Construcción de Matrices de Covarianza Robustas para Optimización de Portafolios de Inversión: Una Aplicación del Teorema de Marcenko-Pastur.

Omar Bairan

21 de julio, 2022

Abstract

La estructuración de portafolios óptimos ha sido un tema central en las finanzas. Desde las ideas presentadas por Harry Markowitz en su ensayo de 1952, el marco de análisis de media-varianza para la optimización de portafolios se ha aplicado ampliamente. Sin embargo, la inestabilidad de las matrices de covarianza para rendimientos de activos ha sido uno de los muchos problemas presentes en la metodología de media-varianza. Dado que las matrices de correlación generalmente se estiman a partir de datos de series de tiempo financieras históricas, que en general no son deterministas ni estacionarias, se espera que haya cierta cantidad de ruido en la estimación de las matrices de correlación. Una metodología que se ha propuesto para reducir dicho ruido consiste en aplicar la distribución de probabilidad de Marcenko-Pastur para separar los valores propios (autovalores) de la matriz de covarianza que se relacionan con el ruido de aquellos que representan señales reales. Generamos matrices de correlación aleatorias a partir de series temporales, obteniendo sus valores propios, así como sus vectores propios correspondientes y comparamos la distribución de los valores propios resultantes con la distribución esperada propuesta por Marcenko-Pastur. Finalmente, se aplica un procedimiento propuesto para eliminación de ruido a la matriz de covarianza obtenida de muestras aleatorias correlacionadas usando la descomposición de Cholesky para evaluar la efectividad de esta metodología. Al final del documento se incluye el código en Python utilizado.

Palabras clave: distribución de Marcenko-Pastur; matriz de covarianza; teoría moderna de portafolios; reducción de ruido; teoría de matrices aleatorias.

1 Introducción.

Las ideas introducidas por Harry Markowitz en su ensayo de 1952 sobre la selección de portafolios (Markowitz, 1952) pueden parecer evidentes en los tiempos actuales, pero representaron un gran avance en la teoría financiera de portafolios. Markowitz rechazó la regla de que el inversor sólo debería maximizar los rendimientos esperados descontados y postuló que los inversores deberían considerar los rendimientos esperados como un elemento deseable y la varianza como un elemento indeseable en sus portafolios (Markowitz, 1952).

Por tanto, el proceso de selección de portafolios implica resolver un problema de optimización en el que la asignación de capital limitado a un conjunto de activos debe maximizar la rentabilidad esperada del portafolio y minimizar la varianza del portafolio, que representa su riesgo.

Sin embargo, cuando se utiliza el marco de optimización de la varianza media para la optimización del portafolio, uno de los problemas que se presentan consiste en la inestabilidad de los parámetros. Dicha inestabilidad está relacionada con el comportamiento aleatorio de los datos de series temporales de activos financieros, que exhiben un comportamiento aleatorio y, en muchos casos, no son estacionarios. (Lo, 2017).

En particular, considerando que los datos financieros son finitos y no deterministas, la estimación de matrices de covarianza y correlación captura una gran cantidad de ruido derivado de los procesos aleatorios que subyacen al mercado. Dicho ruido se combina con las señales que se requieren para estimar correctamente las matrices de covarianza y correlación. Es posible reducir la cantidad de ruido en la matriz de correlación aplicando el Teorema de Marcenko-Pastur (López de Prado, 2020).

Las reglas generales de optimización de la media-varianza del portafolio basadas en el marco de Markovitz se pueden aplicar a cualquier tipo de activos financieros. En este trabajo utilizo variables aleatorias simuladas para crear un portafolio sintético con el fin de probar un proceso de eliminación de ruido que usa la función de densidad de probabilidad de Marcenko-Pastur para separar los valores propios de las matrices de correlación y covarianza que están relacionadas con el ruido de aquellas que transmiten información (señal).

