 o falso en caso contrario.

### 6.1.3 Variables de tipo secuencia

Python incluye varios tipos de variable en el que se pueden almacenar varios valores a la vez. En este apartado se tratarán la *lista* y la *tupla*. El tipo *string* también puede recibir un trato bastante similar al tipo secuencia. Sin embargo, al ser el tipo *string* una cadena de caracteres, no se tratará en este apartado.

En primer lugar, Python introduce las listas. Se trata de un tipo de variable muy versátil. Para definir una lista, se escribe el nombre de la variable, seguido del signo de igualdad y a continuación unos corchetes. De esta forma se crea una lista vacía. Para incluir otros datos dentro de la lista al crearla, solo hay que escribir los valores que se deseen dentro de los corchetes, de la misma forma que si se estuviera creando la variable en la que se guarda dicho valor. A continuación, se muestra un ejemplo en el que se crea una lista con un número entero, un flotante, una cadena de caracteres y un booleano y se le pide al intérprete que muestre lo que contiene la lista por pantalla:

```
# nombre_lista = [ valor1 , valor2 , ... ]

lista = [ 13 , 3.78 , "Hello World!", bool( 0 ) ]

# print es una función para mostrar su argumento por pantalla
# print ( variable ) o print ( valor )
print(lista)
```

[13, 3.78, 'Hello World!', False]

Ilustración 8

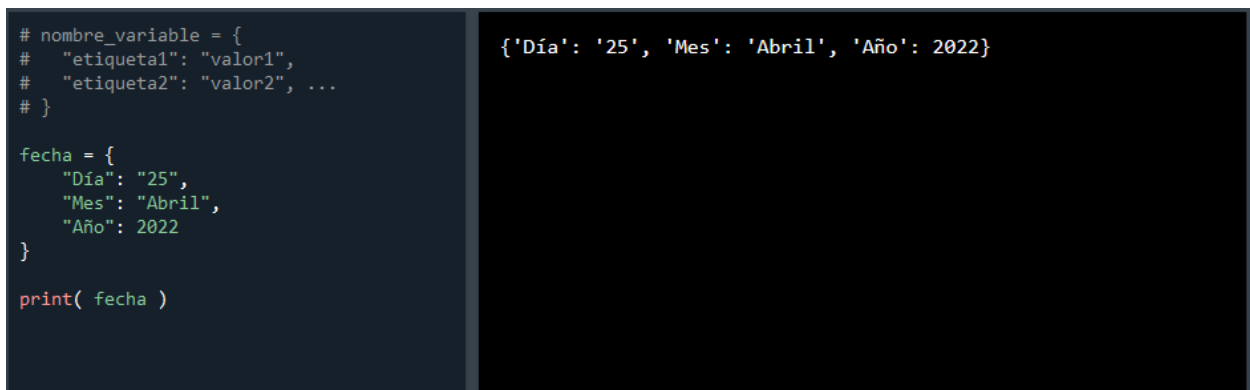


Las listas presentan un carácter heterogéneo y mutable. Esto significa que puede estar formada por varios elementos de distintos tipos, incluyendo en estos tipos la lista, y que todos sus elementos son modificables una vez se ha creado la lista. Se puede acceder a las componentes de una lista usando el nombre de la variable seguido del número del hueco que ocupa dentro de la lista entre corchetes. La primera componente tendrá el índice 0 y la última será el tamaño de la lista menos 1.

En segundo lugar, Python proporciona la posibilidad de crear *tuplas*. Las tuplas son listas pero inmutables, es decir, una vez se ha creado una tupla, no se pueden modificar sus componentes. De esta manera, para crear una tupla, se procede de la misma manera que para una lista, pero usando, en lugar de los corchetes, paréntesis a ambos lados de los valores que se quieran incluir en la tupla.

#### 6.1.4 Variables de tipo mapeo

En el lenguaje Python se incluyen los diccionarios. Los diccionarios son variables que recogen cualquier tipo de valor. Además, estos valores se guardan junto a una etiqueta que recibe el nombre de *key*. Este tipo de variable es similar a las estructuras de datos del lenguaje C. Para crear este tipo de variable, se hace introduciendo las etiquetas y los valores entre unas llaves. Se muestra a continuación un ejemplo:



```
# nombre_variable = {
#   "etiqueta1": "valor1",
#   "etiqueta2": "valor2", ...
# }

fecha = {
    "Día": "25",
    "Mes": "Abril",
    "Año": 2022
}

print( fecha )
```

```
{'Día': '25', 'Mes': 'Abril', 'Año': 2022}
```

Ilustración 9

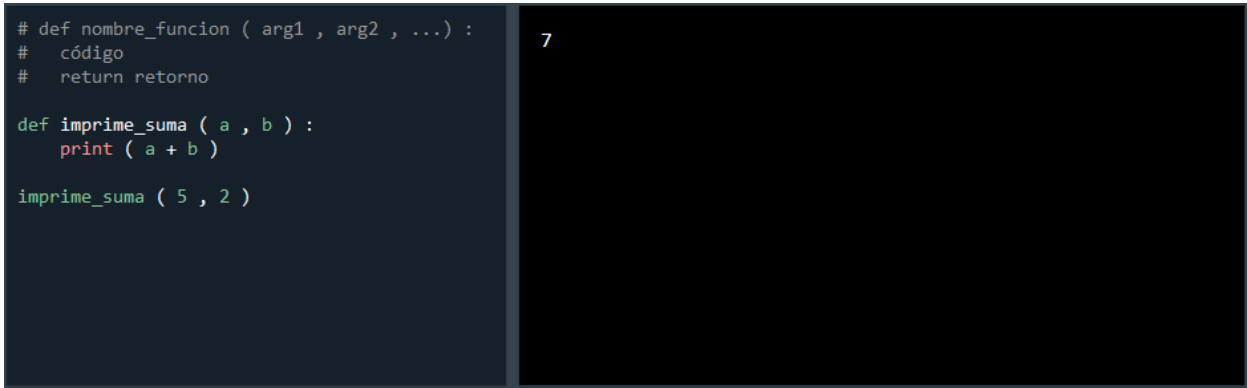
Las características de los diccionarios son:

- Mantienen el orden en el que las etiquetas son escritas en el código.
- Son mutables.
- Son anidables, es decir, un diccionario puede contener a otro.

Esta variable es muy útil puesto que se le puede poner cualquier etiqueta, por lo que es más manejable que los índices de los vectores de C o las listas y las tuplas de Python.

## 6.2 Funciones en Python

Las funciones en Python se definen con el comando *def*. Para definir una función, no es necesario hacerlo al principio del programa, se puede definir y llamar a esa función en la siguiente línea de código. La forma de crear una función será escribir el comando *def*, seguido del nombre que se le desee dar a la función junto con tantos argumentos como se le quieran introducir separados por comas y finalizando con dos puntos. Las sentencias que se asignen a la función deben estar indentadas, ya que el intérprete no considera parte de la función lo que no esté indentado de igual manera. Si se desea que la función devuelva un valor, se utiliza el comando *return*. Para utilizar una función, se escribe el nombre de esta seguido de tantos valores como argumentos se hayan escrito en el momento de la definición. Este resultado se puede guardar en una variable, aplicarlo en otra expresión matemática, imprimirlo por pantalla, etc.



```
# def nombre_funcion ( arg1 , arg2 , ... ) :
#     código
#     return retorno

def imprime_suma ( a , b ) :
    print ( a + b )

imprime_suma ( 5 , 2 )
```

7

Ilustración 10

Por otra parte, Python ofrece la posibilidad de crear funciones *lambda*. Este tipo de funciones son funciones con un código pequeño. En la referencia [9] se definen estas funciones como: " *Python lambdas are only a shorthand notation if you're too lazy to define a function*". Lo que se puede traducir como que las funciones lambda de Python son solo una notación abreviada si eres demasiado vago para definir una función. Estas funciones pueden tener tantos argumentos como se quiera pero solo puede tener una expresión. Estas funciones también son llamadas anónimas. Para definir una función lambda, se escribe el nombre de la función con el signo de igual y del comando *lambda* acompañado por sus argumentos separados por comas. En la misma línea, se escriben dos puntos, seguidos por la expresión que se le quiera asignar a la función.



```
# nombre_función = lambda arg1 , arg2 , ... : expresion

cuadrado = lambda a : a*a

print ( cuadrado ( 7 ) )
```

49

Ilustración 11

### 6.3 Clases en Python

La programación orientada a objetos que soporta el lenguaje Python permite al usuario crear *clases*. El objetivo de estas clases es organizar el código. Las clases son una forma de empaquetar datos y funcionalidades juntos. Una clase genera un nuevo tipo de objeto. Estas clases en Python mantienen todas las características propias de la programación orientada a objetos, como son: la herencia, los objetos, los atributos y los métodos según la referencia [10].

Para crear clases en Python se utiliza el comando *class* seguido por el nombre que se le quiera dar a la clase. A continuación, el código que se quiera incluir dentro de la clase irá indentado. Para asignar un método a la clase, se escribirá una función dentro de la clase. También las clases pueden tener atributos. Los atributos son características que tiene la clase y estas se guardan como variables. Por último, Python permite crear clases dentro de otras clases, pudiendo invocar las clases derivadas a los métodos de la clase base.

### 6.4 Librerías en Python

Una de las principales características de Python es que tiene una gran comunidad que es participe en el desarrollo del software. Esto significa que es muy fácil generar código y que este pueda ser utilizado por

cualquier usuario. Cuando un usuario genera unas líneas de comandos que son de gran utilidad, incluyendo variables, funciones, etc, se guarda todo este en una librería.

Los tipos de librerías se pueden clasificar según la referencia [11] en:

- Deep learning
- Machine learning
- Cálculo numérico
- Visualización
- Inteligencia artificial explicable
- Procesamiento del lenguaje natural

Para utilizar estas librerías, lo primero que se hace es instalarlas desde la ventana de comandos del sistema. Una vez se ha accedido a la ventana de comandos, se escribe la siguiente sentencia, donde *Somepackage* se sustituye por el nombre de la librería que se quiera instalar según la referencia [12]: `python -m pip install SomePackage`. Acto seguido, se ejecutará la descarga e instalación del modulo.

