

Universidad Nacional de Ingeniería
Facultad de Ingeniería Económica, Estadística y Ciencias
Sociales



TESIS

**Reducción del sesgo en modelos factoriales de baja dimensión:
una aplicación a sistemas de información en el mercado
financiero**

Para obtener el Título Profesional de Ingeniero Estadístico.

Elaborado por

Martin Andres Ochoa Celis

 [0009-0003-2065-5952](https://orcid.org/0009-0003-2065-5952)

Asesor

Dra. Anna Sikov Sikov

 [0000-0001-9869-5952](https://orcid.org/0000-0001-9869-5952)

LIMA – PERÚ

2025

Citar/How to cite	(Ochoa, 2025)
Referencia/Reference	Ochoa, M. (2025). <i>Reducción del sesgo en modelos factoriales de baja dimensión: una aplicación a sistemas de información en el mercado financiero</i> . [Tesis de pregrado, Universidad Nacional de Ingeniería]. Repositorio institucional Cybertesis UNI.
Estilo/Style: APA (7ma ed.)	
Estilo/Style: IEEE	M. Ochoa. "Reducción del sesgo en modelos factoriales de baja dimensión: una aplicación a sistemas de información en el mercado financiero", tesis de licenciatura, Fac. Ing. Económica y Estadística, UNI, Lima, Perú, 2025

Dedicatoria

A Dios que es mi fortaleza y guía en esta vida hasta el día de hoy.

*A mis padres, hermana y abuelos por su confianza, valiosos consejos, darme
alegría, apoyo y comprensión en estos años.*

*A mi abuelo Andres que descansa en el Cielo y le hubiese gustado ver culminado este
trabajo.*

Agradecimientos

A la Universidad Nacional de Ingeniería, por la educación de calidad y exigencia; a la escuela de ingeniería estadística por permitirme formarme como profesional.

A la Unidad de Investigación IECOS por el financiamiento otorgado para poder realizar la investigación.

A los profesores que me brindaron conocimiento y lecciones de vida a lo largo de la carrera.

A los profesores Anna Sikov y Jose Cerda Hernandez por su guía en el camino de la investigación, sus valiosos consejos y apoyo en la elaboración de la investigación presente.

A mi familia por su apoyo incondicional a lo largo de estos años.

Resumen

En la presente investigación se aplicó el Bootstrap paramétrico y no paramétrico a los estimadores de Fama-Macbeth, Shanken y Bai-Zhou para obtener estimadores con el sesgo y el error cuadrático medio relativos reducidos respecto a los mencionados utilizando para el estudio simulaciones generadas a partir de datos generados por investigadores extranjeros dado que a nivel nacional no se posee acceso a datos de ese tipo, también se generaron intervalos de confianza para los estimadores mencionados y los obtenidos. Se simularon datos utilizando diferentes tipos de residuales, tamaños de cartera y tamaños de periodo de investigación.

Se aplicó Bootstrap en el primer paso del modelo multifactorial en todos los casos, resultando que el método Bootstrap paramétrico logra mejoras solo en el caso de aplicarlo al estimador de Fama-Macbeth cuando la distribución de los errores es asimétrica. En el resto de los casos obtiene resultados similares a los estimadores de Bai-Zhou y en determinadas situaciones estimadores con pésimas características en términos de sesgo y error cuadrático medio.

Se observó que las estimaciones puntuales son mejores cuando se considera aumentos en el periodo de tiempo, pero cambios en el tamaño de la cartera no tiene impactos significativos. En los intervalos de confianza bajo el enfoque de Fama-Macbeth se obtiene en los dos casos Bootstrap menor variabilidad cuando el tiempo es pequeño y se incrementa el tamaño de la cartera; mientras que en el enfoque Shanken se presenta menor variabilidad en todos los casos respecto al enfoque de Fama-Macbeth.

Palabras clave — Fama-Macbeth, Shanken, Bootstrap, Bai-Zhou, sesgo, error cuadrático medio, rendimiento de un activo, prima de riesgo.

Abstract

In this study, parametric and nonparametric bootstraps were applied to the Fama-Macbeth, Shanken, and Bai-Zhou estimators to obtain estimators with reduced bias and root mean square error compared to the cited estimators. Simulations were generated from data generated by foreign researchers, given that such data are not available at the national level. Confidence intervals were also generated for the cited and the obtained estimators. Data were simulated using different types of residuals, portfolio sizes, and research period sizes.

Bootstrapping was applied in the first step of the multifactor model in all cases. It turned out that the parametric bootstrap method achieves parsimony only when applied to the Fama-Macbeth estimator when the error distribution is skewed. In the remaining cases, it obtains results similar to the Bai-Zhou estimators, and in certain situations, it produces estimators with poor characteristics in terms of bias and root mean square error.

It was observed that point estimates are better when considering increases in the time period, but changes in portfolio size do not have significant impacts. Confidence intervals under the Fama-Macbeth approach show lower variability in both bootstrap cases when the time period is short and the portfolio size increases; while the Shanken approach shows lower variability in all cases compared to the Fama-Macbeth approach.

Keywords — Fama-Macbeth, Shanken, Bootstrap, Bai-Zhou, bias, mean square error, asset return, risk premium.

Keywords — Scientific article, review article, research, citation styles.

Tabla de Contenido

	Pág.
Resumen.....	v
Abstract.....	vi
Introducción.....	xvi
Capítulo I. Parte introductoria del trabajo	1
1.1 Generalidades.....	1
1.2 Descripción del problema de investigación.....	1
1.3 Objetivos del estudio	3
1.3.1 Objetivo general.....	3
1.3.2 Objetivos específicos	3
1.4 Antecedentes investigativos	4
Capítulo II. Marcos teórico y conceptual.....	7
2.1 Marco teórico	7
2.2 Marco conceptual.....	13
Capítulo III. Desarrollo del trabajo de investigación	18
3.1 Población y muestra	18
3.2 Técnicas de análisis e instrumentos	20
3.3 Procedimiento	20
Capítulo IV. Análisis y discusión de resultados.....	27
4.1 Descripción de los datos	27
4.2 Estimador Fama-Macbeth con Bootstrap paramétrico	47
4.2.1 T = 60 y N = 100	47

4.2.2	T = 120 y N = 100	47
4.2.3	T = 60 y N = 300	49
4.2.4	T = 120 y N = 300	50
4.3	Estimador Fama-Macbeth con Bootstrap no paramétrico	52
4.3.1	T = 60 y N = 100	52
4.3.2	T = 120 y N = 100	53
4.3.3	T = 60 y N = 300	54
4.3.4	T = 120 y N = 300	55
4.4	Estimador Shanken con Bootstrap paramétrico.....	57
4.4.1	T = 60 y N = 100	57
4.4.2	T = 120 y N = 100	58
4.4.3	T = 60 y N = 300	58
4.4.4	T = 120 y N = 300	59
4.5	Estimador Shanken con Bootstrap no paramétrico.....	61
4.5.1	T = 60 y N = 100	61
4.5.2	T = 120 y N = 100	62
4.5.3	T = 60 y N = 300	63
4.5.4	T = 120 y N = 300	64
4.6	Aplicación.....	65
4.6.1	Fama-Macbeth.....	65

4.6.2 Shanken.....	66
Conclusiones.....	68
Recomendaciones.....	70
Referencias bibliográficas	71
ANEXOS	74

Lista de Tablas

Pág.

Tabla 1: Resultados obtenidos con $T = 60$ y $N = 100$	47
Tabla 2: Intervalos de confianza Bootstrap con $T = 60$ y $N = 100$	48
Tabla 3: Resultados obtenidos con $T = 120$ y $N = 100$	48
Tabla 4: Intervalos de confianza Bootstrap con $T = 120$ y $N = 100$	49
Tabla 5: Resultados obtenidos con $T = 60$ y $N = 300$	49
Tabla 6: Intervalos de confianza Bootstrap con $T = 60$ y $N = 300$	50
Tabla 7: Resultados obtenidos con $T = 120$ y $N = 300$	50
Tabla 8: Intervalos de confianza Bootstrap con $T = 120$ y $N = 300$	51
Tabla 9: Resultados obtenidos con $T = 60$ y $N = 100$	52
Tabla 10: Intervalos de confianza Bootstrap con $T = 60$ y $N = 100$	53
Tabla 11: Resultados obtenidos con $T = 120$ y $N = 100$	53
Tabla 12: Intervalos de confianza Bootstrap con $T = 120$ y $N = 100$	54
Tabla 13: Resultados obtenidos con $T = 60$ y $N = 300$	54
Tabla 14: Intervalos de confianza Bootstrap con $T = 60$ y $N = 300$	55
Tabla 15: Resultados obtenidos con $T = 120$ y $N = 300$	55
Tabla 16: Intervalos de confianza Bootstrap con $T = 120$ y $N = 300$	56

Tabla 17: Resultados obtenidos con $T = 60$ y $N = 100$	57
Tabla 18: Intervalos de confianza Bootstrap con $T = 60$ y $N = 100$	57
Tabla 19: Resultados obtenidos con $T = 120$ y $N = 100$	58
Tabla 20: Intervalos de confianza Bootstrap con $T = 120$ y $N = 100$	58
Tabla 21: Resultados obtenidos con $T = 60$ y $N = 300$	59
Tabla 22: Intervalos de confianza Bootstrap con $T = 60$ y $N = 300$	59
Tabla 23: Resultados obtenidos con $T = 60$ y $N = 300$	60
Tabla 24: Intervalos de confianza Bootstrap con $T = 60$ y $N = 300$	60
Tabla 25: Resultados obtenidos con $T = 60$ y $N = 100$	61
Tabla 26: Intervalos de confianza Bootstrap con $T = 60$ y $N = 100$	61
Tabla 27: Resultados obtenidos con $T = 120$ y $N = 100$	62
Tabla 28: Intervalos de confianza Bootstrap con $T = 120$ y $N = 100$	62
Tabla 29: Resultados obtenidos con $T = 60$ y $N = 300$	63
Tabla 30: Intervalos de confianza Bootstrap con $T = 60$ y $N = 300$	63
Tabla 31: Resultados obtenidos con $T = 120$ y $N = 300$	64
Tabla 32: Intervalos de confianza Bootstrap con $T = 120$ y $N = 00$	64
Tabla 33: Resultados de aplicar Fama-Macbeth, Bai-Zhou y Bootstrap	66
Tabla 34: Intervalo de confianza con Bootstrap	66
Tabla 35: Resultados de aplicar Shanken, Bai-Zhou y Bootstrap	67

Tabla 36: Intervalo de confianza con Bootstrap.....	67
---	----

Lista de Figuras

	Pág.
Figura 1: Muestra de algunas de las carteras.....	18
Figura 2: Muestra de los factores	19
Figura 3: Distribución de los residuales	23
Figura 4: Distribución de los residuales con mezclas	23
Figura 5: Esquema de Bootstrap paramétrico	25
Figura 6: Esquema de Bootstrap no paramétrico.....	25
Figura 7: Algoritmo de simulaciones.....	26
Figura 8: Evolución del portafolio 1 entre los años 2014 y 2018 medidos en forma mensual.....	27
Figura 9: Correlogramas parcial y normal respectivamente del portafolio 1 entre los años 2014 y 2018 medidos en forma mensual	28
Figura 10: Evolución del portafolio 30 entre los años 2014 y 2018 medidos en forma mensual.....	29
Figura 11: Correlogramas parcial y normal respectivamente del portafolio 30 entre los años 2014 y 2018 medidos en forma mensual	30
Figura 12: Evolución del portafolio 60 entre los años 2014 y 2018 medidos en forma mensual.....	31

Figura 13: Correlogramas parcial y normal respectivamente del portafolio 60 entre los años 2014 y 2018 medidos en forma mensual	32
Figura 14: Evolución del factor Mkt-RF entre los años 2014 y 2018 medidos en forma mensual	33
Figura 15: Correlogramas parcial y normal respectivamente del factor Mkt-RF entre los años 2014 y 2018 medidos en forma mensual	34
Figura 16: Evolución del factor HML entre los años 2014 y 2018 medidos en forma mensual	35
Figura 17: Correlogramas parcial y normal respectivamente del factor HML entre los años 2014 y 2018 medidos en forma mensual	36
Figura 18: Evolución del portafolio 1 entre los años 2009 y 2018 medidos en forma mensual	37
Figura 19: Correlogramas parcial y normal respectivamente del portafolio 1 entre los años 2009 y 2018 medidos en forma mensual	38
Figura 20: Evolución del portafolio 30 entre los años 2009 y 2018 medidos en forma mensual	39
Figura 21: Correlogramas parcial y normal respectivamente del portafolio 30 entre los años 2009 y 2018 medidos en forma mensual	40
Figura 22: Evolución del portafolio 60 entre los años 2009 y 2018 medidos en forma mensual	41
Figura 23: Correlogramas parcial y normal respectivamente del portafolio 60 entre los años 2009 y 2018 medidos en forma mensual	42

Figura 24: Evolución del factor Mkt-RF entre los años 2009 y 2018 medidos en forma mensual	43
Figura 25: Correlogramas parcial y normal respectivamente del factor Mkt-RF entre los años 2009 y 2018 medidos en forma mensual	44
Figura 26: Evolución del factor HML entre los años 2009 y 2018 medidos en forma mensual	45
Figura 27: Correlogramas parcial y normal respectivamente del factor HML entre los años 2009 y 2018 medidos en forma mensual	46
Figura 28: Primer paso de Fama-Macbeth en el software R	65
Figura 29: Primer paso de Fama-Macbeth en el software R	67

Introducción

La prima de riesgo es importante para la toma de decisiones en el ámbito empresarial. En ese sentido es importante su cálculo, entre las alternativas el modelo de Fama-Macbeth permite relacionar la prima de riesgo de factores con los rendimientos de activos de la empresa, determinando así el denominado estimador de Fama-Macbeth de prima de riesgo de factores. Este estimador ha sido mejorado a través de los aportes de Shanken el cual utilizó mínimos cuadrados generalizados; y Bai-Zhou, donde este último logró una disminución considerable del sesgo demostrando matemáticamente que el sesgo depende del tamaño del periodo de investigación y no del número de rendimientos considerados, proponiendo un nuevo estimador basándose en el estimador de Fama-Macbeth que toma en cuenta al tamaño del periodo de tiempo y produciendo así menor sesgo, sin embargo este nuevo estimador posee un error cuadrático medio elevado respecto al estimador de Fama-Macbeth, característica similar comparte el estimador de Shanken el cual también fue modificado teniendo en cuenta el mismo criterio que el estimador Fama-Macbeth y este nuevo estimador posee la misma característica en el sesgo y error cuadrático medio: bajo sesgo, pero error cuadrático medio alto respecto al estimador de Shanken.

El presente estudio aplica los métodos Bootstrap (paramétrico como no paramétrico) con la intención de lograr un equilibrio entre el error cuadrático medio y el sesgo a través de un estudio vía simulaciones utilizando diferentes tamaños de cartera, periodo de tiempo y distribuciones de los errores.

El principal aporte es mostrar en que situaciones y cuál de los enfoques del método Bootstrap logra encontrar equilibrio entre el sesgo y el error cuadrático medio.

La investigación consta de 7 partes:

En el capítulo 1 se encuentra la parte inicial del trabajo que consta de generalidades del estudio, antecedentes y objetivos de la investigación.

El capítulo 2 contiene los marcos teóricos y marco conceptual.

En el capítulo 3 se encuentra el desarrollo metodológico que guiara la obtención de resultados: población y muestra, los softwares que son la herramienta para los cálculos, y el procedimiento que se realizara para la obtención de resultados indicado paso a paso.

El capítulo 4 contiene análisis y discusión de los resultados obtenidos aplicando el procedimiento especificado en el capítulo 3.

Siguiendo se tienen las conclusiones y recomendaciones elaboradas a partir del capítulo 4.

Y por último se tienen las referencias usadas en la investigación.

Capítulo I. Parte introductoria del trabajo

1.1 Generalidades

Se define a una cartera de activos como los activos financieros de una empresa, donde los precios fluctúan a lo largo del tiempo y son de suma importancia en el análisis del riesgo total de la cartera de inversión. Diversos estudios han concluido que existe un conjunto de factores que influyen en las fluctuaciones de los precios de los activos. Debido a esto, es de interés su identificación y cálculo de primas de riesgo. En este sentido, los modelos asset pricing proporcionan una relación “aproximada” para el retorno esperado de un activo. Este retorno se puede escribir como una función de un número desconocido de factores no identificados.

Esto último es muy general y no nos permite realizar inferencia ni probar ciertas hipótesis o supuestos de algunas teorías de asset pricing. El modelo de Sharpe-Lintner(Sharpe: Capital Asset Prices: A Theory of Market Equilibrium under Condition of Risk, 1964; Lintner: The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portafolios and Capital Budgets, 1965; y Security Prices, Risk and Maximal gains from Diversification,1965) asume que el retorno medio de un activo es una función lineal de un único factor, el exceso de retorno del mercado. El segundo modelo de riesgo que es ampliamente estudiado en la literatura es el propuesto por Fama y French (1993), y es conocido como el modelo de tres factores de Fama-French. Este modelo incorpora dos factores de riesgo adicionales que capturan los rendimientos asociados con el efecto del tamaño(size) y el efecto del valor(value). Para ver otros modelos de factores se puede ver Cochrane (2009).

1.2 Descripción del problema de investigación

En el presente trabajo se realiza, primero, el estudio y el análisis el estimador de la prima de riesgo que fue propuesto por Fama y Macbeth, con diversos usos econométricos, para después realizar una corrección utilizando el trabajo de Bai y Zhou (2015). Cabe mencionar que muchos analistas que utilizaron el método de Fama-Macbeth

suponían que para obtener buenas estimaciones del riesgo era necesario solamente una gran cantidad de activos. Sin embargo, esta creencia fue invalidada en el trabajo de Bai y Zhou (2015). En dicho trabajo demostraron que el sesgo del estimador propuesto por Fama y Macbeth no solamente depende de la cantidad de los activos, que generalmente es grande, sino también del tamaño de la serie de tiempo, de modo que si este no es grande el sesgo del estimador puede ser significativo. Es decir, para obtener una buena de estimación de las primas de riesgo utilizando el método de Fama Macbeth no es suficiente con incluir una gran cantidad de activos en la cartera sin tener en cuenta el tamaño de la serie. En su trabajo los autores recalcan que el tamaño de la serie no es grande en la práctica. Para resolver este inconveniente desarrollaron de forma analítica una aproximación del sesgo del estimador original de Fama-Macbeth y a su vez proponen un estimador modificado asintóticamente insesgado para la prima de riesgo.

Los autores probaron mediante un estudio empírico que el nuevo estimador tiene un sesgo empírico muy pequeño cuando el tamaño de la serie de tiempo es pequeño. De igual modo probaron que el estimador desarrollado por Shanken (1992) tiene un sesgo notable cuando el tamaño del periodo de investigación es pequeño y también proponen una estimador para reducir dicho sesgo. Es de notar que en ambos casos el nuevo estimador se halló restando al riesgo estimador el sesgo obteniéndose estimadores asintóticamente insesgados. Como un detalle adicional los autores proponen estimadores asintóticos para la varianza que nos permite construir intervalos de confianza y realizar pruebas de hipótesis.

En situaciones prácticas, notamos que no solo la reducción del sesgo es importante, sino que también el error cuadrático medio nos permite identificar al mejor estimador. El estudio empírico de Bai y Zhou (2015) nos muestra que el error cuadrático medio(ECM) de los estimadores modificados es más alto comparado con el error cuadrático medio de los estimadores de Fama-Macbeth. Observando lo expuesto, el aporte de esta tesis sería el estudio del comportamiento de los estimadores originales desarrollados en Fama y MacBeth (1973) y Shanken (1992) con sus versiones

modificadas, introducidos en Bai y Zhou (2015), en términos del sesgo y ECM, en dependencia del número de activos, tamaño de la serie y el número de factores de riesgo considerados por el modelo. Esto nos permitiría evaluar el sesgo y el ECM de los estimadores para diferentes escenarios e identificar el tamaño de la serie óptimo (quizá como una función del número de activos), que resulta en estimadores con los valores de sesgo y ECM deseables. Además, en base a los resultados empíricos se va a sugerir una corrección para los estimadores desarrollados por Bai y Zhou (2015) para reducir el ECM de los mismos. Para alcanzar estos fines, vamos a realizar varios estudios empíricos, empleando las técnicas de Monte Carlo y Bootstrap. Finalmente, el modelo será ajustado a los datos de una cartera de activos que se negocian en la bolsa de valores de los Estados Unidos.

1.3 Objetivos del estudio

1.3.1 Objetivo general

Realizar un estudio empírico del comportamiento de los estimadores originales propuestos en Fama y MacBeth (1973) y Shanken (1992) con sus versiones modificadas, introducidas en Bai y Zhou (2015), en términos del sesgo y ECM, en dependencia del número de activos y tamaño de la serie.

1.3.2 Objetivos específicos

- Estudiar la eficiencia de los métodos de Bootstrap paramétrico y no paramétrico en términos de la reducción de sesgo de los estimadores de Fama-Macbeth y de Shanken.
- Encontrar equilibrio entre el error cuadrático medio y el sesgo de los estimadores de Bai-Zhou, basándose en simulaciones. Para realizar todos los cálculos necesarios se emplearán los métodos de Monte Carlo y métodos de remuestreo.
- Investigar el impacto del tamaño de cartera y del periodo de investigación en la eficiencia de los estimadores estudiados.

- Realizar un estudio comparativo de los estimadores de Bai-Zhou, Shanken y Fama-Macbeth implementándolos a los datos reales.

1.4 Antecedentes investigativos

1. Bai y Zhou (2015) analizaron el modelo de Fama-Macbeth propuesto en 1973 y observaron la falta de una fórmula asintótica para el estimador de dos pasos, su objetivo fue la búsqueda de una fórmula y proponer un nuevo estimador para las primas de riesgos de factores obteniendo que para el caso usual donde la cantidad de factores es mayor al de series de tiempo, un término significativo para el sesgo, para reducir ese efecto propusieron una mejora a el estimador Fama-Macbeth original también hallando para ese caso las varianzas asintóticas que permite obtener intervalos de confianza y pruebas asintóticas. A través de simulación se obtuvo que este nuevo estimador es mejor que el estimador clásico de Fama-Macbeth obteniéndose así un menor sesgo.
2. Giglio y Xiu (2021) observaron que la identificación de los verdaderos factores que influyen en el rendimiento de los activos es difícil de seleccionar, frente a esta dificultad su objetivo fue proponer una nueva metodología para analizar la prima de riesgo en estos casos. Para esto propusieron una modificación al método de dos pasos de Fama Macbeth en el cual añaden el paso de extraer información de los datos disponibles a través del método de componentes principales y modificando apropiadamente el método de Fama Macbeth llegaron a una metodología de 3 pasos para el cual los autores hallaron una fórmula asintótica. Prueban vía simulación que la nueva metodología mejora el cálculo de las primas de riesgo obteniéndose menor sesgo.
3. Efron (1979) realizó un estudio del método jackknife para intentar explicarlo en términos del método Bootstrap. El autor mostró que el método Bootstrap es útil en una variedad de problemas de estimación.

Mostró que el jackknife es un método de aproximación lineal para el Bootstrap. Se obtuvo que Bootstrap estima correctamente la varianza de la mediana donde el caso jackknife falla. También se mostró en el caso de la discriminación lineal, el método Bootstrap estima con buen desempeño las tasas de error superando a la estimación cruzada que es también un método no paramétrico de estimación.

4. Bellido (2019) buscó determinar el mejor método para estimar el riesgo bursátil del Índice S&P/BVL Mining Index de la Bolsa de Valores de Lima para el periodo 2008-2017, para ello analizó el Value at Risk que se calculó bajo los métodos de bootstrapping, simulación de Monte Carlo y simulación histórica. Utilizando las pruebas de Bcktesting (pruebas de verificación en el análisis cuantitativo de riesgo) obtuvo que el método más adecuado para estimar el riesgo bursátil del Índice S&P/BVL Mining Index de la Bolsa de Valores de Lima es el bootstrapping. El mejor modelo determinado fue el modelo AR (1)-EGARCH(1,1). El valor que se obtuvo del Value at Risk aplicando las metodologías de simulación de Montecarlo, simulación histórica y el método de Bootstrap es de manera respectiva del 3.33822 %, 4.5851 % y 3.160 %. La autora sugirió la aplicación de la metodología Bootstrap para la estimación del Value at risk de la cartera de instrumentos financieros que son de interés para poder tomar una mejor decisión que genere mayor rentabilidad.
5. Gavilán (2021) realizó el análisis y evaluación del método Bootstrap en modelos heterocedásticos para la predicción del índice de la bolsa de valores de Lima en el periodo 2017-2019. Se realizó la predicción con el método paramétrico y Bootstrap para evaluar su desempeño. El modelo que mejor se adecuó fue el modelo ARMA(1,1)-EGARCH(1,1), obteniéndose mejores resultados con el supuesto de distribución de tipo t-student con 5 grados de libertad de los residuales en cada modelo. La

comparación de ambos procedimientos permitió mostrar que la aplicación de Bootstrap permite obtener para el caso del retorno del índice de la bolsa de valores de Lima, intervalos de predicciones con mayores o iguales amplitudes en algunos horizontes de tiempo en comparación a la forma paramétrica.

6. Haro Pimentel (2016) hizo uso de la metodología desarrollada por Fama y Macbeth (metodología de dos pasos) en una cartera que incluyó a las acciones más líquidas de la Bolsa de Valores de Lima para buscar evidencia empírica de la influencia de factores elegidos a priori. Se eligieron cinco factores a priori utilizados como variables independientes (fueron la emisión primaria determinada por el BCRP, la inflación no anticipada, el tipo de cambio no anticipado, el índice de precios de los metales, y el Emerging Markets Bond Index), y el retorno accionario como la variable dependiente.

Capítulo II. Marcos teórico y conceptual

2.1 Marco teórico

Distribución normal

Sea una variable aleatoria X con densidad

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}, x \in \mathbb{R}$$

Entonces se dice que X tiene distribución normal y se escribe $X \sim N(\mu, \sigma^2)$. En el caso de $\mu = 0$ y $\sigma^2 = 1$ se dice que tiene distribución normal estándar.

Algunas propiedades son:

- $E(X) = \mu$
- $Var(X) = \sigma^2$
- Función característica $\phi(t) = \exp(i\mu t - \frac{1}{2}t^2\sigma^2)$
- Distribución simétrica

Distribución t

Sea $X \sim N(0,1)$ independiente de $Y \sim \chi_p^2$. Entonces

$$T = X/\sqrt{(Y/p)}$$

se dice que tiene distribución t con p grados de libertad. Se denota $T \sim t_p$. Su densidad es

$$f(x) = \frac{\Gamma(\frac{1}{2}p + \frac{1}{2})}{\sqrt{p\pi}\Gamma(\frac{1}{2}p)} \left(1 + \frac{t^2}{p}\right)^{-(p+1)/2}, t \in \mathbb{R}$$

Algunas de sus propiedades son:

- $E(T) = 0$ para $p > 1$.
- $Var(T) = \frac{p}{p-2}$ para $p > 2$.
- Distribución simétrica.

Distribución normal sesgada

De Azzalini (2013) tenemos la siguiente proposición:

Sea f_0 una función de densidad en \mathbb{R}^d . $G_0(\cdot)$ una función de distribución en \mathbb{R} y $w(\cdot)$ una función real en \mathbb{R}^d tal que

$$f_0(x) = f_0(-x), w(-x) = -w(x), G_0(-y) = 1 - G_0(y)$$

Para todo $x \in \mathbb{R}^d$ y $y \in \mathbb{R}$ entonces

$$f(x) = 2f_0(x)G_0(w(x))$$

Es una función de densidad en \mathbb{R}^d .

La prueba se puede ver en la obra citada

Si tomamos f_0 como la densidad normal estándar que denotaremos por ϕ , $G_0 = \Phi$ la función de distribución de la normal estándar y $w(x) = ax$ para algún real a , vemos que cumple las condiciones de la proposición mencionada y entonces

$$f(x) = 2\phi(x)\Phi(ax) \quad (1)$$

Es una función de densidad que se denotará $\phi(x, a)$ y su función de distribución por $\Phi(x, a)$.

Sea una variable aleatoria Z que se distribuye según la densidad de la (1). Entonces dado $\varepsilon \in \mathbb{R}$ y $\omega \in \mathbb{R}^+$, la variable:

$$Y = \varepsilon + \omega Z$$

Será llamada variable normal sesgada y su densidad para todo $x \in \mathbb{R}$ es

$$\frac{2}{\omega} \phi\left(\frac{x - \varepsilon}{\omega}\right) \Phi\left(a \frac{x - \varepsilon}{\omega}\right)$$

Y escribimos

$$Y \sim SN(\varepsilon, \omega^2, \alpha)$$

Algunas de sus propiedades:

- $E(Y) = \varepsilon + \omega\delta \sqrt{\frac{2}{\pi}}$ donde $\delta = \frac{a}{\sqrt{1+a^2}}$
- $Var(Y) = \omega^2 \left(1 - \frac{2\delta^2}{\pi}\right)$
- Presenta asimetría cuando $a \neq 0$.

Mezcla de distribuciones

De Feller (1989): Sea una función de densidad $f(x, \theta)$ que depende de un parámetro θ , si se escogen de forma arbitraria $\theta_1, \theta_2, \dots$ y si $\sum p_k = 1$ donde $p_k \geq 0$; entonces:

$$w(x) = \sum_k f(x, \theta_k) p_k$$

Define una nueva densidad de probabilidad. A estas densidades se le denominan mezclas.

De forma más general si F es una distribución que depende de un parámetro θ , y μ es una densidad de probabilidad, entonces

$$W(x) = \int_{-\infty}^{+\infty} F(x, \theta) \mu(\theta) d\theta$$

Es monótona creciente y el rango va de 0 a 1 y en consecuencia una función de distribución. Si F tiene densidad entonces W tiene densidad:

$$w(x) = \int_{-\infty}^{+\infty} f(x, \theta) \mu(\theta) d\theta$$

Regresión de Fama-Macbeth

Asumimos que los rendimientos de las acciones siguen un modelo multifactorial:

$$R_{it} = \alpha_i + \beta_{i1}f_{1t} + \dots + \beta_{ik}f_{kt} + \varepsilon_{it}, i = 1, \dots, N, t = 1, \dots, T \quad (2)$$

Donde:

R_{it} : rendimiento del activo i en el periodo t ($1 \leq i \leq N$)

f_{jt} : realización del j –ésimo factor en el periodo t ($1 \leq j \leq k$)

ε_{it} : término de perturbación aleatoria

N : número de activos y T es el número de observaciones de series de tiempo.

Con $R_t = (R_{1t}, \dots, R_{Nt})'$, $\beta_i = (\beta_{i1}, \dots, \beta_{ik})$, $B = (\beta_1, \beta_2, \dots, \beta_k)$, $f_t = (f_{1t}, \dots, f_{kt})$, $\alpha = (\alpha_1, \dots, \alpha_N)'$ y sea $\varepsilon = (\varepsilon_1, \dots, \varepsilon_T)$ hacemos la formulación vectorial: $R_t = \alpha + \beta_1 f_{1t} + \dots + \beta_k f_{kt} + \varepsilon_t = \alpha + B f_t + \varepsilon_t$.

Se tiene además como supuesto que las perturbaciones son idénticamente distribuidas con media cero e independientes sobre el tiempo y $E(\varepsilon_t \varepsilon_t') = \Sigma$.

Se tiene además para el procedimiento el siguiente supuesto:

$$H_0: E(R_t) = \gamma_0 \mathbf{1}_N + \gamma_1 \beta_1 + \dots + \gamma_k \beta_k \quad (3)$$

Donde γ_i son las primas de riesgo.

El procedimiento consiste en primero, estimar las betas aplicando MCO en la (1) para cada activo; sea $\hat{B} = (\hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_k)$ la matriz resultante. En el segundo paso se ejecuta una regresión transversal de R_t en $\hat{X} = [1_N, \hat{B}]$ en cada periodo para obtener una estimación de $\Gamma = (\gamma_0, \gamma_1, \dots, \gamma_k)'$, en este caso es $\hat{\Gamma}_t = (\hat{X}'\hat{X})^{-1}\hat{X}'R_t$ y como estimador final de Γ se toma

$$\hat{\Gamma} = \frac{1}{T} \sum_{i=1}^T \hat{\Gamma}_t = (\hat{X}'\hat{X})^{-1} \hat{X}'\bar{R} \quad (4)$$

Donde \bar{R} es la media de los rendimientos de los activos.

Estimador de Shanken

Shanken (1992) aplicó mínimos cuadrados generalizados en el segundo paso de la regresión de Fama Macbeth obteniéndose

$$\hat{\Gamma}^{GLS} = (\hat{X}'\hat{\Sigma}^{-1}\hat{X})^{-1}\hat{X}'\hat{\Sigma}^{-1}\bar{R} \quad (5)$$

Donde $\hat{\Sigma}$ es un estimador consistente de Σ y asumido invertible.