Para probar la efectividad del procedimiento propuesto, comparo la varianza mínima obtenida usando un portafolio construido con la matriz de covarianza real, una estimación de la matriz de covarianza obtenida de muestras aleatorias correlacionadas y la matriz de covarianza sin ruido obtenida de muestras aleatorias correlacionadas. Mostraré que el uso de la matriz de covarianza sin ruido permite la reducción de la varianza total del portafolio de inversión.

2 Teoría de Matrices Aleatorias.

El estudio formal de matrices aleatorias comenzó en la década de 1930, pero inicialmente el tema no atrajo mucha atención. Desde entonces, las matrices aleatorias se han aplicado en muchas áreas, como la física nuclear para comprender el comportamiento de resonancias de neutrones lentos y la caracterización de sistemas caóticos, así como en matemáticas para la distribución de los valores de la función zeta de Riemann (Mehta, 2004).

Para una matriz M, de tamaño N x N, con valores aleatorios como componentes, tales que tienen $\mu = 0$ y $\sigma^2 = 1/n$, la ley circular de Girko-Ginibri establece que la densidad de valores propios es constante dentro de un disco centrado en cero y de radio sigma (Potters et al., 2021). Esto se puede ver en la Figura 1.

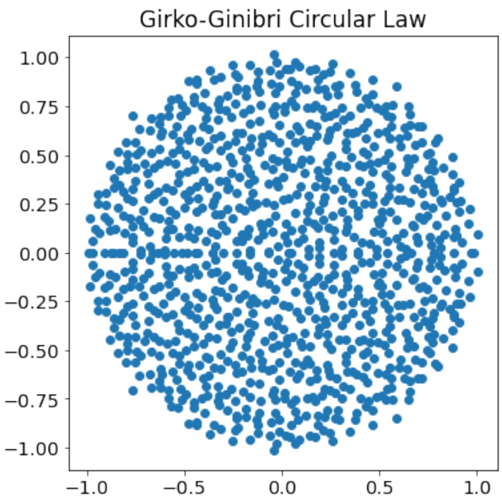

Figure 1: Girko-Ginibri Circular Law. Eigenvalues of a non-symmetric, random matrix, with $n = 1,000$, $\sigma^2 = 1/n$ and $\mu = 0$.

Consideremos también la Matriz de Wigner, que es una matriz simétrica con dimensiones N x N, con elementos reales independientes e idénticamente distribuidos (matriz hermítica) que satisface:

$$E[X_{ij}^2] \leq \infty, 1 \leq j \leq i \leq \infty, X_{ij}^2 = X_{ji}^2 \tag{1}$$

De acuerdo con la Ley del Semicírculo de Wigner (Aldridge el al., 2021), se puede observar que, conforme $N \to \infty$, los valores propios de una matriz de Wigner siguen una distribución característica. Un Conjunto Ortogonal Gaussiano (COG) es una matriz de Wigner que tiene elementos que sólo toman valores 0 o 1, los cuales siguen una distribución Gaussiana. Si ajustamos el tamaño de todos los valores propios, dividiéndolos entre $\sqrt{N}$, los mismos forman un semicírculo. Esto se puede ver en la Figura 2.

Para una matriz grande, aleatoria y simétrica, de acuerdo con la ley del semicírculo de Wigner, podemos esperar que la densidad de probabilidad de los valores propios siga una forma semicircular (Potters, 2021). Cuando observamos una gran cantidad de matrices de tamaño N x N, donde los elementos x_{ij}^2 son variables aleatorias independientes extraídas de una distribución idéntica e independiente con varianza finita, entonces los valores propios resultantes siguen una distribución que tiene la forma de un semicírculo. Esto se ha verificado empíricamente mediante la generación de matrices aleatorias de $N = 1000$, con $\sigma^2 = 1/n$ y $\mu = 0$, utilizando el código Python provisto en el Apéndice.