Cuando se instala un modulo en Python, se pueden incluir librerías en un archivo Python. Para ello, se utiliza la palabra reservada *import*. Este comando se escribe seguido por el nombre de la librería que se quiera utilizar. Para acceder a las funciones de la librería hay que utilizar el nombre de la librería y el nombre de la función que se quiera utilizar con un punto entre ambos. Por ejemplo, se muestra cómo utilizar una función de una librería y su salida:

```
# import nombre_librería
import numpy
identidad = numpy.identity ( 3 )
print ( identidad )
```

```
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]
```

Ilustración 12



# 7 IMPLEMENTACIÓN EN PYTHON DE LA TEORÍA DE LA CARTERA MEDIA-VARIANZA

---

## 7.1 Introducción

Hasta aquí se han ido introduciendo todos los conceptos necesarios para conocer el modelo matemático que se haya detrás de las inversiones en el mercado de valores. En el presente capítulo, se va realiza una introducción a una implementación de la Teoría de la cartera media-varianza. Esta implementación ha sido desarrollada por Ivan Lysenko y fue presentada en enero de 2021 con objetivos meramente educativos o de entretenimiento [13].

En la primera parte del artículo, se hace una introducción al modelo de Markowitz en base a la entrada en *Wikipedia*. No se profundizará en las bases del modelo de Markowitz porque ya se ha hecho a lo largo de los capítulos 3, 4 y 5. Sin embargo, se hará mención a ciertas consideraciones que se harán de cara a facilitar el entendimiento del código que se mostrará posteriormente.

En primer lugar, es importante mencionar que en la implementación, los retornos no tendrán la misma definición que en la introducción teórica que se hace en el capítulo 3, donde los retornos se definían:

$$R = \frac{x_1}{x_0}$$

Donde:

- R: retorno del activo
- $x_i$ : valor del activo en el instante  $i$

En el artículo, este concepto se expresa mediante el logaritmo natural de la razón anterior. Esta consideración, consigue que sea más fácil lidiar con los retornos y además reporta resultados positivos. Entonces se define de la siguiente manera:

$$R = \ln\left(\frac{x_1}{x_0}\right)$$

Gracias a esta suposición, se puede aproximar el retorno esperado usando los datos históricos de retornos de un activo. Para ello, se calcula la esperanza matemática de los retornos diarios mediante la media de estos, suponiendo que el retorno diario de un activo es una variable aleatoria con una media y una varianza determinadas. De esta forma, para obtener una estimación del retorno anual de un activo, solo habría que calcular el retorno diario esperado por el número de días en los que se puede comprar y vender en la bolsa.

Por otra parte, se introduce la razón de Sharpe. La razón de Sharpe es una medida de la optimalidad de un activo. Esta razón compara los retornos de un activo con los retornos de los activos libres de riesgo. También incluye la desviación típica del retorno de dicho activo, por lo que incluye un factor que condiciona su valor según la volatilidad del activo. La razón de Sharpe viene dada por la siguiente expresión:

$$S = \frac{E \cdot (R - R_f)}{\sigma}$$

Donde:

- $S$ : razón de Sharpe
- $R$ : retorno del activo
- $R_f$ : retorno del activo libre de riesgo
- $\sigma$ : desviación típica del retorno del activo

Así, a mayor valor de esta razón, mejor será un activo. Cuando un par de activos tengan misma rentabilidad esperada, se puede hacer uso del valor de la razón de Sharpe de cada uno de los activos y de esta forma comprobar cual tiene mejor relación rentabilidad-riesgo. De igual manera, se puede hacer para dos activos con misma desviación típica y el que mayor razón de Sharpe tenga será el mejor de entre ambos.

## 7.2 Implementación en Python

### 7.2.1 Librerías y constantes

Para empezar, como suele ser habitual en cuando se crea un archivo a partir de algún lenguaje de programación, se empieza importando las librerías necesarias para llevar a cabo las acciones posteriores. También se crean dos variables con dos valores constantes que serán incluidas dentro del modelo matemático. Las variables son la rentabilidad de las letras del Tesoro y el número de días en los que el mercado de valores se encuentra abierto.

```
import numpy as np
import pandas as pd
from scipy.optimize import minimize
import matplotlib.pyplot as plt
from typing import List, Tuple
from functools import cache # Python 3.9 required
from IPython.display import display
import yfinance as yf

from matplotlib import rcParams
rcParams['figure.figsize'] = 12, 9

TREASURY_BILL_RATE = 2.12 #%, mayo 2022
TRADING_DAYS_PER_YEAR = 250
```

La librería *numpy* es una de las más utilizadas en Python. Se usa para el cálculo numérico y el análisis de datos. Además con esta librería se utilizan *arrays* que tienen una velocidad de procesamiento mucho más rápido que las listas de Python. Por otra parte, más adelante se llamará a este módulo con las letras *np*.

En segundo lugar, la librería *pandas* se utiliza generalmente en los campos de Data Science y Machine Learning, ya que ofrece estructuras muy poderosas y flexibles para facilitar la manipulación y el tratamiento de grandes bloques de datos. Estos bloques de datos son series (arrays) o DataFrame. Las series son unidimensionales, y los DataFrame son bidimensionales, donde cada columna es una serie distinta.

En tercer lugar, se importa la librería *scipy*. Esta librería tiene varios subpaquetes, en concreto, se usará el conjunto de herramientas de optimización *optimize*. En esta librería se encuentran funciones de minimización que incluyen *solvers* para problemas no lineales, programación lineal y algunas otras. Esto se utilizará más

adelante para optimizar la función objetivo del modelo matemático.

En cuarto lugar, se llama a la librería *matplotlib* que es un módulo que permite al usuario crear visualizaciones estáticas, animadas e interactivas en Python. En el caso que concierne a este proyecto, se utilizará la función *pyplot*. La función *pyplot* del módulo *matplotlib* tiene asignado el nombre *plt* que le otorga el usuario. Este nombre podría cambiarse por otro.

En quinto lugar, se necesitarán las funciones *list* y *tuple* del módulo *typing*. Este módulo proporciona soporte en tiempo de ejecución (*runtime*) para anotaciones de tipado.

En sexto lugar, se importa, del módulo *functools*, la función *cache*. Esta función está soportada por intérpretes de Python de versión 3.9 o superior. El módulo *functools* es para funciones de orden superior, es decir, funciones que actúan o retornan otras funciones.

En séptimo lugar, el módulo *IPython* permite la lectura, evaluación e impresión de bucles para el desarrollo interactivo de Python. De este módulo, se importa la función *display*.

En octavo lugar, será necesario utilizar el módulo *yfinance*. Este módulo es una API (Application Programming Interface o Interfaz de Programación de Aplicaciones) que ofrece la posibilidad de volcar datos directamente desde la web de la referencia [14]. Esta web proporcionará los datos sobre el histórico del valor de los activos que se introducirán como variables del modelo de Harry Markowitz.

Por último, se importa la función *rcParams* del módulo *matplotlib*. Ésta se utiliza para definir el tamaño de una figura, en el caso que nos aplica, se fija un tamaño de 12 pulgadas de ancho por 9 de ancho.

Para terminar, en esta primera parte del código se establece que el rendimiento de los bonos del Estado español a 10 años en mayo de 2022 es del 2,12%. Este valor será el valor de la rentabilidad de los activos libres de riesgo. Por otro lado, el número de días en los que el mercado de valores se encuentra abierto para comprar o vender activos es de 250 días anuales. Esta cifra permite hacer una estimación de la rentabilidad anual de un activo, conociendo previamente la rentabilidad diaria esperada.

## 7.2.2 Clase Asset

A continuación, se muestra la definición de la clase *Asset* (activo):

```
# Needed for type hinting
class Asset:
    pass

def get_log_period_returns(price_history: pd.DataFrame):
    close = price_history['Close'].values
    return np.log(close[1:] / close[:-1]).reshape(-1, 1)

# daily_price_history has to at least have a column, called 'Close'
class Asset:
    def __init__(self, name: str, daily_price_history: pd.DataFrame):
        self.name = name
        self.daily_returns = get_log_period_returns(daily_price_history)
        self.expected_daily_return = np.mean(self.daily_returns)

    @property
    def expected_return(self):
        return TRADING_DAYS_PER_YEAR * self.expected_daily_return

    def __repr__(self):
        return f'<Asset name={self.name}, expected return={self.expected_return}>'
```

```

@staticmethod
@cache
def covariance_matrix(assets: Tuple[Asset]): # tuple for hashing in the cache
    product_expectation = np.zeros((len(assets), len(assets)))
    for i in range(len(assets)):
        for j in range(len(assets)):
            if i == j:
                product_expectation[i][j] = np.mean(assets[i].daily_returns *
assets[j].daily_returns)
            else:
                product_expectation[i][j] = np.mean(assets[i].daily_returns @
assets[j].daily_returns.T)

    product_expectation *= (TRADING_DAYS_PER_YEAR - 1) ** 2

    expected_returns = np.array([asset.expected_return for asset in
assets]).reshape(-1, 1)
    product_of_expectations = expected_returns @ expected_returns.T

    return product_expectation - product_of_expectations

```

A parte de la clase *Asset*, se crea la función `get_log_period_returns` que se utiliza dentro de la clase *Asset* para conseguir el retorno diario del activo de la clase *Asset*. En esta clase, se crean los métodos de esta clase. Particularmente, se hace mención sobre el método estático. Un método estático es un tipo de método. Este tipo de método tiene la particularidad de que no permiten modificar el estado de la clase ni de la instancia. Esto se debe a que no pueden recibir como argumentos de entrada a ninguna de las dos. Consecuentemente, son métodos son utilizados para hacer notar que el método no realiza modificación alguna ni a la clase ni a la instancia y así evitar futuros problemas.

