

# Optimización de Carteras

## Cálculo de la Frontera eficiente

Silvia García Lara

## Índice

|                                    |   |
|------------------------------------|---|
| 1. Presentación del problema ..... | 1 |
| 2. Descripción de los datos.....   | 1 |
| 3. Gráficas de evolución .....     | 2 |
| 4. Modelo de Markowitz .....       | 3 |
| 5. Conclusiones .....              | 6 |
| Anexo .....                        | 7 |
| Código R.....                      | 7 |

## **1. Presentación del problema**

En este trabajo se aborda la optimización de una cartera de inversión utilizando el modelo de Markowitz, con el objetivo de encontrar la combinación óptima de activos que maximice el rendimiento esperado, dada una cierta cantidad de riesgo.

Se utilizarán datos históricos de precios de acciones de varias empresas tecnológicas. A partir de estos datos, se calcularán los rendimientos y riesgos de las distintas combinaciones de activos y se obtendrá la frontera eficiente, que muestra las mejores combinaciones de activos que ofrecen el mayor rendimiento para un nivel de riesgo dado.

## **2. Descripción de los datos**

Se han elegido 10 empresas para la realización de este trabajo:

- Apple Inc. (AAPL)
- Microsoft Corporation (MSFT)
- Amazon.com, Inc. (AMZN)
- Meta Platforms, Inc. (META) (anteriormente Facebook Inc.)
- NVIDIA Corporation (NVDA)
- Tesla, Inc. (TSLA)
- Adobe Inc. (ADBE)
- Intel Corporation (INTC)
- Netflix, Inc. (NFLX)
- Advanced Micro Devices, Inc. (AMD)

Los datos utilizados en este trabajo provienen de la plataforma Yahoo Finance a través del paquete quantmod en R. Para cada empresa, se han extraído los precios de cierre ajustados desde enero de 2020 hasta abril de 2025. Los datos incluyen:

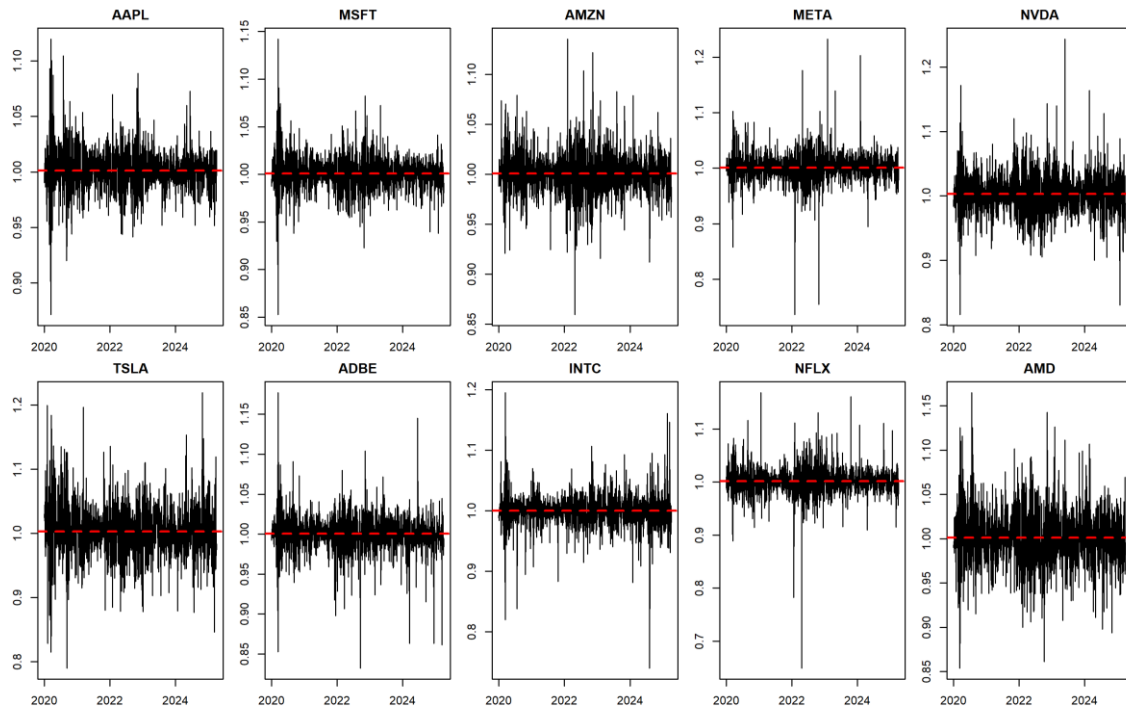
- Apertura: Precio de apertura de la acción en cada día.
- Máximo: El precio más alto alcanzado por la acción durante el día.
- Mínimo: El precio más bajo alcanzado por la acción durante el día.
- Cierre: El precio de cierre de la acción al final de la jornada.
- Volumen: El número de acciones negociadas durante el día.
- Ajustado: Precio de cierre ajustado por dividendos y splits.