Una matriz de Wishart es una matriz cuadrada de la forma $W = XX^T$, donde X es una matriz N x T y los valores de X son variables aleatorias independientes distribuidas de forma idéntica con σ^2 finitos y $\mu = 0$. La matriz de Wishart W tiene valores propios que convergen a la función de densidad de probabilidad de Marcenko-Pastur, según $N \to \infty$, $T \to \infty$, con $T > N$ y $1 < T/N < \infty$ (Aldridge et al., 2021).

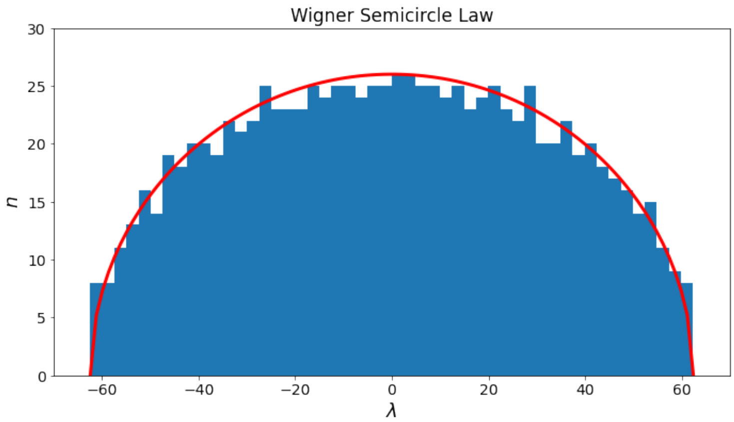

Figure 2: Wigner Semicircle Law.

La función de densidad de probabilidad Marcenko-Pastur se puede expresar como la Función 2:

$$f(\lambda) = \begin{cases} \frac{T}{N}\frac{\sqrt{(\lambda_{+}-\lambda)(\lambda-\lambda_{-})}}{2\pi\lambda\sigma^2} & \text{if } \lambda \in [\lambda_{-}, \lambda_{+}] \\ 0 & \text{if } \lambda \notin [\lambda_{-}, \lambda_{+}] \end{cases} \tag{2}$$

En el sector financiero se ha aplicado la Distribución Marcenko-Pastur de los valores propios de matrices aleatorias con la finalidad de reducir el componente aleatorio (ruido) presente en las matrices de correlación (Lalley, 2019).

El código de Python para calcular la función de densidad de probabilidad de Marcenko-Pastur se puede encontrar en el Apéndice. La forma de la función de densidad de probabilidad Marcenko-Pastur se puede observar en la Figura 3.

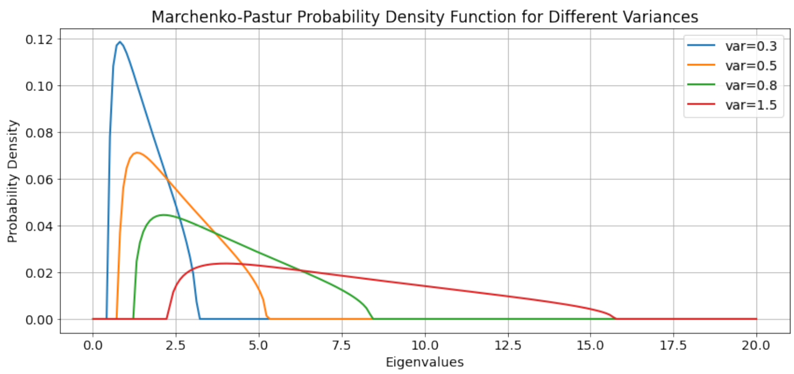

Figure 3: Función de Densidad de Probabilidad de Marcenko-Pastur.