Este método estático se incluye para definir la matriz de covarianza de los activos que pertenezcan a la clase *Asset*. Para ello, utiliza la expresión de la covarianza:

$$\text{cov}(X, Y) = E(X \cdot Y) - E(X) \cdot E(Y)$$

### 7.2.3 Clase *Portfolio*

Después de definir la clase para los activos *Asset*, se crea la clase *Portfolio*. Esta clase estará formada por los activos que compongan la cartera, los retornos esperados de estos activos, la matriz de covarianza de los activos que componen la clase y los pesos asociados a los activos. Se define de la siguiente manera:

```

def random_weights(weight_count):
    weights = np.random.random((weight_count, 1))
    weights /= np.sum(weights)
    return weights.reshape(-1, 1)

class Portfolio:
    def __init__(self, assets: Tuple[Asset]):
        self.assets = assets
        self.asset_expected_returns = np.array([asset.expected_return for asset in
assets]).reshape(-1, 1)

```



```

self.covariance_matrix = Asset.covariance_matrix(assets)
self.weights = random_weights(len(assets))

def unsafe_optimize_with_risk_tolerance(self, risk_tolerance: float):
    res = minimize(
        lambda w: self._variance(w) - risk_tolerance * self._expected_return(w),
        random_weights(self.weights.size),
        constraints=[
            {'type': 'eq', 'fun': lambda w: np.sum(w) - 1.},
        ],
        bounds=[(0., 1.) for i in range(self.weights.size)]
    )

    assert res.success, f'Optimization failed: {res.message}'
    self.weights = res.x.reshape(-1, 1)

def optimize_with_risk_tolerance(self, risk_tolerance: float):
    assert risk_tolerance >= 0.
    return self.unsafe_optimize_with_risk_tolerance(risk_tolerance)

def optimize_with_expected_return(self, expected_portfolio_return: float):
    res = minimize(
        lambda w: self._variance(w),
        random_weights(self.weights.size),
        constraints=[
            {'type': 'eq', 'fun': lambda w: np.sum(w) - 1.},
            {'type': 'eq', 'fun': lambda w: self._expected_return(w) -
expected_portfolio_return},
        ],
        bounds=[(0., 1.) for i in range(self.weights.size)]
    )

    assert res.success, f'Optimization failed: {res.message}'
    self.weights = res.x.reshape(-1, 1)

def optimize_sharpe_ratio(self):
    # Maximize Sharpe ratio = minimize minus Sharpe ratio
    res = minimize(
        lambda w: -(self._expected_return(w) - TREASURY_BILL_RATE / 100) /
np.sqrt(self._variance(w)),
        random_weights(self.weights.size),
        constraints=[
            {'type': 'eq', 'fun': lambda w: np.sum(w) - 1.},
        ],
        bounds=[(0., 1.) for i in range(self.weights.size)]
    )

    assert res.success, f'Optimization failed: {res.message}'
    self.weights = res.x.reshape(-1, 1)

```

```

def _expected_return(self, w):
    return (self.asset_expected_returns.T @ w.reshape(-1, 1))[0][0]

def _variance(self, w):
    return (w.reshape(-1, 1).T @ self.covariance_matrix @ w.reshape(-1,
1))[0][0]

@property
def expected_return(self):
    return self._expected_return(self.weights)

@property
def variance(self):
    return self._variance(self.weights)

def __repr__(self):
    return f'<Portfolio assets={ [asset.name for asset in self.assets] }, expected
return={self.expected_return}, variance={self.variance}>'

```

En esta clase, también se crean 3 métodos de optimización. Estos métodos buscarán la mejor cartera posible según un parámetro. El primero de ellos fija el riesgo máximo que puede contener la cartera. El segundo, obliga a la cartera a tener una rentabilidad, por lo que elegirá la relación de pesos que minimice el riesgo de la cartera. El último método, es una función para obtener una cartera que minimice la razón de Sharpe que se introdujo en el anterior apartado. De esta manera se obtendrán 3 carteras optimizadas según distintos criterios. Esto proporcionará mayor cantidad de datos para un análisis posterior.

### 7.2.4 Función auxiliar *yfinance*

Para poder utilizar los datos de la librería *yfinance*, se define una función. Esta función recibe como argumento de entrada una lista de activos que compondrán la cartera. La función devuelve una lista (DataFrame) con el histórico de la rentabilidad de cada activo que componen la cartera. La función se define como sigue:

```

def yf_retrieve_data(tickers: List[str]):
    dataframes = []

    for ticker_name in tickers:
        ticker = yf.Ticker(ticker_name)
        history = ticker.history(period='1y')

        if history.isnull().any(axis=1).iloc[0]: # the first row can have NaNs
            history = history.iloc[1:]

        assert not history.isnull().any(axis=None), f'history has NaNs in
{ticker_name}'
        dataframes.append(history)

    return dataframes

```

El comentario que menciona a los “NaNs” hace referencia a que en la primera fila pueden aparecer datos que no son numéricos (*Not a Number*).

### 7.2.5 Representación gráfica

Este código sirve para representar carteras tanto aleatorias como optimizadas. Las carteras aleatorias se utilizan para poder localizar dónde está la frontera eficiente del problema de Markowitz. Por otra parte, también se representan 4 carteras formadas según distintas formas de optimización. En la referencia del artículo, el autor selecciona una cartera formada por Apple Inc., Amazon.com Inc., Google LLC, Berkshire Hathaway Inc., Johnson & Johnson y JPMorgan Chase & Co. También se incluyen varias sentencias para representar la frontera eficiente.

```
stocks = ['AAPL', 'AMZN', 'GOOG', 'BRK-B', 'JNJ', 'JPM']
daily_dataframes = yf_retrieve_data(stocks)
assets = tuple([Asset(name, daily_df) for name, daily_df in zip(stocks,
daily_dataframes)])

X = []
y = []

# Drawing random portfolios
for i in range(3000):
    portfolio = Portfolio(assets)
    X.append(np.sqrt(portfolio.variance))
    y.append(portfolio.expected_return)

plt.scatter(X, y, label='Random portfolios')

# Drawing the efficient frontier
X = []
y = []
for rt in np.linspace(-300, 200, 1000):
    portfolio.unsafe_optimize_with_risk_tolerance(rt)
    X.append(np.sqrt(portfolio.variance))
    y.append(portfolio.expected_return)

plt.plot(X, y, 'k', linewidth=3, label='Efficient frontier')

# Drawing optimized portfolios
portfolio.optimize_with_risk_tolerance(0)
plt.plot(np.sqrt(portfolio.variance), portfolio.expected_return, 'm+',
markeredgewidth=5, markersize=20, label='optimize_with_risk_tolerance(0)')

portfolio.optimize_with_risk_tolerance(20)
plt.plot(np.sqrt(portfolio.variance), portfolio.expected_return, 'r+',
markeredgewidth=5, markersize=20, label='optimize_with_risk_tolerance(20)')

portfolio.optimize_with_expected_return(0.25)
plt.plot(np.sqrt(portfolio.variance), portfolio.expected_return, 'g+',
markeredgewidth=5, markersize=20, label='optimize_with_expected_return(0.25)')

portfolio.optimize_sharpe_ratio()
```

```
plt.plot(np.sqrt(portfolio.variance), portfolio.expected_return, 'y+',
markedgedwidth=5, markersize=20, label='optimize_sharpe_ratio()')

plt.xlabel('Portfolio standard deviation')
plt.ylabel('Portfolio expected (logarithmic) return')
plt.legend(loc='lower right')
plt.show()
```

Esta lista con los activos se puede modificar para crear carteras con activos distintos. También se puede cambiar los parámetros de optimización como son la tolerancia del riesgo y la rentabilidad esperada. Además, se puede cambiar desde la función *yfinance* el periodo que se recoge en el histórico de cada activo. De esta forma se puede elegir si tener en cuenta una época que incluya cierto evento histórico o no, como por ejemplo la crisis mundial generada por el impacto del COVID-19.

A continuación se muestra la gráfica que resulta de ejecutar el código anterior:

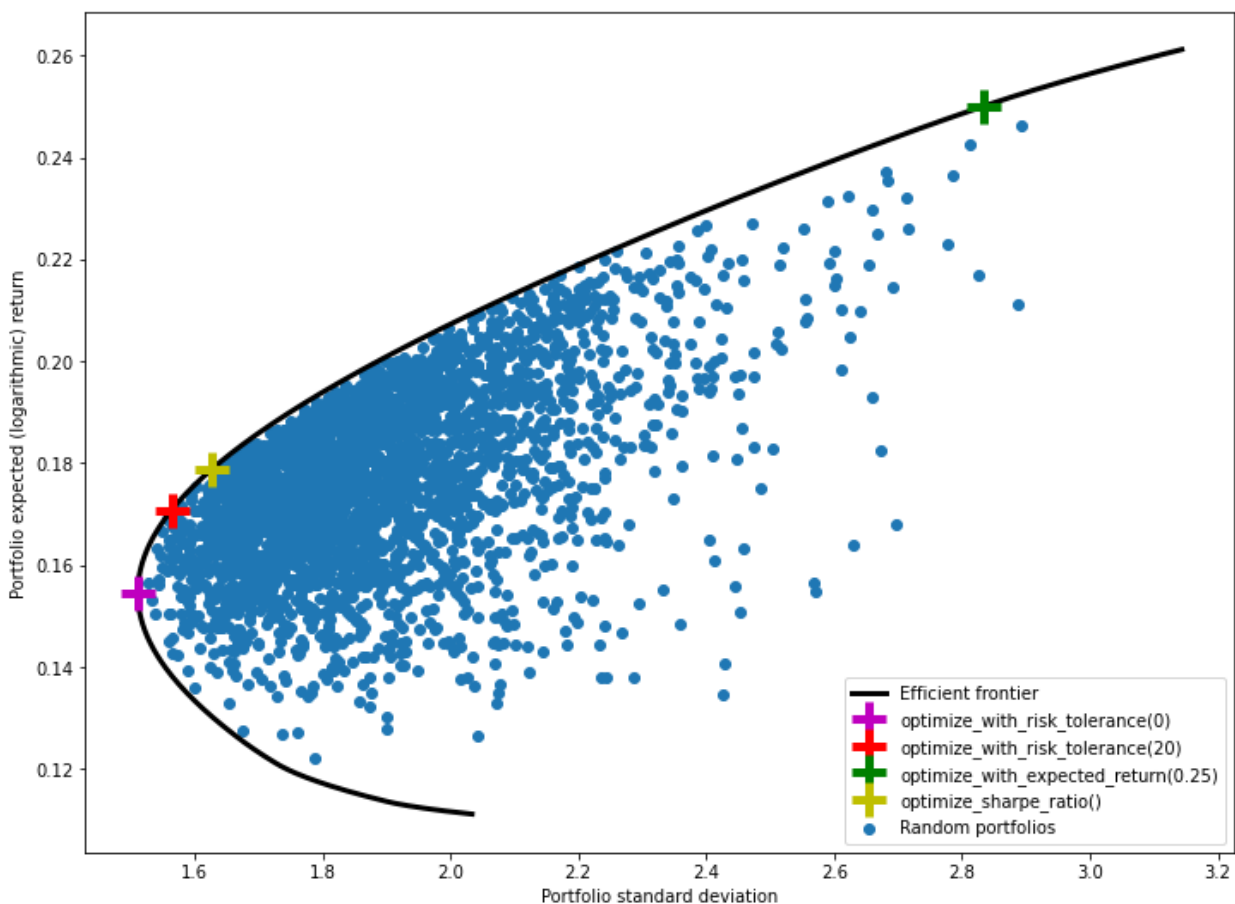


Ilustración 13

### 7.2.6 Obtención de los pesos de cada cartera optimizada

Por último, es necesario para poder conseguir estas carteras en la realidad conocer la relación de los pesos de cada activo. En concreto, este conjunto de comandos muestra por pantalla la proporción de cada activo que habría que adquirir para formar carteras con las características de los métodos de optimización empleados para hallarlas.