Distribución asintótica del estimador Fama Macbeth

Sea $F = (f_1, \dots, f_T)'$, $\dot{F} = F - \mathbf{1}_T \bar{f}$, $\bar{f} = F' \mathbf{1}_T / T$. Asumimos que $E(\bar{f}) = E(f_t)$ denotado por $E(f)$ y particionamos Γ de forma que

$$\Gamma = \begin{pmatrix} \gamma_0 \\ \gamma \end{pmatrix} \quad (6)$$

Bai y Zhou (2015) obtuvieron que el estimador de la regresión Fama Macbeth dado por la (4) tiene la representación asintótica siguiente cuando N es muy grande:

$$\hat{\Gamma} - \Gamma = \Omega^{-1} \begin{pmatrix} 0 \\ -\bar{\sigma}^2 \left(\frac{\dot{F}'\dot{F}}{T} \right)^{-1} \frac{\gamma}{T} \end{pmatrix} + \left(I_{k+1} - \Omega^{-1} \begin{pmatrix} 0 & 0 \\ 0 & \bar{\sigma}^2 \left(\frac{\dot{F}'\dot{F}}{T} \right)^{-1} \frac{\mathbf{1}}{T} \end{pmatrix} \right) \begin{pmatrix} \bar{f} \\ 0 \end{pmatrix} - E(f)$$

$$+ \frac{1}{NT} \Omega^{-1} \begin{pmatrix} -1'_N \varepsilon \left(\frac{\dot{F}' \dot{F}}{T} \right)^{-1} & \gamma + 1'_N \varepsilon 1_T \\ -B' \varepsilon \dot{F} \left(\frac{\dot{F}' \dot{F}}{T} \right)^{-1} & \gamma + B' \varepsilon 1_T \end{pmatrix} + o_p \left(\frac{1}{\sqrt{NT}} \right) \quad (7)$$

Donde:

$$\Omega = \frac{1}{N} \begin{pmatrix} 1'_N 1_N & 1'_N B \\ B' 1_N & B' B \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \bar{\sigma}^2 \left(\frac{\dot{F}' \dot{F}}{T} \right)^{-1} \frac{1}{T} \end{pmatrix}, \bar{\sigma}^2 = \frac{1}{N} \sum_{i=1}^N \sigma_{ii}^2 \text{ donde } E(\varepsilon_{ii}^2) = \sigma_{ii}^2$$

En base a esto propusieron el siguiente estimador:

$$\hat{\Gamma}^* = \hat{\Gamma} + \Omega^{-1} \begin{pmatrix} 0 \\ -\bar{\sigma}^2 \left(\frac{\dot{F}' \dot{F}}{T} \right)^{-1} \frac{\gamma}{T} \end{pmatrix} \quad (8)$$

Asimismo, bajo algunas condiciones de regularidad tenemos para el estimador de Shanken:

$$\begin{aligned} \hat{\Gamma}^{GLS} - \Gamma &= \phi^{-1} \begin{pmatrix} 0 \\ -\left(\frac{\dot{F}' \dot{F}}{T} \right)^{-1} \frac{\gamma}{T} \end{pmatrix} + \left(I_{k+1} - \Phi^{-1} \begin{pmatrix} 0 & 0 \\ 0 & \left(\frac{\dot{F}' \dot{F}}{T} \right)^{-1} \frac{1}{T} \end{pmatrix} \right) (\bar{f} - E(f)) \\ &+ \frac{1}{NT} \Phi^{-1} \begin{pmatrix} -1'_N \Sigma^{-1} \varepsilon \left(\frac{\dot{F}' \dot{F}}{T} \right)^{-1} & \gamma + 1'_N \Sigma^{-1} \varepsilon 1_T \\ -B \Sigma^{-1} \varepsilon \dot{F} \left(\frac{\dot{F}' \dot{F}}{T} \right)^{-1} & \gamma + B' \Sigma^{-1} \varepsilon 1_T \end{pmatrix} + o_p \left(\frac{1}{\sqrt{NT}} \right) \end{aligned} \quad (9)$$

Donde:

$$\Phi = \frac{1}{N} \begin{pmatrix} 1'_N \Sigma^{-1} 1_N & 1'_N \Sigma^{-1} B \\ B' \Sigma^{-1} 1_N & B' \Sigma^{-1} B \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \left(\frac{\dot{F}' \dot{F}}{T} \right)^{-1} \frac{1}{T} \end{pmatrix},$$

Método de Monte Carlo

Tenemos el problema de evaluar la integral

$$E_f(h(X)) = \int_H h(x) f(x) dx \quad (10)$$

Donde H es el soporte de la variable aleatoria X . El método de Monte Carlo aproxima la integral a través de una muestra aleatoria (X_1, \dots, X_n) de f y proponiendo como aproximación

$$\bar{h}_n = \frac{1}{n} \sum_{i=1}^n h(x_i) \quad (11)$$

Bootstrap

Según Efron (1979), el Bootstrap no paramétrico consiste:

Sea $X_i \sim F$, $i = 1, \dots, N$ independientes y $X = (X_1, X_2, \dots, X_N)$. El problema al que nos enfrentamos es estimar la distribución muestral de $R(X, F)$ usando una realización de X .

El método se describe:

1. Construye la distribución de probabilidad muestral \hat{F} , colocando masa $\frac{1}{n}$ en cada punto x_1, x_2, \dots, x_n .
2. Con \hat{F} fijo, realiza una muestra aleatoria de tamaño n de \hat{F} , digamos

$$X_i^* = x_i^*, X_i^* \sim_{ind} \hat{F}, i = 1, \dots, n$$

Llamada muestra Bootstrap, $X^* = (X_1^*, \dots, X_n^*)$, $x^* = (x_1^*, \dots, x_n^*)$. Los valores de X^* se seleccionan con reemplazo del conjunto $\{x_1, x_2, \dots, x_n\}$

3. Se aproxima la distribución $R(X, F)$ mediante la distribución Bootstrap de

$$R^* = R(X^*, \hat{F})$$

Para el caso Bootstrap paramétrico solo cambia la forma de generar \hat{F} , la cual ahora usa el modelo paramétrico de F , es decir la distribución.

En el caso usual la variable R es el sesgo de estimación de una función del parámetro de F .

Según Efron y Tibshirani (1994), para el sesgo tenemos que dado *bias*, estimador Bootstrap de $bias(\hat{\theta}, \theta)$, es natural pensar como estimador corregido por el sesgo:

$$\bar{\theta} = \hat{\theta} - \widehat{bias}$$

Además $\widehat{bias} = \hat{\theta}^* - \hat{\theta}$ donde $\hat{\theta}^*$ es el promedio de los estimadores de $\hat{\theta}$ en las muestras Bootstrap. Entonces

$$\bar{\theta} = 2\hat{\theta} - \hat{\theta}^*$$

Bootstrap en regresión lineal

Según Efron y Tibshirani (1994):

Tenemos que el modelo P de la regresión tiene las siguientes componentes:

$$P = (\beta, F)$$

Donde β son los parámetros de la regresión y F es la distribución de los errores, β es estimado por mínimos cuadrados ordinarios y β se estima a través de $\hat{\varepsilon}_i = \hat{y}_i - \hat{\beta}^t x_i$ y se obtiene así la distribución empírica a través de estos residuales. Siendo así:

$$\hat{F}: \text{probabilidad } \frac{1}{n} \text{ para } \hat{\varepsilon}_i \text{ con } i = 1, \dots, n$$

Siendo así la muestra Bootstrap $\varepsilon^* = (\varepsilon_1^*, \dots, \varepsilon_n^*)$ donde es muestreo por reposición de \hat{F} . Así obteniendo Bootstrap de la respuesta:

$$y_i^* = \hat{\beta}^t x_i + \varepsilon_i^* \text{ para } i = 1, \dots, n$$

Y aplicando MCO a esto último obtenemos el estimador Bootstrap de $\hat{\beta}: \hat{\beta}^*$ y el estimador corregido puede ser obtenido de igual que en el caso clásico, es decir 2 veces el estimado menos el promedio de los estimadores Bootstrap.

La forma anterior es el caso no paramétrico, para el caso paramétrico solo cambia la forma de generación de \hat{F} que se obtiene a partir de la distribución parametral de los residuales.

Intervalo percentil

Según Efron y Tibshirani (1994) se define el intervalo percentil para θ como sigue:

Generamos B muestras Bootstrap: $x^{*1}, x^{*2}, \dots, x^{*B}$ y se calcula los estimadores Bootstrap $\hat{\theta}^*(b) = s(x^{*1})$ donde s es una función como la media o varianza muestral. Sea $\hat{\theta}_B^{*(\alpha)}$ el $100(\alpha)$ percentil empírico de los valores de $\hat{\theta}^*(b)$, es decir, el valor $B\alpha$ en la lista ordenada de las B replicaciones $\hat{\theta}^*(b)$. Entonces el intervalo percentil $1 - 2\alpha$ aproximado es:

$$[\hat{\theta}_{\%lo}, \hat{\theta}_{\%up}] \approx [\hat{\theta}_B^{*(\alpha)}, \hat{\theta}_B^{*(1-\alpha)}]$$

2.2 Marco conceptual

Los siguientes conceptos se extrajeron de James (2004), Perú (2024), Cramér (1999), Feller (1989) y Robert et al. (2010).

Convergencia en probabilidad: Sean X, X_1, X_2, \dots variables aleatorias en un mismo espacio de probabilidad (Ω, \mathcal{A}, P) . X_n converge a X en probabilidad si para todo $\epsilon > 0$

$$P(|X_n - X| \geq \epsilon) \rightarrow 0 \text{ cuando } n \rightarrow \infty$$

Notación: $X_n \xrightarrow{P} X$

Convergencia en distribución: Sean X, X_1, X_2, \dots variables aleatorias con funciones de distribución F, F_1, F_2, \dots respectivamente. X_n converge en distribución a X cuando $n \rightarrow \infty$, si $F_n(x) \rightarrow F(x)$ para todo x punto de continuidad de F . Se denota $X_n \xrightarrow{D} X$.

Notación o_p : Sea $\{X_n\}$ una secuencia de variables aleatorias. $X_n = o_p(a_n)$ donde a_n es una sucesión, significa que

$$\frac{X_n}{a_n} \xrightarrow{P} 0$$

Notación O_p : Sea $\{X_n\}$ una secuencia de variables aleatorias. $X_n = O_p(a_n)$ donde a_n es una sucesión, significa que X_n/a_n es acotado estocásticamente, es decir

$$\forall \epsilon > 0, \exists (M > 0 \text{ y } N > 0): P\left(\left|\frac{X_n}{a_n}\right| > M\right) < \epsilon, \forall n > N$$

Sesgo: Sea v un parámetro y $\hat{v} = \hat{v}(X)$ nuestro estimador. El sesgo de \hat{v} es

$$bias(\hat{v}) = E(\hat{v}) - v$$

Error cuadrático medio (ECM): Sea v un parámetro y $\hat{v} = \hat{v}(X)$ nuestro estimador. El error cuadrático medio es

$$ECM(\hat{v}) = E(\hat{v} - v)^2$$

Consistencia: Supongamos que X_1, X_2, \dots, X_n es una muestra de F_θ donde $\theta \in \Theta$ y queremos estimar $q(\theta)$. Decimos que un estimador $\hat{q}_n(X_1, \dots, X_n)$ de $q(\theta)$ es consistente si

$$\hat{q}_n(X_1, \dots, X_n) \xrightarrow{P} q(\theta)$$

Series de tiempo: Es la realización de un proceso estocástico y estas variables están en periodos equidistantes. Notación: $\{Z_t, t \in \mathbb{N}\}$

Modelos factoriales lineales: Es aquel modelo identificado mediante:

$$Y_{ij} = \beta_j' X_i + \alpha_j' \theta_i + \varepsilon_{ij}, i = 1, \dots, n; j = 1, \dots, J$$

Donde Y_{ij} son las respuestas, X_i es un vector de regresores observados, θ_i es un vector de factores latentes y ε_{ij} es perturbación aleatoria.

Métodos de Monte Carlo: Método que permite la aproximación de expresiones que no se pueden expresar fácilmente de manera analítica. Consiste en generar datos Y_1, \dots, Y_N de una simulación independientes e idénticamente distribuidos según una conocida o desconocida densidad f . Nuestros datos son obtenidos de N ejecuciones independientes donde Y_i es el resultado de la i -ésima corrida. Supongamos que el objetivo es hallar $EY = l$ con $Y \sim f$. Tenemos que un estimador insesgado es:

$$\bar{Y} = \frac{1}{N} \sum_{i=1}^N Y_i$$

Asumiendo $|l| < \infty$ y la varianza de Y finita, digamos σ^2 , tenemos que un estimador de la varianza de Y es:

$$S^2 = \frac{1}{N-1} \sum_{i=1}^N (Y_i - \bar{Y})^2$$

Entonces se halla el intervalo aproximado de $1 - \alpha$ confianza para l :

$$\left(\bar{Y} - z_{1-\frac{\alpha}{2}} \frac{S}{\sqrt{N}}, \bar{Y} + z_{1-\frac{\alpha}{2}} \frac{S}{\sqrt{N}} \right)$$

Este procedimiento es el método de Monte Carlo.

Métodos de remuestreo: Son procedimientos estadísticos que reutilizan los datos de la muestra con fines de inferencia estadística. Sin embargo, no requieren supuestos paramétricos que puedan ser difíciles de verificar en la práctica. Entre ellos tenemos los métodos Bootstrap y Croos validation.

Mínimos cuadrados ordinarios: Sea el modelo lineal:

$$Y_{t \times 1} = X_{t \times k} \beta_{k \times 1} + \varepsilon_{t \times 1}$$

Con $var(\varepsilon) = \sigma^2 I_k$ entonces los mínimos cuadrados ordinarios consisten en hallar un estimador de β a través de la minimización de

$$\hat{\varepsilon}' \hat{\varepsilon}$$

Es decir minimizar la suma de cuadrados de los errores, cuyo estimador a través de esa minimización es:

$$\hat{\beta}_{MCO} = (X'X)^{-1}X'Y$$

Mínimos cuadrados generalizados: Sea el modelo lineal

$$Y_{t \times 1} = X_{t \times k} \beta_{k \times 1} + \varepsilon_{t \times 1}$$

Con $var(\varepsilon) = \sigma^2 \Omega$ donde Ω no es necesariamente la matriz identidad pero es definida positiva. Entonces podemos transformar el modelo lineal mediante el uso del álgebra lineal obteniendo así un modelo lineal con el supuesto clásico de homocedasticidad. Entonces los mínimos cuadrados generalizados consisten en realizar dicha transformación y hallar un estimador de β en el nuevo modelo vía mínimos cuadrados ordinarios:

$$\hat{\beta}_{MCG} = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}Y$$

Distribución empírica: La función de distribución empírica de una muestra aleatoria de tamaño n , denotada por $F_n(x)$ y definida para todos los números reales, denotando por $X_{(1)}, X_{(2)}, \dots, X_{(n)}$ las estadísticas de orden de una muestra aleatoria, es

$$F_n(x) = \begin{cases} 0, & x < X_{(1)} \\ \frac{k}{n}, & X_{(k)} \leq x < X_{(k+1)} \quad k = 1, \dots, n-1 \\ 1, & x \geq X_{(n)} \end{cases}$$

Datos panel: Un panel de N unidades observadas a lo largo de T periodos se representa de la siguiente manera:

$$Y_{ij}, \text{ con } i = 1, \dots, N; t = 1, \dots, T$$

Donde:

- i denota a la unidad o individuo, mientras que t hace referencia al tiempo
- N es el número total de unidades o individuos del panel
- T es el número de periodos de observación
- Y_{ij} representa a la variable de interés para la unidad i en el periodo de tiempo t .

Activo: son recursos tangibles e intangibles que una empresa posee y que tienen valor monetario.

Rendimiento de un activo: es el beneficio, rentabilidad, interés que produce un activo que se expresa en términos porcentuales.

Prima de riesgo: Diferencial entre el tipo de interés exigido por el mercado a un título de renta fija emitido por una empresa privada, y a otro de plazo equivalente por un emisor sin riesgo de crédito.

Cartera de activos: Conjunto de inversiones en títulos, valores de renta fija o variable y otros activos financieros que pertenecen a una persona natural o jurídica.

Capítulo III. Desarrollo del trabajo de investigación

3.1 Población y muestra

Tenemos una base de datos que contiene información sobre 100 portafolios de activos (ver Figura 1), 5 factores (ver Figura 2) a lo largo del periodo de enero de 1964 hasta junio del 2019 (medidos de forma mensual). Para la muestra usaremos los datos entre 2009 y 2018(120 periodos) con 5 factores y 100 portafolios de activos. La fuente de los datos es https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.htm.

Figura 1

Muestra de algunas de las carteras

Month	port8	port9	port10	port11	port12	port13	port14	port15	port16	port17	port18	port19	port20	port21	port22	port23	port24	port25	port26	port27
196401	3.5696	6.4239	5.3482	4.1409	-2.2196	2.7036	2.0104	3.4519	4.2909	2.829	4.8886	3.846	4.4879	-4.3985	-3.0654	-0.5295	4.2988	3.9355	6.684	1.5982
196402	1.4714	3.5593	2.9118	3.6045	3.3614	2.332	3.7665	3.0111	1.0605	-0.6467	-1.8346	6.8595	5.0928	0.065	2.3345	4.2637	0.6427	3.9144	2.7787	-0.0597
196403	4.1225	1.3155	1.8305	-0.579	1.4821	0.6634	1.0085	4.1953	0.2047	3.5628	2.1655	5.2375	2.3537	1.7177	2.4444	3.3009	2.0996	3.7886	3.7256	5.9057
196404	-1.8796	0.3607	0.2112	-0.4711	-5.8144	5.5005	4.5612	-1.4713	1.3543	-0.241	-0.1357	-2.3687	-1.8011	-3.5863	-2.0507	-1.7654	-1.1576	0.1624	-2.1281	-2.767
196405	0.5255	-0.0409	3.7843	-0.0276	0.3697	-1.9447	-1.3551	-2.3685	0.4158	0.7913	-0.9153	1.4604	0.3639	0.1976	-0.8679	2.2039	-0.8042	3.2546	3.2141	1.0099
196406	1.3306	1.2707	0.4277	1.1336	-1.0104	-0.0846	-0.9755	2.2873	-0.7919	-0.1456	1.362	1.6981	1.2042	0.6818	-1.2001	1.8774	0.7027	3.086	3.2147	0.4327
196407	5.0835	2.7434	7.6628	-2.7874	7.404	3.3312	5.0153	1.8017	6.0059	1.4252	1.981	4.161	4.2565	2.7312	4.2378	2.7582	3.3996	3.847	4.2125	2.9456
196408	-1.0202	-0.3009	-1.1568	-1.7716	3.1164	1.3024	-3.2182	-6.835	-0.5639	-0.4975	-1.1585	0.762	-0.9565	-0.6393	0.2766	-0.5759	-0.1218	-4.7119	0.006	1.1046
196409	4.7597	1.7852	3.4166	3.6442	7.3325	1.3761	0.5797	3.5486	3.5864	0.6595	1.5197	3.073	7.6913	-1.2945	3.435	2.265	3.7836	6.8436	1.248	2.2546
196410	1.1641	3.8844	2.405	0.1426	3.1545	-2.1184	1.1152	2.9926	0.0349	2.0436	2.9181	3.7846	3.6	0.8447	-0.1459	-1.046	-0.4492	4.7627	1.3177	1.0021
196411	-0.8225	-1.124	1.0427	-1.3227	2.7141	-1.5348	0.6208	-0.9643	-0.1355	1.6646	2.657	2.5574	0.3026	-0.5676	6.8258	2.0284	-1.0944	4.3223	-1.4974	0.8139
196412	-2.0794	-0.1979	-1.0908	-1.1194	-2.0497	-0.9699	-2.9848	-1.6552	-2.2055	-4.1654	-2.9847	-2.5616	-3.5144	-0.2497	0.2648	1.163	-1.9122	2.8061	1.2469	-3.9318
196501	8.5756	6.734	9.4033	8.4989	9.3822	0.9644	8.2632	3.4279	1.4769	3.3985	6.8757	10.9223	8.5335	16.8656	8.4164	7.8259	6.21	2.3484	3.426	5.7024
196502	4.6867	4.5242	3.0153	2.3918	3.468	2.0384	0.2264	1.7504	4.2568	2.4358	1.5338	4.3986	2.2779	8.3142	0.1655	4.3597	6.3678	3.0907	3.1337	2.4767
196503	1.1427	0.5752	4.5789	-1.9687	2.3872	2.4715	1.6812	0.7854	5.827	1.5406	3.9917	-0.3295	7.3133	8.1758	-0.4176	1.7795	-3.7818	4.4529	0.1739	2.2175
196504	4.7959	5.9915	4.8771	2.5571	5.1829	3.5851	7.0085	5.646	4.1757	-1.0575	6.5114	4.1955	3.179	0.3262	-0.8946	5.314	3.068	2.258	1.1569	5.7326
196505	-0.3267	0.1705	-0.5224	-2.7929	-1.7119	-3.5618	-4.1583	-1.4599	3.1569	-1.0442	-1.0794	-2.0048	-3.0888	2.9063	2.1661	1.8304	-3.1114	-4.3693	0.9231	-0.3398
196506	-8.3066	-7.7545	-9.8159	-11.6828	-13.0247	-9.4575	-12.5808	-8.8179	-7.932	-6.1152	-7.5873	-8.7235	-11.946	-14.4228	-5.7844	-8.7725	-8.4738	-10.3782	-7.4379	-8.3696
196507	0.8624	2.9049	5.0715	6.8247	-0.6493	5.5564	3.0549	4.2426	2.2817	3.2127	2.3448	3.509	5.9283	8.6282	2.6738	-1.1674	1.1039	3.2901	3.5348	3.2321
196508	3.814	2.9433	3.3166	3.4551	0.9403	8.6896	1.987	6.87	4.6666	6.0203	2.145	4.0122	6.0435	6.8664	4.8436	5.1424	3.6248	3.4777	5.3172	2.2486

Nota. Muestra de los primeros meses de los factores. Obtenido de la página web https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

Figura 2

Muestra de los factores

Month	Mkt-RF	SMB	HML	RMW	CMA
196401	2.24	-0.19	1.59	0.17	1.51
196402	1.54	0.1	2.83	-0.05	0.9
196403	1.41	0.99	3.32	-2.21	3.19
196404	0.1	-1.37	-0.55	-1.27	-1.04
196405	1.42	-0.9	1.98	-0.16	0.14
196406	1.27	-0.26	-0.68	-0.28	-0.15
196407	1.74	0.28	0.68	0.04	1.94
196408	-1.44	0.09	0.09	0.15	0.33
196409	2.69	-0.51	1.65	-0.54	0.61
196410	0.59	0.43	1.14	-0.38	0.43
196411	0	0.61	-1.98	0.62	-0.26
196412	0.03	-0.26	-2.55	1.04	-1.48
196501	3.54	2.7	0.18	0.89	0.1
196502	0.44	3.48	0.22	0.2	-0.65
196503	-1.34	1.79	1.09	-0.36	0.72
196504	3.11	1.2	0.71	0.41	-2.27
196505	-0.77	0.04	-1.63	-0.4	0.5
196506	-5.51	-4.35	0.55	0.21	0.37
196507	1.43	0.87	2.18	-1.37	0.03
196508	2.73	2.81	-0.96	1.9	-0.74
196509	2.86	0.61	-0.1	-0.89	0.8
196510	2.6	2.48	1.52	1.17	-0.65
196511	-0.03	4.68	0.21	-1.01	-0.94
196512	1.01	2.08	1.95	-1.14	-0.53
196601	0.72	3.89	3.49	-2.82	-0.11
196602	-1.21	4.54	0.27	-0.17	-1.49
196603	-2.51	0.94	-2.07	1.3	0.06
196604	2.14	3.45	-0.54	0.39	-0.95
196605	-5.66	-4.7	-1.55	1.64	-1.52

Nota. Muestra de los primeros meses de los rendimientos. Obtenido de la página web https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html.

Resaltamos que las variables incluidas en la investigación son:

- Port: Portafolio de 100 activos.
- SMB: Rendimiento promedio de las nueve carteras de acciones pequeñas menos el rendimiento promedio de las nueve carteras de acciones grandes.
- HML: Rendimiento promedio de las dos carteras de valor menos el rendimiento promedio de las dos carteras de crecimiento.
- RMW: Rendimiento promedio de las dos carteras de rentabilidad operativa sólidas menos el rendimiento promedio de las dos carteras de rentabilidad operativa débil.
- CMA: Rendimiento promedio de las dos carteras de inversión conservadoras menos el rendimiento promedio de las dos carteras de inversión agresivas.
- Mkt-RF: Rendimiento excedente en el mercado

3.2 Técnicas de análisis e instrumentos

Se hará uso del software R para la simulación de los modelos multifactoriales del rendimiento de los activos y factores de riesgo, y cálculos de los estimadores de Fama Macbeth, Shanken y Bai-Zhou.

Generación de muestras de rendimientos de activos y factores de riesgo se realizará a través de simulación estocástica, usando los modelos multifactoriales, presentados en el Capítulo 2. Asimismo, se emplearán los métodos de Bootstrap para desarrollar mejoras de los estimadores de Fama-MacBeth y de Shanken.

Se hará uso de las librerías ggplot2 de Wickham (2016) para la generación de gráficos que nos servirán para la comparación de los resultados y mejor visualización; R Core Team (2023) para la generación de muestras aleatorias provenientes de diversas distribuciones de probabilidad, boot de Canty y Ripley (2022) para aplicación de la metodología de Bootstrap. Adicionalmente, se programará algunos procedimientos propios de los modelos y estimadores estudiados.

También se hará uso del software Julia que permite optimizar cálculos de simulaciones en menor tiempo. Se usará Noris (2023) para importar los datos de Excel, Language (2023) para cálculos de estadística descriptiva, Breloff y others (2024) para la realización de gráficos, Collaborators (2023a) para la elaboración de la estructura DataFrame, Collaborators (2023b) para generar distribuciones de probabilidad, Collaborators (2023c) para realizar regresión, Collaborators (2023d) para la generación de números aleatorios y Collaborators (2023e) para cálculos descriptivos complementario.

3.3 Procedimiento

El procedimiento por realizar para obtener resultados va a ser mencionado a continuación:

1. Se va a realizar la extracción de datos de la página web de French, siendo así nuestra base de datos conformada por un portafolio de 100 rendimientos y 5 factores, medidos en forma mensual desde Julio de 1963 hasta Junio del 2019.

2. Se va a realizar un análisis exploratorio de las diferentes características que poseen los datos, por ejemplo, la estacionariedad de los rendimientos para verificar si se cumple algunos supuestos como la no autocorrelación en el tiempo.
3. Se procede a realizar simulaciones para tamaños de muestra de T (tiempo) y N (tamaño de portafolio) usando para ello los datos obtenidos de French, se usará un solo factor (Rendimiento excedente en el mercado). El algoritmo será el siguiente:

a) Si $N \leq 100$ pasar al paso b) sino continuar leyendo. Se procede a generar los rendimientos que faltan, para ello en cada tiempo hallamos la media de los 100 rendimientos y su varianza, posteriormente se generan rendimientos si $N > 100$ a partir de una muestra aleatoria normal con las medias y varianzas halladas. Matemáticamente, sea $\bar{\mu}_{100,t}$ y $\bar{\sigma}_{100,t}^2$ la media y varianza muestral de los 100 rendimientos en el tiempo t , entonces el i -ésimo portafolio donde $i = 101, \dots, N$ en el tiempo t se genera como a partir de una muestra aleatoria de distribución $N(\bar{\mu}_{100,t}, \bar{\sigma}_{100,t}^2)$.

b) Procedemos a estimar los parámetros del modelo, es decir, α_i , los β_{ij} y σ_{ii}

- Para estimar los β se utiliza el supuesto de la (3) utilizando como estimadores de la esperanza la media de los factores en el tiempo y los valores $\gamma_0 = 0,083333$, $\gamma_1 = 0,66667$ obtenidos de Shanken y Zhou (2007)
- Con los valores obtenidos de β se observa la (2) y se procede a la estimación de α_i como el promedio de la diferencia de los $R_{it} - \beta_{i1} f_{1t} - \dots - \beta_{ik} f_{it}$ en el tiempo. Y la varianza de las mismas diferencias es el estimador de los σ_{ii} (en el caso de Shanken debido a que nos interesa

cuando $N > T$, usaremos la regresión ponderada, caso especial de mínimos cuadrados generalizados, pues en este caso según Bai y Zhou (2015) la matriz de covarianzas como estimador es no invertible).

c) Realizamos la simulación de los rendimientos usando los parámetros estimados, los factores y con diferentes distribuciones de los errores en la

(2). Sea para este caso $D_{1i} = N(0, \sigma_{ii}^2)$, $D_{2i} = \sqrt{\frac{\sigma_{ii}^2}{3}} t(3)$, $D_{3i} =$

$\sqrt{\frac{\sigma_{ii}^2}{1 - \frac{25}{13\pi}}} (SN(0,1,10) - \frac{5}{\sqrt{13\pi}})$ las distribuciones que usaremos, se puede ver

en la Figura 3 sus densidades. Además, se añadirá dos mezclas de distribuciones normales para generar una variable con colas pesadas:

$$A_1 = M_1|Y = 0 \sim N(-0.5,1), A_2 = M_1|Y = 1 \sim N(0.5,1), Y \sim Ber(0.5)$$

Que cumple con $E(M_1) = 0$ y $Var(M_1) = 5/4$.

Además, también se puede generar una variable con asimetría negativa usando las siguientes dos mezclas:

$$B_1 = M_2|Y = 0 \sim N(-0.5,1), B_2 = M_2|Y = 1 \sim N(0.5,0.45^2), Y \sim Ber(0.8)$$

Que cumple con $E(M_2) = 0.522$ y $Var(M_2) = 0.3$. La variable que nos permite ver su generación es en ambos casos $(1 - Y) \times A_1 + Y \times A_2$ y $(1 - Y) \times B_1 + Y \times B_2$ respectivamente. En la Figura 4 se puede observar sus densidades. Por tanto, usaremos para la distribución de los errores

$$D_{4i} = \sqrt{\frac{4}{5}} \sigma_{ii}^2 M_1 \text{ y } D_{5i} = \sqrt{\frac{1}{var(M_2)}} \sigma_{ii}^2 M_2.$$

Figura 4

Distribución de los residuales

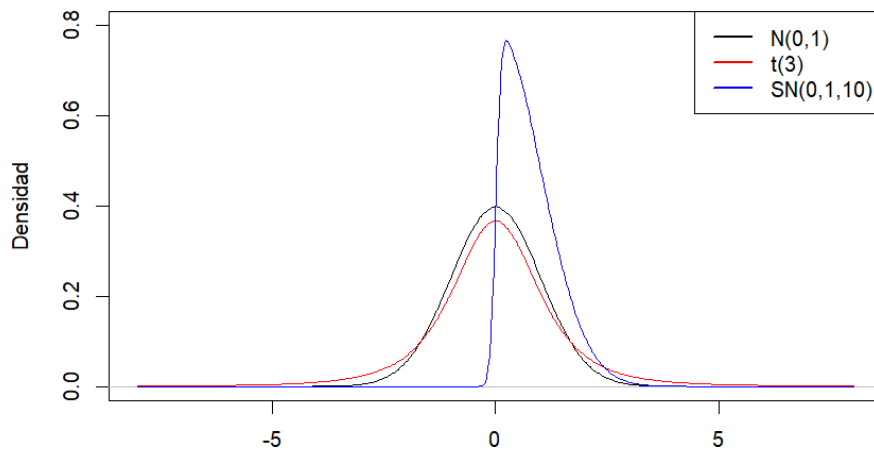
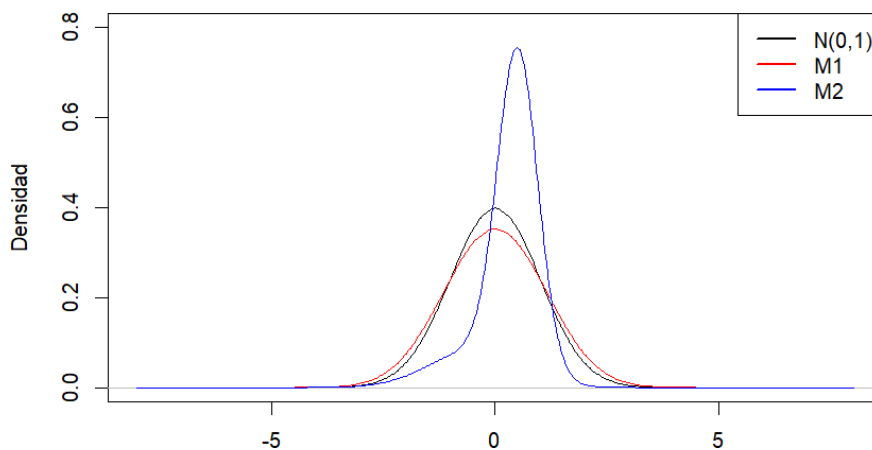


Figura 3

Distribución de los residuales con mezclas



- d) Una vez simulados se procede al cálculo de los estimadores de Fama-Macbeth, Shanken y Bai-Zhou. Estos resultados se almacenan.
- e) Con los estimadores de Fama-Macbeth y Shanken se procede a realizar estimadores Bootstrap de tamaño 200 de muestra; tanto el enfoque paramétrico como no paramétrico. La forma de realizar el Bootstrap para ambos casos se muestra en la Figura 5 y Figura 6. Para estos sea $\hat{D}_{1i} =$

$$N(0, \hat{\sigma}_{ii}^2), \hat{D}_{2i} = \sqrt{\frac{\hat{\sigma}_{ii}^2}{3}} t(3), \hat{D}_{3i} = \sqrt{\frac{\hat{\sigma}_{ii}^2}{1 - \frac{25}{13\pi}}} \left(SN(0,1,10) - \frac{5}{\sqrt{13\pi}} \right), \hat{D}_{4i} = \sqrt{\frac{4}{5}} \hat{\sigma}_{ii}^2 M_1 \text{ y}$$

$$\hat{D}_{5i} = \sqrt{\frac{1}{var(M_2)}} \hat{\sigma}_{ii}^2 M_2. \text{ Estos estimadores se almacenan.}$$

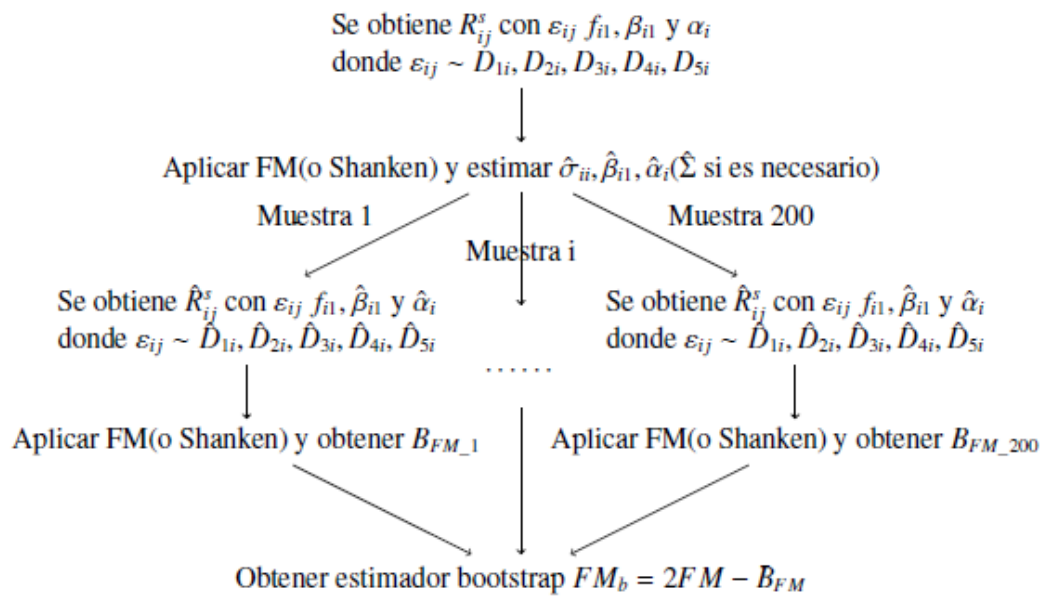
- f) Se repiten los pasos c), d) y e) hasta lograr 1000 simulaciones. Con estos resultados hallamos el sesgo relativo (razón entre el sesgo y el verdadero valor), promedio de estimadores, error cuadrático medio relativo para cada factor y se procede a realizar tablas para su posterior análisis. También para cada caso se construirán intervalos usando percentiles.