3 Aplicación a una Matriz de Correlación Aleatoria.

En esta sección, para realizar un experimento controlado, se genera una matriz de correlación aleatoria para verificar que sigue la distribución Marcenko-Pastur. El primer paso es crear una matriz T x N (500 por 200). Para generar los datos aleatorios se ha utilizado la función random.normal() de la biblioteca Python Numpy. Asimismo, se usan las funciones que se encuentran en el Apéndice para crear una matriz de correlación aleatoria. Ver figura 4 para observar los resultados de este proceso.

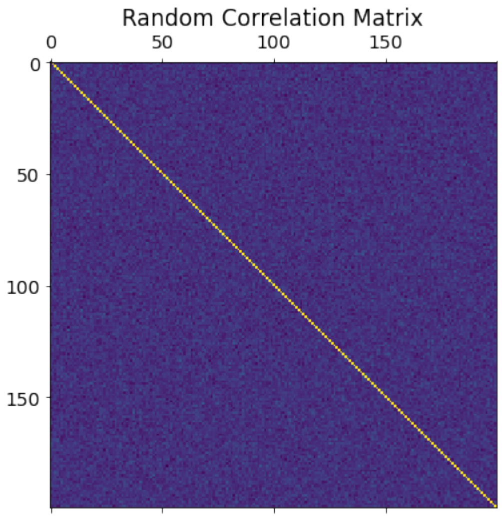

Figure 4: Matriz de Correlación Aleatoria.

Luego, se obtienen los valores propios correspondientes a la matriz usando la función linalg.eigh() de la biblioteca Python Numpy. Posteriormente, se utiliza la función de densidad de probabilidad de Marcenko-Pastur y se comparan los resultados con los de una matriz de correlación aleatoria. El código de python se puede encontrar en el Apéndice. Como se puede ver en la Figura 5, los valores propios obtenidos a partir de los dato aleatorios siguen la distribución de Marcenko-Pastur.

4 Creación de Matrices de Correlación y Covarianza con Estructura Conocida a Partir de Datos Aleatorios.

Adicionalmente, para demostrar cómo se puede aplicar el teorema de Marcenko-Pastur para separar los valores propios relacionados con el ruido aleatorio de otros que representan información relevante, se utilizan matrices de covarianza y correlación deterministas para generar múltiples series de puntos

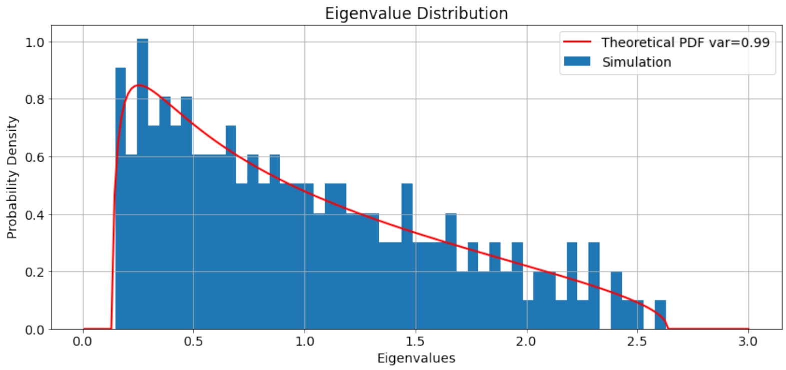

Figure 5: Distribución de Valores Propios de Matriz Aleatoria.

de datos aleatorios correlacionados, aplicando una descomposición de Cholesky de la matriz determinista simétrica. Después de eso, se calculan nuevas matrices de covarianza y correlación a partir de los datos aleatorios correlacionados. La matriz resultante incluye la información determinista que estaba presente en la matriz original, pero también presenta ruido del proceso aleatorio.