Se puede obtener los pesos (con una definida precisión de 5 decimales) de cada activo de la siguiente forma:

```
pd.options.display.float_format = "{:, .5f}".format
```

```

portfolio = Portfolio(assets)
portfolio.optimize_with_risk_tolerance(0)
riskless_weights = portfolio.weights.flatten()

portfolio.optimize_with_risk_tolerance(20)
weights_risk_tolerance = portfolio.weights.flatten()

portfolio.optimize_with_expected_return(0.25)
weights_return = portfolio.weights.flatten()

portfolio.optimize_sharpe_ratio()
weights_sharpe = portfolio.weights.flatten()

display(
    pd.DataFrame(
        list(
            zip(
                [asset.name for asset in portfolio.assets],
                riskless_weights,
                weights_risk_tolerance,
                weights_return,
                weights_sharpe,
            )
        ),
        columns=[
            'asset',
            'optimize_with_risk_tolerance(0)',
            'optimize_with_risk_tolerance(20)',
            'optimize_with_expected_return(0.25)',
            'optimize_sharpe_ratio()',
        ],
    )
)

```

De esta forma, se obtiene las siguientes relaciones de pesos para las carteras optimizadas formadas por los activos de Apple Inc., Amazon.com Inc., Google LLC, Berkshire Hathaway Inc., Johnson & Johnson y JPMorgan Chase & Co. Habiendo optimizado con tolerancias de riesgo 0 y 20, con un retorno esperado del 25% y con la mejor razón de Sharpe:

	asset	optimize_with_risk_tolerance(0)	optimize_with_risk_tolerance(20)	optimize_with_expected_return(0.25)	optimize_sharpe_ratio()
0	AAPL	0.11428	0.16286	0.40720	0.18626
1	AMZN	0.09335	0.14528	0.41457	0.17040
2	GOOG	0.14037	0.15165	0.17666	0.15667
3	BRK-B	0.23204	0.17985	0.00000	0.15426
4	JNJ	0.30786	0.26009	0.00000	0.23798
5	JPM	0.11210	0.10027	0.00156	0.09443

Ilustración 14



# 8 APLICACIÓN DEL MODELO DE MARKOWITZ EN LENGUAJE DE PROGRAMACIÓN PYTHON

---

## 8.1 Modificaciones del código original

Una vez que se ha introducido el código que implementa la teoría de la cartera de la media-varianza, se procede a realizar algunos cambios para facilitar los posteriores resultado y análisis. Estos cambios son:

- Cambio en la selección de activos, tanto cualitativa como cuantitativamente, para obtener distintos resultados y así poder compararlos y analizarlos.
- Cambio en los parámetros de los comandos que representaban las carteras optimizadas para mejorar la visualización, tanto colores como tamaños.
- Cambio en el argumento que se le pasa al método `portfolio.optimize_with_expected_return()` que pasa de ser 0,25 a 0,20 para mejorar el proceso de minimización.
- Cambio en la ubicación del comando `plt.show()` para que el gráfico sea lo último que nos devuelva el programa y así poder leer los demás datos.
- Cambio en la función `display()` que nos muestra por pantalla además de los activos y sus pesos según la cartera que se tenga que formar y se incluye la columna que proporciona el retorno esperado de cada activo añadiendo la línea `[TRADING_DAYS_PER_YEAR * asset.expected_daily_return for asset in portfolio.assets]`, en la línea donde se escriben las filas del DataFrame.

## 8.2 Simulaciones

En este apartado, se mostrarán tablas y gráficas que son el resultado ejecutar el programa escrito en Python que se mostró en el capítulo anterior. Se ha optado por generar distintas carteras sectoriales de España. Las carteras han sido optimizadas según distintos métodos de optimización. Estas carteras optimizadas tienen que estar sobre la frontera eficiente por definición. Las características de los distintos métodos de optimización son:

- Optimización con tolerancia de riesgo 0: este método sirve para hallar la cartera que consigue la solución de mínima varianza al modelo. Esta solución representa la cartera eficiente que menos riesgo conlleva al inversor.
- Optimización con tolerancia de riesgo 20: este método encuentra la cartera con mayor rentabilidad dentro del rango de carteras que desde la mínima varianza hasta las que tengan un 20% más de esa varianza. De esta manera, esta cartera representa a los inversores que no se conforman con la mínima

rentabilidad esperada eficiente, si no que están dispuestos a correr un mayor riesgo con el fin de poder obtener una mayor rentabilidad.

- Optimización con un retorno esperado fijado: utilizando este método se consigue hallar la cartera que nos permita obtener la rentabilidad esperada que fijemos al menor riesgo posible. Con esta forma de optimizar se consigue simular a un inversor que está dispuesto a asumir riesgos mayores si se le estima una rentabilidad esperada que él considere.
- Optimización de la razón de Sharpe: este método se emplea para conseguir una cartera que optimiza la relación entre la diferencia del retorno esperado con la rentabilidad libre de riesgo y la desviación típica como medida del riesgo de la cartera. Este tipo de carteras representa a los inversores que querrían aprovechar el concepto de economía de escala. Esto es que si arriesga un poco, consigue mejorar bastante la rentabilidad esperada. La deseconomía de escala se encontraría si se tuviera que arriesgar mucho para conseguir un leve aumento de la rentabilidad esperada.

De esta forma, se considera que es interesante poder valorar distintas carteras eficientes para poder hacer un análisis de los beneficios y las desventajas que cada una conlleva.

### 8.2.1 Simulación de ejemplo

Esta primera simulación se hace para establecer una referencia externa al mercado español. En este caso, se opta por una cartera compuesta por 6 activos:

- Apple
- Amazon
- Google
- Berkshire Hathaway
- Johnson & Johnson
- JP Morgan

Esta cartera no está formada por empresas de un mismo sector. Sí son empresas que componen el mismo índice, el S&P500. Una vez se ejecuta el código introduciendo los tickers de estas empresas se obtiene la siguiente gráfica:



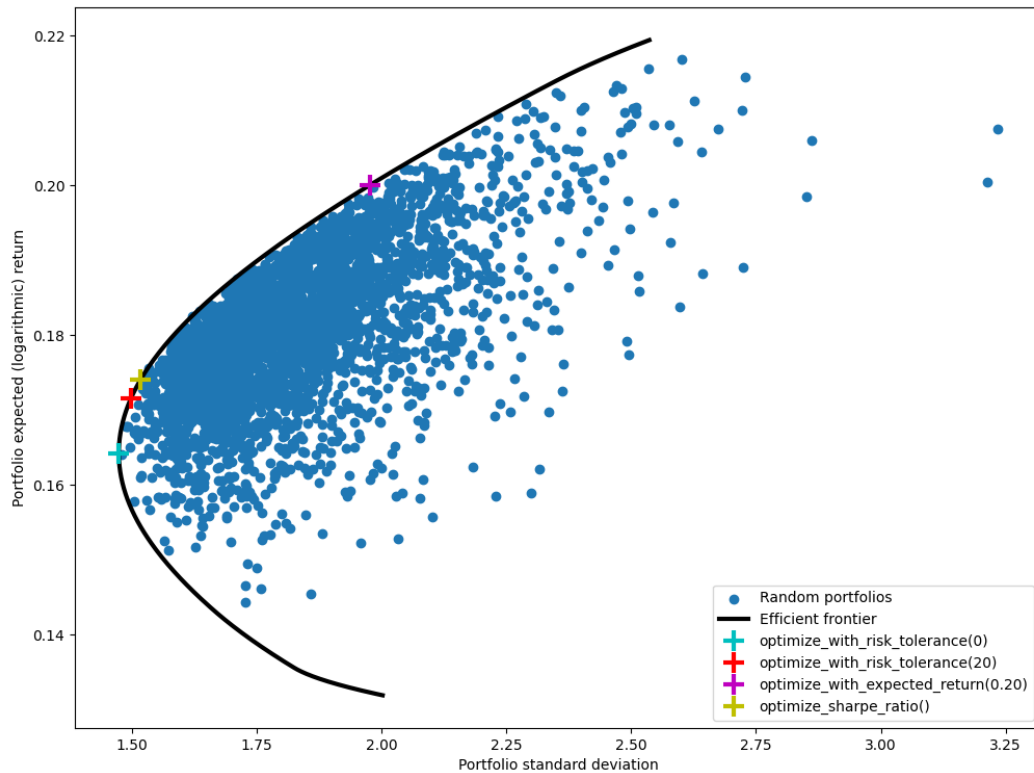


Ilustración 15

Se puede comprobar que los algoritmos de optimización han funcionado observando que el conjunto de carteras aleatorias junto con la frontera eficiente forman una region convexa. Por otro lado, la optimización con tolerancia de riesgo 0 proporciona la solución de mínima varianza para esta cartera. Además, se sabe que son carteras optimizadas porque se encuentran sobre la frontera eficiente. Al pedir una cartera con una rentabilidad esperada del 20%, además de cumplir esta condición, se puede notar que tiene una desviación típica o riesgo considerablemente mayor que las otras carteras optimizadas.

A continuación, se recoge el DataFrame de salida que incluye el retorno esperado de cada active y su correspondiente peso dentro de la cartera optimizada por cada método empleado:

Tabla 8-1: Relación de pesos de las carteras optimizadas con activos de ejemplo

Activo	Ticker	Retorno esperado	Optimización con tolerancia de riesgo 0	Optimización con tolerancia de riesgo 20	Optimización con retorno esperado 20%	Optimización de la razón de Sharpe
Apple	AAPL	0,21166	0,10778	0,13130	0,22178	0,13874
Amazon	AMZN	0,24059	0,09121	0,12324	0,24651	0,13351
Google	GOOG	0,20856	0,13308	0,16015	0,26457	0,16888
Berkshire Hathaway	BRK-B	0,13308	0,25282	0,21645	0,07701	0,20475
Johnson & Johnson	JNJ	0,13102	0,28997	0,24574	0,07501	0,23115
JP Morgan	JPM	0,16067	0,12513	0,12312	0,11512	0,12297

Analizando esta table se pueden advertir diversidad de características de la cartera. En primer lugar, la columna de retorno esperado proporciona una visión sobre qué activos tendrán más pesos que otros según la rentabilidad global de la cartera. Por ejemplo, se puede comprobar que los activos con mayor rentabilidad son aquellas que componen la cartera optimizada para un retorno esperado del 20%. También, se puede deducir que los activos con menor riesgo serán aquellos que tengan mayores pesos dentro de la cartera que sea la solución de mínima varianza, en este caso Berkshire Hathaway y Johnson & Johnson son los activos cuyas rentabilidades menos fluctuaciones presentan a lo largo del tiempo. Por otra parte, comparando los distintos pesos entre Apple, Amazon y Google, se puede comprobar que aparentemente una inversión en Google presenta menor riesgo que las otras dos. Principalmente, hasta para conseguir una cartera con la rentabilidad del 20% es la que mayor proporción representa de esa cartera a pesar de no tener el mayor retorno esperado. También resalta que JP Morgan al ser la que menor peso representa en la cartera optimizada en base a la razón de Sharpe. Es por resto que se puede concluir, en virtud de la definición de esta razón, que tiene la peor relación entre rentabilidad esperada y riesgo.