Se utilizarán los precios ajustados para calcular los rendimientos diarios de las acciones.

### 3. Gráficas de evolución

Para poder ver la evolución de las acciones desde el año 2020 al año 2025 calculamos los retornos para cada una de las empresas.

Se realiza un gráfico para cada empresa representando los retornos.



Estos retornos representan la variación en el precio de sus acciones a lo largo del tiempo, se calcula también una media de los retornos representada con una línea roja.

Para tener una visión más concreta de cada empresa calculamos el rendimiento y el riesgo que tienen de media.

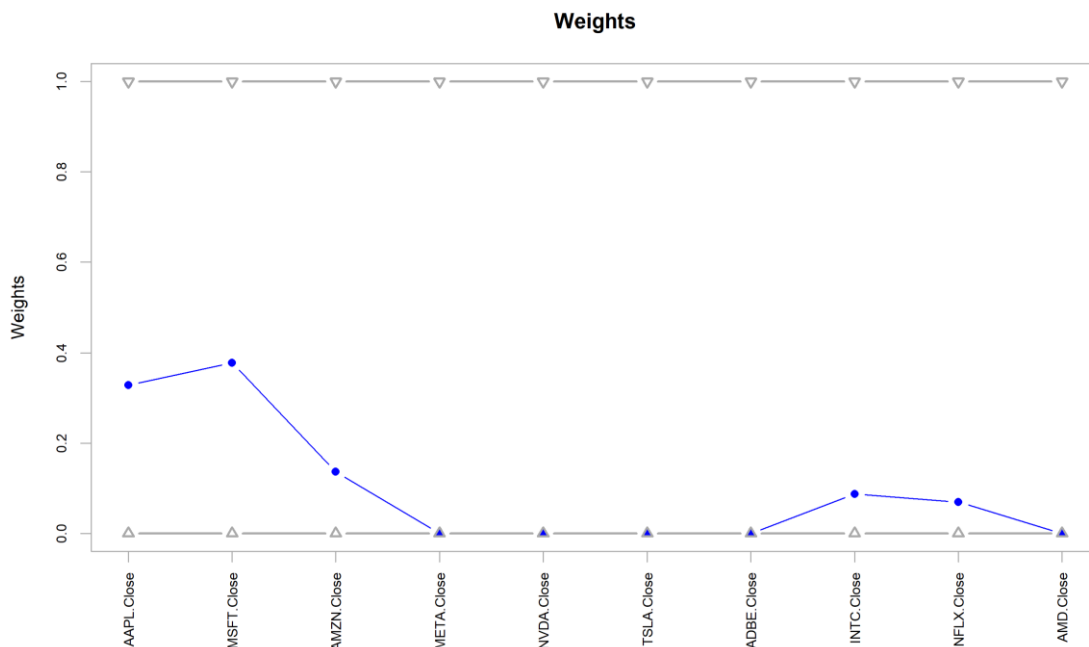
| Empresa   | Rendimiento | Riesgo       |
|-----------|-------------|--------------|
| Apple     | 1.00115     | 0.0002390423 |
| Microsoft | 1.000444    | 0.0002101761 |
| Amazon    | 1.000016    | 0.0002977797 |
| Meta      | 1.000915    | 0.0003623551 |
| Nvidia    | 1.000987    | 0.0007338006 |
| Tesla     | 1.000361    | 0.001093345  |
| Adobe     | 1.000254    | 0.0003242414 |
| Intel     | 0.999857    | 0.0004365701 |
| Netflix   | 1.000644    | 0.0004365701 |
| AMD       | 1.000063    | 0.0007320669 |