El procedimiento anterior se puede realizar para diversos escenarios de T y N . Esos resultados nos permitirán obtener resultados. Se muestra el diagrama de flujo en la Figura 7.

4. Con los resultados obtenidos en el paso anterior se procede a realizar análisis para obtener conclusiones y recomendaciones según sea el caso.

Figura 5

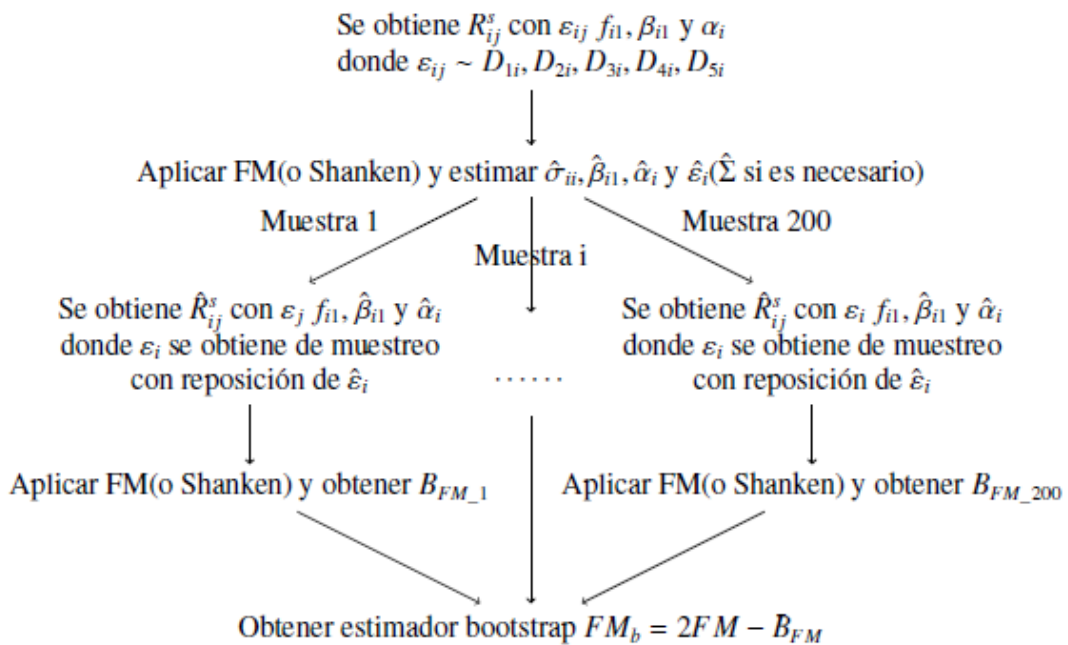
Esquema de Bootstrap paramétrico



Nota. Elaboración propia

Figura 6

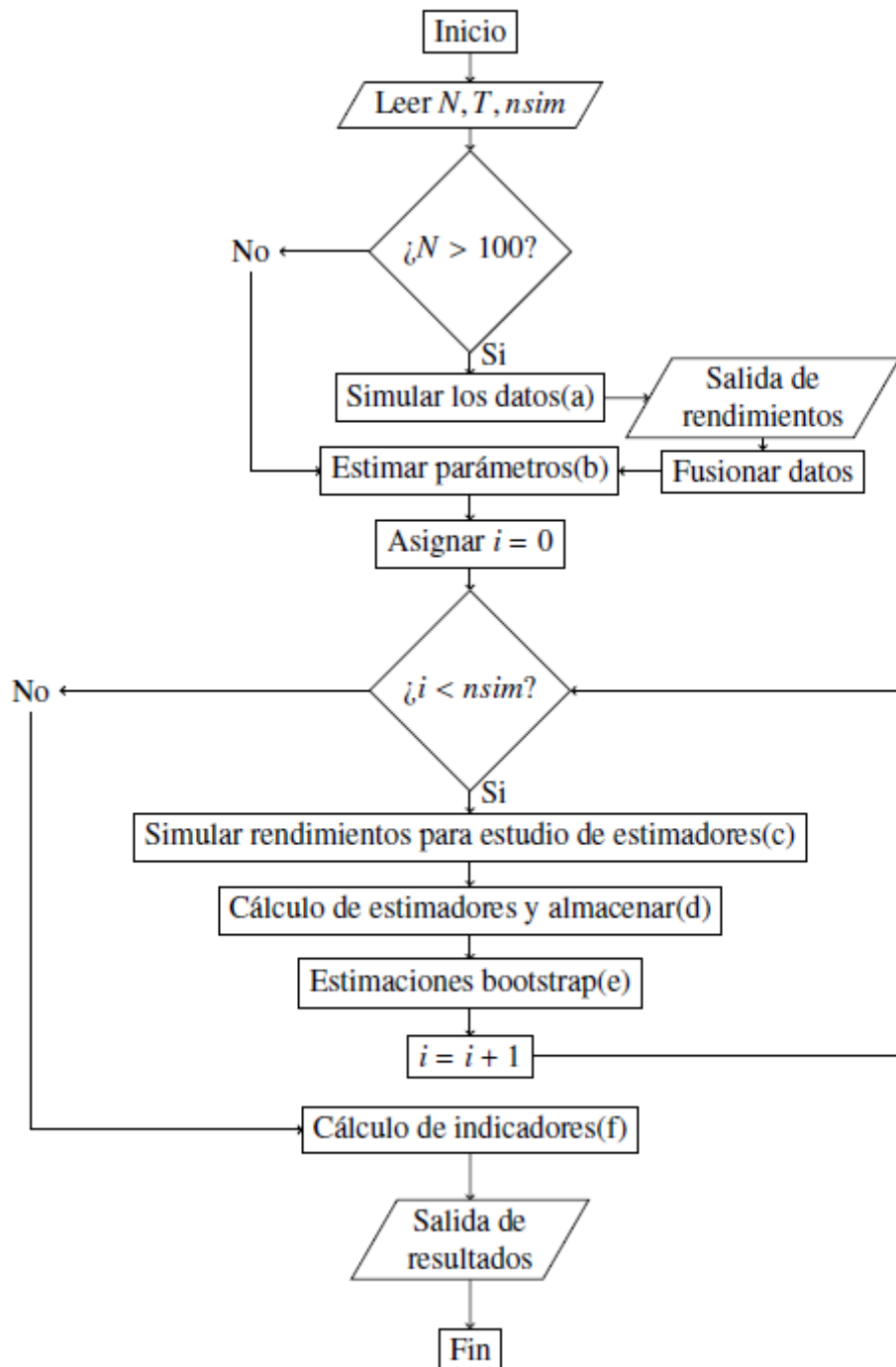
Esquema de Bootstrap no paramétrico



Nota. Elaboración propia

Figura 7

Algoritmo de simulaciones



Nota. Elaboración propia

Capítulo IV. Análisis y discusión de resultados

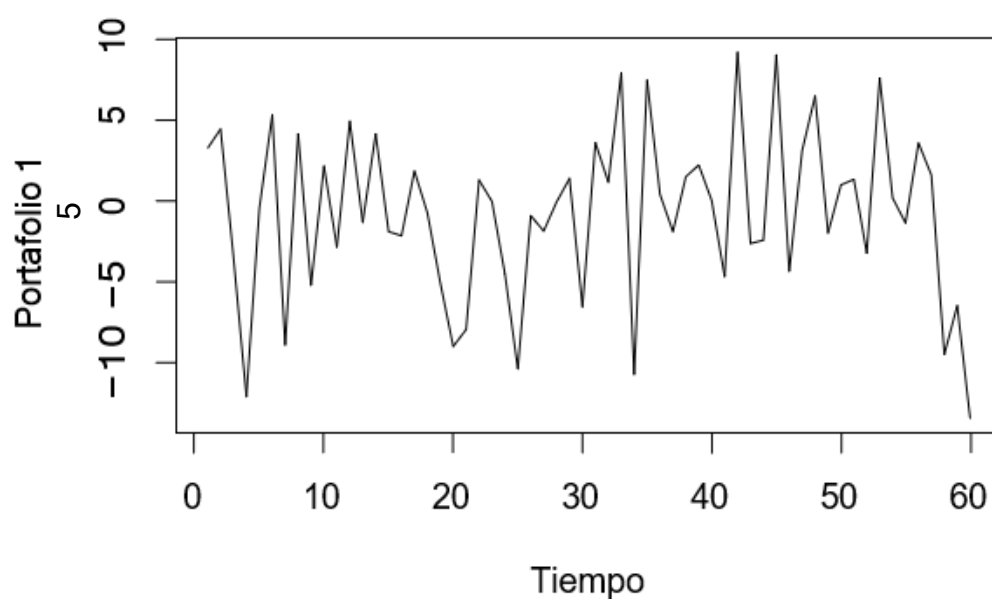
4.1 Descripción de los datos

Dado que tenemos alrededor de 100 portafolios y 5 factores, realizaremos el análisis descriptivo de 3 portafolios y 2 factores. Se realizará en dos ventanas temporales: del año 2009 al 2018, y del año 2014 al 2018.

En el caso del periodo del 2014 al 2018, se puede observar en la Figura 8 que el portafolio 1 es aparentemente estacionario, lo cual se puede contrastar en la Figura 9 donde los correlogramas refuerzan la estacionariedad en este caso.

Figura 8

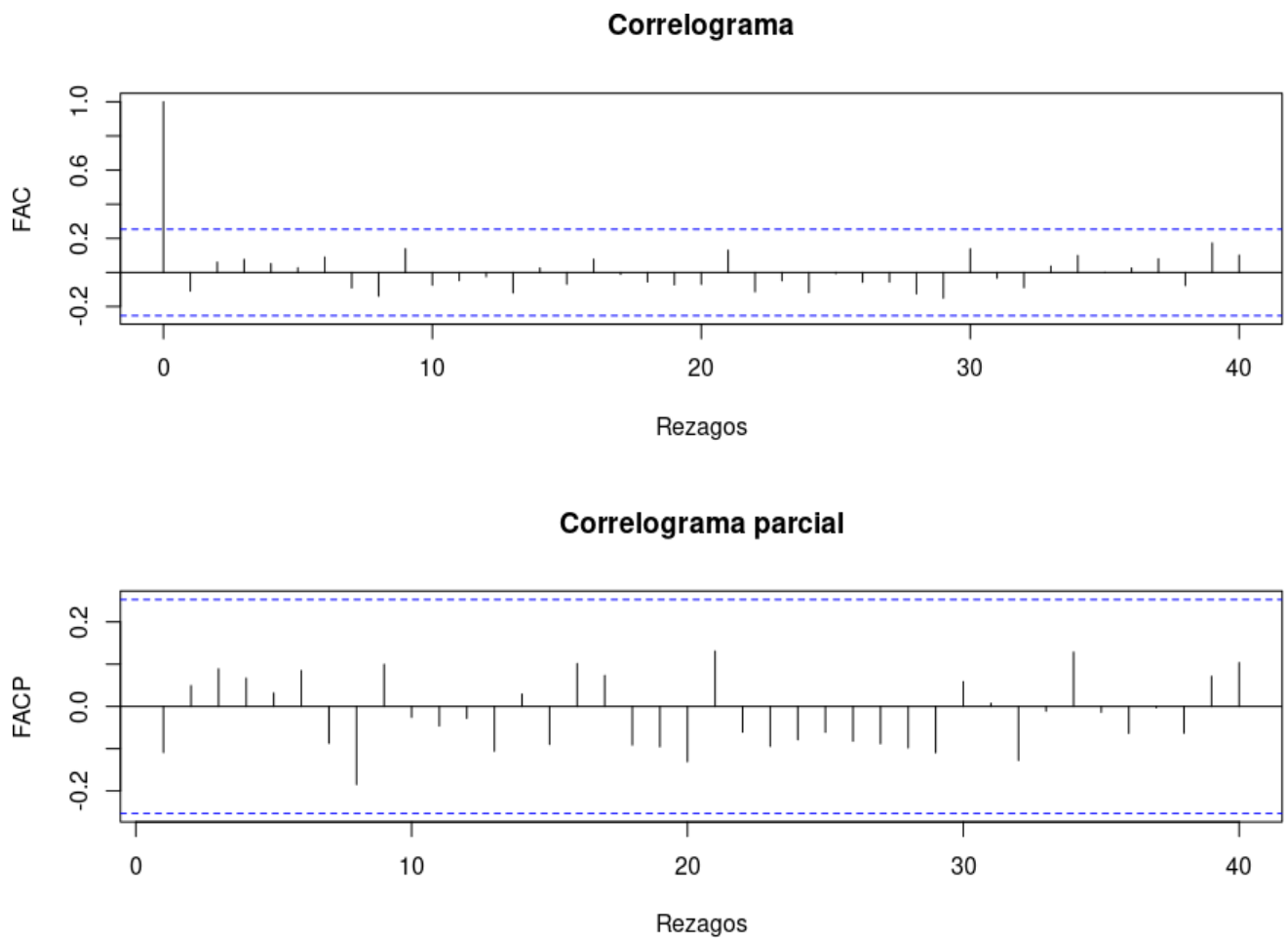
Evolución del portafolio 1 entre los años 2014 y 2018 medidos en forma mensual



Nota. Elaboración propia

Figura 9

Correlogramas parcial y normal respectivamente del portafolio 1 entre los años 2014 y 2018 medidos en forma mensual

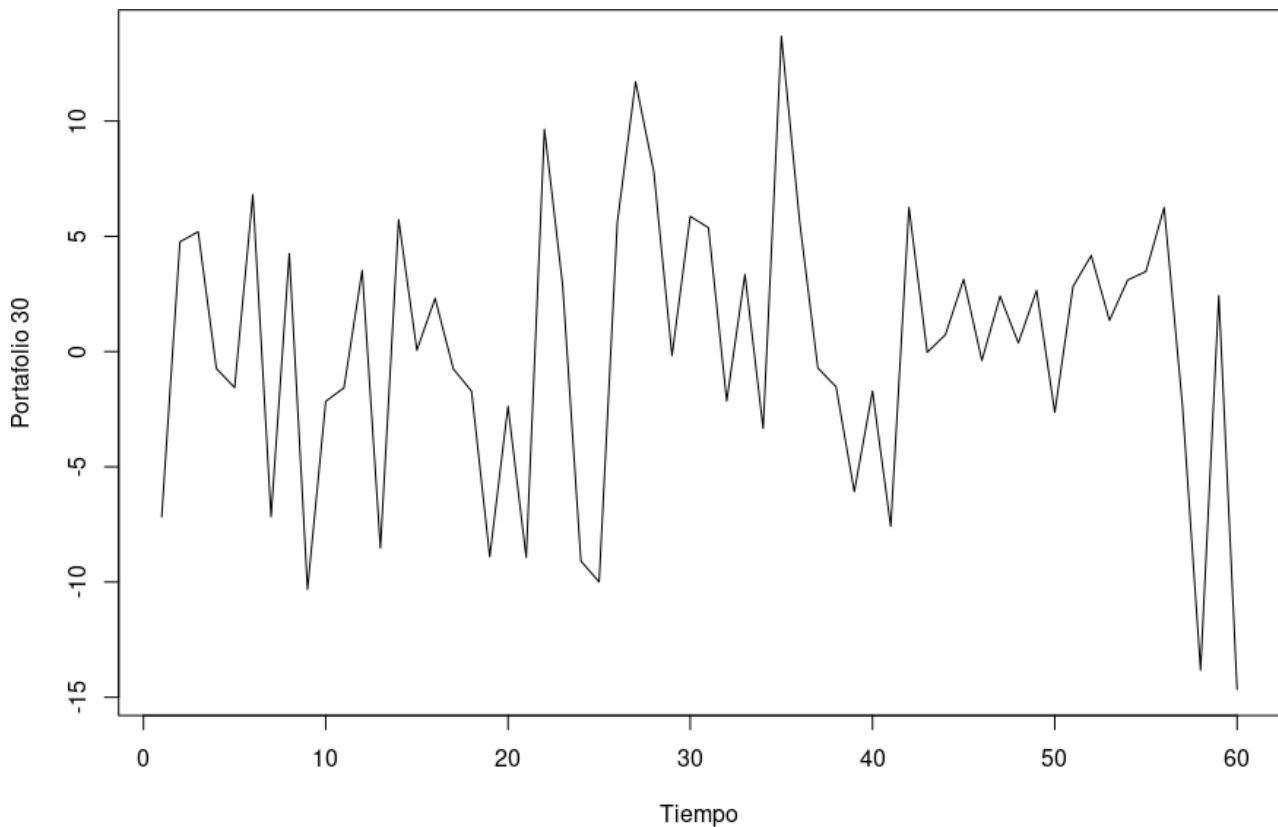


Nota. Elaboración propia

Observemos ahora que pasa con el portafolio 30: en la Figura 10 vemos un comportamiento aparentemente estacionario, para corroborar esto se observa los correlogramas en la Figura 11 donde se observa un comportamiento estacionario.

Figura 10

Evolución del portafolio 30 entre los años 2014 y 2018 medidos en forma mensual

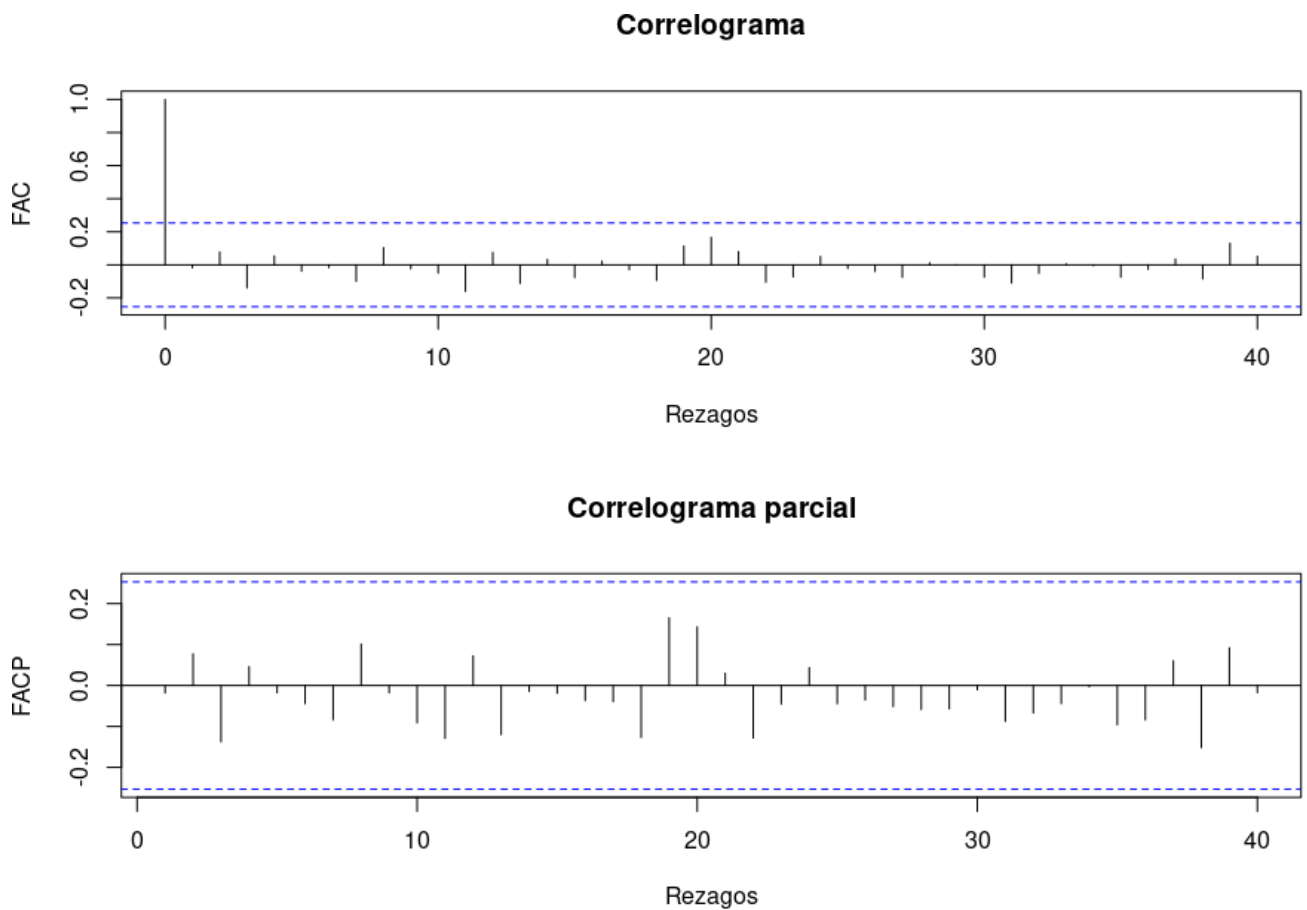


Nota. Elaboración propia

Veamos ahora que pasa con el portafolio 60: en la Figura 12 vemos un comportamiento aparentemente estacionario, para corroborar esto se observa los correlogramas en la Figura 13 donde se observa un comportamiento estacionario.

Figura 11

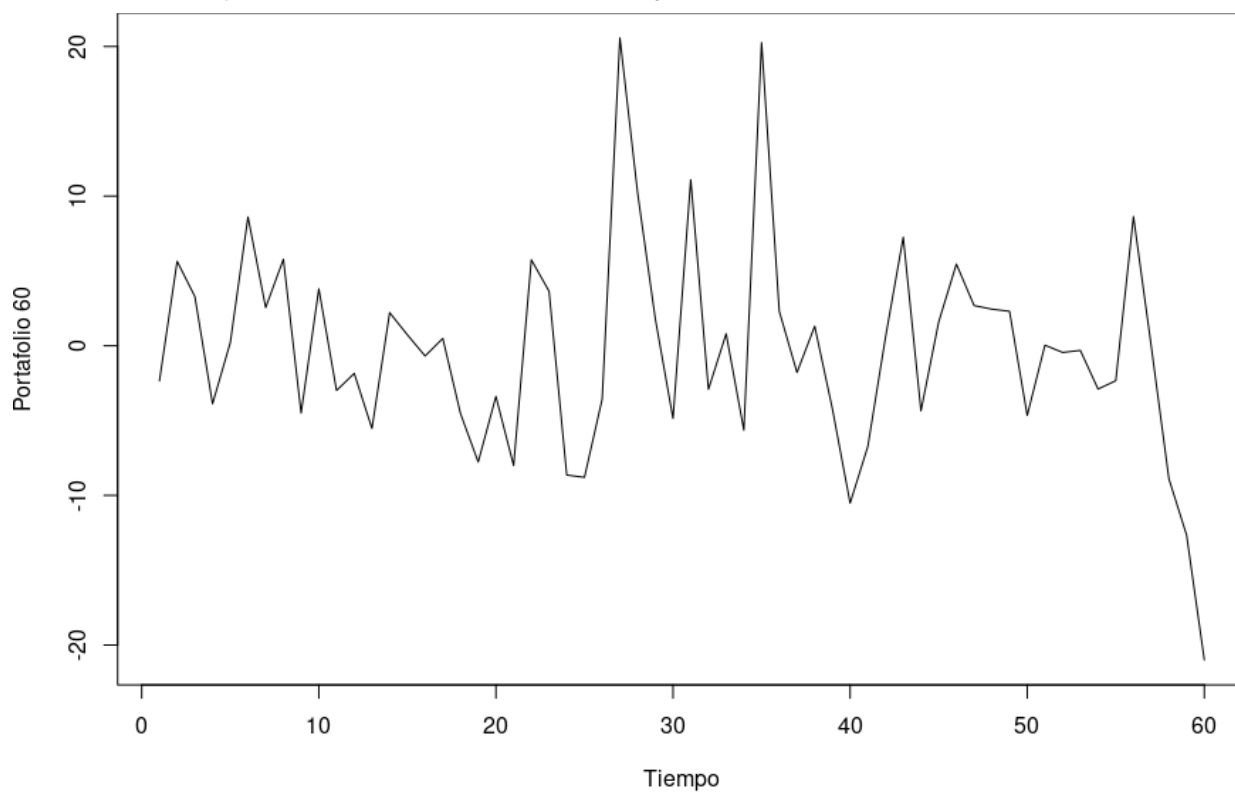
Correlogramas parcial y normal respectivamente del portafolio 30 entre los años 2014 y 2018 medidos en forma mensual



Nota. Elaboración propia

Figura 12

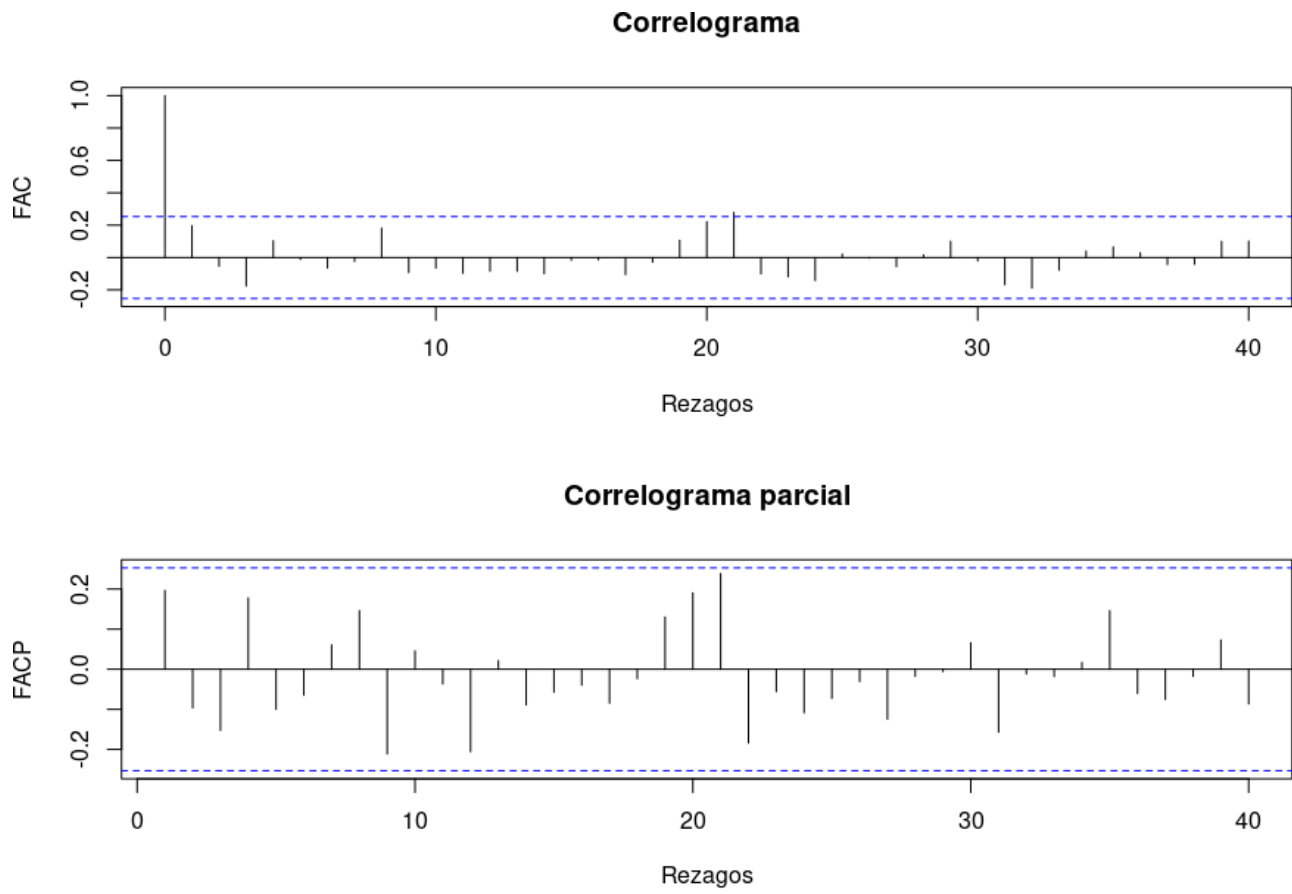
Evolución del portafolio 60 entre los años 2014 y 2018 medidos en forma mensual



Nota. Elaboración propia.

Figura 13

Correlogramas parcial y normal respectivamente del portafolio 60 entre los años 2014 y 2018

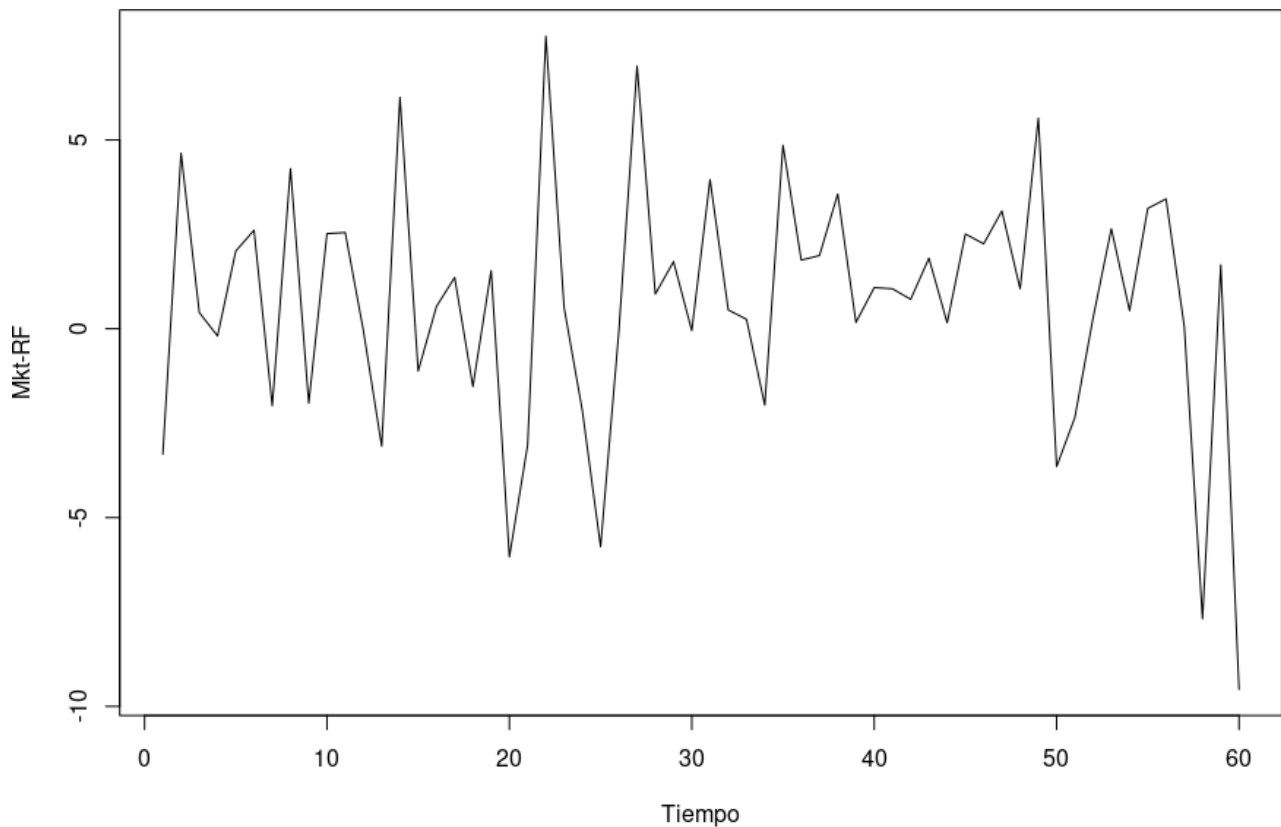


Nota. Elaboración propia.

Veamos ahora que pasa con el factor Mkt-RF: en la Figura 14 vemos un comportamiento aparentemente estacionario, para corroborar esto se observa los correlogramas en la Figura 15 donde se observa un comportamiento estacionario.