## 8.2.2 Simulación en el sector de las empresas energéticas y petrolíferas en España

En esta segunda simulación, se han seleccionado 6 activos pertenecientes al sector de las empresas que dedican sus actividades al sector de la energía y el sector del petróleo. Estas empresas se han seleccionado entre las que se encuentran en el Mercado Continuo. Estas son:

- Acciona Energía
- Enagas
- Endesa
- Iberdrola
- Naturgy
- Repsol

Al llevar a cabo la ejecución del archivo Python, se obtiene como salida la siguiente gráfica:

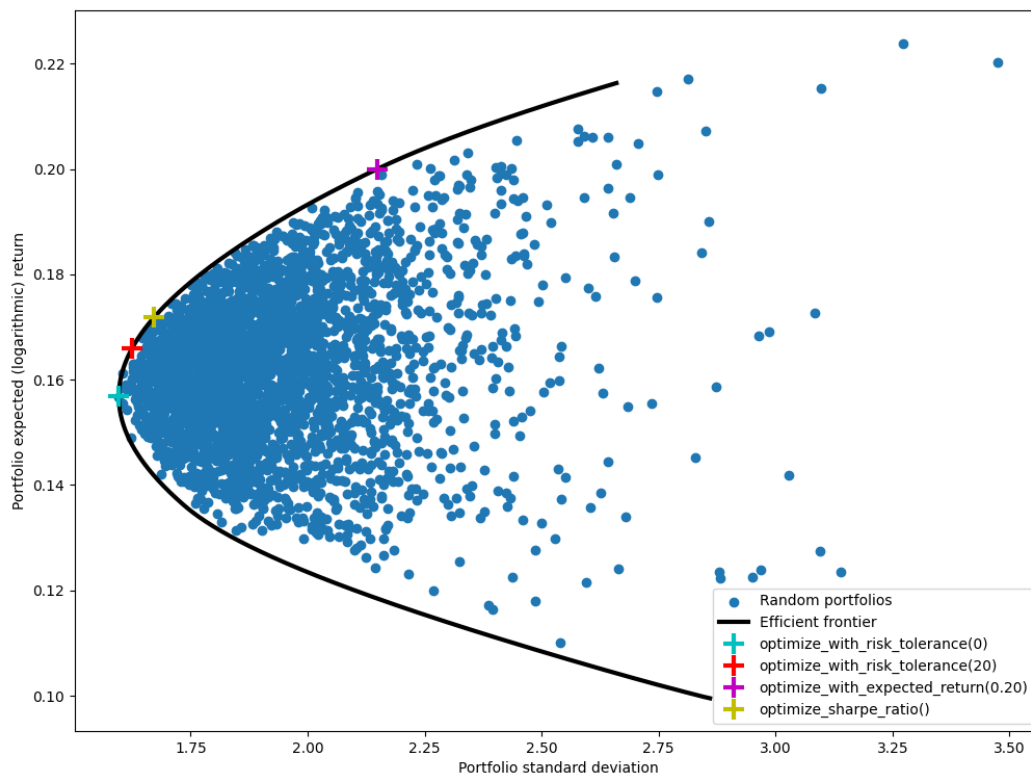


Ilustración 16

Analizando la posición de las carteras formadas y la frontera eficiente se puede afirmar que este conjunto de empresas no permiten mejorar las condiciones de rentabilidad esperada y riesgo de las posibles carteras que se formaron en el apartado anterior. A continuación, se muestra la relación de pesos que corresponde a cada cartera que ha sido optimizada por cada uno de los métodos empleados:

Tabla 8-2: Relación de pesos de las carteras optimizadas con activos del sector de las energéticas y petrolíferas

Activo	Ticker	Retorno esperado	Optimización con tolerancia de riesgo 0	Optimización con tolerancia de riesgo 20	Optimización con retorno esperado 20%	Optimización de la razón de Sharpe
Acciona Energía	ANE.MC	0,26986	0,09563	0,13805	0,30610	0,16513
Enagas	ENG.MC	0,10852	0,22355	0,18123	0	0,15415
Endesa	ELE.MC	0,17071	0,20148	0,21246	0,24774	0,21924
Iberdrola	IBE-MC	0,16444	0,20979	0,21606	0,23181	0,22006
Naturgy	NTGY.MC	0,17256	0,16925	0,17972	0,21436	0,18666
Repsol	REP.MC	0,08607	0,10030	0,07247	0	0,05475

Lo más llamativo de esta tabla es encontrar que los activos de Enagas y de Repsol no formarían parte de la cartera eficiente que proporciona una rentabilidad esperada del 20%. El motivo de esto es el bajo retorno esperado de estos activos. También, es destacable que sea Enagas el mayor activo que compone la cartera con menor riesgo. De esta afirmación se puede deducir que estos activos representan el menor riesgo, junto con una de las menores rentabilidades esperadas. Por otra parte, se puede comprobar que Acciona Energía estima la mayor rentabilidad para el inversor, sin embargo, no representa un activo con la mejor proporción entre rentabilidad esperada y riesgo. Los activos que poseen la mejor razón de Sharpe son Endesa e Iberdrola puesto que entre ambas conforman casi la mitad de la cartera optimizada en base a esta relación.

### 8.2.3 Simulación en el sector de las industrias en España

En esta tercera simulación, se genera una cartera compuesta por empresas del sector de la industria en España. Esta cartera está compuesta por activos de la siguiente relación de empresas:

- Acciona
- Acerinox
- Airbus
- Elecnor
- Ferrovial
- Fluidra

Una vez que se han incorporado estos activos incluyendo en el código de Python un vector que contiene los tickers de las empresas que formarán las distintas carteras se obtiene la siguiente gráfica:

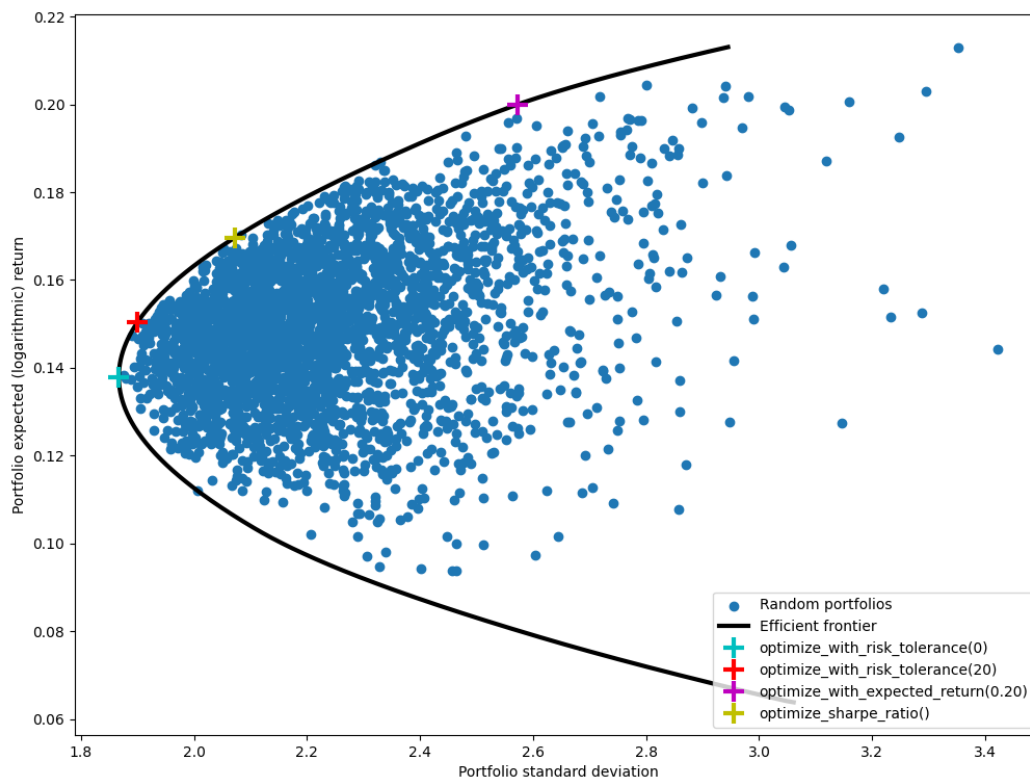


Ilustración 17

Observando el resultado se puede comprobar que el intervalo de rentabilidades en las que se encuentran las carteras eficientes tiene un límite inferior que en las simulaciones anteriores. Esto se puede comprobar observando la rentabilidad esperada del 14% de la solución de mínima varianza. También, la longitud de la frontera eficiente es mayor que en los otros sectores ya que el sector de la industria permite generar también carteras con un retorno esperado del 20%. Por otra parte, las carteras formadas no mejoran las condiciones de riesgo que presentan las anteriores simulaciones. Por todo esto se puede concluir que este sector no mejora al sector de las energéticas y petrolíferas de España.

Tabla 8-3: Relación de pesos de las carteras optimizadas con activos del sector de las industrias

Activo	Ticker	Retorno esperado	Optimización con tolerancia de riesgo 0	Optimización con tolerancia de riesgo 20	Optimización con retorno esperado 20%	Optimización de la razón de Sharpe
Acciona	ANA.MC	0,17652	0,14282	0,15877	0,22036	0,18409
Acerinox	ACX.MC	0,08051	0,14380	0,12022	0	0,08383
Airbus	AIR.MC	0,15470	0,11488	0,12049	0,13059	0,12951
Elecnor	ENO.MC	0,06065	0,23653	0,18429	0	0,10351
Ferrovial	FER.MC	0,15334	0,23903	0,24970	0,26571	0,26566
Fluidra	FDR.MC	0,26127	0,12293	0,16653	0,38334	0,23340

Analizando la tabla con los pesos de cada activo según el método de optimización que se emplee, resulta digno de mención el caso de la empresa Elecnor. Lo más llamativo es que sea la una de las 2 empresas mayoritarias en la cartera que es la solución de mínima varianza y también que no forme parte de la cartera que consigue una rentabilidad esperada del 20%. A partir de esto se puede deducir que su histórico de valores a lo largo del tiempo presenta las menores oscilaciones entre los demás activos. Por otra parte, se estudia el caso de Acerinox, la cual tampoco forma parte de la cartera a la que se le fija la esperanza del retorno al 20%. Además de no formar parte de esa cartera, es en los otros casos siempre una de las que menor proporción tiene. Esto puede darse debido a que una inversión en ese activo conlleve más riesgo que en ninguna de las otras. También se debe remarcar que Ferrovial siempre tomará uno de los pesos más altos dentro de cada cartera. Esto se puede explicar debido a que su relación rentabilidad esperada riesgo es la mejor dentro de esta cartera como bien lo confirma que sea el mayor peso en la cartera que optimiza la razón de Sharpe de la cartera.