Para generar una matriz de correlación con ruido aleatorio se sigue el siguiente procedimiento:

- Crear matrices de covarianza y correlación, con dimensiones de 200 por 200 y una estructura conocida (ver Figura 6).
- Aplicar la descomposición de Cholesky a la matriz de covarianza usando la función cholesky() de la biblioteca scipy.linalg para obtener la matriz de covarianza descompuesta inferior L.
- Generar una matriz aleatoria M con dimensiones de 200 por 500 a partir de muestras aleatorias (ver la Figura 7).
- Transformar las muestras no correlacionadas en la matriz M en muestras correlacionadas multiplicando la matriz L por la matriz M.
- Calcular las nuevas matrices de covarianza y correlación a partir de las muestras correlacionadas. (ver Figura 8).

Una ventaja importante de usar este procedimiento es que se conoce la estructura de la matriz de correlación que subyace a la generación de variables aleatorias correlacionadas. Esto es útil para poder verificar la efectividad del proceso para eliminar el ruido de las matrices de covarianza y correlación.

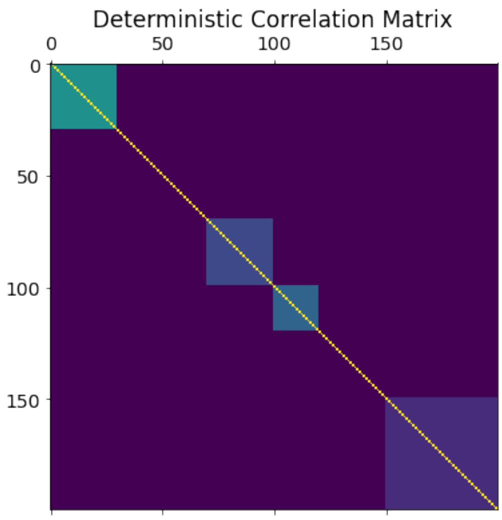

Figure 6: Matriz de Correlación Original.

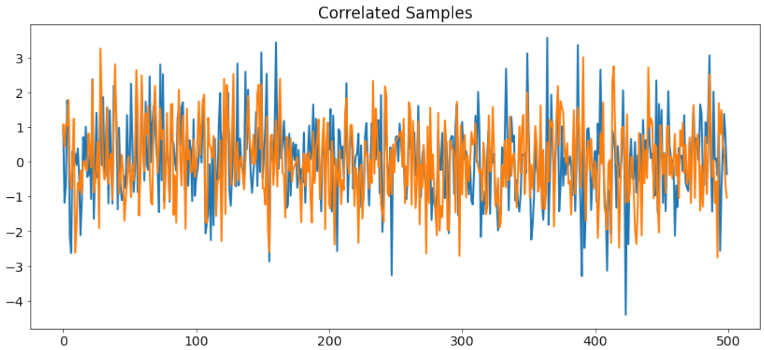

Figure 7: Muestras Aleatorias Correlacionadas.

5 Implementación de la Metodología para Reducción de Ruido en los Datos Aleatorios Correlacionados.

El siguiente paso consiste en seleccionar qué valores propios ajustar y cuáles deben permanecer iguales. Como se puede ver en la Figura 9, hay algunos valores propios grandes que se pueden considerar como señal y múltiples valores propios pequeños que se tratarán como ruido. En figura 10 es fácil observar que algunos autovalores están fuera del área bajo la función de densidad de probabilidad de Marcenko-Pastur representada por la línea de color rojo.

Se han propuesto diferentes métodos para separar los valores propios aleatorios de los que se relacionan con la señal. Un método útil parte la función de densidad de probabilidad Marcenko-Pastur. (Aldridge et al, 2021). Dado que la función de densidad de probabilidad de Marcenko-Pastur proporciona