#### 4. Modelo de Markowitz

Para la resolución del problema, se define un portafolio con una función objetivo que minimiza el riesgo y las correspondientes restricciones que indican que todo el dinero disponible se debe invertir (full\_investment) y todos los pesos serán positivos o cero (long\_only).

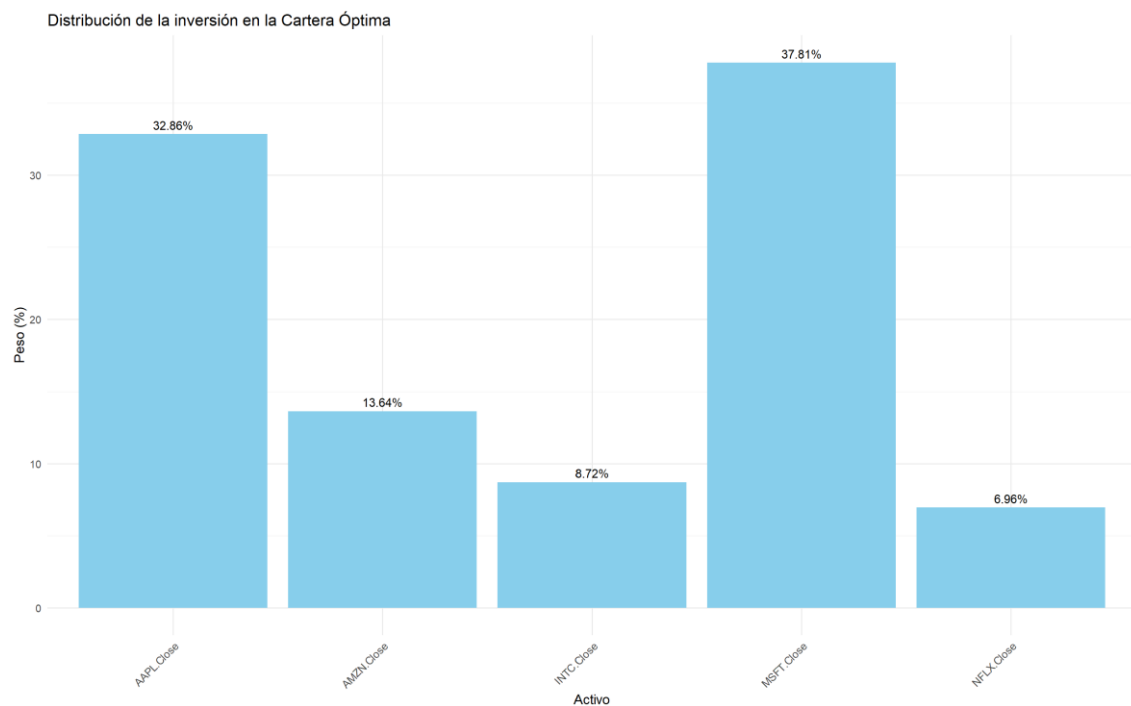
```
#####  
##### PORTFOLIO #####  
#####  
  
# Definir la especificación del portafolio para todas las empresas  
portSpec0 <- portfolio.spec(colnames(retornos_xts))  
  
##### Restricciones #####  
  
portSpec0 <- add.constraint(portfolio = portSpec0, type = "full_investment")  
portSpec0 <- add.constraint(portfolio = portSpec0, type = "long_only")  
  
##### Funcion objetivo #####  
  
# Objetivo: Minimizar riesgo  
portMinRisk <- add.objective(portfolio = portSpec0, type = "risk", name = "StdDev")  
  
##### Optimizacion #####  
  
optMinRisk <- optimize.portfolio(R = retornos_xts, portfolio = portMinRisk,  
                                optimize_method = "ROI")
```

Seguidamente, obtenemos la composición de la cartera que minimiza el riesgo de las 10 empresas, así como un gráfico que muestra los pesos de cada una de ellas:



En el gráfico observamos que las empresas como Apple o Microsoft son las que mayor peso tienen en la cartera seguidas por Amazon, Intel y Netflix.