#### 8.2.4 Simulación en el sector de las tecnológicas en España

En esta cuarta simulación, se ha optado por generar carteras que dediquen sus actividades empresariales al sector tecnológico español. Los activos que compondrán estas carteras se dedican en su mayoría al campo de las telecomunicaciones entre otros. Las empresas que se han considerado en este caso son:

- Cellnex
- Indra
- Telefónica
- Amadeus
- Ezentis
- Dominion

Cuando se incorporan los tickers de estas empresas en la variable `stocks` dentro del programa Python y se ejecuta, se obtiene el siguiente gráfico:

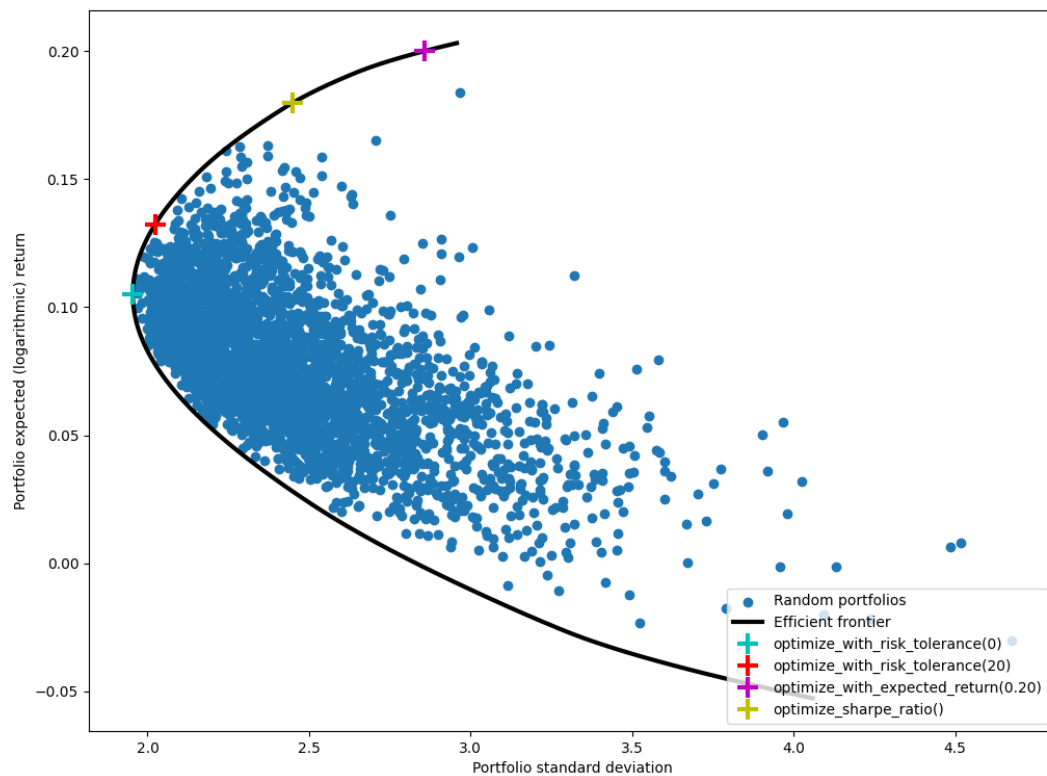


Ilustración 18

Se puede apreciar que la gráfica resultante es menos atractiva para los inversores. En primer lugar, la solución de mínima varianza posee una desviación típica mayor a la que se habían obtenido anteriormente. En segundo lugar, la curva de la frontera de las carteras aleatorias llega a tener una rentabilidad negativa. Por otro lado, parece que este sector presenta una Buena proporción rentabilidad riesgo pues aumentando el riesgo se consigue una mayor rentabilidad esperada en proporción para desviaciones típicas entre 2 y 2,5. Esto puede permitir al inversor valorar más posibilidades entre la cartera con tolerancia de riesgo 0 y la cartera optimizada según la razón de Sharpe.

Tabla 8-4: Relación de pesos de las carteras optimizadas con activos del sector de las tecnológicas

Activo	Ticker	Retorno esperado	Optimización con tolerancia de riesgo 0	Optimización con tolerancia de riesgo 20	Optimización con retorno esperado 20%	Optimización de la razón de Sharpe
Cellnex	CLNX.MC	0,15924	0,19891	0,22673	0,24167	0,27456
Indra	IDR.MC	0,05443	0,12600	0,10901	0	0,05901
Telefónica	TEF.MC	-0,01188	0,21282	0,14731	0	0
Amadeus	AMS.MC	0,14451	0,18972	0,20895	0,17154	0,23800
Ezentis	EZE.MC	-0,11386	0,06092	0,02587	0	0
Dominion	DOM.MC	0,23301	0,21162	0,28214	0,58679	0,42844

Analizando la tabla que proporciona los pesos exactos de cada cartera además de las rentabilidades esperadas de cada activo, se puede verificar que en la muestra de empresas que se han seleccionado no se espera ganar dinero si se invirtiera exclusivamente en ellas. Esto se puede afirmar ya que 2 empresas tienen un retorno esperado negativo. Con esto, se puede suponer que estos activos no formarán parte de aquellas carteras con mayor rentabilidad esperada. Sin embargo, es posible comprobar que Telefónica, a pesar de no formar parte de las carteras con un retorno esperado mayor, es el activo principal de la cartera de mínimo riesgo. Esto se debe a que la evolución del valor de sus acciones presenta la menor de las fluctuaciones. Por el contrario, Dominion presenta en el resto de los casos el mayor peso de la cartera. La explicación de esto radica en poseer la mejor rentabilidad esperada de todos los activos seleccionados y además implicar un riesgo menor que el resto de los activos.

### 8.2.5 Simulación en el sector de los bienes de consumo en España

En la quinta simulación de este estudio, se ha optado por elegir una representación de empresas que pertenecen al sector que se dedica a generar bienes de consumo. Las empresas que conformarán las distintas carteras optimizadas son:

- Coca Cola Europacific Partners
- Ence
- Faes Farma
- Iberpapel
- Inditex
- Vidrala

A continuación se presenta la gráfica que resulta de ejecutar el programa en lenguaje Python utilizando los identificativos de estas empresas como entrada.



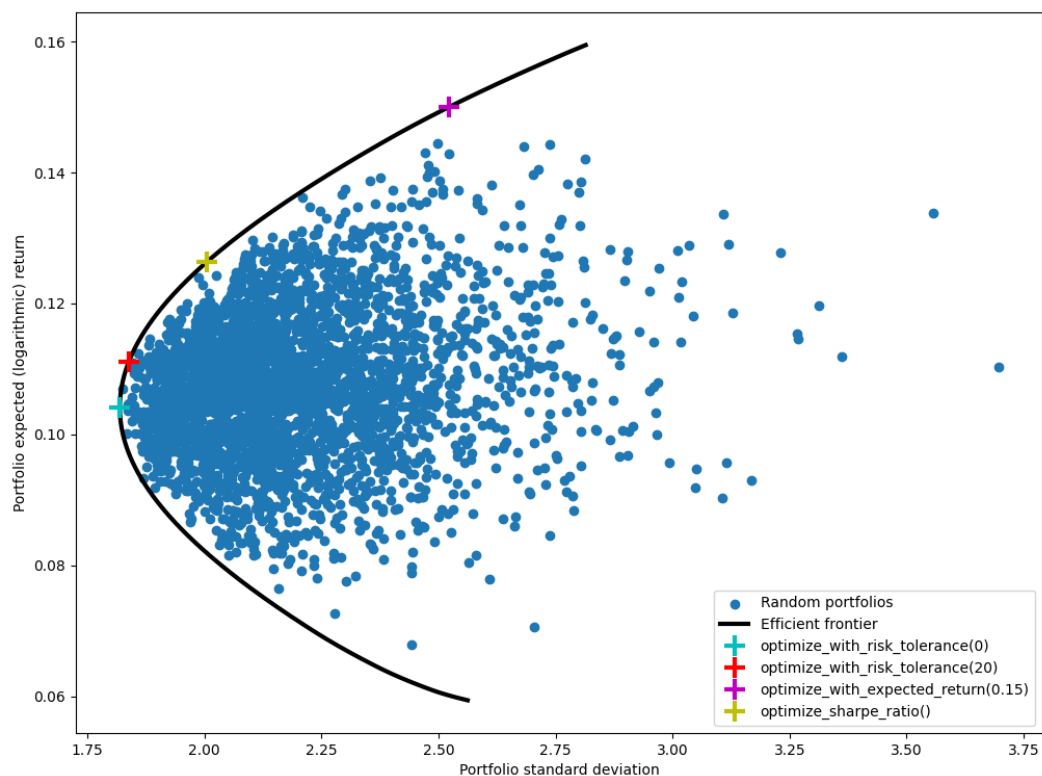


Ilustración 19

En primer lugar, hay que notificar que se ha disminuido el retorno esperado de la cartera que se optimiza a partir de conseguir ese retorno. La causa de esto es que el algoritmo de optimización que se utiliza no era capaz de encontrar una cartera eficiente con una rentabilidad del 20%. Para solventar esta situación se opta por pedir un retorno esperado del 15%. Observando la gráfica, a parte de obtener peores rentabilidades que anteriormente, se consigue unas mejores condiciones de riesgo. Sin embargo, no mejora las desviaciones típicas de la simulación del sector de las empresas energéticas y petrolíferas. Cabe remarcar la proximidad de las carteras con tolerancias de riesgo del 0% y 20%. Como consecuencia de esto, un inversor que no quiera asumir riesgos mayores pueda encontrar una solución que no sea la de mínima varianza pero que mantenga una rentabilidad esperada mayor.