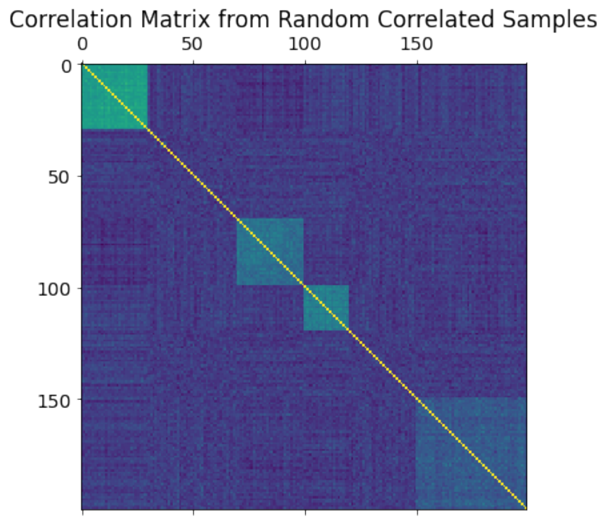

Figure 8: Matriz de Correlación con Muestra Aleatoria.

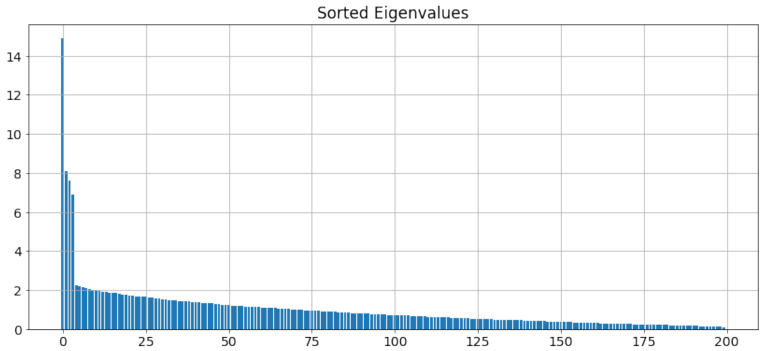

Figure 9: Valores Propios.

los valores propios más grandes y más pequeños que se pueden encontrar en una matriz de Wishart, podemos considerar una matriz de correlación aleatoria como un caso especial al que podemos aplicar esta función de densidad de probabilidad.

La función para el valor propio máximo esperado se puede ver en la Fórmula 3 y el valor propio mínimo esperado se puede estimar con la fórmula 4. Entonces, cualquier valor propio que esté fuera de ese intervalo se puede considerar como señal (es decir, que representa datos que no provienen de un proceso generado aleatoriamente).

$$\lambda_+ = \sigma^2\left(1 + \sqrt{\frac{N}{T}}\right)^2 \tag{3}$$

$$\lambda_- = \sigma^2\left(1 - \sqrt{\frac{N}{T}}\right)^2 \tag{4}$$

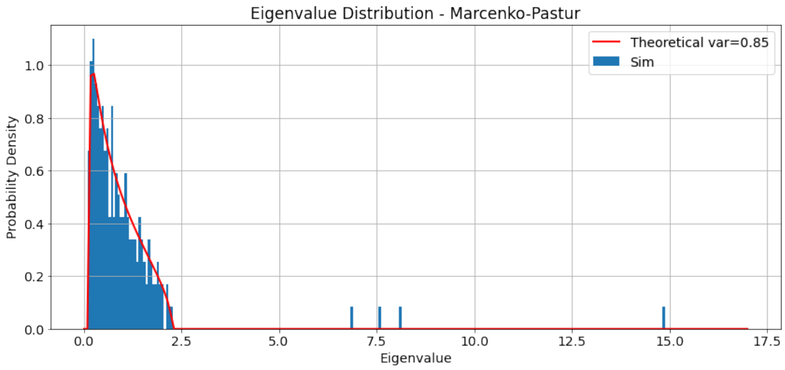

Figure 10: Distribución de Valores Propios.

El siguiente paso es ajustar todos los valores propios que se encuentran dentro de la función de densidad de probabilidad de Markcenko-Pastur asignándoles su valor promedio, mientras se mantienen intactos los otros valores propios. Cuando se realiza este proceso nos queda una nueva estructura de valores propios que se puede observar en la Figura 11.