Figura 14

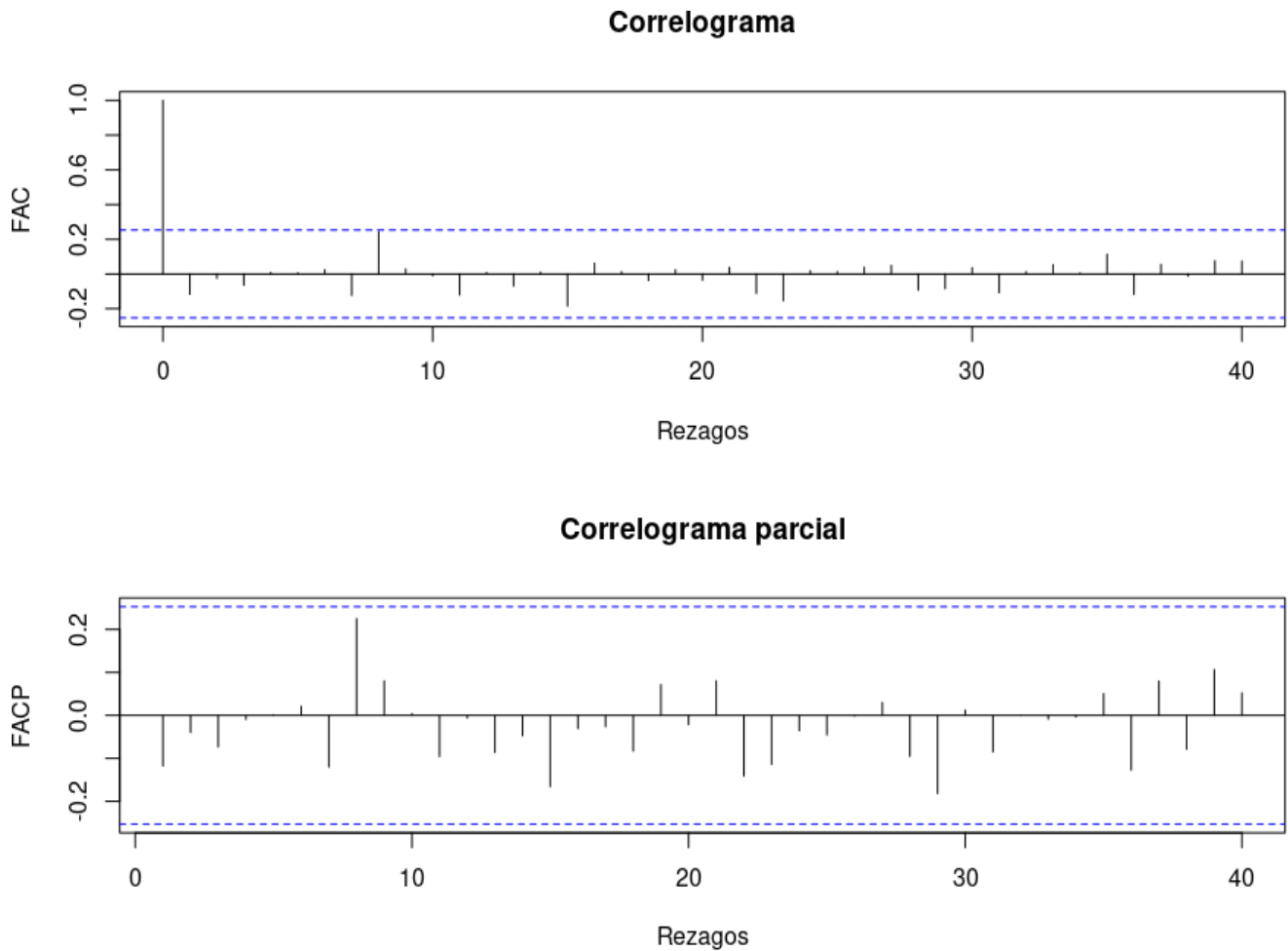
Evolución del factor Mkt-RF entre los años 2014 y 2018 medidos en forma mensual



Nota. Elaboración propia.

Figura 15

Correlogramas parcial y normal respectivamente del factor Mkt-RF entre los años 2014 y 2018 medidos en forma mensual

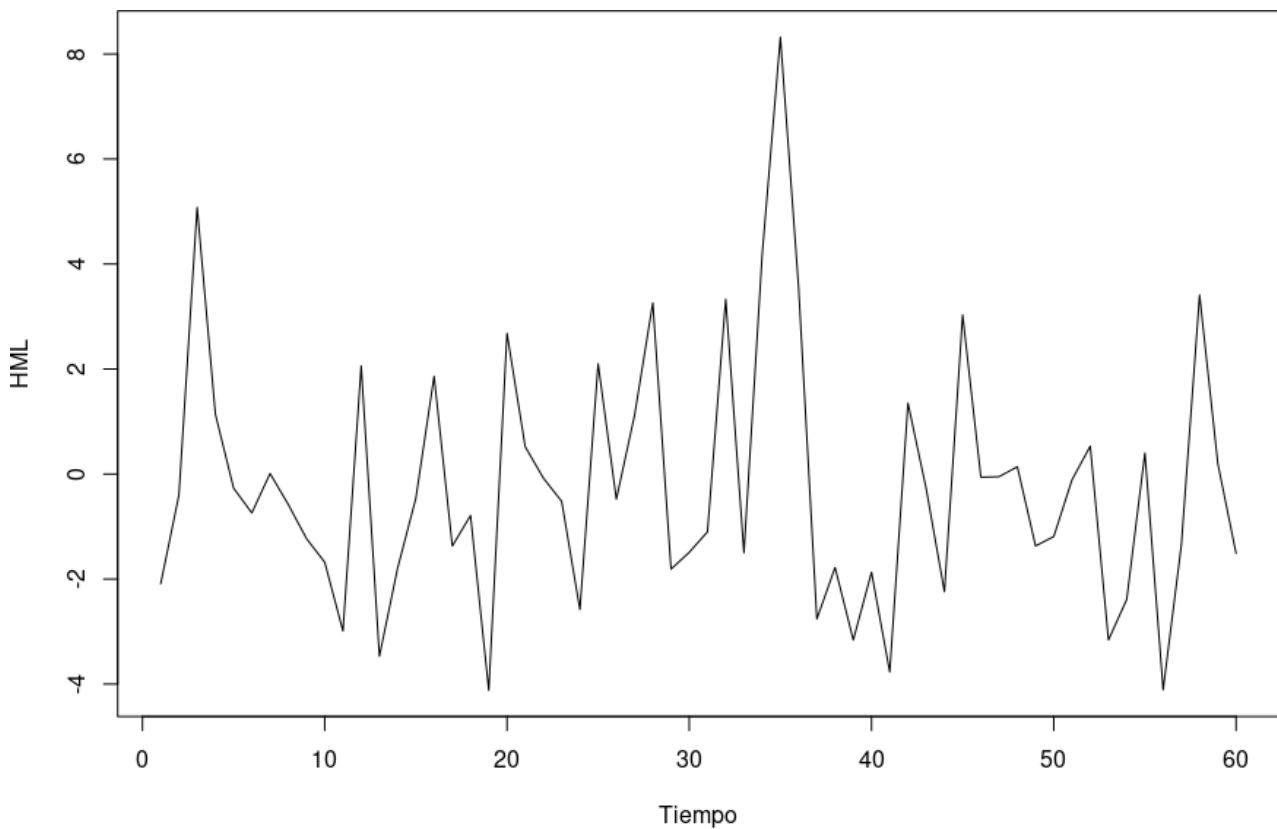


Nota. Elaboración propia.

Veamos ahora que pasa con el factor HML: en la Figura 16 vemos un comportamiento aparentemente no estacionario, para corroborar esto se observa los correlogramas en la Figura 17 donde se observa rezagos significativos, pero de forma leve, pero dentro de los supuestos no figura restricción por parte de los factores.

Figura 16

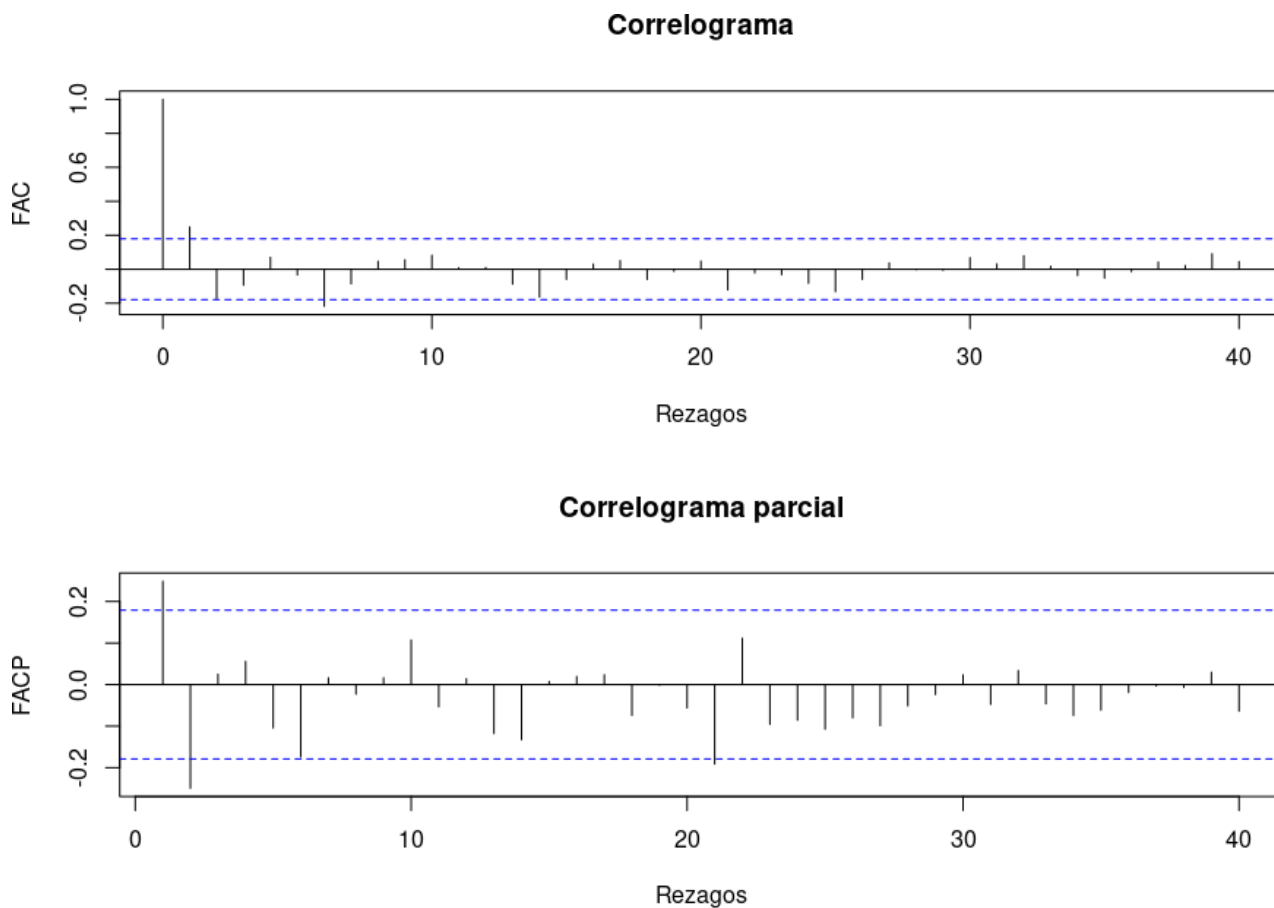
Evolución del factor HML entre los años 2014 y 2018 medidos en forma mensual



Nota. Elaboración propia.

Figura 17

Correlogramas parcial y normal respectivamente del factor HML entre los años 2014 y 2018 medidos en forma mensual

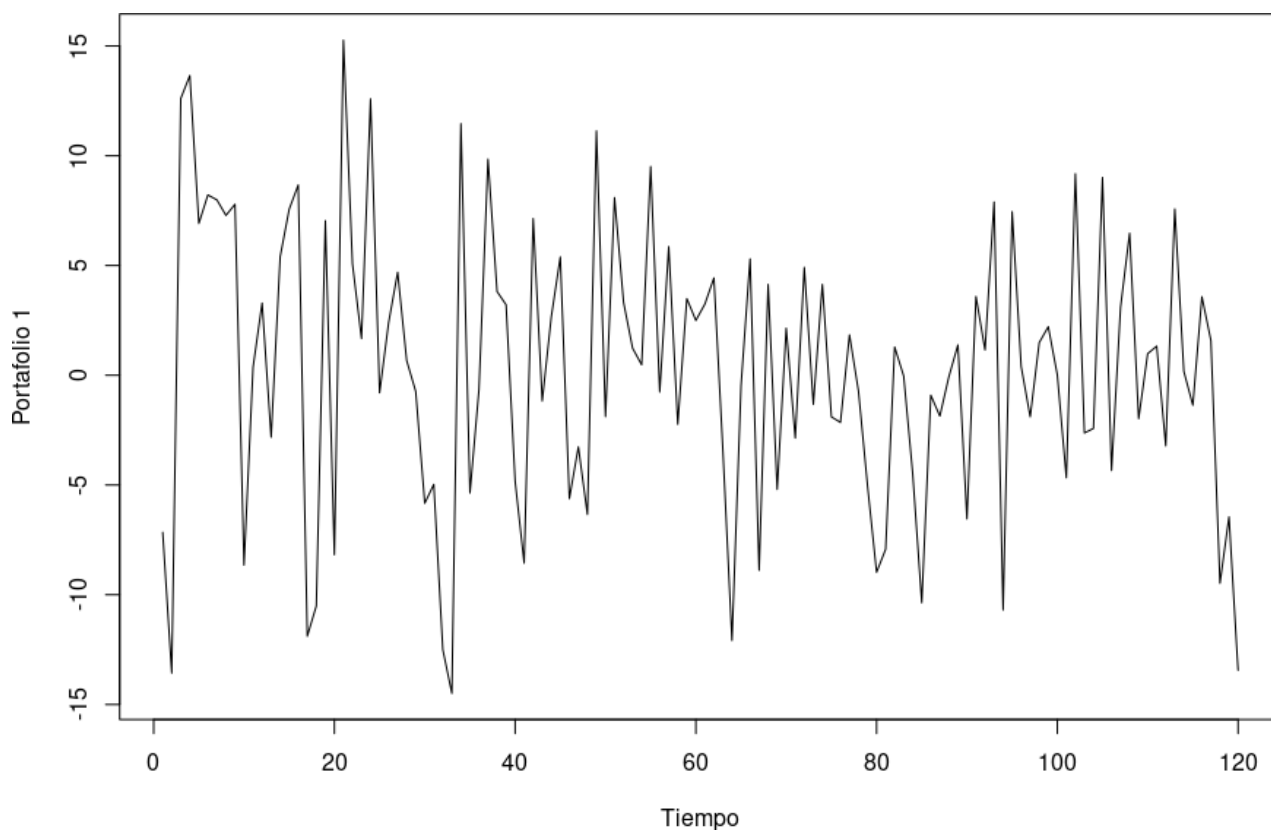


Nota. Elaboración propia.

Veamos ahora que pasa con el portafolio 1: en la Figura 18 vemos un comportamiento aparentemente no estacionario, para corroborar esto se observa los correlogramas en la Figura 19 donde se observa un comportamiento estacionario pues los rezagos son no significativos.

Figura 18

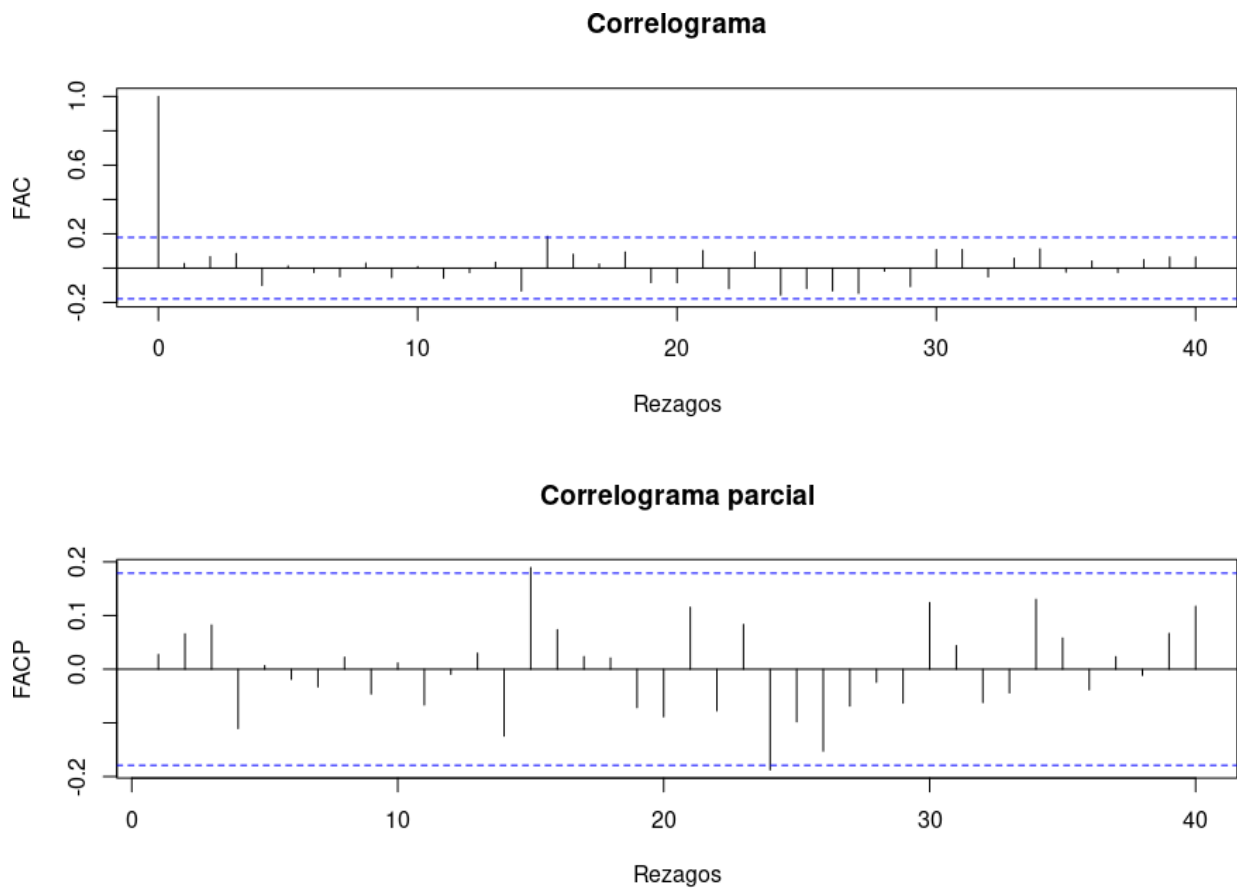
Evolución del portafolio 1 entre los años 2009 y 2018 medidos en forma mensual



Nota. Elaboración propia.

Figura 19

Correlogramas parcial y normal respectivamente del portafolio 1 entre los años 2009 y 2018 medidos en forma mensual

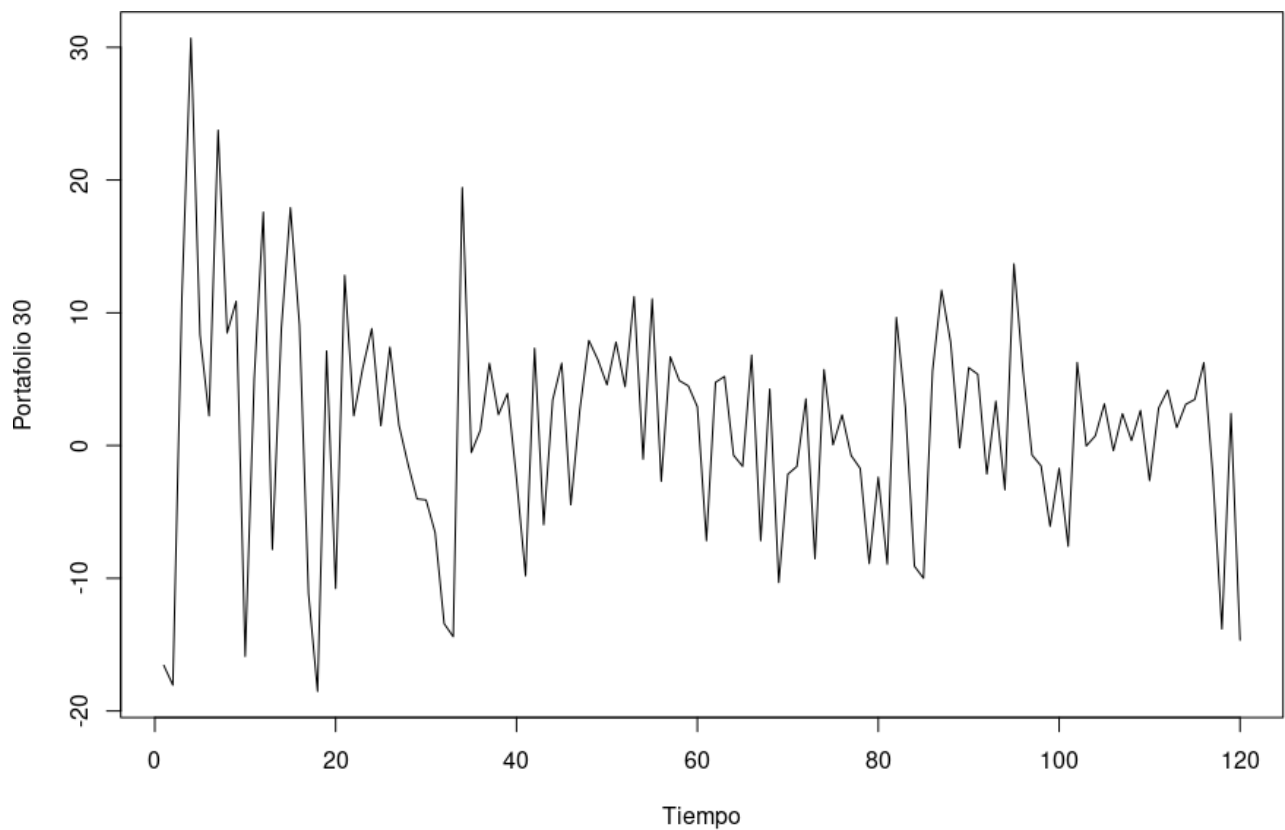


Nota. Elaboración propia.

Veamos ahora que pasa con el portafolio 30: en la Figura 20 vemos un comportamiento aparentemente no estacionario, para corroborar esto se observa los correlogramas en la Figura 21 donde se observa un rezago significativo en ambos correlogramas y debido a que no son los primeros rezagos podemos considerar estos datos adecuados en términos de estacionariedad (es decir estacionarios).

Figura 20

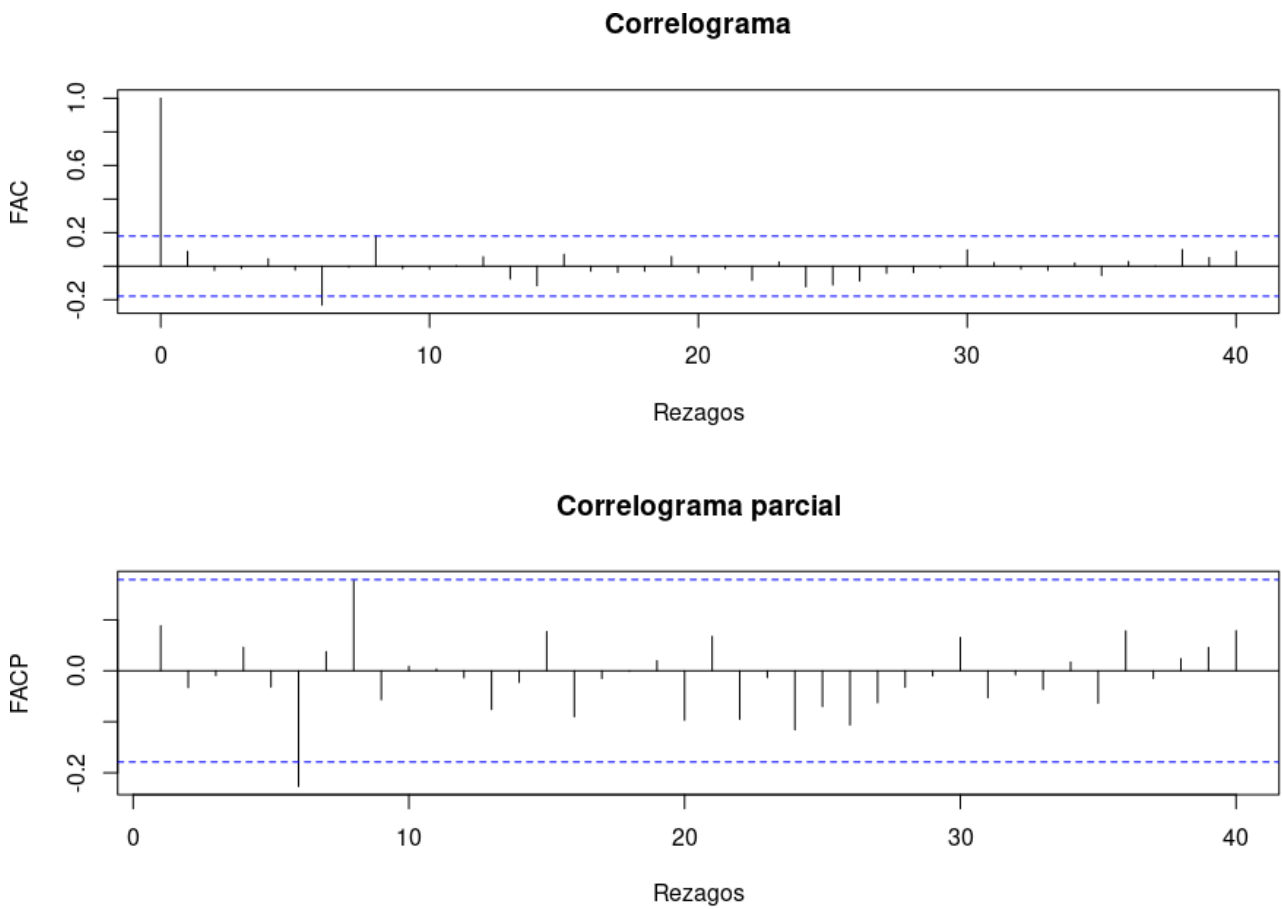
Evolución del portafolio 30 entre los años 2009 y 2018 medidos en forma mensual



Nota. Elaboración propia

Figura 21

Correlogramas parcial y normal respectivamente del portafolio 30 entre los años 2009 y 2018 medidos en forma mensual

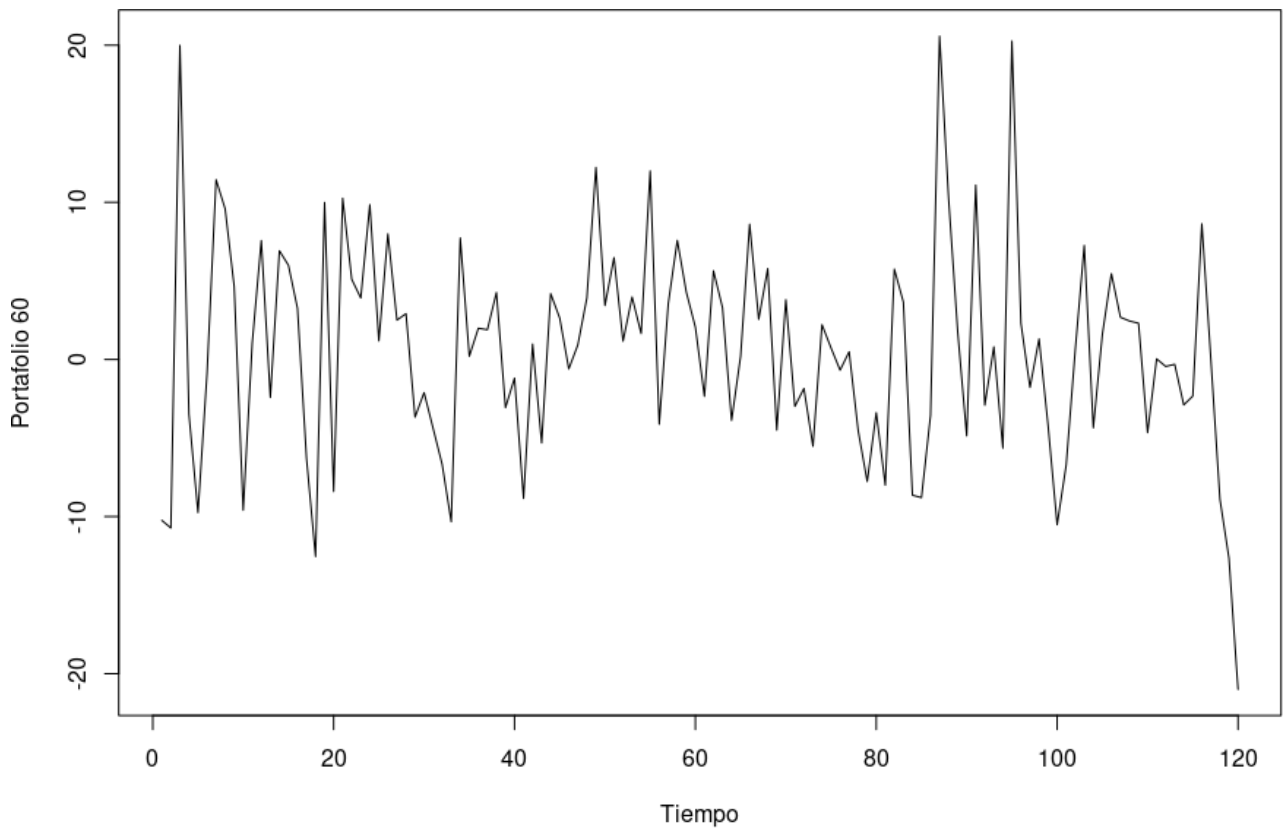


Nota. Elaboración propia

Veamos ahora que pasa con el portafolio 60: en la Figura 22 vemos un comportamiento aparentemente estacionario, para corroborar esto se observa los correlogramas en la Figura 23 donde se observa un comportamiento estacionario.

Figura 22

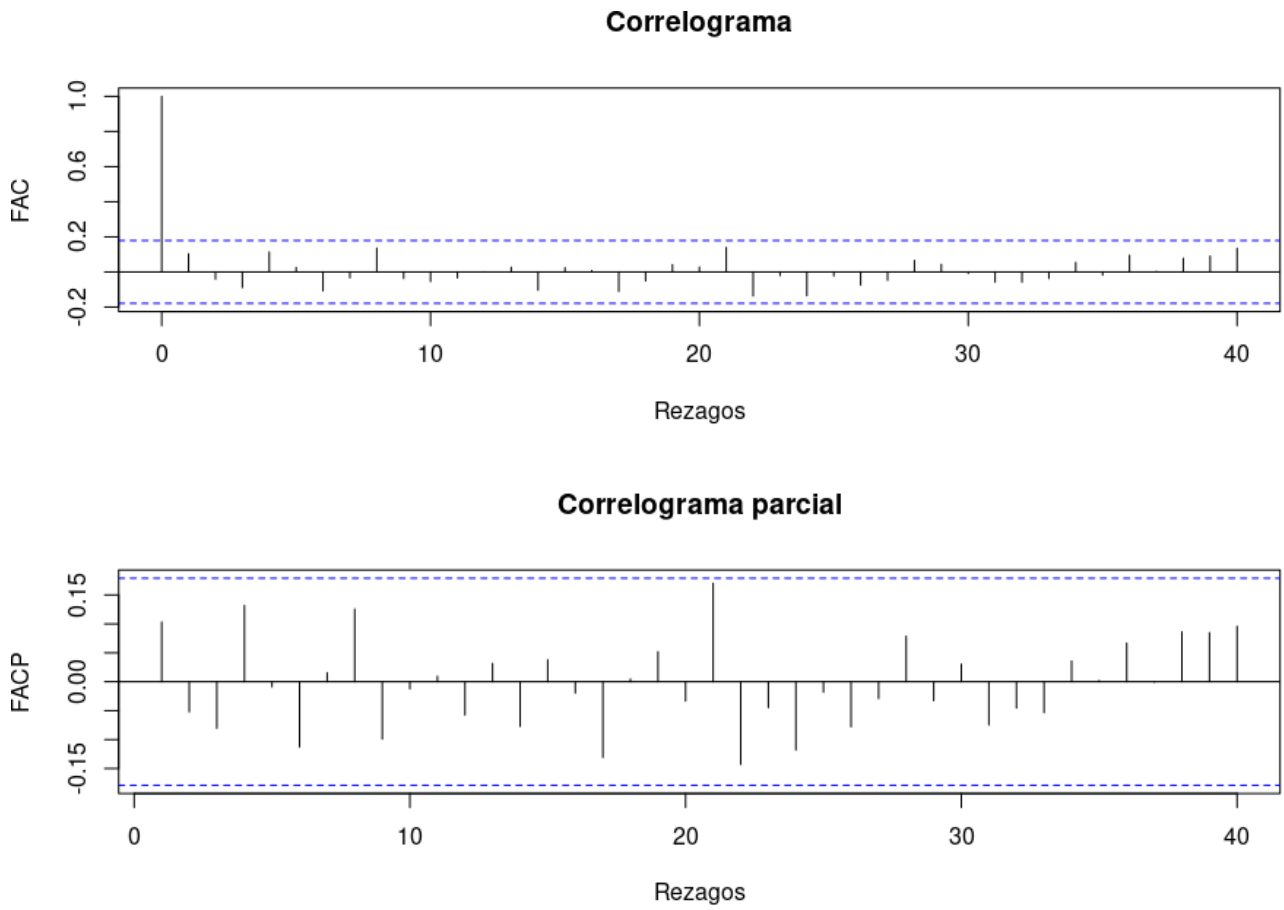
Evolución del portafolio 60 entre los años 2009 y 2018 medidos en forma mensual



Nota. Elaboración propia

Figura 23

Correlogramas parcial y normal respectivamente del portafolio 60 entre los años 2009 y 2018 medidos en forma mensual

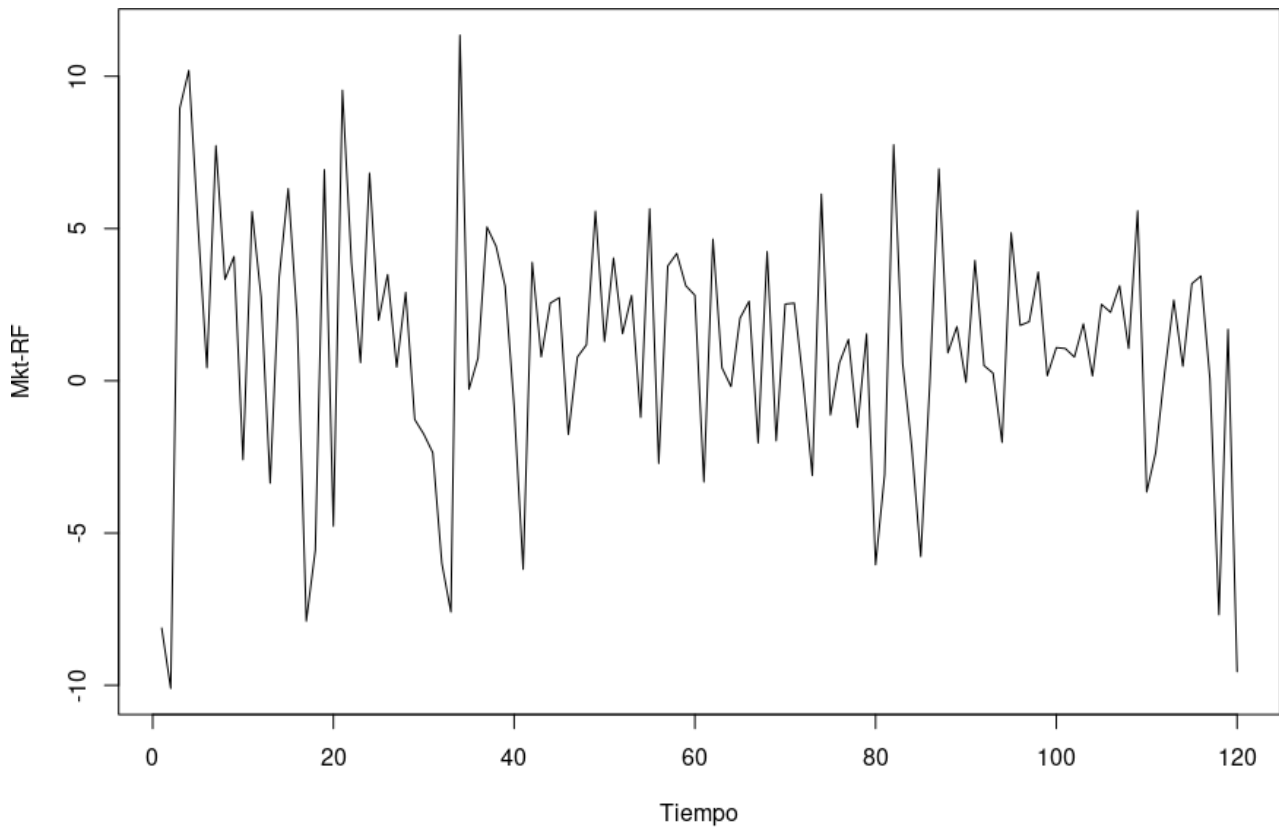


Nota. Elaboración propia

Veamos ahora que pasa con el factor Mkt-RF: en la Figura 24 vemos un comportamiento aparentemente estacionario, para corroborar esto se observa los correlogramas en la Figura 25 donde se observa un comportamiento estacionario.