En el siguiente gráfico de barras se describen los porcentajes de inversión de la cartera óptima.



Para la obtención de la cartera de mínimo riesgo se debe invertir el 32,86% del dinero en Apple, un 13,64% en Amazon, un 8,72% en Intel, un 37,81% en Microsoft y un 6,96% en Netflix.

```
> print(optMinRisk$objective_measures)
$StdDev
  StdDev
0.01765559
> minReturn
[1] 1.000809
```

Con esta cartera se obtiene un riesgo del 1,7% y una rentabilidad del 0.08%

Una vez encontrada la cartera que minimiza el riesgo, nos centramos en conseguir la cartera que maximice la rentabilidad y a la vez minimice el riesgo teniendo así dos objetivos.

```
##### DOBLE OBJETIVO: MEAN-RISK #####
#Objetivo 1: Maximizar el retorno medio esperado de la cartera.
portMeanRisk <- add.objective(portfolio = portSpec0, type = "return", name = "mean")
#Objetivo 2: Minimizar el riesgo
portMeanRisk <- add.objective(portfolio = portMeanRisk, type = "risk", name = "StdDev",
                             risk_aversion = 10)
optMeanRisk <- optimize.portfolio(R = retornos_xts, portfolio = portMeanRisk,
                                optimize_method = "ROI")
print(optMeanRisk)
```

maxSR

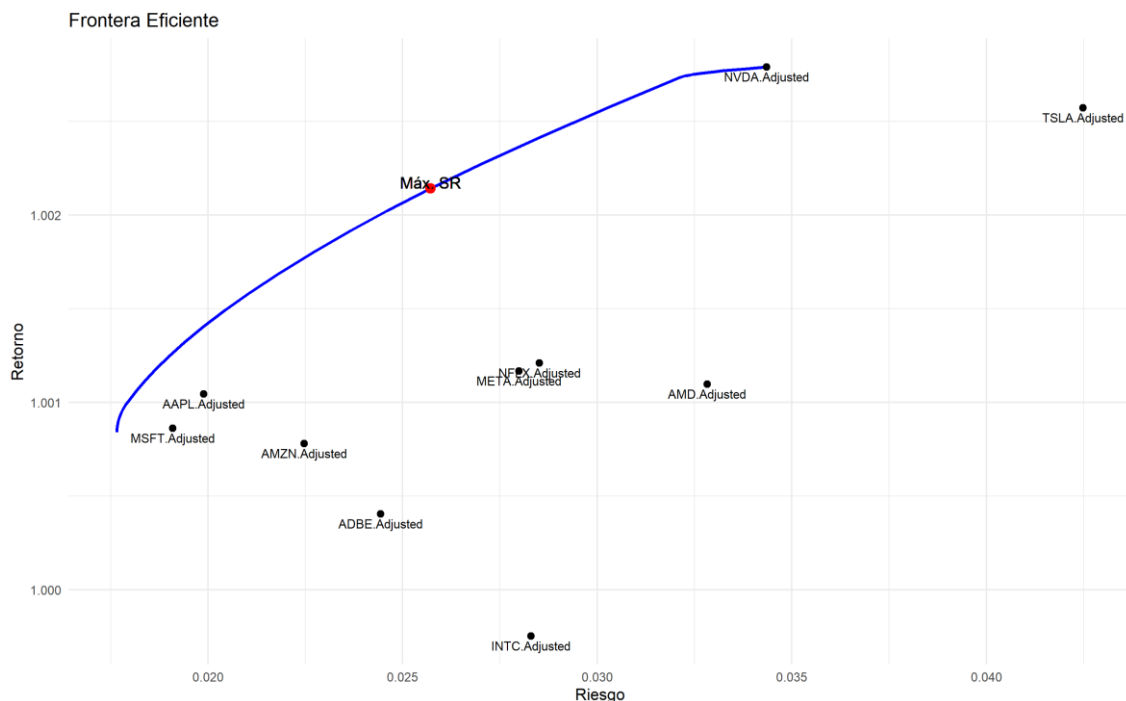
| Return   | Risk       | SR        |
|----------|------------|-----------|
| 1.002142 | 0.02571643 | 0.1027292 |