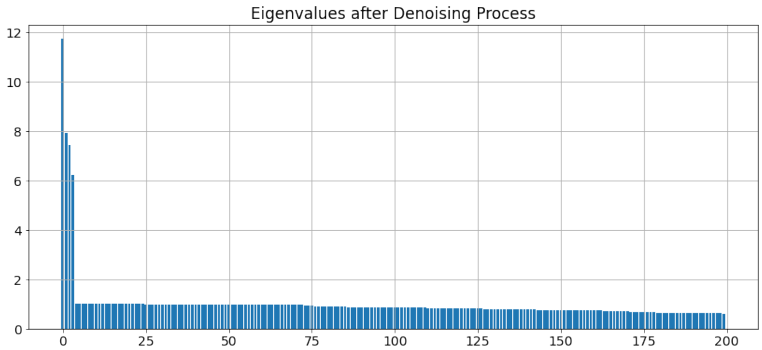

Figure 11: Valores Propios luego de Proceso para Eliminación de Ruido.

Finalmente, estimamos el portafolio de varianza mínima aplicando el procedimiento de optimización de varianza media propuesto por Markowitz (Markowitz, 1952) y comparamos los resultados en la Figura 10 y en la Tabla 1.

Desviación Estándar del Portafolio	
Correlaciones Reales	St. Dev. = 0.0894
Correlaciones según Muestreo (Ruido)	St. Dev. = 0.117
Correlaciones luego de Extracción de Ruido	St. Dev. = 0.0911

Table 1

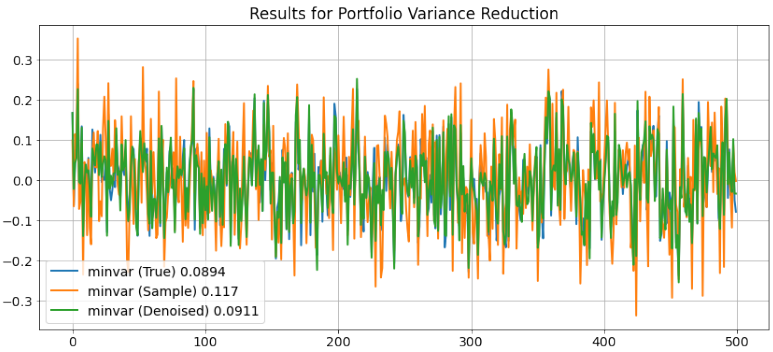

Figure 12: Resultados para Reducción de Varianza de Portafolio.

6 Conclusión.

Luego de aplicar el procedimiento para eliminación de ruido, aplicando los ajustes correspondientes a los autovalores obtenidos a partir de una matriz de correlación generada con una mezcla de datos aleatorios y deterministas, se puede observar en la Figura 10 y en la Tabla 1, que fue posible obtener una reducción considerable en la desviación estándar del portafolio total. Esta reducción representa más del 22% de la desviación estándar que estaba presente en el portafolio que se creó a partir de la matriz de correlación obtenida con los datos observados.