Figura 24

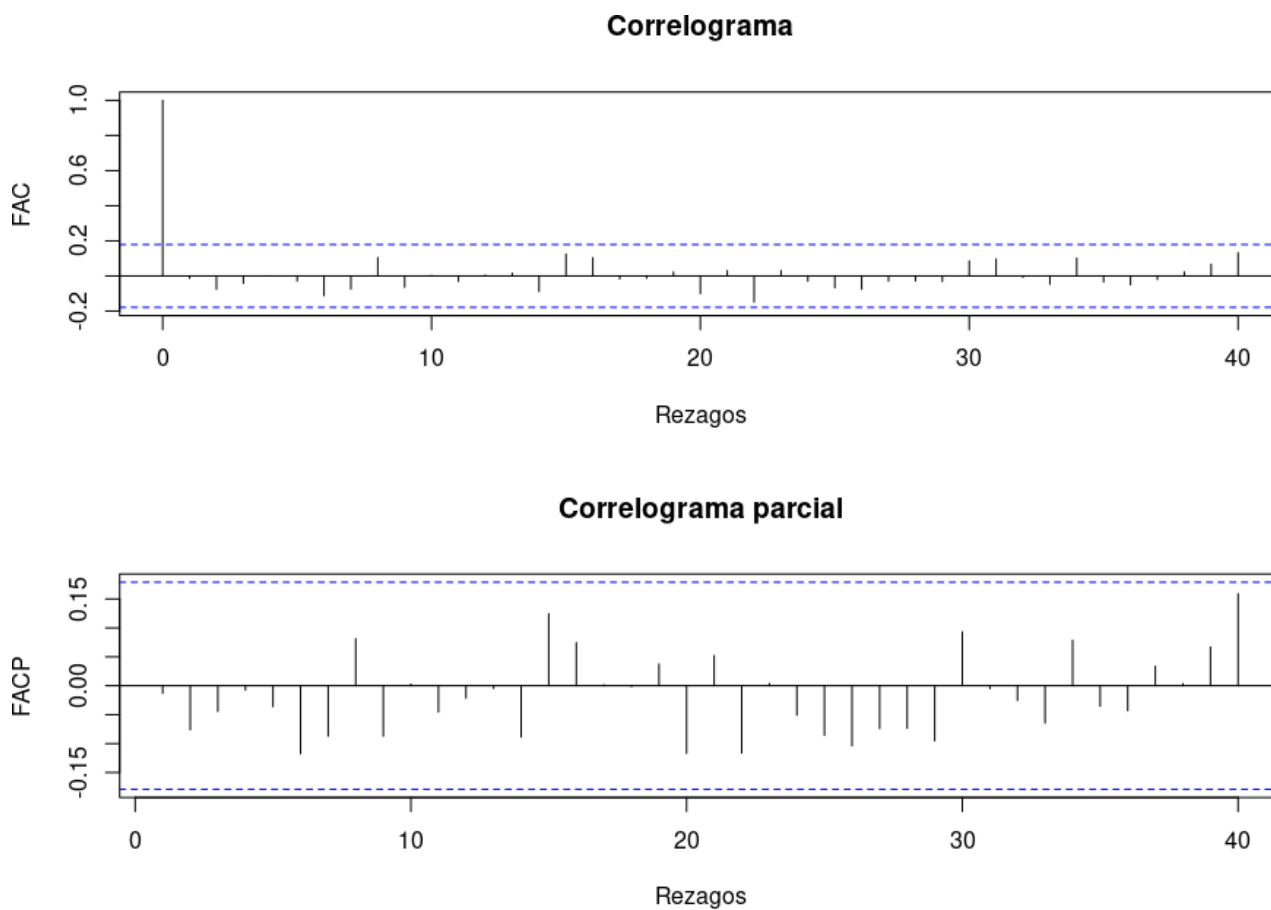
Evolución del factor Mkt-RF entre los años 2009 y 2018 medidos en forma mensual



Nota. Elaboración propia

Figura 25

Correlogramas parcial y normal respectivamente del factor Mkt-RF entre los años 2009 y 2018 medidos en forma mensual

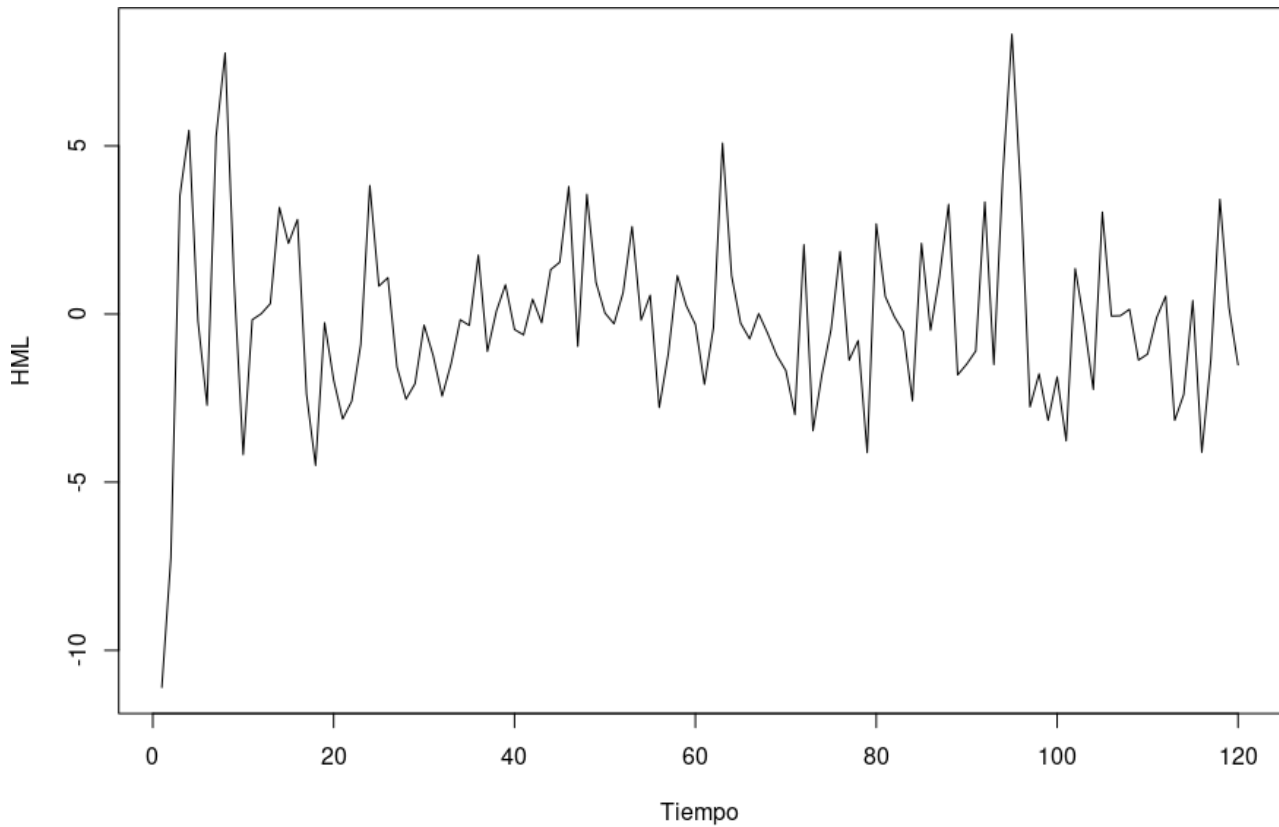


Nota. Elaboración propia

Veamos ahora que pasa con el factor HML: en la Figura 26 vemos un comportamiento aparentemente no estacionario, para corroborar esto se observa los correlogramas en la Figura 27 donde se observa rezagos significativos, sin embargo, en los supuestos no se requiere la estacionariedad de los factores.

Figura 26

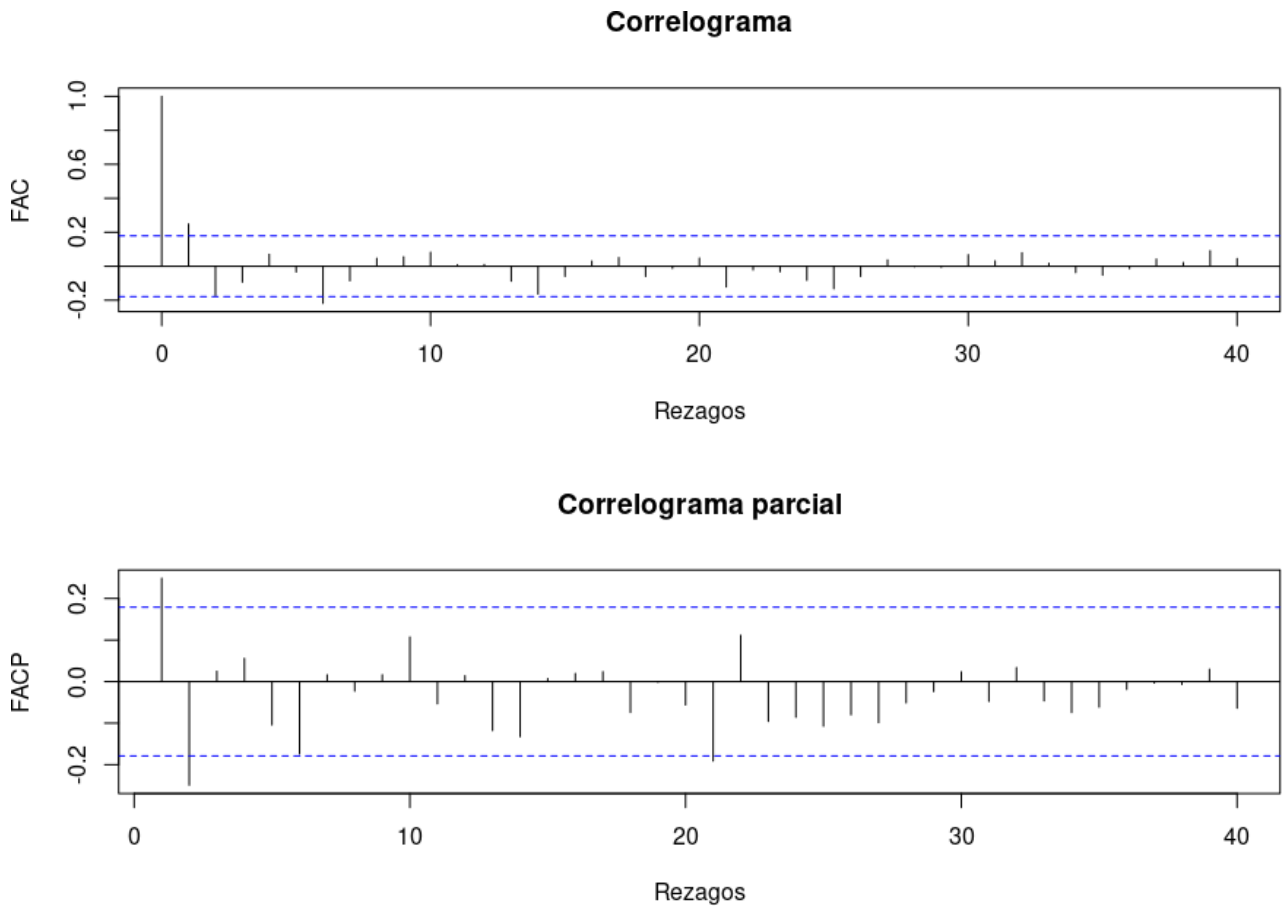
Evolución del factor HML entre los años 2009 y 2018 medidos en forma mensual



Nota. Elaboración propia

Figura 27

Correlogramas parcial y normal respectivamente del factor HML entre los años 2009 y 2018 medidos en forma mensual



Nota. Elaboración propia

4.2 Estimador Fama-Macbeth con Bootstrap paramétrico

4.2.1 T = 60 y N = 100

En la Tabla 1 a través de 1000 simulaciones se obtiene que la aplicación del método Bootstrap al estimador Fama-Macbeth es similar a los resultados obtenidos por Bai-Zhou, es decir no logra reducir el error cuadrático medio lo que significa que no logra obtener mejoras, esto se ve a lo largo de las diferentes distribuciones excepto para el caso de la distribución asimétrica donde si se ve mejoras tanto en el sesgo como error cuadrático medio. En el caso de aplicar Bootstrap al estimador de Bai-Zhou se obtiene peores resultados a los estimadores anteriores. Mientras que en la Tabla 2 se observa que el estimador Bai-Zhou con Bootstrap provee de intervalos de confianza con mayor variabilidad.

Tabla 1

Resultados obtenidos con T = 60 y N = 100 con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM	SR	8.1305 %	8.1203 %	13.7157 %	7.8537 %	9.2692 %
	ME	0.61247	0.6125	0.5752	0.6143	0.6049
	ECMR	17.9987 %	18.3631 %	19.6205 %	17.3802 %	17.6287 %
BZ	SR	0.7791 %	0.7951 %	7.0909 %	0.4535 %	1.9832 %
	ME	0.6615	0.6614	0.6194	0.6636	0.6534
	ECMR	18.1498 %	19.4347 %	18.325 %	17.5395 %	17.4034 %
FM-B	SR	1.1276 %	1.2534 %	2.8459 %	0.7222 %	2.2245 %
	ME	0.6592	0.6583	0.6477	0.6619	0.6518
	ECMR	18.1076 %	20.1674 %	17.4087 %	17.5476 %	17.4764 %
BZ-B	SR	7.3406 %	7.2963 %	5.2986 %	7.8103 %	6.1687 %
	ME	0.7156	0.7153	0.7020	0.7187	0.7078
	ECMR	20.6248 %	24.8825 %	19.2736 %	20.1911 %	19.6988 %

Nota. FM=Fama-Macbeth, BZ=Bai-Zhou, FM-B: FM Bootstrap, BZ-B: BZ Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

4.2.2 T = 120 y N = 100

En la Tabla 3 a través de 1000 simulaciones se obtiene que la aplicación del método Bootstrap al estimador Fama-Macbeth produce mejores resultados que el estimador de Bai-Zhou en los casos de distribución de colas pesadas, mezcla 1 y

asimétrica, en los otros casos son similares o no logran mejoras.

Tabla 2

Intervalos de confianza Bootstrap al 95 % de confianza con $T = 60$ y $N = 100$ con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM-B	2.5	0.3725	0.3547	0.3816	0.4003	0.3643
	97.5	0.9706	0.9510	0.9129	0.9601	0.9293
BZ-B	2.5	0.4042	0.4055	0.4117	0.4325	0.3935
	97.5	1.0568	1.0447	0.9895	1.0391	1.0126

Nota. FM-B: FM Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia.

Y en el caso de aplicar Bootstrap al estimador de Bai-Zhou, observamos que solo se obtiene mejoras en el caso de la distribución asimétrica. Mientras que en la Tabla 4 se observa que en todos los casos los intervalos poseen una variabilidad demasiado alta.

Tabla 3

Resultados obtenidos con $T = 120$ y $N = 100$ con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM	SR	0.8985 %	0.0329 %	12.153 %	0.0795 %	3.9745 %
	ME	0.6607	0.6665	0.5857	0.6672	0.6402
	ECMR	45.2846 %	48.8727 %	45.2063 %	46.247 %	44.9284 %
BZ	SR	1.1728 %	1.7536 %	10.4391 %	2.1523 %	1.9132 %
	ME	0.6745	0.6784	0.5971	0.6810	0.6539
	ECMR	46.2946 %	52.0663 %	45.7598 %	47.2798 %	45.9255 %
FM-B	SR	0.5106 %	0.2683 %	2.5343 %	0.4427 %	3.4913 %
	ME	0.6633	0.6685	0.6498	0.6696	0.6434
	ECMR	45.8475 %	51.7340 %	44.3693 %	46.8429 %	45.5081 %
BZ-B	SR	1.5291 %	1.7782 %	0.5362 %	2.4817 %	1.5334 %
	ME	0.6769	0.6785	0.6631	0.6832	0.6564
	ECMR	46.866 %	58.6999 %	45.2119 %	47.8801 %	46.5009 %

Nota. FM=Fama-Macbeth, BZ=Bai-Zhou, FM-B: FM Bootstrap, BZ-B: BZ Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 4

Intervalos de confianza Bootstrap al 95 % de confianza con $T = 120$ y $N = 100$ con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM-B	2.5	-0.0824	-0.0338	-0.0650	-0.1218	-0.0814
	97.5	1.3725	1.4505	1.3384	1.4099	1.3591
BZ-B	2.5	-0.0836	-0.0344	-0.0666	-0.1247	-0.0827
	97.5	1.4057	1.4865	1.3685	1.4370	1.3908

Nota. FM-B: FM Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia.

4.2.3 T = 60 y N = 300

En la Tabla 5, a través de 1000 simulaciones se observa que el estimador Bootstrap de Fama-Macbeth es mejor que el estimador de Bai-Zhou solo en el caso de distribución asimétrica. Mientras que la aplicación de Bootstrap a Bai-Zhou produce peores resultados en todos los casos. Se observa que en la Tabla 6 que los intervalos de confianza no poseen una variabilidad excesiva a comparación de lo obtenido en las Tabla 4 y Tabla 2, observándose que el método Bootstrap a Bai-Zhou posee la mayor variabilidad en todas las distribuciones.

Tabla 5

Resultados obtenidos con $T = 60$ y $N = 300$ con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM	SR	8.1319 %	8.1410 %	11.522 %	8.1693 %	7.9004 %
	ME	0.6125	0.6124	0.5899	0.6122	0.614
	ECMR	10.797 %	11.7003 %	12.5514 %	10.8686 %	10.6762 %
BZ	SR	1.1760 %	1.1973 %	4.9858 %	1.0978 %	0.9934 %
	ME	0.6588	0.6587	0.6334	0.6594	0.6600
	ECMR	9.2523 %	10.718 %	9.8165 %	9.3137 %	9.2409 %
FM-B	SR	1.2262 %	1.4139 %	1.8148 %	1.1738 %	1.0853 %
	ME	0.6585	0.6572	0.6546	0.6588	0.6594
	ECMR	9.2545 %	11.08 %	8.9888 %	9.3190 %	9.2704 %
BZ-B	SR	6.7881 %	6.6196 %	6.1336 %	6.9875 %	6.8598 %
	ME	0.7119	0.7108	0.7076	0.7133	0.7124
	ECMR	11.4374 %	13.5999 %	10.8363 %	11.5804 %	11.5038 %

Nota. FM=Fama-Macbeth, BZ=Bai-Zhou, FM-B: FM Bootstrap, BZ-B: BZ Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 6

Intervalos de confianza Bootstrap al 95 % de confianza con $T = 60$ y $N = 300$ con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM-B	2.5	0.5103	0.5023	0.5099	0.5104	0.5091
	97.5	0.8058	0.8098	0.7929	0.8092	0.7972
BZ-B	2.5	0.5527	0.5446	0.5520	0.5515	0.5502
	97.5	0.8713	0.8780	0.8577	0.8770	0.8638

Nota. FM-B: FM Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia.

4.2.4 $T = 120$ y $N = 300$

En la Tabla 7, a través de 1000 simulaciones se observa que la aplicación de Bootstrap al estimador de Fama-Macbeth es mejor al estimador de Bai-Zhou solo en el caso de tener distribución asimétrica. Mientras que en el caso de aplicar Bootstrap al estimador de Bai-Zhou no produce mejoras. Mientras que en la Tabla 8 se tiene una variabilidad considerable, pero observamos que es menor a lo observado en la Tabla 4.

Tabla 7

Resultados obtenidos con $T = 120$ y $N = 300$ con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM	SR	2.4036 %	1.8657 %	6.1879 %	3.0535 %	3.571 %
	ME	0.6506	0.6542	0.6254	0.6463	0.6429
	ECMR	27.7675 %	25.9034 %	27.7792 %	27.0743 %	26.981 %
BZ	SR	0.4860 %	0.8543 %	3.4853 %	0.2499 %	0.7908 %
	ME	0.6699	0.6724	0.6434	0.6650	0.6614
	ECMR	28.5713 %	26.6392 %	28.2542 %	27.7842 %	27.6786 %
FM-B	SR	0.0096 %	0.5261 %	0.0368 %	0.7837 %	1.2849 %
	ME	0.6666	0.6702	0.6664	0.6614	0.6581
	ECMR	28.5502 %	26.6952 %	27.9247 %	27.7555 %	27.7029 %
BZ-B	SR	3.0131 %	3.3866 %	3.0214 %	2.1460 %	1.6018 %
	ME	0.6868	0.6892	0.6868	0.6810	0.6773
	ECMR	29.5765 %	27.6618 %	28.9016 %	28.6754 %	28.6096 %

Nota. FM=Fama-Macbeth, BZ=Bai-Zhou, FM-B: FM Bootstrap, BZ-B: BZ Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 8

Intervalos de confianza Bootstrap al 95 % de confianza con $T = 120$ y $N = 300$ con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM-B	2.5	0.2021	0.2191	0.2105	0.2035	0.2152
	97.5	1.1122	1.0808	1.0930	1.1136	1.1271
BZ-B	2.5	0.2072	0.2244	0.2152	0.2085	0.2208
	97.5	1.1480	1.1177	1.1278	1.1457	1.1569

Nota. FM-B: FM Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia.

4.3 Estimador Fama-Macbeth con Bootstrap no paramétrico

4.3.1 T = 60 y N = 100

En la Tabla 9, a través de 1000 se observa que la aplicación de Bootstrap al estimador de Fama- Macbeth produce resultados similares al estimador de Bai-Zhou con lo cual no obtenemos mejoras en las diferentes distribuciones, solo logra mejorar el sesgo en el caso de la mezcla 2 pero el error cuadrático medio no es menor al del estimador de Bai-Zhou. Mientras que la aplicación de Bootstrap en el caso de Bai-Zhou solo produce mejoras en el sesgo en el caso de distribución asimétrica y mezcla 2, mientras que el error cuadrático medio no mejora. Mientras que en la Tabla 10 se observa que el estimador Bai-Zhou con Bootstrap provee de intervalos de confianza con mayor variabilidad.

Tabla 9

Resultados obtenidos con T = 60 y N = 100 con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM	SR	8.7561 %	8.5605 %	13.0859 %	8.0621 %	7.5996 %
	ME	0.6083	0.6096	0.5794	0.6129	0.616
	ECMR	17.5144 %	20.7387 %	19.3038 %	18.0528 %	17.3599 %
BZ	SR	1.4467 %	1.3673 %	6.3908 %	0.6949 %	4.2788 %
	ME	0.6570	0.6576	0.6241	0.662	0.6381
	ECMR	17.3633 %	21.7609 %	18.1133 %	18.2356 %	17.1977 %
FM-B	SR	1.8043 %	1.8650 %	6.7396 %	1.1604 %	0.4981 %
	ME	0.6546	0.6542	0.6217	0.6589	0.6633
	ECMR	17.4276 %	22.4349 %	18.231 %	18.1947 %	17.6123 %
BZ-B	SR	6.7061 %	6.5091 %	1.0283 %	7.4091 %	0.0749 %
	ME	0.7114	0.7101	0.6735	0.7161	0.6662
	ECMR	19.7275 %	25.8021 %	18.8958 %	20.7494 %	17.6756 %

Nota. FM=Fama-Macbeth, BZ=Bai-Zhou, FM-B: FM Bootstrap, BZ-B: BZ Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 10

Intervalos de confianza Bootstrap al 95 % de confianza con $T = 60$ y $N = 100$ con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM-B	2.5	0.3772	0.3474	0.3408	0.374	0.3804
	97.5	0.9246	0.9432	0.9046	0.9678	0.9385
BZ-B	2.5	0.4085	0.3750	0.3700	0.4074	0.3819
	97.5	1.0047	1.0271	0.9811	1.0514	0.944

Nota. FM-B: FM Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia.

4.3.2 T = 120 y N = 100

En la Tabla 11, a través de 1000 simulaciones se observa que la aplicación de Bootstrap al estimador de Fama-Macbeth es mejor que el estimador de Bai-Zhou solo en el caso de tener distribución Normal. En el caso de aplicar Bootstrap al estimador de Bai-Zhou se obtiene mejoras solo en el caso de mezcla 2 donde tiene reducción de sesgo y error cuadrático medio bajo. Mientras que en la Tabla 12 se observa que los intervalos de confianza poseen gran variabilidad.

Tabla 11

Resultados obtenidos con $T = 120$ y $N = 100$ con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM	SR	2.9333 %	2.5084 %	7.4572 %	1.3850 %	1.5397 %
	ME	0.6862	0.6499	0.6170	0.6574	0.6769
	ECMR	45.0316 %	44.1558 %	43.8916 %	45.7539 %	45.9828 %
BZ	SR	5.1078 %	0.5325 %	5.6553 %	0.6810 %	2.0649 %
	ME	0.7007	0.6631	0.6290	0.6712	0.6804
	ECMR	46.175 %	45.521 %	44.5644 %	46.7488 %	46.2575 %
FM-B	SR	3.4651 %	2.1074 %	7.1339 %	0.8548 %	2.0398 %
	ME	0.6898	0.6526	0.6191	0.661	0.6803
	ECMR	45.5615 %	45.3324 %	44.3121 %	46.2081 %	46.5599 %
BZ-B	SR	5.6238 %	0.1384 %	5.2958 %	1.1984 %	0.8075 %
	ME	0.7042	0.6657	0.6314	0.6747	0.6721
	ECMR	46.7207 %	46.6614 %	44.9994 %	47.2203 %	45.8645 %

Nota. FM=Fama-Macbeth, BZ=Bai-Zhou, FM-B: FM Bootstrap, BZ-B: BZ Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 12

Intervalos de confianza Bootstrap al 95 % de confianza con $T = 120$ y $N = 100$ con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM-B	2.5	-0.0340	-0.0828	-0.0655	-0.0627	-0.0008
	97.5	1.4117	1.3401	1.3324	1.4225	1.4899
BZ-B	2.5	-0.0349	-0.0841	-0.0667	-0.0642	-0.0004
	97.5	1.4501	1.3720	1.3566	1.4548	1.4737

Nota. FM-B: FM Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia.

4.3.3 $T = 60$ y $N = 300$

En la Tabla 13, a través de 1000 simulaciones se observa que la aplicación de Bootstrap al estimador de Fama-Macbeth obtiene similares resultados al estimador de Bai-Zhou en todos los casos menos en el caso de mezcla 2 donde si se obtiene mejoras respecto al sesgo y error cuadrático medio. Mientras que la aplicación de Bootstrap al estimador de Bai-Zhou se obtiene mejoras en el sesgo en el caso de distribución asimétrica pero no en el error cuadrático medio (son similares) y en la mezcla 2 se obtiene mejoras tanto en términos de sesgo como error cuadrático medio. Se observa que en la Tabla 14 que los intervalos de confianza no poseen una variabilidad excesiva a comparación de lo obtenido en las Tabla 12 y Tabla 10, observándose que el método Bootstrap a Bai-Zhou posee la mayor variabilidad en todas las distribuciones.

Tabla 13

Resultados obtenidos con $T = 60$ y $N = 300$ con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM	SR	7.2321 %	8.3939 %	12.5048 %	7.7061 %	8.3922 %
	ME	0.6185	0.6107	0.5833	0.6153	0.6107
	ECMR	9.8681 %	11.5173 %	13.2565 %	10.4725 %	10.2682 %
BZ	SR	0.7537 %	1.1121 %	5.9548 %	0.6756 %	5.1829 %
	ME	0.6616	0.6593	0.6270	0.6622	0.6321
	ECMR	8.5142 %	10.1341 %	10.3341 %	9.0602 %	8.9953 %
FM-B	SR	0.921 %	1.3657 %	6.1611 %	0.8188 %	1.6873 %
	ME	0.6605	0.6576	0.6256	0.6612	0.6554
	ECMR	8.5303 %	10.3001 %	10.4271 %	9.0874 %	8.4128 %
BZ-B	SR	6.5521 %	7.1505 %	1.4472 %	7.3665 %	1.0194 %
	ME	0.7104	0.7143	0.6763	0.7158	0.6599
	ECMR	10.6276 %	12.6971 %	9.9879 %	11.5484 %	8.397 %

Nota. FM=Fama-Macbeth, BZ=Bai-Zhou, FM-B: FM Bootstrap, BZ-B: BZ Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 14

Intervalos de confianza Bootstrap al 95 % de confianza con $T = 60$ y $N = 300$ con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM-B	2.5	0.5204	0.4942	0.4792	0.5168	0.5259
	97.5	0.791	0.8215	0.7681	0.7985	0.7917
BZ-B	2.5	0.5585	0.5360	0.5189	0.5592	0.5290
	97.5	0.8507	0.8927	0.8311	0.8673	0.7970

Nota. FM-B: FM Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia.

4.3.4 $T = 120$ y $N = 300$

En la Tabla 15 a través de 1000 simulaciones se observa que la aplicación de Bootstrap al estimador de Fama-Macbeth obtiene resultados similares o peores al estimador de Bai-Zhou en todos los casos excepto en el sesgo en casos como la distribución normal y mezcla (pero no error cuadrático medio). En el caso de aplicar Bootstrap al estimador de Bai-Zhou, vemos que solo se obtiene mejoras en el sesgo, pero no error cuadrático medio en el caso de distribución asimétrica. Mientras que en la Tabla 16 se tiene una variabilidad considerable, pero observamos que es menor a lo observado en la Tabla 12.

Tabla 15

Resultados obtenidos con $T = 120$ y $N = 300$ con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM	SR	2.2802 %	1.4819 %	6.2915 %	2.4137 %	3.8179 %
	ME	0.6515	0.6568	0.6247	0.6506	0.6412
	ECMR	26.0973 %	28.7828 %	26.5697 %	25.4543 %	25.9946 %
BZ	SR	0.5524 %	1.4258 %	3.6965 %	0.2671 %	3.3879 %
	ME	0.6704	0.6762	0.6420	0.6685	0.6441
	ECMR	26.8126 %	29.7534 %	26.9585 %	26.1032 %	26.0779 %
FM-B	SR	0.0024 %	0.9060 %	4.301 %	0.1790 %	1.6643 %
	ME	0.6667	0.6727	0.6380	0.6655	0.6556
	ECMR	26.8285 %	29.8991 %	27.1114 %	26.0774 %	26.6599 %
BZ-B	SR	2.9836 %	3.9767 %	1.5656 %	2.6408 %	3.4145 %
	ME	0.6866	0.6932	0.6562	0.6843	0.6439
	ECMR	27.7677 %	31.0267 %	27.6816 %	26.9338 %	26.2499 %

Nota. FM=Fama-Macbeth, BZ=Bai-Zhou, FM-B: FM Bootstrap, BZ-B: BZ Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia

Tabla 16

Intervalos de confianza Bootstrap al 95 % de confianza con $T = 120$ y $N = 300$ con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
FM-B	2.5	0.2573	0.1737	0.1891	0.2282	0.2338
	97.5	1.0956	1.1463	1.0501	1.0623	1.0746
BZ-B	2.5	0.2643	0.1795	0.1939	0.2357	0.2301
	97.5	1.1372	1.1843	1.0837	1.0922	1.0574

Nota. FM-B: FM Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia.

4.4 Estimador Shanken con Bootstrap paramétrico

4.4.1 T = 60 y N = 100

En la Tabla 17, a través de 1000 simulaciones se obtiene que la aplicación del método Bootstrap al estimador Shanken es similar o peor a los resultados obtenidos por Bai-Zhou excepto en el caso de la mezcla 2 donde si se obtiene mejoras en el sesgo y error cuadrático medio; mientras que la aplicación de Bootstrap al estimador de Bai-Zhou no produce mejoras. Mientras que en la Tabla 18 se observa que el estimador Bai-Zhou con Bootstrap provee de intervalos de confianza con mayor variabilidad, y con respecto a los resultados de la Tabla 2, estos intervalos poseen menor variabilidad.