> weights

| AAPL.Adjusted | MSFT.Adjusted | AMZN.Adjusted | INTC.Adjusted | NFLX.Adjusted |
|---------------|---------------|---------------|---------------|---------------|
| 0.32873373    | 0.37892390    | 0.13581727    | 0.08648726    | 0.07003784    |

La mejor cartera es aquella que tiene el mayor Ratio Sharpe (0.1027). Las empresas que componen la cartera siguen siendo las mismas con porcentaje de inversión ligeramente diferentes. Observamos que el riesgo ha aumentado de un 1.7% a un 2.5%, sin embargo, con ese pequeño aumento en el riesgo se ha conseguido obtener un rendimiento de 0.2% en lugar de 0.08%.

Obtenemos el gráfico de la frontera eficiente que representa el rendimiento frente a riesgo, así como las distintas empresas y el valor de la mejor cartera (representado mediante el círculo rojo sobre la línea azul de la frontera eficiente).



## 5. Conclusiones

Las empresas situadas a la izquierda del gráfico son las que tienen un menor riesgo a la hora de invertir en carteras y las que se encuentran a la derecha tienen un mayor rendimiento.

La empresa Tesla es la que más lejos se encuentra de la frontera eficiente, en comparación con el resto de las empresas.

Es una buena opción invertir en Nvidia, ya que es la que tiene un mayor rendimiento medio aun que el riesgo sea elevado.

Si damos más importancia a minimizar el riesgo que a aumentar el rendimiento, la mejor empresa para invertir sería Microsoft, ya que es la que tiene menor riesgo de todas las empresas.

Vemos que las empresas que forman parte de la cartera óptima están todas cerca de la frontera óptima. Podemos concluir con estos resultados que no conviene invertir en las empresas AMD ni Tesla ya que están muy lejos de la frontera eficiente.



## Anexo

### Código R

```
#install.packages("quantmod")
#install.packages("PerformanceAnalytics")
#install.packages("PortfolioAnalytics")
#install.packages("ROI.plugin.quadprog")
#install.packages("ROI.plugin.glpk")
library(quantmod)
library(ggplot2)
library(PerformanceAnalytics)
library(PortfolioAnalytics)
library(ROI.plugin.quadprog)
library(ROI.plugin.glpk)
library(readxl)
library(xts)

# Obtener datos de las empresas de Enero 2020 a Abril 2025
# Apple
getSymbols("AAPL", src = "yahoo", from = "2020-01-01", to = "2025-04-01")
write.csv(AAPL, "AAPL_data.csv")

# Microsoft
getSymbols("MSFT", src = "yahoo", from = "2020-01-01", to = "2025-04-01")
write.csv(MSFT, "MSFT_data.csv")

# Amazon
getSymbols("AMZN", src = "yahoo", from = "2020-01-01", to = "2025-04-01")
write.csv(AMZN, "AMZN_data.csv")

# Meta
getSymbols("META", src = "yahoo", from = "2020-01-01", to = "2025-04-01")
write.csv(META, "META_data.csv")

# NVIDIA
getSymbols("NVDA", src = "yahoo", from = "2020-01-01", to = "2025-04-01")
write.csv(NVDA, "NVDA_data.csv")
```

```

# Tesla
getSymbols("TSLA", src = "yahoo", from = "2020-01-01", to = "2025-04-01")
write.csv(TSLA, "TSLA_data.csv")

# Adobe
getSymbols("ADBE", src = "yahoo", from = "2020-01-01", to = "2025-04-01")
write.csv(ADBE, "ADBE_data.csv")

# Intel
getSymbols("INTC", src = "yahoo", from = "2020-01-01", to = "2025-04-01")
write.csv(INTC, "INTC_data.csv")

# Netflix
getSymbols("NFLX", src = "yahoo", from = "2020-01-01", to = "2025-04-01")
write.csv(NFLX, "NFLX_data.csv")

# AMD
getSymbols("AMD", src = "yahoo", from = "2020-01-01", to = "2025-04-01")
write.csv(AMD, "AMD_data.csv")

# Lista de empresas
empresas <- c("AAPL", "MSFT", "AMZN", "META", "NVDA", "TSLA", "ADBE", "INTC",
"NFLX", "AMD")

# Lista para almacenar datos y retornos
datos_empresas <- list()
retornos_empresas <- list()
rendimiento <- list()
riesgo<- list()

# Cargar los datos en la lista y cálculo de retornos
for (empresa in empresas) {
  datos_empresas[[empresa]] <- get(empresa)
  precios_xts <- (Cl(datos_empresas[[empresa]]) /
Op(datos_empresas[[empresa]]))