7 Apéndice: Códigos.

```
Codigo en Python utilizado:

#Ley Circular de Girko-Ginibri
import numpy as np
import matplotlib.pyplot as plt

n = 1000
M = np.random.random((n,n)) - 0.5
M *= (12/n)**0.5
w = np.linalg.eigvals(M)
plt.scatter(np.real(w), np.imag(w))
plt.axes().set_aspect(1)
plt.title('Girko-Ginibri Circular Law')
plt.show()


# Ley de Semic rculo de Wigner
import numpy as np
import matplotlib.pyplot as plt

N = 1000
A = np.random.normal(0, 1, (N, N))
B = (A + A.T)/np.sqrt(2)
eigenvalues = np.linalg.eigvalsh(B)
print(max(eigenvalues), min(eigenvalues))

plt.hist(eigenvalues, bins=70)
plt.show()

# Funcion de covarianza para dos variables
def calcCovar(x, y):
    mean_x = x.mean()
    mean_y = y.mean()
    n = len(x)
    return sum((x - mean_x) * (y - mean_y)) / n

# Matriz de covarianza
def covarMat(data):
    row, col = data.shape
    covar_mat = np.zeros((col, col))
    for i in range(col):
        for j in range(col):
            covar_mat[i][j] = calcCovar(data[:, i], data[:, j])
    return covar_mat

# Matriz de correlacion
```

```python
def corrMat(data):
    rows, cols = data.shape
    corr_mat = np.zeros((cols, cols))
    for i in range(cols):
        for j in range(cols):
            x, y = data[:, i], data[:, j]
            corr_mat[i][j] = calcCovar(x, y) / (x.std() * y.std())
    return corr_mat


# Funcion de Densidad de Probabilidad de Marcenko Pastur
def marcenko_pastur_pdf(var, time_steps, num_series, eVal):
    #var = variance
    #eVal = eigenvalue
    T = time_steps
    N = num_series
    q = T/N

    eMin = var * (1 - (1. / q) ** 0.5) ** 2
    eMax = var * (1 + (1. / q) ** 0.5) ** 2

    pdf =  (eMax - eVal) * (eVal - eMin)
    pdf[pdf<0] = 0
    pdf =  q / (2 * np.pi * var * eVal) * pdf**0.5

    return pdf,eMin, eMax


#Funcion para generacion de muestras correlacionadas
def gen_corr_samples(num_samples, cov):
    d = cov.shape[0]

    # Paso 1: Generar muestras aleatorias
    x = np.random.normal(size=(d, num_samples))

    # Paso 2: Descomposicion de la matriz de covarianza
    cov_decomp = cholesky(cov, lower=True)

    # Paso 3: Transformaci n de muestras no correlacionadas
    y = np.dot(cov_decomp, x)

    return y

#Funcion para obtener matriz de correlacion sin ruido
def denoisedCorr(eVal, eVec, nFacts):
    eVal_ = eVal.copy()
    eVal_[nFacts:] = eVal_[nFacts:].sum() / float(eVal_.shape[0]-nFacts)
    eVal_ = np.diag(eVal_)

    covar_ = np.dot(eVec,eVal_).dot(eVec.T)
```

```
    corr = covar2corr(covar_)
    return corr

# Funcion para obtener vector de pesos para varianza minima

def min_variance_weights(cov):
    inv = np.linalg.inv(cov)
    ones = np.ones(shape=(inv.shape[0],1))
    w = np.dot(inv,ones)
    w /= np.dot(ones.T,w)
    return w
```

Referencias bibliográficas.

[1] Alridge, Irene et al. *Big Data Science in Finance*. Wiley (2021).

[2] Bun, J. et al. *Cleaning large correlation matrices: Tools from Random Matrix Theory*. Elsevier. Physics Reports. (2016).

[3] Livan, G. et al. *Introduction to Random Matrices Theory and Practice*. Springer Briefs in Mathematical Physics (2018).

[4] Lalley, Steven P. *Random Matrices: Wigner and Marchenko-Pastur Theorems*. (2019).

[5] Lo, Andrew W. *Adaptive Markets: Financial Evolution at the Speed of Thought*. Princeton University Press (2017).

[6] López de Prado, Marcos. *Machine Learning for Asset Managers*. Cambridge University Press (2020).

[7] Markowitz, H. *Portfolio Selection*. The Journal of Finance, Vol. 7, No. 1, pp.77-91. March. (1952).

[8] Mehta, Madan Lal. *Random Matrices*. Elsevier Academic Press (2004).

[9] O'Rourke, S., M. et al *Eigenvectors of random matrices: A survey*. Elsevier Journal of Combinatorial Theory. (2016).

[10] Potters, M. et al. *A First Course in Random Matrix Theory for Physicists, Engineers and Data Scientists*. Cambridge University Press (2021).