Tabla 17

Resultados obtenidos con T = 60 y N = 100 con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH	SR	8.1683 %	7.1142 %	8.4081 %	7.9531 %	13.6744 %
	ME	0.6122	0.6192	0.6106	0.6136	0.5755
	ECMR	13.3977 %	12.0557 %	13.6946 %	13.5833 %	16.579 %
BZ	SR	1.1054 %	1.2732 %	1.6898 %	0.8488 %	7.2661 %
	ME	0.6593	0.6582	0.6554	0.6610	0.6182
	ECMR	12.5996 %	11.3177 %	12.8293 %	12.9178 %	14.4939 %
SH-B	SR	1.3297 %	2.5222 %	1.6265 %	1.0423 %	2.9563 %
	ME	0.6578	0.6499	0.6558	0.6597	0.6470
	ECMR	12.6133 %	11.3412 %	12.7774 %	12.9565 %	13.562 %
BZ-B	SR	6.9828 %	5.1152 %	6.5534 %	7.3183 %	5.0574 %
	ME	0.7132	0.7008	0.7104	0.7155	0.7004
	ECMR	14.8395 %	12.7927 %	14.8223 %	15.3047 %	15.1206 %

Nota. SH=Shanken, BZ=Bai-Zhou, FM-B: FM Bootstrap, SH-B: SH Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 18

Intervalos de confianza Bootstrap al 95 % de confianza con T = 60 y N = 100 con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH-B	2.5	0.467	0.4655	0.4507	0.4425	0.4412
	97.5	0.8684	0.8255	0.8632	0.8637	0.8728
BZ-B	2.5	0.5065	0.5001	0.4840	0.4801	0.4804
	97.5	0.9451	0.8914	0.9335	0.939	0.9429

Nota. SH-B: SH Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia.

4.4.2 T = 120 y N = 100

En la Tabla 19, a través de 1000 simulaciones se obtiene que la aplicación del método Bootstrap al estimador Shanken es similar a los resultados obtenidos por Bai-Zhou, solo obteniendo mejoras en el caso de distribución mezcla 2 en el sesgo. Mientras que la aplicación de Bootstrap al caso de Bai-Zhou solo produce peores resultados en todos los casos. Mientras que en la Tabla 20 se observa que en todos los casos los intervalos poseen una variabilidad baja en comparación de Tabla 4.

Tabla 19

Resultados obtenidos con T = 120 y N = 100 con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH	SR	5.9594 %	4.8427 %	4.6575 %	5.1298 %	4.0228 %
	ME	0.6269	0.6344	0.6356	0.6325	0.6399
	ECMR	13.1225 %	12.2045 %	12.3966 %	13.3143 %	13.0472 %
BZ	SR	0.5504 %	0.128 %	0.5683 %	0.3308 %	1.3948 %
	ME	0.663	0.6658	0.6705	0.6689	0.6760
	ECMR	12.9229 %	12.1357 %	12.4746 %	13.4007 %	13.4428 %
SH-B	SR	0.7134 %	0.9208 %	1.4326 %	0.1915 %	0.5999 %
	ME	0.6619	0.6605	0.6571	0.6679	0.6627
	ECMR	12.9057 %	12.0625 %	12.4363 %	13.3908 %	13.4248 %
BZ-B	SR	5.373 %	4.8973 %	4.5578 %	6.3337 %	5.4859 %
	ME	0.7025	0.6993	0.6971	0.7089	0.7032
	ECMR	14.4044 %	13.3839 %	13.6763 %	15.1623 %	14.989 %

Nota. SH=Shanken, BZ=Bai-Zhou, FM-B: FM Bootstrap, SH-B: SH Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 20

Intervalos de confianza Bootstrap al 95 % de confianza con T = 120 y N = 100 con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH-B	2.5	0.4596	0.4685	0.4619	0.4466	0.4470
	97.5	0.8655	0.8490	0.8481	0.8838	0.8802
BZ-B	2.5	0.4879	0.4975	0.4916	0.475	0.4753
	97.5	0.9176	0.8995	0.9044	0.9380	0.9346

Nota. SH-B: SH Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia.

4.4.3 T = 60 y N = 300

En la Tabla 21 a través de 1000 simulaciones se obtiene que la aplicación del

método Bootstrap al estimador Shanken es similar a los resultados obtenidos por Bai-Zhou, esto se ve a lo largo de las diferentes distribuciones obteniendo solo mejoras en el caso de la mezcla 2 tanto en términos de sesgo como error cuadrático medio. En el caso de aplicar Bootstrap al estimador de Bai-Zhou se obtiene peores resultados a los estimadores anteriores. Mientras que en la Tabla 22 se observa que los intervalos de confianza no poseen gran variabilidad y es ligeramente menor a lo observado en la Tabla 6

Tabla 21

Resultados obtenidos con $T = 60$ y $N = 300$ con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH	SR	9.1036 %	6.7438 %	9.0820 %	9.0938 %	12.2134 %
	ME	0.606	0.6217	0.6061	0.6060	0.5852
	ECMR	10.326 %	8.4589 %	10.3904 %	10.4377 %	12.2 %
BZ	SR	1.7516 %	1.0042 %	2.1068 %	1.9353 %	5.9934 %
	ME	0.655	0.660	0.6526	0.6538	0.6267
	ECMR	7.9042 %	6.883 %	8.0552 %	8.0995 %	8.9998 %
SH-B	SR	1.8253 %	2.1694 %	2.1291 %	2.0225 %	2.4882 %
	ME	0.6545	0.6522	0.6525	0.6532	0.6501
	ECMR	7.9462 %	6.9976 %	8.0417 %	8.1262 %	7.8578 %
BZ-B	SR	6.9064 %	5.3520 %	6.4083 %	6.4472 %	5.2298 %
	ME	0.7127	0.7024	0.7094	0.7097	0.7015
	ECMR	10.2326 %	8.5243 %	10.0532 %	10.1576 %	9.2779 %

Nota. SH=Shanken, BZ=Bai-Zhou, FM-B: FM Bootstrap, SH-B: SH Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 22

Intervalos de confianza Bootstrap al 95 % de confianza con $T = 60$ y $N = 300$ con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH-B	2.5	0.5362	0.5442	0.5271	0.5286	0.5278
	97.5	0.7869	0.7673	0.7723	0.7769	0.7694
BZ-B	2.5	0.5828	0.5859	0.5718	0.5749	0.5692
	97.5	0.8580	0.8275	0.8420	0.8438	0.8318

Nota. SH-B: SH Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia.

4.4.4 $T = 120$ y $N = 300$

En la Tabla 23 a través de 1000 simulaciones se obtiene que la aplicación del método Bootstrap al estimador Shanken es similar o peor a los resultados obtenidos por

Bai-Zhou, excepto en el caso de distribución asimétrica donde el caso asimétrico obtiene mejores métricas. Mientras que la aplicación de Bootstrap al estimador de Bai-Zhou solo produce peores resultados en todos los casos. Mientras que en la Tabla 24 se observa que la variabilidad es mucho menor a los obtenido en la Tabla 8.

Tabla 23

Resultados obtenidos con $T = 120$ y $N = 300$ con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH	SR	5.3151 %	4.5214 %	3.0062 %	4.8151 %	4.2031 %
	ME	0.6312	0.6365	0.6466	0.6346	0.6386
	ECMR	7.5966 %	7.3214 %	6.3868 %	7.7944 %	7.8072 %
BZ	SR	0.7985 %	0.1973 %	1.1371 %	0.1555 %	0.8285 %
	ME	0.6613	0.6654	0.6743	0.6656	0.6722
	ECMR	6.5791 %	6.6186 %	6.2285 %	7.0723 %	7.4315 %
SH-B	SR	0.8647 %	0.9056 %	0.3859 %	0.1992 %	0.7533 %
	ME	0.6609	0.6606	0.6641	0.6653	0.6616
	ECMR	6.6059 %	6.6078 %	6.1505 %	7.089 %	7.4446 %
BZ-B	SR	4.1547 %	4.4247 %	4.3100 %	4.9877 %	4.9031 %
	ME	0.6944	0.6962	0.6954	0.6999	0.6994
	ECMR	7.7055 %	7.8163 %	7.3435 %	8.5082 %	8.8264 %

Nota. SH=Shanken, BZ=Bai-Zhou, FM-B: FM Bootstrap, SH-B: SH Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 24

Intervalos de confianza Bootstrap al 95 % de confianza con $T = 120$ y $N = 300$ con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH-B	2.5	0.5511	0.5541	0.5717	0.5537	0.5472
	97.5	0.766	0.7693	0.7566	0.7749	0.7803
BZ-B	2.5	0.5781	0.5841	0.5989	0.5819	0.5761
	97.5	0.8055	0.8123	0.7934	0.8164	0.8248

Nota. SH-B: SH Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia.

4.5 Estimador Shanken con Bootstrap no paramétrico

4.5.1 T = 60 y N = 100

En la Tabla 25, a través de 1000 simulaciones se obtiene que la aplicación del método Bootstrap al estimador Shanken es similar o peor a los resultados obtenidos por Bai-Zhou, y solo en el caso de distribución mezcla se obtienen mejoras en términos de sesgo y error cuadrático medio. Mientras que la aplicación de Bootstrap al estimador de Bai-Zhou solo obtiene peores resultados. Se observa en la Tabla 26 menor variabilidad en los intervalos de confianza respecto a la Tabla 10.

Tabla 25

Resultados obtenidos con T = 60 y N = 100 con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH	SR	8.2758 %	6.5289 %	8.5938 %	8.8124 %	13.4887 %
	ME	0.6115	0.6231	0.6094	0.6079	0.5767
	ECMR	13.9016 %	12.4234 %	13.7384 %	14.0976 %	16.7095 %
BZ	SR	1.2156 %	0.6186 %	1.9196 %	1.7972 %	7.0998 %
	ME	0.6586	0.6625	0.6539	0.6547	0.6193
	ECMR	13.1847 %	12.0032 %	12.8155 %	13.1872 %	14.7443 %
SH-B	SR	1.4202 %	1.0230 %	5.5634 %	2.147 %	3.3453 %
	ME	0.6572	0.6599	0.6296	0.6524	0.6444
	ECMR	13.2162 %	12.7219 %	13.4701 %	13.2238 %	14.1657 %
BZ-B	SR	6.9691 %	6.1487 %	2.1545 %	6.1743 %	4.6283 %
	ME	0.7131	0.7077	0.6810	0.7078	0.6975
	ECMR	15.4604 %	14.5727 %	13.8814 %	15.156 %	15.6057 %

Nota. SH=Shanken, BZ=Bai-Zhou, FM-B: FM Bootstrap, SH-B: SH Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 26

Intervalos de confianza Bootstrap al 95 % de confianza con T = 60 y N = 100 con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH-B	2.5	0.4396	0.4606	0.4165	0.4458	0.4346
	97.5	0.8600	0.8727	0.8289	0.8806	0.8732
BZ-B	2.5	0.4757	0.4933	0.4500	0.4812	0.4692
	97.5	0.9371	0.9385	0.8953	0.9556	0.9480

Nota. SH-B: SH Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia

4.5.2 T = 120 y N = 100

En la Tabla 27, a través de 1000 simulaciones se obtiene que la aplicación del método Bootstrap al estimador Shanken es similar a los resultados obtenidos por Bai-Zhou excepto en la mezcla 2 donde se obtiene solo mejoras en el sesgo pero no error cuadrático medio, mientras que la aplicación de Bootstrap al estimador de Bai-Zhou solo obtiene peores resultados en todos los casos. Se observa en la Tabla 28 menor variabilidad notable en los intervalos de confianza respecto a la Tabla 12.

Tabla 27

Resultados obtenidos con T = 120 y N = 100 con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH	SR	5.8663 %	5.7761 %	4.171 %	6.5534 %	2.7683 %
	ME	0.6276	0.6282	0.6389	0.6230	0.6482
	ECMR	13.575 %	12.6724 %	12.6256 %	13.4339 %	12.7019 %
BZ	SR	0.4291 %	1.1168 %	1.0895 %	1.1684 %	2.744 %
	ME	0.6638	0.6592	0.6739	0.6589	0.6850
	ECMR	13.4716 %	12.3926 %	12.8651 %	13.0986 %	13.4572 %
SH-B	SR	0.5980 %	1.4027 %	2.0483 %	1.3526 %	0.8629 %
	ME	0.6627	0.6573	0.6803	0.6577	0.6724
	ECMR	13.4641 %	13.0526 %	12.9874 %	13.077 %	13.438 %
BZ-B	SR	5.5552 %	3.9944 %	8.0590 %	4.7372 %	7.0872 %
	ME	0.7037	0.6933	0.7204	0.6983	0.7139
	ECMR	15.0301 %	14.1088 %	15.1569 %	14.4035 %	15.4461 %

Nota. SH=Shanken, BZ=Bai-Zhou, FM-B: FM Bootstrap, SH-B: SH Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 28

Intervalos de confianza Bootstrap al 95 % de confianza con T = 120 y N = 100 con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH-B	2.5	0.4444	0.4533	0.475	0.4515	0.4506
	97.5	0.8798	0.8593	0.8880	0.8684	0.8869
BZ-B	2.5	0.4726	0.4790	0.5009	0.4797	0.4767
	97.5	0.9368	0.9060	0.9402	0.9245	0.9410

Nota. SH-B: SH Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia

4.5.3 T = 60 y N = 300

En la Tabla 29, a través de 1000 simulaciones se obtiene que la aplicación del método Bootstrap al estimador Shanken es similar o peor a los resultados obtenidos por Bai-Zhou, y solo en el caso de la mezcla 2 se obtiene buenos resultados en términos del sesgo y error cuadrático medio. Mientras que la aplicación de Bootstrap al estimador de Bai-Zhou solo provoca peores resultados. En la Tabla 30 intervalos de confianza con baja variabilidad en todos los casos, variabilidad ligeramente menor a lo observado en la Tabla 14.

Tabla 29

Resultados obtenidos con T = 60 y N = 300 con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH	SR	8.7302 %	7.0532 %	8.6779 %	9.3649 %	12.4683 %
	ME	0.6085	0.6196	0.6088	0.6042	0.5835
	ECMR	10.2525 %	8.5541 %	10.0503 %	10.5327 %	12.3933 %
BZ	SR	1.5092 %	1.1748 %	1.9713 %	2.0813 %	6.2620 %
	ME	0.6566	0.6588	0.6535	0.6528	0.6249
	ECMR	8.0822 %	6.8224 %	7.8490 %	8.0517 %	9.1559 %
SH-B	SR	1.6674 %	1.3917 %	4.5084 %	2.2509 %	3.486 %
	ME	0.6556	0.6574	0.6366	0.6517	0.6434
	ECMR	8.1271 %	7.277 %	8.4981 %	8.0868 %	8.2721 %
BZ-B	SR	6.9608 %	5.7680 %	3.341 %	6.4601 %	4.1942 %
	ME	0.7131	0.7051	0.6889	0.7097	0.6946
	ECMR	10.4467 %	9.0637 %	8.7627 %	10.1205 %	9.0944 %

Nota. SH=Shanken, BZ=Bai-Zhou, FM-B: FM Bootstrap, SH-B: SH Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 30

Intervalos de confianza Bootstrap al 95 % de confianza con T = 60 y N = 300 con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH-B	2.5	0.5265	0.5413	0.5128	0.5300	0.5195
	97.5	0.7846	0.7711	0.7623	0.7868	0.7717
BZ-B	2.5	0.5716	0.5808	0.5545	0.5775	0.5621
	97.5	0.8524	0.8285	0.8264	0.8604	0.8346

Nota. SH-B: SH Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia

4.5.4 T = 120 y N = 300

En la Tabla 31, a través de 1000 simulaciones se obtiene que la aplicación del método Bootstrap al estimador Shanken es similar a los resultados obtenidos por Bai-Zhou. Mientras que la aplicación de Bootstrap al estimador de Bai-Zhou solo provoca peores resultados. Se observa en la Tabla 32 intervalos de confianza de baja variabilidad y variabilidad notablemente menor a los obtenidos en Tabla 16.

Tabla 31

Resultados obtenidos con T = 120 y N = 300 con diferentes distribuciones de los errores

Estimador	Métrica	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH	SR	4.9464 %	4.5209 %	3.1295 %	5.0057 %	3.5075 %
	ME	0.6337	0.6365	0.6458	0.6333	0.6433
	ECMR	7.5375 %	7.3598 %	7.2689 %	7.6455 %	7.1723 %
BZ	SR	0.6228 %	0.7384 %	1.212 %	0.5520 %	0.9268 %
	ME	0.6625	0.6617	0.6748	0.6630	0.6728
	ECMR	6.6863 %	6.6614 %	7.1903 %	6.7954 %	6.9346 %
SH-B	SR	0.6872 %	0.8993 %	1.9877 %	0.6052 %	0.6294 %
	ME	0.6621	0.6607	0.6799	0.6626	0.6625
	ECMR	6.7040 %	7.0426 %	7.3239 %	6.8364 %	6.9954 %
BZ-B	SR	4.1293 %	3.4373 %	6.8735 %	4.3655 %	4.3271 %
	ME	0.6942	0.6896	0.7125	0.6958	0.6955
	ECMR	7.7889 %	7.8453 %	9.3669 %	8.0147 %	8.1482 %

Nota. SH=Shanken, BZ=Bai-Zhou, FM-B: FM Bootstrap, SH-B: SH Bootstrap, SR=Sesgo relativo, ME: media de estimadores, ECMR: error cuadrático relativo. Elaboración propia.

Tabla 32

Intervalos de confianza Bootstrap al 95 % de confianza con T = 120 y N = 300 con diferentes distribuciones de los errores

Estimador	Percentil	Normal	T-student	Normal Sesg.	Mezcla 1	Mezcla 2
SH-B	2.5	0.5541	0.5409	0.5654	0.5515	0.5442
	97.5	0.7708	0.7725	0.7958	0.7766	0.7736
BZ-B	2.5	0.5808	0.5649	0.5916	0.5792	0.5706
	97.5	0.8089	0.8066	0.8338	0.8161	0.8119

Nota. SH-B: SH Bootstrap, BZ-B: BZ Bootstrap. Elaboración propia

4.6 Aplicación

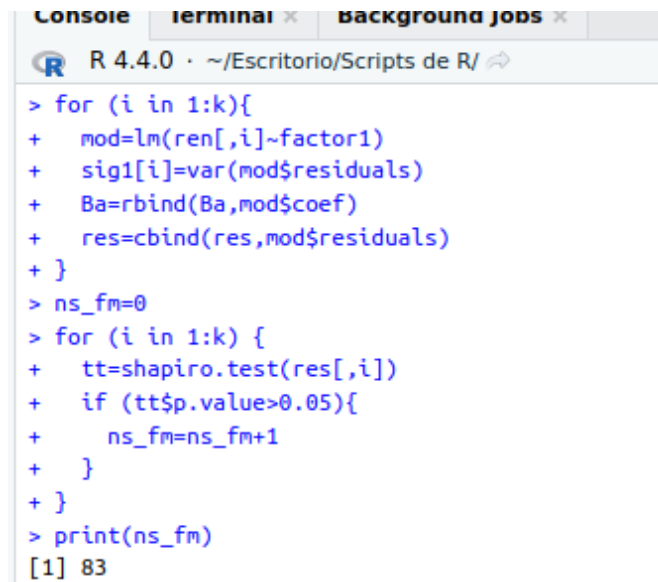
Realizaremos la aplicación de los métodos Bootstrap a los datos que tenemos usando las 100 carteras y además tomando como periodo de tiempo entre enero del 2013 y diciembre del 2018, y el factor rendimiento excedente en el mercado.

4.6.1 Fama-Macbeth

Realizamos el primer paso obteniendo los residuales correspondientes que sometidos a la prueba de normalidad de Shapiro-Wilks obtenemos 83 de los 100 casos normalidad, tal como se muestra en la Figura 28.

Figura 28

Primer paso de Fama-Macbeth en el software R



```
Console | Terminal x | Background jobs x
R 4.4.0 · ~/Escritorio/Scripts de R/ ↗
> for (i in 1:k){
+   mod=lm(ren[,i]~factor1)
+   sig1[i]=var(mod$residuals)
+   Ba=rbind(Ba,mod$coef)
+   res=cbind(res,mod$residuals)
+ }
> ns_fm=0
> for (i in 1:k) {
+   tt=shapiro.test(res[,i])
+   if (tt$p.value>0.05){
+     ns_fm=ns_fm+1
+   }
+ }
> print(ns_fm)
[1] 83
```

Nota. Elaboración propia.

Caso en el cuál aplicaremos Bootstrap paramétrico con 200 muestras Bootstrap y errores normales para la construcción de intervalos de confianza, se calcularán 1000 estimadores Bootstrap. Obteniendo luego de la aplicación del segundo paso de Fama Macbeth y Bootstrap la Tabla 33; en la cual se puede decir que la aplicación de Bootstrap en el estimador Fama-Macbeth es similar al de Bai-Zhou. Además, tenemos en la Tabla 34 los percentiles 2.5 y 97.5 para obtener un intervalo de confianza del 95 %.

Tabla 33*Resultados de aplicar Fama-Macbeth, Bai-Zhou y Bootstrap*

	FM	FM-B	FM-b
Intercepto	0.97489176	0.99522587	0.99533084
Mkt-RF	-0.05485256	-0.07270278	-0.07286532

Nota. FM: Fama-Macbeth, FM-B: FM con Bai-Zhou, FM-b: FM Bootstrap. Elaboración propia.

Tabla 34*Intervalo de confianza con Bootstrap*

Percentil 2.5	Percentil 97.5
-0.1033218	-0.04250424

Nota. Elaboración propia.

4.6.2 Shanken

Realizamos el primer paso obteniendo los residuales correspondientes que sometidos a la prueba de normalidad de Shapiro-Wilks obtenemos 83 de los 100 casos normalidad, tal como se muestra en la Figura 29

Caso en el cuál aplicaremos Bootstrap paramétrico con 200 muestras Bootstrap y errores normales para la construcción de intervalos de confianza, se calcularán 1000 estimadores Bootstrap. Obteniendo luego de la aplicación del segundo paso de Fama Macbeth (en Shanken) y Bootstrap la Tabla 35; en la cual se puede decir que la aplicación de Bootstrap en el estimador Shanken es similar al de Bai-Zhou. Además, tenemos en la Tabla 36 los percentiles 2.5 y 97.5 para obtener un intervalo de confianza del 95 %.

Figura 29

Primer paso de Fama-Macbeth en el software R

```
R 4.4.0 · ~/Escritorio/Scripts de R/ ↗  
> for (i in 1:k){  
+   mod=lm(ren[,i]~factor1)  
+   Ba=rbind(Ba,mod$coef)  
+   sig1[i,i]=var(mod$residuals)  
+   res=cbind(res,mod$residuals)  
+ }  
> ns_sh=0  
> for (i in 1:k) {  
+   tt=shapiro.test(res[,i])  
+   if (tt$p.value>0.05){  
+     ns_sh=ns_sh+1  
+   }  
+ }  
> ns_sh  
[1] 83
```

Nota. Elaboración propia.

Tabla 35

Resultados de aplicar Shanken, Bai-Zhou y Bootstrap

	SH	SH-B	SH-b
Intercepto	0.84085735	0.82764493	0.82011982
Mkt-RF	0.04465638	0.05687448	0.06230702

Nota. SH: Shanken, SH-B: SH con Bai-Zhou, SH-b: SH Bootstrap. Elaboración propia.

Tabla 36

Intervalo de confianza con Bootstrap

Percentil 2.5	Percentil 97.5
0.02292715	0.09903258

Nota. Elaboración propia.

Conclusiones

En el trabajo que se presenta se realizó la aplicación de dos versiones del método Bootstrap: paramétrico y no paramétrico, para disminuir el sesgo y error cuadrático medio en los estimadores de Fama-Macbeth, Shanken y de Bai-Zhou en diferentes escenarios de comportamiento del residual, para ello se usó datos externos (del extranjero) para realizar simulaciones de rendimiento de portafolio a su vez solo considerando un solo factor para el estudio; así como también investigar el impacto del tamaño de cartera y periodo de tiempo en los estimadores. Luego de la aplicación de las metodologías y según el resultado obtenido en el Capítulo 4 se han obtenido las siguientes conclusiones:

Conclusión general

El estudio empírico reveló que el método Bootstrap que produjo mejores resultados fue el Bootstrap paramétrico.

Conclusión específica 1

El método Bootstrap paramétrico en el estimador de Fama-Macbeth presentó mejores resultados que el estimador Bai-Zhou en términos de sesgo y error cuadrático medio relativo en los diferentes tamaños de periodo de tiempo, tamaño de la cartera en el caso de tener distribución de errores asimétricas pero en los otros casos solo se obtuvo resultados similares, mientras que la aplicación de los método Bootstrap en el estimador de Bai-Zhou provoca peores resultados o resultados similares respecto al estimador de Bai-Zhou; mientras que la aplicación del Bootstrap no paramétrico al estimador de Fama-Macbeth en este caso no consigue mejoras en los diferentes escenarios de tamaño de cartera, periodo de tiempo y distribuciones de los residuales, y en el caso de la aplicación al estimador de Bai-Zhou solo logró mejoras en el sesgo en el caso de distribución asimétrica (normal sesgada) pero no en error cuadrático medio y en el resto de casos obtuvo peores resultados. La aplicación del método Bootstrap paramétrico al estimador de Shanken logró mejoras respecto al

estimador de Bai-Zhou cuando el periodo de tiempo es elevado distribuciones de los errores es asimétrica (normal sesgada) tanto en sesgo como error cuadrático medio; mientras que en el caso de Bootstrap no paramétrico no se obtuvo ninguna mejora respecto al estimador de Bai-Zhou ni Fama-Macbeth.

Conclusión específica 2

Se encontró equilibrio entre el error cuadrático medio y el sesgo solo en el caso de aplicar el Bootstrap paramétrico en el estimador Fama-Macbeth cuando los errores se distribuyen de forma asimétrica (normal sesgada). En los casos de aplicar Bootstrap al estimador de Bai-Zhou no se encuentra equilibrio.

Conclusión específica 3

Se obtuvo que las estimaciones puntuales son mejores cuando consideramos mayor periodo de tiempo de investigación, mientras que el aumento de tamaño de cartera no obtuvo impacto significativo. Además, se observa que los intervalos de confianza del enfoque Fama-Macbeth tanto en Bootstrap paramétrico como no paramétrico presentan menor variabilidad cuando el tiempo a considerar es pequeño y se incrementa el tamaño del portafolio. En el caso del enfoque Shanken se presentan menor variabilidad en todos los casos comparados con el enfoque Fama-Macbeth.

Conclusión específica 4

Se realizó la aplicación del método Bootstrap paramétrico a los datos que se usaron para el estudio de simulación, pero ahora teniendo en cuenta los rendimientos obteniéndose así tanto en el caso de Fama-Macbeth como Shanken que estos estimadores Bootstrap son similares a los de Bai-Zhou pues en este caso se tiene normalidad.

Recomendaciones

Después de haber culminado el presente trabajo junto con sus resultados se puede recomendar tener en cuenta:

- A través del estudio de simulación se demuestra que es necesario tener en cuenta mayor cantidad de periodo de tiempo para obtener buenos estimadores en términos de sesgo.
- Se sugiere aplicar el método de Bootstrap paramétrico al estimador de Fama-Macbeth para obtener un equilibrio entre el error cuadrático medio y sesgo en el caso de tener distribución asimétrica de los residuales.
- Se sugiere realizar el análisis con más de un factor para estudios posteriores, utilizando computadoras con mejores rendimientos ya que las simulaciones toman tiempo considerable, así como también incluir en el análisis la correlación entre rendimientos.

Referencias bibliográficas

- Azzalini, A. (2013). *The skew-normal and related families* (Vol. 3). Cambridge University Press.
- Bai, J., & Zhou, G. (2015). Fama–MacBeth two-pass regressions: Improving risk premia estimates. *Finance Research Letters*, 15, 31–40.
- Bellido, L. K. (2019). *Bootstrapping como metodología para la estimación del riesgo bursátil del rendimiento del índice S&P/BVL Mining Índice de la Bolsa de Valores de Lima 2008-2017* [Tesis de pregrado]. Universidad de San Martín de Porres.
- Breloff, T. & others. (2024). *Plots.jl: Powerful Convenience for Julia Visualization*.
<https://github.com/JuliaPlots/Plots.jl>
- Canty, A., & Ripley, B. D. (2022). *boot: Bootstrap R (S-Plus) Functions*.
- Ceroni Galloso, M. (2010). ¿ Investigación básica, aplicada o sólo investigación? *Revista de la sociedad Química del Perú*, 76(1), 5–6.
- Cochrane, J. (2009). *Asset pricing: Revised edition*. Princeton university press.
- Collaborators, J. (2023a). *DataFrames.jl: Tools for working with tabular data in Julia*.
<https://github.com/JuliaStats/Statistics.jl>
- Collaborators, J. (2023b). *Distributions.jl: A Julia package for probability distributions and associated functions*. <https://github.com/JuliaStats/Distributions.jl>
- Collaborators, J. (2023c). *GLM.jl: Linear and generalized linear models in Julia*.
<https://github.com/JuliaStats/GLM.jl>
- Collaborators, J. (2023d). *Random.jl*.
<https://github.com/JuliaLang/julia/tree/master/stdlib/Random>

- Collaborators, J. (2023e). *StatsBase.jl: Is a Julia package that provides basic support for statistics*. <https://github.com/JuliaStats/StatsBase.jl/tree/master>
- Cramér, H. (1999). *Mathematical methods of statistics* (Vol. 26). Princeton university press.
- Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *Ann. Statist.*, 7(1), 1–26.
- Efron, B., & Tibshirani, R. J. (1994). *An introduction to the bootstrap*. Chapman and Hall/CRC.
- Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of financial economics*, 33(1), 3–56.
- Fama, E. F., & MacBeth, J. D. (1973). Risk, return, and equilibrium: Empirical tests. *Journal of political economy*, 81(3), 607–636.
- Feller, W. (1989). *Introducción a la Teoría de Probabilidades y sus Aplicaciones, vol. II*. Limusa.
- Gavilán, J. C. O. (2021). Metodología bootstrap en la predicción del índice de la bolsa de valores de lima, Perú (S&P/BVL). *Revista de Ciencia y Tecnología*, 37(1), 30–36.
- Giglio, S., & Xiu, D. (2021). Asset pricing with omitted factors. *Journal of Political Economy*, 129(7), 1947–1990.
- Good, P. I. (2006). *Resampling methods*. Springer.
- Haro Pimentel, E. E. (2016). *Modelo de valorización por arbitraje aplicado al índice S&P/BVL Perú Select de la Bolsa de Valores de Lima para el periodo Agosto 1999 a Agosto 2015* [Tesis de pregrado]. Universidad Andina del Cuzco.