Tabla 8-5: Relación de pesos de las carteras optimizadas con activos del sector de los bienes de consumo

Activo	Ticker	Retorno esperado	Optimización con tolerancia de riesgo 0	Optimización con tolerancia de riesgo 20	Optimización con retorno esperado 15%	Optimización de la razón de Sharpe
Coca Cola Europacific Partners	CCEP.MC	0,07694	0,14183	0,13015	0,04976	0,10447
Ence	ENC.MC	0,12549	0,08602	0,09151	0,11956	0,10443
Faes Farma	FAE.MC	0,15010	0,14071	0,16015	0,26963	0,20277
Iberpapel	IBG.MC	0,04758	0,22534	0,18679	0	0,09931
Inditex	ITX.MC	0,07391	0,19230	0,17462	0,05501	0,13661
Vidrala	VID.MC	0,17119	0,21380	0,25678	0,50604	0,35242

Analizando la tabla que muestra las rentabilidades esperadas de los activos y la composición de las carteras optimizadas se puede comprobar que como ningún activo supera el 18% del retorno esperado, no hay ninguna posible combinación de estos activos que consiga una rentabilidad esperada promedio del 20%. En el caso de cada activo, Iberpapel e Inditex muestran características similares ya que sus pesos son parejos en cada una de las carteras. Ambas son mayoritarias en las carteras que limitan el riesgo y, por el contrario, minoritarias en la cartera que impone una rentabilidad esperada mínima. Esto viene por el poco riesgo que supone una inversión en sus activos y por el escaso retorno que se le supone a los mismos. Por otra parte, Faes Farma no es un activo que conlleve poco riesgo en comparación con los demás que conforman las carteras, sin embargo, la relación entre la rentabilidad esperada de sus activos y el riesgo que conllevan hacen que sea uno de los principales componentes de la cartera optimizada según la razón de Sharpe.

### 8.2.6 Simulación en el sector de los servicios financieros en España

En esta penúltima simulación, se realiza una selección de empresas dedicadas al sector financiero, como pueden ser los bancos, compañías de seguros, de contabilidad, etc. En concreto, las empresas cuyos activos formarán parte de este estudio son:

- Banco Santander
- Banco Sabadell
- Banco BBVA
- Caixabank
- Mapfre
- Grupo Catalana Occidente

Como se hizo anteriormente, se introducen estos activos dentro de el código que permite obtener las carteras optimizadas según los 4 métodos distintos. Como resultado se obtiene la siguiente figura:

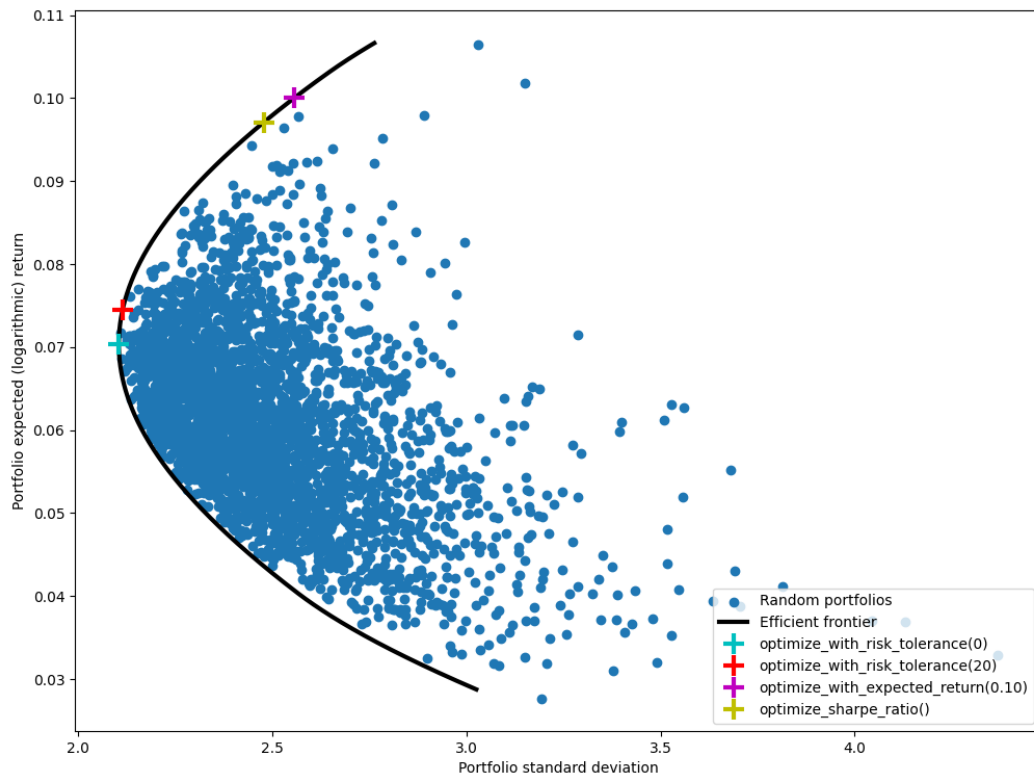


Ilustración 20

En primer lugar, el sector financiero no ofrece rentabilidades elevadas, por lo que se ha limitado para el método que busca la cartera óptima con un retorno esperado fijado al 10% dicho retorno. Como consecuencia de esto, ya se tiene la primera conclusión que se puede sacar de la gráfica. Por el contrario, este sector no presenta mucha amplitud de desviaciones típicas entre las carteras optimizadas. Esto implica que el histórico de valor de las acciones de estos activos no presenta tantas fluctuaciones como en otros sectores. Otra conclusión que se puede sacar analizando la gráfica es la proximidad de las carteras con tolerancias de riesgo y las carteras que optimizan según la razón de Sharpe y para un retorno esperado dado. Lo primero es consecuencia de el poco riesgo que conllevan los activos de estas empresas. Esto último se debe a la buena relación entre la rentabilidad esperada y el riesgo de las empresas de este sector.

A continuación, se muestra una tabla con la composición de cada una de las carteras optimizadas por los distintos métodos:

Tabla 8-6: Relación de pesos de las carteras optimizadas con activos del sector de los servicios financieros

Activo	Ticker	Retorno esperado	Optimización con tolerancia de riesgo 0	Optimización con tolerancia de riesgo 20	Optimización con retorno esperado 10%	Optimización de la razón de Sharpe
Santander	SAN.MC	0,01554	0,15462	0,13548	0,01875	0,03324
Sabadell	SAB.MC	0,00988	0,09548	0,08244	0,00293	0,01281
BBVA	BBVA.MC	0,04590	0,15643	0,14777	0,09510	0,10139
Caixabank	CXBK.MC	0,08322	0,14912	0,15347	0,17992	0,17600
Mapfre	MAP.MC	0,07725	0,20518	0,20839	0,22789	0,22645
Catalana Occidente	GCO.MC	0,13196	0,23918	0,27245	0,47541	0,45011

Analizando la tabla conjuntamente con la gráfica, se puede comprobar la poca diferencia que hay en las composiciones de las carteras que se encuentran más próximas. En el caso de las carteras con tolerancias de riesgo. Cabe destacar la diferencia de los pesos de el Grupo Catalana Occidente que ocurre debido a que es el activo con mayor retorno esperado, por lo que para aumentar la rentabilidad de la cartera, se deberá aumentar la proporción de este activo. Como era de esperar, es el activo mayoritario para las carteras con mayores rentabilidades esperadas. Por el contrario, los activos de Sabadell son los que menos ocupan en las carteras. Probablemente, esto se deba a que es el activo con menor retorno esperado. Además parece la empresa con peor relación entre rentabilidad esperada y riesgo ya que también es el activo con menor proporción en las carteras que consiguen mayores rentabilidades.

### 8.2.7 Simulación en el sector servicios en España

En esta última simulación del estudio, se ha decidido optar por empresas que realizan sus actividades económicas en el sector servicios. Este sector pasa por cadenas hoteleras, cadenas de medios de comunicación y clínicas sanitarias entre otras. Las empresas que se han elegido para realizar la simulación son:

- Aena
- Clínica Baviera
- Grupo Atresmedia
- Mediaset
- NH Hoteles
- Meliá Hoteles

Cuando se introducen los datos correspondientes a estos activos dentro del programa escrito en lenguaje Python, se obtiene la siguiente gráfica:

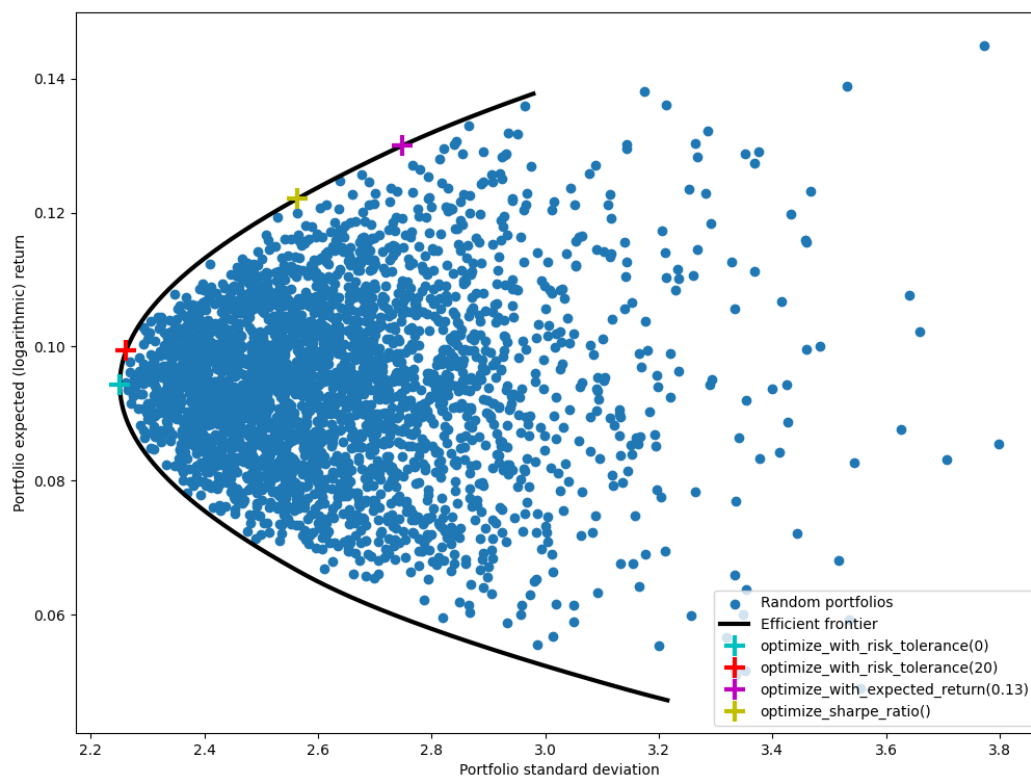


Ilustración 21

En primer lugar, se remarca que las rentabilidades de este sector son mayores que en el anterior pero no se aproximan a las rentabilidades de otros sectores como el de las energéticas y petrolíferas. Como consecuencia de esto, se usa como entrada para el método de optimización para un retorno esperado fijado de antemano una rentabilidad esperada del 13%. En esta gráfica se puede comprobar que según la naturaleza del inversor hay distintas opciones para componer carteras con distintas rentabilidades según el riesgo que se esté dispuesto a asumir. En este caso, hay carteras eficientes con rentabilidades esperadas que oscilan entre el 9,5% aproximadamente hasta un 14%. También, las desviaciones típicas de estas carteras se encuentran entre 2,3 y 2,8.