```

```

    retornos_empresas[[empresa]]
na.omit(Return.calculate(Ad(datos_empresas[[empresa]])))+1
    rendimiento[[empresa]] <-mean(na.omit(precios_xts))
    riesgo[[empresa]]<-var(na.omit(precios_xts))
}

```

rendimiento

riesgo

```

# Mostrar los primeros valores de los retornos de cada empresa
for (empresa in empresas) {
  cat("\nRetornos de", empresa, ":\n")
  print(head(retornos_empresas[[empresa]]))
}

```

```

par(mfrow = c(2, 5), mar = c(2, 2, 2, 1))

```

```

# Graficar los retornos y la media para cada empresa
for (empresa in empresas) {
  if (!is.null(retornos_empresas[[empresa]])) {
    ret <- retornos_empresas[[empresa]]
    media_ret <- mean(ret, na.rm = TRUE)
    plot.zoo(ret,
      main = empresa,
      ylab = "Retorno",
      xlab = "",
      ylim = range(ret, na.rm = TRUE))

    abline(h = media_ret, col = "red", lty = 2, lwd = 2)
  }
}

```

```

# Crear un objeto xts con los retornos de todas las empresas
retornos_xts <- do.call(merge, retornos_empresas)

```

```
#####
#####

##### PORTFOLIO
#####

#####

#####

# Definir la especificación del portafolio para todas las empresas
portSpec0 <- portfolio.spec(colnames(retornos_xts))

##### Restricciones
#####

portSpec0 <- add.constraint(portfolio = portSpec0, type = "full_investment")
portSpec0 <- add.constraint(portfolio = portSpec0, type = "long_only")

##### Funcion objetivo
#####

# Objetivo: Minimizar riesgo
portMinRisk <- add.objective(portfolio = portSpec0, type = "risk", name = "StdDev")

##### Optimizacion
#####

optMinRisk <- optimize.portfolio(R = retornos_xts, portfolio = portMinRisk,
                                optimize_method = "ROI")

# Pesos óptimos
while (!is.null(dev.list())) dev.off()
dev.new()
par(mfrow = c(1, 1))
chart.Weights(optMinRisk)
pesos <- optMinRisk$weights
print(round(pesos, 4))

# Valor óptimo de la función objetivo
print(optMinRisk$objective_measures)
```

```

# Retorno mínimo
medias <- colMeans(retornos_xts, na.rm = TRUE)
minReturn <- sum(pesos * medias, na.rm = TRUE)

# Filtrar activos con pesos positivos y significativos
filtered_weights <- optMinRisk$weights[optMinRisk$weights > 1e-3]

# Crear un dataframe con los datos ajustados
weights_df <- data.frame(
  Activo = names(filtered_weights),
  Peso = filtered_weights*100
)

# Gráfico distribución de la inversión en la cartera óptima
ggplot(weights_df, aes(x = Activo, y = Peso)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  geom_text(aes(label = paste0(round(Peso, 2), "%")), vjust = -0.5, size = 3.5) +
  labs(title = "Distribución de la inversión en la Cartera Óptima",
    y = "Peso (%)",
    x = "Activo") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

##### DOBLE OBJETIVO: MEAN-RISK
#####
#Objetivo 1: Maximizar el retorno medio esperado de la cartera.
portMeanRisk <- add.objective(portfolio = portSpec0, type = "return", name =
"mean")
#Objetivo 2: Minimizar el riesgo
portMeanRisk <- add.objective(portfolio = portMeanRisk, type = "risk", name =
"StdDev",
  risk_aversion = 10)
optMeanRisk <- optimize.portfolio(R = retornos_xts, portfolio = portMeanRisk,
  optimize_method = "ROI")
print(optMeanRisk)

# Diversificación:  $1 - \sum(w^2)$ 
diversificacion <- diversification(optMeanRisk$weights)