- James, B. R. (2004). *Probabilidad: Un curso de nivel intermedio*. IMCA.
- Kroese, D. P., & Rubinstein, R. Y. (2012). Monte carlo methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(1), 48–58.
- Language, T. J. (2023). *Statistics.jl: The Statistics stdlib that ships with Julia*.
<https://github.com/JuliaStats/Statistics.jl>
- Noris, F. (2023). *XLSX.jl: Excel file reader/writer for the Julia language*.
<https://github.com/felipenoris/XLSX.jl>
- Perú, B. C. de R. del. (2024). *Glosario de términos económicos*.
<https://www.bcrp.gob.pe/publicaciones/glosario.html>
- R Core Team. (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>
- Robert, C. P., Casella, G., & Casella, G. (2010). *Introducing monte carlo methods with r* (Vol. 18). Springer.
- Rodríguez Jiménez, A., & Pérez Jacinto, A. O. (2017). Métodos científicos de indagación y de construcción del conocimiento. *Revista Ean*, 82, 179–200.
- Shanken, J. (1992). On the estimation of beta-pricing models. *The review of financial studies*, 5(1), 1–33.
- Shanken, J., & Zhou, G. (2007). Estimating and testing beta pricing models: Alternative methods and their performance in simulations. *Journal of Financial Economics*, 84(1), 40–86.
- Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>

ANEXOS

Simulación

2026-01-15

Código para obtener mezcla de distribuciones

```
require(sn)
n=5000000
x1<-rnorm(n,0,1)
x2 <- rt(n,3)
x3 <- rsn(n,0,1,10)
y1<-rbinom(n,1,0.5)
y2<-rbinom(n,1,0.8)
x4 <- (1-y1)*rnorm(n,-0.5,1)+y1*rnorm(n,0.5,1)
x5 <- (1-y2)*rnorm(n,-0.5,1)+y2*rnorm(n,0.5,0.45)

plot(density(x1,from = -8, to = 8),col="black",
      ylim=c(0,0.8),ylab="Densidad",xlab="",main="")
lines(density(x2,from = -8, to = 8),col="red")
lines(density(x3,from = -8, to = 8),col="blue")
legend("topright",
       legend = c("N(0,1)", "t(3)", "SN(0,1,10)"),
       col = c("black", "red", "blue"),
       lty = 1, lwd = 2)

plot(density(x1,from = -8, to = 8),col="black",
      ylim=c(0,0.8),ylab="Densidad",xlab="",main="")
lines(density(x4,from = -8, to = 8),col="red")
lines(density(x5,from = -8, to = 8),col="blue")
legend("topright",
       legend = c("N(0,1)", "M1", "M2"),
       col = c("black", "red", "blue"),
       lty = 1, lwd = 2)
```

Análisis descriptivo

2026-01-15

Carga de Librerías y datos

```
library(readxl)
setwd("Colocar ruta de enlace")
base=read_excel("100_Portfolios_10x10.xlsx",sheet = "Data")
fac=5#numero de factores a tomar
r=100#numero de rendimientos a tomar en entrenamiento
t=120#tiempo
factor =as.data.frame(base[(661-t):660,102:(101+fac)])
rend = as.data.frame(base[(661-t):660,2:(1+r)])
factor1=factor[61:120,]
rend1=rend[61:120,]
```

Primera ventana temporal(2014-2018)

```
plot(ts(rend1[,1]),xlab="Tiempo",ylab="Portafolio 1")
par(mfrow=c(2,1))
acf(ts(rend1[,1]),xlab="Rezagos",main="Correlograma")
pacf(ts(rend1[,1]),xlab="Rezagos",main="Correlograma Parcial")
dev.off()

plot(ts(rend1[,30]),xlab="Tiempo",ylab="Portafolio 30")
par(mfrow=c(2,1))
acf(ts(rend1[,30]),xlab="Rezagos",main="Correlograma")
pacf(ts(rend1[,30]),xlab="Rezagos",main="Correlograma Parcial")
dev.off()

plot(ts(rend1[,60]),xlab="Tiempo",ylab="Portafolio 60")
par(mfrow=c(2,1))
acf(ts(rend1[,60]),xlab="Rezagos",main="Correlograma")
pacf(ts(rend1[,60]),xlab="Rezagos",main="Correlograma Parcial")
dev.off()

plot(ts(factor1[,1]),xlab="Tiempo",ylab="Mkt-RF")
par(mfrow=c(2,1))
acf(ts(factor1[,1]),xlab="Rezagos",main="Correlograma")
pacf(ts(factor1[,1]),xlab="Rezagos",main="Correlograma Parcial")
dev.off()

plot(ts(factor1[,3]),xlab="Tiempo",ylab="HML")
par(mfrow=c(2,1))
acf(ts(factor1[,3]),xlab="Rezagos",main="Correlograma")
pacf(ts(factor1[,3]),xlab="Rezagos",main="Correlograma Parcial")
dev.off()
```

Segunda ventana temporal(2009-2018)

```

plot(ts(rend[,1]),xlab="Tiempo",ylab="Portafolio 1")
par(mfrow=c(2,1))
acf(ts(rend[,1]),xlab="Rezagos",main="Correlograma")
pacf(ts(rend[,1]),xlab="Rezagos",main="Correlograma Parcial")
dev.off()

plot(ts(rend[,30]),xlab="Tiempo",ylab="Portafolio 30")
par(mfrow=c(2,1))
acf(ts(rend[,30]),xlab="Rezagos",main="Correlograma")
pacf(ts(rend[,30]),xlab="Rezagos",main="Correlograma Parcial")
dev.off()

plot(ts(rend[,60]),xlab="Tiempo",ylab="Portafolio 60")
par(mfrow=c(2,1))
acf(ts(rend[,60]),xlab="Rezagos",main="Correlograma")
pacf(ts(rend[,60]),xlab="Rezagos",main="Correlograma Parcial")
dev.off()

plot(ts(factor[,1]),xlab="Tiempo",ylab="Mkt-RF")
par(mfrow=c(2,1))
acf(ts(factor[,1]),xlab="Rezagos",main="Correlograma")
pacf(ts(factor[,1]),xlab="Rezagos",main="Correlograma Parcial")
dev.off()

plot(ts(factor[,3]),xlab="Tiempo",ylab="HML")
par(mfrow=c(2,1))
acf(ts(factor[,3]),xlab="Rezagos",main="Correlograma")
pacf(ts(factor[,3]),xlab="Rezagos",main="Correlograma Parcial")
dev.off()

```

Esquema general

```

#Ventana 1
i=1
i=i+1
plot(ts(rend1[,i]),xlab="Tiempo",ylab="Portafolio i")
par(mfrow=c(2,1))
acf(ts(rend1[,i]),xlab="Rezagos",main="Correlograma")
pacf(ts(rend1[,i]),xlab="Rezagos",main="Correlograma Parcial")
dev.off()

#Ventana 2
i=1
i=i+1
plot(ts(rend[,i]),xlab="Tiempo",ylab="Portafolio i")
par(mfrow=c(2,1))
acf(ts(rend[,i]),xlab="Rezagos",main="Correlograma")
pacf(ts(rend[,i]),xlab="Rezagos",main="Correlograma Parcial")
dev.off()

```

Esquema Fama Macbeth – Bootstrap paramétrico

2026-01-15

Importar librerías

```
using XLSX
using Statistics
using Plots
using DataFrames
using Distributions
using GLM
using Random
using StatsBase
using JLD2

cd("Ingresar ruta")

#Dado que Las iteraciones requieren demasiado tiempo se tuvo que guardar
La información
@Load "FM_BP_1_fact_M1.jld2" FM_1_i FM_1_v FM_2_i FM_2_v FM_3_i FM_3_v F
M_4_i FM_4_v
```

Carga de datos y simulación de rendimientos

```
#Parametros a cambiar para cada diferente iteracion
fac=1#numero de factores a tomar
r=100#numero de rendimientos a tomar en entrenamiento
t=120#tiempo
rn=300#cantidad de rendimientos totales que se necesitan
nsim=1000;#numero de simulaciones

gam=[0.083333,0.66667,0.33333,0.16667]
base=XLSX.readtable("Ingresar ruta/100_Portfolios_10x10.xlsx","Data")
base=DataFrame(base)
#### Paso 1 ####
#Generar Los alfa, beta y sigma "verdaderos"#
factor =base[(661-t):660,102:(101+fac)]
rend = base[(661-t):660,2:(1+r)]
#simulamos 200 portafolios alrededor
rm=reshape([mean(rend[i,:]) for i in 1:t],t,1)
drm=reshape([std(rend[i,:]) for i in 1:t],t,1)
r=rn
sig=zeros(300,1)
rend=hcat(rend,DataFrame(zeros(t,200),:auto))
for sim in 1:t
    for ax in 101:300
        rend[sim,ax]=rand(Normal(rm[sim],drm[sim]),1)[1]
    end
end
#usamos el primer paso para La generacion#
B=zeros(300,1+fac)
col_rend=[mean(rend[:,i]) for i in 1:300]
for i in 1:300
    B[i,2]=(col_rend[i]-gam[1])/gam[2]
```

```

end
for i in 1:300
    B[i,1]=mean(rend[:,i]-B[i,2:(1+fac)][1]*Matrix(factor))
    sig[i]=var(rend[:,i]-B[i,2:(1+fac)][1]*Matrix(factor))
end

```

Simulaciones de estimaciones

```

#####Paso2#####
#generamos simulación de Los rendimientos usando Los alpha,beta y sigma
y los factores
n=t#tiempo
k=r#portafolio
factor1=Matrix(factor[1:n,:])
B1=B[1:k,:]
t3=ones(k,1)
f=factor1-ones(n,n)*factor1/n
sig1=zeros(k,1) #sigmas
sigb=zeros(k,1) #sigmas bootstrap
FM=zeros(fac+1,0)
FM_B=zeros(fac+1,0)
FM_b=zeros(fac+1,0)
FM_B_b=zeros(fac+1,0)
for a in 1:nsim
    #Genera simulacion de rendimientos
    ren=zeros(n,0)
    for i in 1:k
        #ren[,i]=B[i,1]+B[i,2]*factor[c(583:(582+n)),1]+B[i,3]*factor[c(583:
(582+n)),2]+rnorm(n)
        #y=rand(Binomial(1,0.5),n)
        #x1=rand(Normal(-0.5,1),n)
        #x2=rand(Normal(0.5,1),n)
        #x=(ones(n,1)-y).*x1+y.*x2
        ren=hcat(ren,B1[i,1][1]*ones(n,1)+B1[i,2:(1+fac)][1]*factor1+sqrt(4*
sig[i]/5)*rand(MixtureModel(Normal[
Normal(-0.5, 1.0),
Normal(0.5, 1.0)]),n))#generacion de rendimientos simulados, modifica
ble según distribución
    end
    Ba=zeros(0,1+fac)
    res=zeros(n,0)
    #####
    for i in 1:k
        bas=DataFrame(hcat(convert(Matrix{Float64},reshape(ren[:,i],n,1)),co
nvert(Matrix{Float64},factor1)),:auto)
        mod=lm(@formula(x1~x2),bas)
        sig1[i]=var(residuals(mod))
        Ba=vcat(Ba,reshape(coef(mod),1,2))
        res=hcat(res,reshape(residuals(mod),n,1))
    end
    #Estimación Fama Macbeth
    col_ren=reshape([mean(ren[:,i]) for i in 1:r],k,1)
    bas1=DataFrame(hcat(col_ren,Ba[:,2:(1+fac)]),:auto)
    FM=hcat(FM,reshape(coef(lm(@formula(x1~x2),bas1)),fac+1,1))

```

```

#####Corrección de Bai-Zhou#####
ax=mean(sig1)*(inv(transpose(f)*f))
g=vcat(hcat(transpose(t3)*t3,transpose(t3)*Ba[:,2:(1+fac)]),hcat(trans
pose(Ba[:,2:(1+fac)])*t3,k*ax+transpose(Ba[:,2:(1+fac)])*Ba[:,2:(1+fac)
])/k

FM_B=hcat(FM_B,FM[:,a]+inv(g)*vcat(0,ax*FM[2:(1+fac),a]))
##### Bootstrap #####
FM_1=zeros(fac+1,0)
FM_B_1=zeros(fac+1,0)
for s in 1:200
    ren_b=zeros(n,0)
    #Genera simulacion de rendimientos
    for i in 1:k
        #y1=rand(Binomial(1,0.5),n)
        #x11=rand(Normal(-0.5,1),n)
        #x21=rand(Normal(0.5,1),n)
        #x1=(ones(n,1)-y1).*x11+y1.*x21
        #ren[:,i]=B[i,1]+B[i,2]*factor[c(583:(582+n)),1]+B[i,3]*factor[c(58
3:(582+n)),2]+rnorm(n)
        ren_b=hcat(ren_b,Ba[i,1][1]*ones(n,1)+Ba[i,2:(1+fac)][1]*factor1+s
qrt(4*sig1[i]/5)*rand(MixtureModel(Normal[
Normal(-0.5, 1.0),
Normal(0.5, 1.0)]),n))#generacion de rendimientos simulados, modifica
ble según distribución
    end
    #Primer paso de la regresion de fama macbeth
    Ba_b=zeros(0,1+fac)
    for i in 1:k
        bas2=DataFrame(hcat(convert(Matrix{Float64},reshape(ren_b[:,i],n,1
)),convert(Matrix{Float64},factor1)),:auto)
        modb=lm(@formula(x1~x2),bas2)
        sigb[i]=var(residuals(modb))
        Ba_b=vcat(Ba_b,reshape(coef(modb),1,2))
    end
    col_ren_b=reshape([mean(ren_b[:,i]) for i in 1:r],k,1)
    bas3=DataFrame(hcat(col_ren_b,Ba_b[:,2:(1+fac)]),:auto)
    FM_1=hcat(FM_1,reshape(coef(lm(@formula(x1~x2),bas3)),fac+1,1))
    #####Corrección de Bai-Zhou#####
    ax=mean(sigb)*(inv(transpose(f)*f))
    g=vcat(hcat(transpose(t3)*t3,transpose(t3)*Ba_b[:,2:(1+fac)]),hcat(t
ranspose(Ba_b[:,2:(1+fac)])*t3,
        k*ax+transpose(Ba_b[:,2:(1+fac)])*Ba_b[:,2:(1+fac)]))/k
    FM_B_1=hcat(FM_B_1,FM_1[:,s]+inv(g)*vcat(0,ax*FM_1[2:(1+fac),s]))
    #print(s)
end
FM_b=hcat(FM_b,2*FM[:,a]-reshape([mean(FM_1[1,:]),mean(FM_1[2,:])],2,1
))
FM_B_b=hcat(FM_B_b,2*FM_B[:,a]-reshape([mean(FM_B_1[1,:]),mean(FM_B_1[
2,:])],2,1))

print(a,"\n") #auxiliar
end

```

Resultados

```
##### Analisis de 1 factor #####
#N=100 y T=60(Listo)
resum1=DataFrame(SESGO=[mean(FM[2, :]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2, :]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2, :]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2, :]-gam[2]*ones(nsim,1))/gam[2]],
                 MEDIA=[mean(FM[2, :]),mean(FM_B[2, :]),mean(FM_b[2, :]),
mean(FM_B_b[2, :])],
                 ECM=[sqrt(mean((FM[2, :]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                    sqrt(mean((FM_b[2, :]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2])]
FM_1_i=DataFrame(FamaMacbeth=FM[1, :],BaiZhou=FM_B[1, :],FM_Bootstrap=FM_b
[1, :],FM_Bai_Boots=FM_B_b[1, :]);
FM_1_v=DataFrame(FamaMacbeth=FM[2, :],BaiZhou=FM_B[2, :],FM_Bootstrap=FM_b
[2, :],FM_Bai_Boots=FM_B_b[2, :]);

resum1

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_1_FM=sort(FM_1_v[:,1])
I_1_BZ=sort(FM_1_v[:,2])
I_1_FM_B=sort(FM_1_v[:,3])
I_1_BZ_B=sort(FM_1_v[:,4])
int_1=DataFrame(METODO=["FM", "FM Bai Zhou", "FM bootstrap", "FM Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_1_FM[25],I_1_BZ[25],I_1_FM_B[25],I_1_BZ_B
[25]],
                PERCENTIL_2=[I_1_FM[975],I_1_BZ[975],I_1_FM_B[975],I_1_B
Z_B[975]])

##### Analisis de 1 factor #####
#N=100 y T=120(Listo)
resum2=DataFrame(SESGO=[mean(FM[2, :]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2, :]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2, :]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2, :]-gam[2]*ones(nsim,1))/gam[2]],
                 MEDIA=[mean(FM[2, :]),mean(FM_B[2, :]),mean(FM_b[2, :]),
mean(FM_B_b[2, :])],
                 ECM=[sqrt(mean((FM[2, :]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                    sqrt(mean((FM_b[2, :]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2])]
FM_2_i=DataFrame(FamaMacbeth=FM[1, :],BaiZhou=FM_B[1, :],FM_Bootstrap=FM_b
[1, :],FM_Bai_Boots=FM_B_b[1, :]);
FM_2_v=DataFrame(FamaMacbeth=FM[2, :],BaiZhou=FM_B[2, :],FM_Bootstrap=FM_b
[2, :],FM_Bai_Boots=FM_B_b[2, :]);

resum2

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
```

```

I_2_FM=sort(FM_2_v[:,1])
I_2_BZ=sort(FM_2_v[:,2])
I_2_FM_B=sort(FM_2_v[:,3])
I_2_BZ_B=sort(FM_2_v[:,4])
int_2=DataFrame(METODO=["FM", "FM Bai Zhou", "FM bootstrap", "FM Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_2_FM[25],I_2_BZ[25],I_2_FM_B[25],I_2_BZ_B
[25]],
                PERCENTIL_2=[I_2_FM[975],I_2_BZ[975],I_2_FM_B[975],I_2_B
Z_B[975]])

#### Analisis de 1 factor #####
#N=300 y T=60(Listo)
resum3=DataFrame(ESGO=[mean(FM[2, :]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2, :]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2, :]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2, :]-gam[2]*ones(nsim,1))/gam[2]],
                MEDIA=[mean(FM[2, :]),mean(FM_B[2, :]),mean(FM_b[2, :]),
mean(FM_B_b[2, :])],
                ECM=[sqrt(mean((FM[2, :]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                    sqrt(mean((FM_b[2, :]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2])]
FM_3_i=DataFrame(FamaMacbeth=FM[1, :],BaiZhou=FM_B[1, :],FM_Bootstrap=FM_b
[1, :],FM_Bai_Boots=FM_B_b[1, :]);
FM_3_v=DataFrame(FamaMacbeth=FM[2, :],BaiZhou=FM_B[2, :],FM_Bootstrap=FM_b
[2, :],FM_Bai_Boots=FM_B_b[2, :]);

resum3

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_3_FM=sort(FM_3_v[:,1])
I_3_BZ=sort(FM_3_v[:,2])
I_3_FM_B=sort(FM_3_v[:,3])
I_3_BZ_B=sort(FM_3_v[:,4])
int_3=DataFrame(METODO=["FM", "FM Bai Zhou", "FM bootstrap", "FM Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_3_FM[25],I_3_BZ[25],I_3_FM_B[25],I_3_BZ_B
[25]],
                PERCENTIL_2=[I_3_FM[975],I_3_BZ[975],I_3_FM_B[975],I_3_B
Z_B[975]])

#### Analisis de 1 factor #####
#N=300 y T=120(Listo)
resum4=DataFrame(ESGO=[mean(FM[2, :]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2, :]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2, :]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2, :]-gam[2]*ones(nsim,1))/gam[2]],
                MEDIA=[mean(FM[2, :]),mean(FM_B[2, :]),mean(FM_b[2, :]),
mean(FM_B_b[2, :])],
                ECM=[sqrt(mean((FM[2, :]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                    sqrt(mean((FM_b[2, :]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2])]

```

```

FM_4_i=DataFrame(FamaMacbeth=FM[1, :], BaiZhou=FM_B[1, :], FM_Bootstrap=FM_b
[1, :], FM_Bai_Boots=FM_B_b[1, :]);
FM_4_v=DataFrame(FamaMacbeth=FM[2, :], BaiZhou=FM_B[2, :], FM_Bootstrap=FM_b
[2, :], FM_Bai_Boots=FM_B_b[2, :]);

resum4

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_4_FM=sort(FM_4_v[:,1])
I_4_BZ=sort(FM_4_v[:,2])
I_4_FM_B=sort(FM_4_v[:,3])
I_4_BZ_B=sort(FM_4_v[:,4])
int_4=DataFrame(METODO=["FM", "FM Bai Zhou", "FM bootstrap", "FM Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_4_FM[25], I_4_BZ[25], I_4_FM_B[25], I_4_BZ_B
[25]],
                PERCENTIL_2=[I_4_FM[975], I_4_BZ[975], I_4_FM_B[975], I_4_B
Z_B[975]])

@save "FM_BP_1_fact_M1.jld2" FM_1_i FM_1_v FM_2_i FM_2_v FM_3_i FM_3_v F
M_4_i FM_4_v

```

Esquema Fama Macbeth – Bootstrap no paramétrico

2026-01-15

Importar librerías

```
using XLSX
using Statistics
using Plots
using DataFrames
using Distributions
using GLM
using Random
using StatsBase
using JLD2

cd("Ingresar ruta")

#Dado que Las iteraciones requieren demasiado tiempo se tuvo que guardar
La información
@Load "FM_BNP_1_fact_asimetrico.jld2" FM_1_i FM_1_v FM_2_i FM_2_v FM_3_i
FM_3_v FM_4_i FM_4_v
```

Carga de datos y simulación de rendimientos

```
#Parametros a cambiar para cada diferente iteracion
fac=1#numero de factores a tomar
r=100#numero de rendimientos a tomar en entrenamiento
t=120#tiempo
rn=300#cantidad de rendimientos totales que se necesitan
nsim=1000;#numero de simulaciones

gam=[0.083333,0.66667,0.33333,0.16667]
base=XLSX.readtable("Ingresar ruta/100_Portfolios_10x10.xlsx","Data")
base=DataFrame(base)
#### Paso 1 ####
#Generar Los alfa, beta y sigma "verdaderos"#
factor =base[(661-t):660,102:(101+fac)]
rend = base[(661-t):660,2:(1+r)]
#simulamos 200 portafolios alrededor
rm=reshape([mean(rend[i,:]) for i in 1:t],t,1)
drm=reshape([std(rend[i,:]) for i in 1:t],t,1)
r=rn
sig=zeros(300,1)
rend=hcat(rend,DataFrame(zeros(t,200),:auto))
for sim in 1:t
    for ax in 101:300
        rend[sim,ax]=rand(Normal(rm[sim],drm[sim]),1)[1]
    end
end
#usamos el primer paso para La generacion#
B=zeros(300,1+fac)
col_rend=[mean(rend[:,i]) for i in 1:300]
for i in 1:300
    B[i,2]=(col_rend[i]-gam[1])/gam[2]
```

```

end
for i in 1:300
    B[i,1]=mean(rend[:,i]-B[i,2:(1+fac)])[1]*Matrix(factor))
    sig[i]=var(rend[:,i]-B[i,2:(1+fac)])[1]*Matrix(factor))
end

```

Simulaciones de estimaciones

```

#####Paso2#####
#generamos simulación de Los rendimientos usando Los alpha,beta y sigma
y los factores
n=t#tiempo
k=r#portafolio
factor1=Matrix(factor[1:n,:])
B1=B[1:k,:]
t3=ones(k,1)
f=factor1-ones(n,n)*factor1/n
sig1=zeros(k,1) #sigmas
sigb=zeros(k,1) #sigmas bootstrap
FM=zeros(fac+1,0)
FM_B=zeros(fac+1,0)
FM_b=zeros(fac+1,0)
FM_B_b=zeros(fac+1,0)
for a in 1:nsim
    #Genera simulacion de rendimientos
    ren=zeros(n,0)
    for i in 1:k
        #ren[,i]=B[i,1]+B[i,2]*factor[c(583:(582+n)),1]+B[i,3]*factor[c(583:
(582+n)),2]+rnorm(n)
        ren=hcat(ren,B1[i,1])[1]*ones(n,1)+B1[i,2:(1+fac)])[1]*factor1+sqrt(si
g[i]/(1-25/(13*pi)))*
            (rand(SkewNormal(0,1,10),n)-5/sqrt(13*pi)*ones(n,1))#genera
cion de rendimientos simulados, modificable según distribución
    end
    Ba=zeros(0,1+fac)
    res=zeros(n,0)
    #####
    for i in 1:k
        bas=DataFrame(hcat(convert(Matrix{Float64},reshape(ren[:,i],n,1)),co
nvert(Matrix{Float64},factor1)),:auto)
        mod=lm(@formula(x1~x2),bas)
        sig1[i]=var(residuals(mod))
        Ba=vcat(Ba,reshape(coef(mod),1,2))
        res=hcat(res,reshape(residuals(mod),n,1))
    end
    #Estimación Fama Macbeth
    col_ren=reshape([mean(ren[:,i]) for i in 1:r],k,1)
    bas1=DataFrame(hcat(col_ren,Ba[:,2:(1+fac)]),:auto)
    FM=hcat(FM,reshape(coef(lm(@formula(x1~x2),bas1)),fac+1,1))
    #####Corrección de Bai-Zhou#####
    ax=mean(sig1)*(inv(transpose(f)*f))
    g=vcat(hcat(transpose(t3)*t3,transpose(t3)*Ba[:,2:(1+fac)]),hcat(trans
pose(Ba[:,2:(1+fac)]*t3,
        k*ax+transpose(Ba[:,2:(1+fac)])*Ba[:,2:(1+fac)]))/k

```

```

FM_B=hcat(FM_B,FM[:,a]+inv(g)*vcat(0,ax*FM[2:(1+fac),a]))
#### Bootstrap ####
FM_1=zeros(fac+1,0)
FM_B_1=zeros(fac+1,0)
for s in 1:200
    ren_b=zeros(n,0)
    #Genera simulacion de rendimientos
    for i in 1:k
        res_1 = sample(res[:,i], n, replace = true)
        #ren[,i]=B[i,1]+B[i,2]*factor[c(583:(582+n)),1]+B[i,3]*factor[c(58
3:(582+n)),2]+rnorm(n)
        ren_b=hcat(ren_b,Ba[i,1][1]*ones(n,1)+Ba[i,2:(1+fac)][1]*factor1+r
es_1)#generacion de rendimientos simulados
    end
    #Primer paso de La regresion de fama macbeth
    Ba_b=zeros(0,1+fac)
    for i in 1:k
        bas2=DataFrame(hcat(convert(Matrix{Float64},reshape(ren_b[:,i],n,1
)),convert(Matrix{Float64},factor1)),:auto)
        modb=lm(@formula(x1~x2),bas2)
        sigb[i]=var(residuals(modb))
        Ba_b=vcat(Ba_b,reshape(coef(modb),1,2))
    end
    col_ren_b=reshape([mean(ren_b[:,i]) for i in 1:r],k,1)
    bas3=DataFrame(hcat(col_ren_b,Ba_b[:,2:(1+fac)]),:auto)
    FM_1=hcat(FM_1,reshape(coef(lm(@formula(x1~x2),bas3)),fac+1,1))
    ####Corrección de Bai-Zhou####
    ax=mean(sigb)*(inv(transpose(f)*f))
    g=vcat(hcat(transpose(t3)*t3,transpose(t3)*Ba_b[:,2:(1+fac)]),hcat(t
ranspose(Ba_b[:,2:(1+fac)])*t3,
        k*ax+transpose(Ba_b[:,2:(1+fac)])*Ba_b[:,2:(1+fac)]))/k
    FM_B_1=hcat(FM_B_1,FM_1[:,s]+inv(g)*vcat(0,ax*FM_1[2:(1+fac),s]))
    #print(s)
end
FM_b=hcat(FM_b,2*FM[:,a]-reshape([mean(FM_1[1,:]),mean(FM_1[2,:])],2,1
))
FM_B_b=hcat(FM_B_b,2*FM_B[:,a]-reshape([mean(FM_B_1[1,:]),mean(FM_B_1[
2,:])],2,1))

print(a,"\n") #auxiliar
end

```

Resultados

```

#### Analisis de 1 factor ####
#N=100 y T=60(Listo)
resum1=DataFrame(ESGO=[mean(FM[2,:]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2,:]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2,:]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2,:]-gam[2]*ones(nsim,1))/gam[2]],
                MEDIA=[mean(FM[2,:]),mean(FM_B[2,:]),mean(FM_b[2,:]),
mean(FM_B_b[2,:])],
                ECM=[sqrt(mean((FM[2,:]-gam[2]*ones(nsim,1)).^2)/gam[

```

```

2]),sqrt(mean((FM_B[2,]-gam[2]*ones(nsim,1)).^2)/gam[2]),
      sqrt(mean((FM_b[2,]-gam[2]*ones(nsim,1)).^2)/gam[2]),
am[2]),sqrt(mean((FM_B_b[2,]-gam[2]*ones(nsim,1)).^2)/gam[2]))
FM_1_i=DataFrame(FamaMacbeth=FM[1,],BaiZhou=FM_B[1,],FM_Bootstrap=FM_b
[1,],FM_Bai_Boots=FM_B_b[1,]);
FM_1_v=DataFrame(FamaMacbeth=FM[2,],BaiZhou=FM_B[2,],FM_Bootstrap=FM_b
[2,],FM_Bai_Boots=FM_B_b[2,]);

resum1

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_1_FM=sort(FM_1_v[:,1])
I_1_BZ=sort(FM_1_v[:,2])
I_1_FM_B=sort(FM_1_v[:,3])
I_1_BZ_B=sort(FM_1_v[:,4])
int_1=DataFrame(METODO=["FM", "FM Bai Zhou", "FM bootstrap", "FM Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_1_FM[25],I_1_BZ[25],I_1_FM_B[25],I_1_BZ_B
[25]],
                PERCENTIL_2=[I_1_FM[975],I_1_BZ[975],I_1_FM_B[975],I_1_B
Z_B[975]])

##### Analisis de 1 factor #####
#N=100 y T=120(Listo)
resum2=DataFrame(ESGO=[mean(FM[2,]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2,]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2,]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2,]-gam[2]*ones(nsim,1))/gam[2]],
                MEDIA=[mean(FM[2,]),mean(FM_B[2,]),mean(FM_b[2,]),
mean(FM_B_b[2,])],
                ECM=[sqrt(mean((FM[2,]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2,]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                    sqrt(mean((FM_b[2,]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2,]-gam[2]*ones(nsim,1)).^2)/gam[2]))
FM_2_i=DataFrame(FamaMacbeth=FM[1,],BaiZhou=FM_B[1,],FM_Bootstrap=FM_b
[1,],FM_Bai_Boots=FM_B_b[1,]);
FM_2_v=DataFrame(FamaMacbeth=FM[2,],BaiZhou=FM_B[2,],FM_Bootstrap=FM_b
[2,],FM_Bai_Boots=FM_B_b[2,]);

resum2

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_2_FM=sort(FM_2_v[:,1])
I_2_BZ=sort(FM_2_v[:,2])
I_2_FM_B=sort(FM_2_v[:,3])
I_2_BZ_B=sort(FM_2_v[:,4])
int_2=DataFrame(METODO=["FM", "FM Bai Zhou", "FM bootstrap", "FM Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_2_FM[25],I_2_BZ[25],I_2_FM_B[25],I_2_BZ_B
[25]],
                PERCENTIL_2=[I_2_FM[975],I_2_BZ[975],I_2_FM_B[975],I_2_B
Z_B[975]])

```

```

##### Analisis de 1 factor #####
#N=300 y T=60(Listo)
resum3=DataFrame(SESGO=[mean(FM[2, :]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2, :]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2, :]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2, :]-gam[2]*ones(nsim,1))/gam[2]],
                MEDIA=[mean(FM[2, :]),mean(FM_B[2, :]),mean(FM_b[2, :]),
mean(FM_B_b[2, :])],
                ECM=[sqrt(mean((FM[2, :]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                    sqrt(mean((FM_b[2, :]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2])]
FM_3_i=DataFrame(FamaMacbeth=FM[1, :],BaiZhou=FM_B[1, :],FM_Bootstrap=FM_b
[1, :],FM_Bai_Boots=FM_B_b[1, :]);
FM_3_v=DataFrame(FamaMacbeth=FM[2, :],BaiZhou=FM_B[2, :],FM_Bootstrap=FM_b
[2, :],FM_Bai_Boots=FM_B_b[2, :]);

resum3

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_3_FM=sort(FM_3_v[:,1])
I_3_BZ=sort(FM_3_v[:,2])
I_3_FM_B=sort(FM_3_v[:,3])
I_3_BZ_B=sort(FM_3_v[:,4])
int_3=DataFrame(METODO=["FM", "FM Bai Zhou", "FM bootstrap", "FM Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_3_FM[25],I_3_BZ[25],I_3_FM_B[25],I_3_BZ_B
[25]],
                PERCENTIL_2=[I_3_FM[975],I_3_BZ[975],I_3_FM_B[975],I_3_B
Z_B[975]])

##### Analisis de 1 factor #####
#N=300 y T=120(Listo)
resum4=DataFrame(SESGO=[mean(FM[2, :]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2, :]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2, :]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2, :]-gam[2]*ones(nsim,1))/gam[2]],
                MEDIA=[mean(FM[2, :]),mean(FM_B[2, :]),mean(FM_b[2, :]),
mean(FM_B_b[2, :])],
                ECM=[sqrt(mean((FM[2, :]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                    sqrt(mean((FM_b[2, :]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2])]
FM_4_i=DataFrame(FamaMacbeth=FM[1, :],BaiZhou=FM_B[1, :],FM_Bootstrap=FM_b
[1, :],FM_Bai_Boots=FM_B_b[1, :]);
FM_4_v=DataFrame(FamaMacbeth=FM[2, :],BaiZhou=FM_B[2, :],FM_Bootstrap=FM_b
[2, :],FM_Bai_Boots=FM_B_b[2, :]);

resum4

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_4_FM=sort(FM_4_v[:,1])
I_4_BZ=sort(FM_4_v[:,2])

```

```

I_4_FM_B=sort(FM_4_v[:,3])
I_4_BZ_B=sort(FM_4_v[:,4])
int_4=DataFrame(METODO=["FM", "FM Bai Zhou", "FM bootstrap", "FM Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_4_FM[25],I_4_BZ[25],I_4_FM_B[25],I_4_BZ_B
[25]],
                PERCENTIL_2=[I_4_FM[975],I_4_BZ[975],I_4_FM_B[975],I_4_B
Z_B[975]])

@save "FM_BNP_1_fact_asimetrico.jld2" FM_1_i FM_1_v FM_2_i FM_2_v FM_3_i
FM_3_v FM_4_i FM_4_v

```

Esquema Shanken – Bootstrap paramétrico

2026-01-15

Importar librerías

```
using XLSX
using Statistics
using Plots
using DataFrames
using Distributions
using GLM
using Random
using StatsBase
using JLD2

cd("Ingresar ruta")

#Dado que Las iteraciones requieren demasiado tiempo se tuvo que guardar
La información
@Load "SH_BP_1_fact_M1.jld2" FM_1_i FM_1_v FM_2_i FM_2_v FM_3_i FM_3_v F
M_4_i FM_4_v
```

Carga de datos y simulación de rendimientos

```
#Parametros a cambiar para cada diferent iteracion
fac=1#numero de factores a tomar
r=100#numero de rendimientos a tomar en entrenamiento
t=120#tiempo
rn=300#cantidad de rendimientos totales que se necesitan
nsim=1000;#numero de simulaciones

gam=[0.083333,0.66667,0.33333,0.16667]
base=XLSX.readtable("Ingresar ruta/100_Portfolios_10x10.xlsx","Data")
base=DataFrame(base)
#### Paso 1 ####
#Generar Los alfa, beta y sigma "verdaderos"#
factor =base[(661-t):660,102:(101+fac)]
rend = base[(661-t):660,2:(1+r)]
#simulamos 200 portafolios alrededor
rm=reshape([mean(rend[i,:]) for i in 1:t],t,1)
drm=reshape([std(rend[i,:]) for i in 1:t],t,1)
r=rn
sig=zeros(300,300)
rend=hcat(rend,DataFrame(zeros(t,200),:auto))
for sim in 1:t
    for ax in 101:300
        rend[sim,ax]=rand(Normal(rm[sim],drm[sim]),1)[1]
    end
end
#usamos el primer paso para La generacion#
```

```

B=zeros(300,1+fac)
col_rend=[mean(rend[:,i]) for i in 1:300]
for i in 1:300
    B[i,2]=(col_rend[i]-gam[1])/gam[2]
end
for i in 1:300
    B[i,1]=mean(rend[:,i]-B[i,2:(1+fac)][1]*Matrix(factor))
    sig[i,i]=var(rend[:,i]-B[i,2:(1+fac)][1]*Matrix(factor))
end