A continuación, se muestra la tabla que contiene la relación de pesos de los activos que compone cada cartera que se optimiza según cada método:

Tabla 8-7: Relación de pesos de las carteras optimizadas con activos del sector servicios

Activo	Ticker	Retorno esperado	Optimización con tolerancia de riesgo 0	Optimización con tolerancia de riesgo 20	Optimización con retorno esperado 13%	Optimización de la razón de Sharpe
Aena	AENA.MC	0,12315	0,23292	0,24613	0,32504	0,30341
Clínica Baviera	CBAV.MC	0,18664	0,14666	0,17346	0,33440	0,29190
Atremedia	A3M.MC	0,08793	0,16032	0,15835	0,14513	0,14984
Mediaset	TL5.MC	0,02070	0,18866	0,16122	0	0,03982
NH Hoteles	NHH.MC	0,08261	0,12708	0,12413	0,10559	0,11123
Meliá Hoteles	MEL.MC	0,06763	0,14436	0,13671	0,08983	0,10380

Observando los resultados obtenidos en la tabla, se puede comprobar que activos como los de las empresas NH Hoteles, Meliá Hoteles, Atresmedia y Mediaset ofrecen rentabilidades esperadas muy bajas. Como consecuencia de esto, no tomarán pesos elevados en las carteras que presenten retornos esperados más elevados. Es destacable dentro de este grupo de empresas que Mediaset es la que mayor peso de las cuatro tiene en las carteras con menos riesgo, sin embargo, en las carteras que mayor rentabilidad presentan es el activo con menor peso, llegando a poseer un peso asociado de 0 en la cartera que se optimiza estableciendo el retorno esperado al 13%. Por otro lado, se puede comprobar que conforme aumenta la rentabilidad esperada de la cartera los pesos que corresponden a los activos de Aena y Clínica Baviera también aumentan. También se puede concluir que invertir en activos de Clínica Baviera conlleva mayores riesgos que si lo fueran de Aena. Esto se justifica porque además de tener mayor peso Aena en las carteras con menor riesgo, también tiene un peso menor para la cartera que se optimiza según la razón de Sharpe. Esto último implica que la relación entre rentabilidad esperada y riesgo de Aena es mejor que la de Clínica Baviera.

## 9 CONCLUSIONES

---

A lo largo del proyecto, se ha realizado un estudio sobre el modelo de la cartera de la media-varianza que le sirvió a Harry Markowitz para conseguir un premio Nobel en el año 1990. En dicho estudio, se analizan los datos a tener en cuenta para realizar la elección de una cartera. Se ha definido que las carteras están compuestas por activos. Estos activos son derechos que tienen un valor monetario y se pueden comprar y vender. Estos activos no tienen un valor fijo, ni con el tiempo va perdiendo valor. Estos activos tienen dos propiedades fundamentales para la aplicación del modelo, la media y la varianza. Estas dos propiedades quedan definidas en base al histórico de valores que tiene cada activo. La longitud del periodo de tiempo en el que se tiene en cuenta los distintos valores del activo queda definida por el inversor. La media se define como la esperanza matemática de una muestra de datos, que viene a resultar el valor medio de la muestra. La varianza se utiliza como una medida de la dispersión de una muestra del histórico de valores. Se define como la esperanza matemática del cuadrado de la desviación del valor muestral respecto a su media.

Una vez se conoce cómo se forman las carteras, se realiza un análisis del modelo matemático el cual permite la obtención de carteras óptimas. Una cartera óptima se define para el modelo como aquella cartera que tiene un riesgo mínimo y una rentabilidad esperada igual o mayor que la referencia que indica el inversor. Junto con este modelo se explica un algoritmo que nos permite generar carteras admisibles y óptimas. Las variables que emplea el modelo son los pesos de los activos que componen la cartera que se quiera optimizar. Por consiguiente, el resultado de emplear el modelo y el algoritmo será las proporciones de cada activo que hay que adquirir para conseguir una cartera que minimice el riesgo y tenga una rentabilidad esperada aceptable para el inversor.

Como consecuencia de este modelo, surgen nuevos conceptos. El primero de ellos es la frontera eficiente, que se define como el conjunto de carteras que consiguen mayores rentabilidades para todo el intervalo de riesgo (varianza o desviación típica) que puedan tener las posibles carteras formadas por los activos seleccionados. Por definición, todas las carteras óptimas se deben encontrar en esta frontera eficiente. En segundo lugar, surge el teorema de los dos fondos. Este teorema establece que dadas dos carteras pertenecientes a la frontera eficiente, se puede conseguir un conjunto de carteras que también se encuentran en dicha frontera. En tercer lugar, aparece el teorema de los fondos mutuos. Para entender este teorema primero se definen los activos libres de riesgo, que son aquellos que tienen un nivel de riesgo muy bajo y se modelan como libres de riesgo. Posteriormente se establece que cuando una cartera posee activos libres de riesgo, dicha cartera se puede formar como combinación lineal de los activos libres de riesgo y el resto de los activos. Por último, se establece un modelo para la valorización de activos financieros. Este modelo se basa también en los activos libres de riesgo. El resultado del modelo es la obtención de un parámetro que permite obtener una estimación del en qué sentido y en qué magnitud afecta una fluctuación en el mercado a un activo o cartera.

Por otra parte, en este proyecto se emplea un archivo escrito en lenguaje Python que consigue implementar el modelo de Markowitz. A lo largo del capítulo 8, se llevan a cabo distintas simulaciones de carteras que se introducen en la implementación del modelo y se obtienen distintas carteras optimizadas. Analizando todos los casos en conjunto, se puede comprobar que las carteras con menor riesgo, son aquellas en las que los pesos de los activos son más equitativas. En base a esta observación se puede deducir que la diversificación de las carteras minimiza el riesgo de éstas. Por otra parte, en varios casos existen activos dentro de carteras optimizadas a los que le corresponde un peso de cero. Esto se debe en gran parte a la diferencia de

rentabilidades esperada de los activos que conforman la cartera. De esta idea se puede afirmar que para conseguir una cartera que nos aporte mejores rendimientos, primero el inversor tiene que observar el mercado para poder estimar el comportamiento futuro del valor de estos activos. Por ejemplo, no conviene invertir en activos cuya rentabilidad esperada sea muy baja o negativa así como tampoco es beneficioso formar carteras con activos que tengan grandes fluctuaciones en su histórico de valores. Por otro lado, conviene conocer cómo se comportan estos activos ante cambios en el mercado. Para ello se puede hacer uso del modelo de valorización de activos financieros (CAPM) y se estima los valores de la beta de cada active y/o de la cartera conjunta. Además, en virtud de la definición de frontera eficiente y del teorema de los dos fondos, se pueden obtener carteras diferentes, compuestas por los mismos activos en distinta proporción, que nos proporcionen niveles de rendimiento y riesgo que se ajusten más a la naturaleza del inversor. También resulta interesante incluir activos libres de riesgo en una cartera ya que representa una forma de conseguir reducir el riesgo de la cartera. Aunque se sacrifique parte de la rentabilidad, estos activos también tienen rentabilidades que serán no muy elevadas. La proporción debe usarse dentro del rango de rentabilidades que considera aceptable el inversor.

Recapitulando, para crear una cartera de inversión será necesario en primer lugar observar y conocer el mercado en el que se quiera invertir. Sólo de esta manera se puede evitar la inclusión de activos que no sean atractivos ya sea por su rentabilidad esperada o por su variabilidad. Una vez se consiguen carteras que no tengan mejores rendimientos para un nivel de riesgo, se considera una cartera eficiente. A partir de 2 carteras eficientes se pueden conocer carteras intermedias a éstas que varíen en rentabilidad esperada y riesgo. Siempre será interesante conocer la cartera de mínima varianza para poder ofrecer esta solución a los inversores más conservadores ya que esta cartera será la que menor riesgo conlleve. Siempre será el inversor el que dicte qué rentabilidades esperadas son aceptables y qué nivel de riesgo está dispuesto a asumir.



# REFERENCIAS

---

- [1] «Markowitz Mean-Variance Portfolio Theory» [En línea]. Available: <https://sites.math.washington.edu/~burke/crs/408f/notes/fin/mark1.pdf>
- [2] «Optimización con restricciones de desigualdad» [En línea]. Available: <http://www.ub.edu/matheopt/optimizacion-economica/condicion-de-khun-tucker>
- [3] «Introducción al lenguaje Python» [En línea]. Available: <https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion1/index.html>
- [4] «Download Python» [En línea]. Available: <https://www.python.org/downloads/>
- [5] «PEP 20 – The Zen of Python» [En línea]. Available: <https://peps.python.org/pep-0020/>
- [6] «Programación en Python - Nivel básico » [En línea]. Available: <https://entrenamiento-python-basico.readthedocs.io>
- [7] «Python Data Types - W3Schools» [En línea]. Available: [https://www.w3schools.com/python/python\\_datatypes.asp](https://www.w3schools.com/python/python_datatypes.asp)
- [8] «Python Online Compiler - W3Schools» [En línea]. Available: [https://www.w3schools.com/python/python\\_compiler.asp](https://www.w3schools.com/python/python_compiler.asp)
- [9] «Design and History FAQ - Python » [En línea]. Available: <https://docs.python.org/3/faq/design.html#why-can-t-lambda-expressions-contain-statements>
- [10] «Clases - Tutorial de Python» [En línea]. Available: <https://tutorial.recursospython.com/clases/>
- [11] «Librerías en Python - Immune Institute » [En línea]. Available: <https://immune.institute/librerias-python-que-son/>
- [12] «Instalando módulos de Python» [En línea]. Available: <https://docs.python.org/es/3/installing/index.html>
- [13] «Modern portfolio theory model implementation in Python» [En línea]. Available: <https://medium.com/analytics-vidhya/modern-portfolio-theory-model-implementation-in-python-e416facabf46>
- [14] «Yahoo finanzas» [En línea]. Available: <https://es.finance.yahoo.com/>