```

```

# Mayor diversificacion: (Depende del numero de activos)
# El maximo para n activos: 1 - 1/n
max <- 1 - 1/7
diversificacionRelativa <- diversificacion/max
##### FRONTERA EFICIENTE
#####
n <- 200
riskAversion <- exp(seq(log(0.001), log(1000000), length.out = n))
fronteraEficiente <- data.frame(Return = numeric(n), Risk = numeric(n), SR =
numeric(n))

for (i in 1:n) {
  portMeanRisk <- portfolio.spec(colnames(retornos_xts))
  portMeanRisk <- add.constraint(portfolio = portMeanRisk, type =
"full_investment")
  portMeanRisk <- add.constraint(portfolio = portMeanRisk, type = "long_only")
  portMeanRisk <- add.objective(portfolio = portMeanRisk, type = "return", name =
"mean")
  portMeanRisk <- add.objective(portfolio = portMeanRisk, type = "risk", name =
"StdDev", risk_aversion = riskAversion[i])
  optMeanRisk <- optimize.portfolio(R = retornos_xts, portfolio = portMeanRisk,
optimize_method = "ROI")
  fronteraEficiente$Return[i] <- optMeanRisk$objective_measures$mean
  fronteraEficiente$Risk[i] <- optMeanRisk$objective_measures$StdDev
}
fronteraEficiente <- na.omit(fronteraEficiente)

# Sharpe ratio
rf <- 0.9995
fronteraEficiente$SR <- (fronteraEficiente$Return - rf) / fronteraEficiente$Risk

# Portafolio que maximiza el Sharpe ratio
maxSR <- fronteraEficiente[which.max(fronteraEficiente$SR), ]
maxSR

weights <- extractWeights(optMeanRisk)
weights <- weights[extractWeights(optMeanRisk) > 1e-3]
weights

```

```

# Crear un data frame con los datos de cada activo
asset_data <- data.frame(
  Risk = apply(retornos_xts, 2, sd, na.rm = TRUE),
  Return = medias,
  Label = colnames(retornos_xts)
)

# Graficar la frontera eficiente con los puntos de cada activo
ggplot(fronteraEficiente, aes(x = Risk, y = Return)) +
  geom_line(color = "blue", size = 1) +
  geom_point(aes(x = maxSR$Risk, y = maxSR$Return), color = "red", size = 3) +
  geom_text(aes(x = maxSR$Risk, y = maxSR$Return, label = "Máx. SR"), vjust = 0) +
  geom_point(data = asset_data, mapping = aes(x = Risk, y = Return), color = "black",
    size = 2) +
  geom_text(data = asset_data, mapping = aes(x = Risk, y = Return, label = Label),
    vjust = 1.5, size = 3) +
  labs(title = "Frontera Eficiente", x = "Riesgo", y = "Retorno") +
  theme_minimal()

```