```

Simulaciones de estimaciones

```

#####Paso2#####
#generamos simulación de Los rendimientos usando Los alpha,beta y sigma
y Los factores
n=t#tiempo
k=r#portafolio
factor1=Matrix(factor[1:n,:])
B1=B[1:k,:]
t3=ones(k,1)
f=factor1-ones(n,n)*factor1/n
sig1=zeros(k,k) #sigmas
sigb=zeros(k,k) #sigmas bootstrap
FM=zeros(fac+1,0)
FM_B=zeros(fac+1,0)
FM_b=zeros(fac+1,0)
FM_B_b=zeros(fac+1,0)
for a in 1:nsim
    #Genera simulacion de rendimientos
    ren=zeros(n,0)
    for i in 1:k
        #ren[,i]=B[i,1]+B[i,2]*factor[c(583:(582+n)),1]+B[i,3]*factor[c(583:
(582+n)),2]+rnorm(n)
        #y=rand(Binomial(1,0.5),n)
        #x1=rand(Normal(-0.5,1),n)
        #x2=rand(Normal(0.5,1),n)
        #x=(ones(n,1)-y).*x1+y.*x2
        ren=hcat(ren,B1[i,1][1]*ones(n,1)+B1[i,2:(1+fac)][1]*factor1+
            sqrt(4*sig[i,i]/5)*rand(MixtureModel(Normal[
Normal(-0.5, 1.0),
Normal(0.5, 1.0)]),n))#generacion de rendimientos simulados, modifica
ble según distribución
    end
    Ba=zeros(0,1+fac)
    res=zeros(n,0)
    #####
    for i in 1:k
        bas=DataFrame(hcat(convert(Matrix{Float64},reshape(ren[:,i],n,1)),
            convert(Matrix{Float64},factor1)),:auto)
        mod=lm(@formula(x1~x2),bas)
        sig1[i,i]=var(residuals(mod))
        Ba=vcat(Ba,reshape(coef(mod),1,2))
    end
end

```

```

    res=hcat(res,reshape(residuals(mod),n,1))
end
#Estimación Fama Macbeth
col_ren=reshape([mean(ren[:,i]) for i in 1:r],k,1)
aux_m=hcat(ones(k,1),Ba[:,2:(1+fac)])
FM=hcat(FM,
    inv(transpose(aux_m)*inv(sig1)*aux_m)*transpose(aux_m)*inv(sig1)
*col_ren)
####Corrección de Bai-Zhou####
ax=inv(transpose(f)*f)
g=vcat(hcat(transpose(t3)*inv(sig1)*t3,transpose(t3)*inv(sig1)*Ba[:,2:
(1+fac)]),
    hcat(transpose(Ba[:,2:(1+fac)])*inv(sig1)*t3,k*ax+
    transpose(Ba[:,2:(1+fac)])*inv(sig1)*Ba[:,2:(1+fac)]))/k

FM_B=hcat(FM_B,FM[:,a]+inv(g)*vcat(0,ax*FM[2:(1+fac),a]))
#### Bootstrap ####
FM_1=zeros(fac+1,0)
FM_B_1=zeros(fac+1,0)
for s in 1:200
    ren_b=zeros(n,0)
    #Genera simulacion de rendimientos
    for i in 1:k
        #y1=rand(Binomial(1,0.5),n)
        #x11=rand(Normal(-0.5,1),n)
        #x21=rand(Normal(0.5,1),n)
        #x1=(ones(n,1)-y1).*x11+y1.*x21
        #ren[:,i]=B[i,1]+B[i,2]*factor[c(583:(582+n)),1]+B[i,3]*factor[c(58
3:(582+n)),2]+rnorm(n)
        ren_b=hcat(ren_b,Ba[i,1][1]*ones(n,1)+Ba[i,2:(1+fac)][1]*factor1+
            sqrt(4*sig1[i,i]/5)*rand(MixtureModel(Normal[
Normal(-0.5, 1.0),
Normal(0.5, 1.0)]),n))#generacion de rendimientos simulados, modifica
ble según distribución
    end
    #Primer paso de la regresion de fama macbeth
    Ba_b=zeros(0,1+fac)
    for i in 1:k
        bas2=DataFrame(hcat(convert(Matrix{Float64},reshape(ren_b[:,i],n,1
)),
            convert(Matrix{Float64},factor1)),:auto)
        modb=lm(@formula(x1~x2),bas2)
        sigb[i,i]=var(residuals(modb))
        Ba_b=vcat(Ba_b,reshape(coef(modb),1,2))
    end
    col_ren_b=reshape([mean(ren_b[:,i]) for i in 1:r],k,1)
    aux_m_b=hcat(ones(k,1),Ba_b[:,2:(1+fac)])
    FM_1=hcat(FM_1,inv(transpose(aux_m_b)*
        inv(sigb)*aux_m_b)*transpose(aux_m_b)*inv(sigb)*col_ren_
b)
    ####Corrección de Bai-Zhou####
    ax=(inv(transpose(f)*f))
    g=vcat(hcat(transpose(t3)*inv(sigb)*t3,transpose(t3)*inv(sigb)*Ba_b[
:,2:(1+fac)]),

```

```

        hcat(transpose(Ba_b[:,2:(1+fac)])*inv(sigb)*t3,k*ax+
            transpose(Ba_b[:,2:(1+fac)])*inv(sigb)*Ba_b[:,2:(1+fac)]
        ))/k
        FM_B_1=hcat(FM_B_1,FM_1[:,s]+inv(g)*vcac(0,ax*FM_1[2:(1+fac),s]))
        #print(s)
    end
    FM_b=hcat(FM_b,2*FM[:,a]-reshape([mean(FM_1[1,:]),mean(FM_1[2,:])],2,1
    ))
    FM_B_b=hcat(FM_B_b,2*FM_B[:,a]-reshape([mean(FM_B_1[1,:]),mean(FM_B_1[
    2,:])],2,1))

    print(a,"\n") #auxiliar
end

```

Resultados

```

##### Analisis de 1 factor #####
#N=100 y T=60(Listo)
resum1=DataFrame(ESGO=[mean(FM[2,:]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2,:]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2,:]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2,:]-gam[2]*ones(nsim,1))/gam[2]],
                MEDIA=[mean(FM[2,:]),mean(FM_B[2,:]),mean(FM_b[2,:]),
mean(FM_B_b[2,:])],
                ECM=[sqrt(mean((FM[2,:]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2,:]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                    sqrt(mean((FM_b[2,:]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2,:]-gam[2]*ones(nsim,1)).^2)/gam[2])])
FM_1_i=DataFrame(FamaMacbeth=FM[1,:],BaiZhou=FM_B[1,:],FM_Bootstrap=FM_b
[1,:],FM_Bai_Boots=FM_B_b[1,:]);
FM_1_v=DataFrame(FamaMacbeth=FM[2,:],BaiZhou=FM_B[2,:],FM_Bootstrap=FM_b
[2,:],FM_Bai_Boots=FM_B_b[2,:]);

resum1

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_1_FM=sort(FM_1_v[:,1])
I_1_BZ=sort(FM_1_v[:,2])
I_1_FM_B=sort(FM_1_v[:,3])
I_1_BZ_B=sort(FM_1_v[:,4])
int_1=DataFrame(METODO=["SH", "SH Bai Zhou", "SH bootstrap", "SH Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_1_FM[25],I_1_BZ[25],I_1_FM_B[25],I_1_BZ_B
[25]],
                PERCENTIL_2=[I_1_FM[975],I_1_BZ[975],I_1_FM_B[975],I_1_B
Z_B[975]])

##### Analisis de 1 factor #####
#N=100 y T=120(Listo)
resum2=DataFrame(ESGO=[mean(FM[2,:]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2,:]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2,:]-gam[2]*ones(nsim,1))/gam[2],me

```

```

an(FM_B_b[2,:]-gam[2]*ones(nsim,1))/gam[2]],
    MEDIA=[mean(FM[2,:]),mean(FM_B[2,:]),mean(FM_b[2,:]),
mean(FM_B_b[2,:])],
    ECM=[sqrt(mean((FM[2,:]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2,:]-gam[2]*ones(nsim,1)).^2)/gam[2]),
    sqrt(mean((FM_b[2,:]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2,:]-gam[2]*ones(nsim,1)).^2)/gam[2])]
FM_2_i=DataFrame(FamaMacbeth=FM[1,:],BaiZhou=FM_B[1,:],FM_Bootstrap=FM_b
[1,:],FM_Bai_Boots=FM_B_b[1,:]);
FM_2_v=DataFrame(FamaMacbeth=FM[2,:],BaiZhou=FM_B[2,:],FM_Bootstrap=FM_b
[2,:],FM_Bai_Boots=FM_B_b[2,:]);

resum2

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_2_FM=sort(FM_2_v[:,1])
I_2_BZ=sort(FM_2_v[:,2])
I_2_FM_B=sort(FM_2_v[:,3])
I_2_BZ_B=sort(FM_2_v[:,4])
int_2=DataFrame(METODO=["SH","SH Bai Zhou","SH bootstrap","SH Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_2_FM[25],I_2_BZ[25],I_2_FM_B[25],I_2_BZ_B
[25]],
                PERCENTIL_2=[I_2_FM[975],I_2_BZ[975],I_2_FM_B[975],I_2_B
Z_B[975]])

#### Analisis de 1 factor #####
#N=300 y T=60(Listo)
resum3=DataFrame(SESAGO=[mean(FM[2,:]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2,:]-gam[2]*ones(nsim,1))/gam[2],
    mean(FM_b[2,:]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2,:]-gam[2]*ones(nsim,1))/gam[2]],
    MEDIA=[mean(FM[2,:]),mean(FM_B[2,:]),mean(FM_b[2,:]),
mean(FM_B_b[2,:])],
    ECM=[sqrt(mean((FM[2,:]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2,:]-gam[2]*ones(nsim,1)).^2)/gam[2]),
    sqrt(mean((FM_b[2,:]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2,:]-gam[2]*ones(nsim,1)).^2)/gam[2])]
FM_3_i=DataFrame(FamaMacbeth=FM[1,:],BaiZhou=FM_B[1,:],FM_Bootstrap=FM_b
[1,:],FM_Bai_Boots=FM_B_b[1,:]);
FM_3_v=DataFrame(FamaMacbeth=FM[2,:],BaiZhou=FM_B[2,:],FM_Bootstrap=FM_b
[2,:],FM_Bai_Boots=FM_B_b[2,:]);

resum3

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_3_FM=sort(FM_3_v[:,1])
I_3_BZ=sort(FM_3_v[:,2])
I_3_FM_B=sort(FM_3_v[:,3])
I_3_BZ_B=sort(FM_3_v[:,4])
int_3=DataFrame(METODO=["SH","SH Bai Zhou","SH bootstrap","SH Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_3_FM[25],I_3_BZ[25],I_3_FM_B[25],I_3_BZ_B

```

```

[25]],
                                PERCENTIL_2=[I_3_FM[975],I_3_BZ[975],I_3_FM_B[975],I_3_B
Z_B[975]])

#### Analisis de 1 factor #####
#N=300 y T=120(Listo)
resum4=DataFrame(ESGO=[mean(FM[2,:]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2,:]-gam[2]*ones(nsim,1))/gam[2],
                                mean(FM_b[2,:]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2,:]-gam[2]*ones(nsim,1))/gam[2]],
                                MEDIA=[mean(FM[2,:]),mean(FM_B[2,:]),mean(FM_b[2,:]),
mean(FM_B_b[2,:])],
                                ECM=[sqrt(mean((FM[2,:]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2,:]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                                sqrt(mean((FM_b[2,:]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2,:]-gam[2]*ones(nsim,1)).^2)/gam[2])])
FM_4_i=DataFrame(FamaMacbeth=FM[1,:],BaiZhou=FM_B[1,:],FM_Bootstrap=FM_b
[1,:],FM_Bai_Boots=FM_B_b[1,:]);
FM_4_v=DataFrame(FamaMacbeth=FM[2,:],BaiZhou=FM_B[2,:],FM_Bootstrap=FM_b
[2,:],FM_Bai_Boots=FM_B_b[2,:]);

resum4

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_4_FM=sort(FM_4_v[:,1])
I_4_BZ=sort(FM_4_v[:,2])
I_4_FM_B=sort(FM_4_v[:,3])
I_4_BZ_B=sort(FM_4_v[:,4])
int_4=DataFrame(METODO=["SH","SH Bai Zhou","SH bootstrap","SH Bai Zhou -
Bootstrap"],
                                PERCENTIL_1=[I_4_FM[25],I_4_BZ[25],I_4_FM_B[25],I_4_BZ_B
[25]],
                                PERCENTIL_2=[I_4_FM[975],I_4_BZ[975],I_4_FM_B[975],I_4_B
Z_B[975]])

@save "SH_BP_1_fact_M1.jld2" FM_1_i FM_1_v FM_2_i FM_2_v FM_3_i FM_3_v F
M_4_i FM_4_v

```

Esquema Shanken – Bootstrap no paramétrico

2026-01-15

Importar librerías

```
using XLSX
using Statistics
using Plots
using DataFrames
using Distributions
using GLM
using Random
using StatsBase
using JLD2
```

```
cd("Ingresar ruta")
```

```
#Dado que las iteraciones requieren demasiado tiempo se tuvo que guardar la información
```

```
@Load "SH_BNP_1_fact_M1.jld2" FM_1_i FM_1_v FM_2_i FM_2_v FM_3_i FM_3_v
FM_4_i FM_4_v
```

Carga de datos y simulación de rendimientos

```
#Parametros a cambiar para cada diferente iteracion
```

```
fac=1#numero de factores a tomar
```

```
r=100#numero de rendimientos a tomar en entrenamiento
```

```
t=120#tiempo
```

```
rn=300#cantidad de rendimientos totales que se necesitan
```

```
nsim=1000;#numero de simulaciones
```

```
gam=[0.083333,0.66667,0.33333,0.16667]
```

```
base=XLSX.readtable("Ingresar ruta/100_Portfolios_10x10.xlsx","Data")
```

```
base=DataFrame(base)
```

```
#### Paso 1 ####
```

```
#Generar Los alfa, beta y sigma "verdaderos"
```

```
factor =base[(661-t):660,102:(101+fac)]
```

```
rend = base[(661-t):660,2:(1+r)]
```

```
#simulamos 200 portafolios alrededor
```

```
rm=reshape([mean(rend[i,:]) for i in 1:t],t,1)
```

```
drm=reshape([std(rend[i,:]) for i in 1:t],t,1)
```

```
r=rn
```

```
sig=zeros(300,300)
```

```
rend=hcat(rend,DataFrame(zeros(t,200),:auto))
```

```
for sim in 1:t
```

```
  for ax in 101:300
```

```
    rend[sim,ax]=rand(Normal(rm[sim],drm[sim]),1)[1]
```

```
  end
```

```
end
```

```
#usamos el primer paso para la generacion
```

```
B=zeros(300,1+fac)
```

```
col_rend=[mean(rend[:,i]) for i in 1:300]
```

```
for i in 1:300
```

```
  B[i,2]=(col_rend[i]-gam[1])/gam[2]
```

```

end
for i in 1:300
    B[i,1]=mean(rend[:,i]-B[i,2:(1+fac)])[1]*Matrix(factor))
    sig[i,i]=var(rend[:,i]-B[i,2:(1+fac)])[1]*Matrix(factor))
end

```

Simulaciones de estimaciones

```

#####Paso2#####
#generamos simulación de Los rendimientos usando Los alpha,beta y sigma
y los factores
n=t#tiempo
k=r#portafolio
factor1=Matrix(factor[1:n,:])
B1=B[1:k,:]
t3=ones(k,1)
f=factor1-ones(n,n)*factor1/n
sig1=zeros(k,k) #sigmas
sigb=zeros(k,k) #sigmas bootstrap
FM=zeros(fac+1,0)
FM_B=zeros(fac+1,0)
FM_b=zeros(fac+1,0)
FM_B_b=zeros(fac+1,0)
for a in 1:nsim
    #Genera simulacion de rendimientos
    ren=zeros(n,0)
    for i in 1:k
        #ren[:,i]=B[i,1]+B[i,2]*factor[c(583:(582+n)),1]+B[i,3]*factor[c(583:
(582+n)),2]+rnorm(n)
        ren=hcat(ren,B1[i,1][1]*ones(n,1)+B1[i,2:(1+fac)])[1]*factor1+sqrt(4*
sig[i,i]/5)*rand(MixtureModel(Normal[
    Normal(-0.5, 1.0),
    Normal(0.5, 1.0)]),n))#generacion de rendimientos simulados, modifica
ble según distribución
    end
    Ba=zeros(0,1+fac)
    res=zeros(n,0)
    #####
    for i in 1:k
        bas=DataFrame(hcat(convert(Matrix{Float64},reshape(ren[:,i],n,1)),
convert(Matrix{Float64},factor1)),:auto)
        mod=lm(@formula(x1~x2),bas)
        sig1[i,i]=var(residuals(mod))
        Ba=vcat(Ba,reshape(coef(mod),1,2))
        res=hcat(res,reshape(residuals(mod),n,1))
    end
    #Estimación Fama Macbeth
    col_ren=reshape([mean(ren[:,i]) for i in 1:r],k,1)
    aux_m=hcat(ones(k,1),Ba[:,2:(1+fac)])
    FM=hcat(FM,inv(transpose(aux_m)*inv(sig1)*aux_m)*
transpose(aux_m)*inv(sig1)*col_ren)
    #####Corrección de Bai-Zhou#####
    ax=inv(transpose(f)*f)
    g=vcat(hcat(transpose(t3)*inv(sig1)*t3,transpose(t3)*inv(sig1)*Ba[:,2:

```

```

(1+fac)]),
    hcat(transpose(Ba[:,2:(1+fac)])*inv(sig1)*t3,
          k*ax+transpose(Ba[:,2:(1+fac)])*inv(sig1)*Ba[:,2:(1+fac)])))/
k

FM_B=hcat(FM_B,FM[:,a]+inv(g)*vcat(0,ax*FM[2:(1+fac),a]))
#### Bootstrap ####
FM_1=zeros(fac+1,0)
FM_B_1=zeros(fac+1,0)
for s in 1:200
    ren_b=zeros(n,0)
    #Genera simulacion de rendimientos
    for i in 1:k
        res_1 = sample(res[:,i], n, replace = true)
        #ren[:,i]=B[i,1]+B[i,2]*factor[c(583:(582+n)),1]+B[i,3]*factor[c(58
3:(582+n)),2]+rnorm(n)
        ren_b=hcat(ren_b,Ba[i,1][1]*ones(n,1)+
                  Ba[i,2:(1+fac)][1]*factor1+res_1)#generacion de rendimie
ntos simulados
    end
    #Primer paso de la regresion de fama macbeth
    Ba_b=zeros(0,1+fac)
    for i in 1:k
        bas2=DataFrame(hcat(convert(Matrix{Float64},reshape(ren_b[:,i],n,1
)),
                       convert(Matrix{Float64},factor1)),:auto)
        modb=lm(@formula(x1~x2),bas2)
        sigb[i,i]=var(residuals(modb))
        Ba_b=vcat(Ba_b,reshape(coef(modb),1,2))
    end
    col_ren_b=reshape([mean(ren_b[:,i]) for i in 1:r],k,1)
    aux_m_b=hcat(ones(k,1),Ba_b[:,2:(1+fac)])
    FM_1=hcat(FM_1,inv(transpose(aux_m_b)*
                          inv(sigb)*aux_m_b)*transpose(aux_m_b)*inv(sigb)*col_ren_
b)
    ####Corrección de Bai-Zhou####
    ax=(inv(transpose(f)*f))
    g=vcat(hcat(transpose(t3)*inv(sigb)*t3,transpose(t3)*inv(sigb)*Ba_b[
:,2:(1+fac)]),
           hcat(transpose(Ba_b[:,2:(1+fac)])*inv(sigb)*t3,
                 k*ax+transpose(Ba_b[:,2:(1+fac)])*inv(sigb)*Ba_b[:,2:(1+
fac)])))/k
    FM_B_1=hcat(FM_B_1,FM_1[:,s]+inv(g)*vcat(0,ax*FM_1[2:(1+fac),s]))
    #print(s)
end
FM_b=hcat(FM_b,2*FM[:,a]-reshape([mean(FM_1[1,:]),mean(FM_1[2,:])],2,1
))
FM_B_b=hcat(FM_B_b,2*FM_B[:,a]-reshape([mean(FM_B_1[1,:]),mean(FM_B_1[
2,:])],2,1))

print(a,"\n") #auxiliar
end

```

Resultados

```

##### Analisis de 1 factor #####
#N=100 y T=60(Listo)
resum1=DataFrame(SESGO=[mean(FM[2, :]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2, :]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2, :]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2, :]-gam[2]*ones(nsim,1))/gam[2]],
                MEDIA=[mean(FM[2, :]),mean(FM_B[2, :]),mean(FM_b[2, :]),
mean(FM_B_b[2, :])],
                ECM=[sqrt(mean((FM[2, :]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                    sqrt(mean((FM_b[2, :]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2])]
FM_1_i=DataFrame(FamaMacbeth=FM[1, :],BaiZhou=FM_B[1, :],FM_Bootstrap=FM_b
[1, :],FM_Bai_Boots=FM_B_b[1, :]);
FM_1_v=DataFrame(FamaMacbeth=FM[2, :],BaiZhou=FM_B[2, :],FM_Bootstrap=FM_b
[2, :],FM_Bai_Boots=FM_B_b[2, :]);

resum1

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_1_FM=sort(FM_1_v[:,1])
I_1_BZ=sort(FM_1_v[:,2])
I_1_FM_B=sort(FM_1_v[:,3])
I_1_BZ_B=sort(FM_1_v[:,4])
int_1=DataFrame(METODO=["SH", "SH Bai Zhou", "SH bootstrap", "SH Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_1_FM[25],I_1_BZ[25],I_1_FM_B[25],I_1_BZ_B
[25]],
                PERCENTIL_2=[I_1_FM[975],I_1_BZ[975],I_1_FM_B[975],I_1_B
Z_B[975]])

##### Analisis de 1 factor #####
#N=100 y T=120(Listo)
resum2=DataFrame(SESGO=[mean(FM[2, :]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2, :]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2, :]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2, :]-gam[2]*ones(nsim,1))/gam[2]],
                MEDIA=[mean(FM[2, :]),mean(FM_B[2, :]),mean(FM_b[2, :]),
mean(FM_B_b[2, :])],
                ECM=[sqrt(mean((FM[2, :]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                    sqrt(mean((FM_b[2, :]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2])]
FM_2_i=DataFrame(FamaMacbeth=FM[1, :],BaiZhou=FM_B[1, :],FM_Bootstrap=FM_b
[1, :],FM_Bai_Boots=FM_B_b[1, :]);
FM_2_v=DataFrame(FamaMacbeth=FM[2, :],BaiZhou=FM_B[2, :],FM_Bootstrap=FM_b
[2, :],FM_Bai_Boots=FM_B_b[2, :]);

resum2

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_2_FM=sort(FM_2_v[:,1])
I_2_BZ=sort(FM_2_v[:,2])

```

```

I_2_FM_B=sort(FM_2_v[:,3])
I_2_BZ_B=sort(FM_2_v[:,4])
int_2=DataFrame(METODO=["SH", "SH Bai Zhou", "SH bootstrap", "SH Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_2_FM[25],I_2_BZ[25],I_2_FM_B[25],I_2_BZ_B
[25]],
                PERCENTIL_2=[I_2_FM[975],I_2_BZ[975],I_2_FM_B[975],I_2_B
Z_B[975]])

#### Analisis de 1 factor #####
#N=300 y T=60(Listo)
resum3=DataFrame(ESGO=[mean(FM[2, :]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2, :]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2, :]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2, :]-gam[2]*ones(nsim,1))/gam[2]],
                MEDIA=[mean(FM[2, :]),mean(FM_B[2, :]),mean(FM_b[2, :]),
mean(FM_B_b[2, :])],
                ECM=[sqrt(mean((FM[2, :]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                    sqrt(mean((FM_b[2, :]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2])]
FM_3_i=DataFrame(FamaMacbeth=FM[1, :],BaiZhou=FM_B[1, :],FM_Bootstrap=FM_b
[1, :],FM_Bai_Boots=FM_B_b[1, :]);
FM_3_v=DataFrame(FamaMacbeth=FM[2, :],BaiZhou=FM_B[2, :],FM_Bootstrap=FM_b
[2, :],FM_Bai_Boots=FM_B_b[2, :]);

resum3

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_3_FM=sort(FM_3_v[:,1])
I_3_BZ=sort(FM_3_v[:,2])
I_3_FM_B=sort(FM_3_v[:,3])
I_3_BZ_B=sort(FM_3_v[:,4])
int_3=DataFrame(METODO=["SH", "SH Bai Zhou", "SH bootstrap", "SH Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_3_FM[25],I_3_BZ[25],I_3_FM_B[25],I_3_BZ_B
[25]],
                PERCENTIL_2=[I_3_FM[975],I_3_BZ[975],I_3_FM_B[975],I_3_B
Z_B[975]])

#### Analisis de 1 factor #####
#N=300 y T=120(Listo)
resum4=DataFrame(ESGO=[mean(FM[2, :]-gam[2]*ones(nsim,1))/gam[2],mean(FM
_B[2, :]-gam[2]*ones(nsim,1))/gam[2],
                    mean(FM_b[2, :]-gam[2]*ones(nsim,1))/gam[2],me
an(FM_B_b[2, :]-gam[2]*ones(nsim,1))/gam[2]],
                MEDIA=[mean(FM[2, :]),mean(FM_B[2, :]),mean(FM_b[2, :]),
mean(FM_B_b[2, :])],
                ECM=[sqrt(mean((FM[2, :]-gam[2]*ones(nsim,1)).^2)/gam[
2]),sqrt(mean((FM_B[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2]),
                    sqrt(mean((FM_b[2, :]-gam[2]*ones(nsim,1)).^2)/g
am[2]),sqrt(mean((FM_B_b[2, :]-gam[2]*ones(nsim,1)).^2)/gam[2])]
FM_4_i=DataFrame(FamaMacbeth=FM[1, :],BaiZhou=FM_B[1, :],FM_Bootstrap=FM_b
[1, :],FM_Bai_Boots=FM_B_b[1, :]);

```

```

FM_4_v=DataFrame(FamaMacbeth=FM[2, :], BaiZhou=FM_B[2, :], FM_Bootstrap=FM_b
[2, :], FM_Bai_Boots=FM_B_b[2, :]);

resum4

## intervalo bootstrap usando 95% de confianza, es decir usaremos los pe
rcentiles 0.025(25) y 0.975(975) ##
I_4_FM=sort(FM_4_v[:,1])
I_4_BZ=sort(FM_4_v[:,2])
I_4_FM_B=sort(FM_4_v[:,3])
I_4_BZ_B=sort(FM_4_v[:,4])
int_4=DataFrame(METODO=["SH", "SH Bai Zhou", "SH bootstrap", "SH Bai Zhou -
Bootstrap"],
                PERCENTIL_1=[I_4_FM[25], I_4_BZ[25], I_4_FM_B[25], I_4_BZ_B
[25]],
                PERCENTIL_2=[I_4_FM[975], I_4_BZ[975], I_4_FM_B[975], I_4_B
Z_B[975]])

@save "SH_BNP_1_fact_M1.jld2" FM_1_i FM_1_v FM_2_i FM_2_v FM_3_i FM_3_v
FM_4_i FM_4_v

```

Aplicación de Fama Macbeth

2026-01-15

Librerías y datos

```
library(readxl)
setwd("colocar ruta")
load("Aplicacion_FM.RData")
base=read_excel("100_Portfolios_10x10.xlsx",sheet = "Data")
fac=1#numero de factores a tomar
r=100#numero de rendimientos a tomar en entrenamiento
t=72#tiempo
factor =as.data.frame(base[(661-t):660,102:(101+fac)])
rend = as.data.frame(base[(661-t):660,2:(1+r)])
```

Fama Macbeth

```
n=t#tiempo
k=r#portafolio
factor1=as.matrix(factor[1:n,])
t3=matrix(1,k,1)
f=factor1-matrix(1,n,n)%*%factor1/n
sig1=rep(0,k) #sigmas
sigb=rep(0,k) #sigmas bootstrap
FM=matrix(0,fac+1,0)
FM_B=matrix(0,fac+1,0)
FM_b=matrix(0,fac+1,0)
ren=rend
Ba=matrix(0,0,1+fac)
res=matrix(0,n,0)
for (i in 1:k){
  mod=lm(ren[,i]~factor1)
  sig1[i]=var(mod$residuals)
  Ba=rbind(Ba,mod$coef)
  res=cbind(res,mod$residuals)
}
ns_fm=0
for (i in 1:k) {
  tt=shapiro.test(res[,i])
  if (tt$p.value>0.05){
    ns_fm=ns_fm+1
  }
}
#Estimación Fama Macbeth
FM=cbind(FM,lm(colMeans(ren)~Ba[,2:(1+fac)])$coef)
###Corrección de Bai-Zhou###
ax=mean(sig1)*(solve(t(f)%*%f))
g=rbind(cbind(1,t(t3)%*%Ba[,2:(1+fac)]/k),cbind(t(Ba[,2:(1+fac)])%*%t3/k
,
ax+t(Ba[,2:(1+fac)])%*%B
a[,2:(1+fac)]/k))
```

```
FM_B=cbind(FM_B,FM[,1]+solve(g)%*%rbind(0,ax%%FM[2:(1+fac),1]))
```

```
FM;FM_B
```

Bootstrap paramétrico

```
FM_B_1=matrix(0, fac+1, 0)
for (p in 1:1000) {
  FM_1=matrix(0, fac+1, 0)
  for (s in 1:200) {
    ren_b=matrix(0, n, 0)
    #Genera simulacion de rendimientos
    for (i in 1:k){
      #ren[,i]=B[i,1]+B[i,2]*factor[c(583:(582+n)),1]+
      #B[i,3]*factor[c(583:(582+n)),2]+rnorm(n)
      ren_b=cbind(ren_b, rnorm(n, Ba[i,1]+apply(Ba[i,2:(1+fac)], 1,
sum),
                                sqrt(sig1[i])))#generacion de rendimientos
simulados
    }
    #Primer paso de la regresion de fama macbeth
    Ba_b=matrix(0, 0, 1+fac)
    for (i in 1:k){
      modb=lm(ren_b[,i]~factor1)
      sigb[i]=var(modb$residuals)
      Ba_b=rbind(Ba_b, modb$coef)
    }
    FM_1=cbind(FM_1, lm(colMeans(ren_b)~Ba_b[,2:(1+fac)])$coef)#estimador
fama macbeth
  }
  FM_B_1=cbind(FM_B_1, rowMeans(FM_1))
  print(p)
}

FM_b=cbind(FM_b, 2*FM[,1]-rowMeans(FM_B_1))
FM_res=data.frame(FM, FM_B, FM_b=FM_b[,1])
#Intervalo de confianza
IC=(2*FM[,1]-(FM_B_1))[2,]
IC=IC[order(IC)]
print(data.frame(P025=c(IC[25]), P975=c(IC[975])))
```

Guardado de datos

```
save.image(file = "Aplicacion_FM.RData")
```

Aplicación de Shanken

2026-01-15

Librerías y datos

```
library(readxl)
setwd("colocar ruta")
load("Aplicacion_FM.RData")
base=read_excel("100_Portfolios_10x10.xlsx",sheet = "Data")
fac=1#numero de factores a tomar
r=100#numero de rendimientos a tomar en entrenamiento
t=72#tiempo
factor =as.data.frame(base[(661-t):660,102:(101+fac)])
rend = as.data.frame(base[(661-t):660,2:(1+r)])
```

Shanken

```
n=t#tiempo
k=r#portafolio
factor1=as.matrix(factor[1:n,])
t3=matrix(1,k,1)
f=factor1-matrix(1,n,n)%*%factor1/n
sig1=matrix(0,k,k)
SH=matrix(0,fac+1,0)
SH_B=matrix(0,fac+1,0)
SH_b=matrix(0,fac+1,0)

#Genera simulacion de rendimientos
ren=rend
Ba=matrix(0,0,1+fac)
res=matrix(0,n,0)
for (i in 1:k){
  mod=lm(ren[,i]~factor1)
  Ba=rbind(Ba,mod$coef)
  sig1[i,i]=var(mod$residuals)
  res=cbind(res,mod$residuals)
}
ns_sh=0
for (i in 1:k) {
  tt=shapiro.test(res[,i])
  if (tt$p.value>0.05){
    ns_sh=ns_sh+1
  }
}

#Estimación Shanken
SH=cbind(SH,lm(colMeans(ren)~Ba[,2:(1+fac)],
              weights = sqrt(diag(sig1)))$coef)

###Corrección de Bai-Zhou###
ax1=solve(sig1)
ax=solve(t(f)%*%f)
g=rbind(cbind(t(t3)%*%ax1%*%t3/k,t(t3)%*%ax1%*%Ba[,2:(1+fac)]/k),
        cbind(t(Ba[,2:(1+fac)])%*%ax1%*%t3/k,
```

```
ax+t(Ba[,2:(1+fac)])%%ax1%%Ba[,2:(1+fac)]/k))
SH_B=cbind(SH_B,SH[,1]+solve(g)%%rbind(0,ax%%SH[2:(1+fac),1]))
```

Bootstrap paramétrico

```
SH_B_1=matrix(0, fac+1,0)
for (p in 1:1000){
  SH_1=matrix(0, fac+1,0)
  for (s in 1:200) {
    ren_b=matrix(0,n,0)
    #Genera simulacion de rendimientos
    for (i in 1:k){
      #ren[,i]=B[i,1]+B[i,2]*factor[c(583:(582+n)),1]+B[i,3]*
      #factor[c(583:(582+n)),2]+rnorm(n)
      ren_b=cbind(ren_b,rnorm(n,Ba[i,1]+
        apply(Ba[i,2:(1+fac)]*factor1,1,sum),
        sqrt(sig1[i,i])))#generacion de rendimientos simulados
    }
    #Primer paso de la regresion de fama macbeth
    Ba_b=matrix(0,0,1+fac)
    sigb=matrix(0,k,k)
    for (i in 1:k){
      modb=lm(ren_b[,i]~factor1)
      sigb[i,i]=var(modb$residuals)
      Ba_b=rbind(Ba_b,modb$coef)
    }
    SH_1=cbind(SH_1,lm(colMeans(ren_b)~Ba_b[,2:(1+fac)],
      weights = sqrt(diag(sigb)))$coef)#estimador fama macbeth
  }
  SH_B_1=cbind(SH_B_1,rowMeans(SH_1))
  print(p)
}

SH_b=cbind(SH_b,2*SH[,1]-rowMeans(SH_B_1))
SH_res=data.frame(SH,SH_B,SH_b=SH_b[,1])

#Intervalo de confianza
IC=(2*SH[,1]-(SH_B_1))[2,]
IC=IC[order(IC)]
print(data.frame(P025=c(IC[25]),P975=c(IC[975])))
```

Guardado de datos

```
save.image(file = "Aplicacion_SH.RData")
```