

TRABAJO FINAL DE MÁSTER

Título: Comparación de Modelos GARCH y Volatilidad Realizada en la Predicción de la Volatilidad del Tipo de Cambio USD/EUR.

Autoría: Haodong Du

Tutoría: Héctor Rufino Alcalde

Curso académico: 2023 - 2024



UNIVERSITAT DE
BARCELONA

Facultat d'Economia
i Empresa

Màster
de Ciències
Actuarials
i Financieres

Facultad de Economía y Empresa

Universidad de Barcelona

Trabajo Final de Máster

Máster en Ciencias Actuariales y Financieras

**Comparación de Modelos GARCH y Volatilidad Realizada
en la Predicción de la Volatilidad del Tipo de Cambio
USD/EUR.**

Autoría: Haodong Du

Tutoría: Héctor Rufino Alcalde.

Resumen

Este trabajo compara dos enfoques principales para la predicción de la volatilidad del tipo de cambio USD/EUR: los modelos GARCH y la volatilidad realizada. Los modelos GARCH, introducidos por Engle y Bollerslev, son ampliamente utilizados en finanzas para modelar y prever la volatilidad debido a su capacidad para capturar la dinámica temporal y la agrupación de la volatilidad. Sin embargo, estos modelos se basan en datos de menor frecuencia y pueden no reflejar con precisión la volatilidad intradía.

Por otro lado, la volatilidad realizada, propuesta por Andersen et al. (2003), utiliza datos de alta frecuencia para proporcionar una medida más precisa de la volatilidad. En este estudio, se aplican técnicas de Machine Learning, específicamente la regresión autoregresiva con redes neuronales (NNAR), para predecir la volatilidad realizada, comparándola con los modelos GARCH.

El análisis revela que los modelos GARCH, a pesar de ser conservadores, pueden resultar en una sobreestimación de la volatilidad y, por lo tanto, en una reserva excesiva de capital en muchas ocasiones. La metodología NNAR demuestra ser efectiva en la captura de patrones no lineales complejos en los datos, ofreciendo predicciones precisas de la volatilidad. La comparación entre ambos enfoques se realiza evaluando el Valor en Riesgo (VaR) al 5% y al 1%, mostrando que el modelo GARCH es más conservador y proporciona mayor seguridad en situaciones de alta volatilidad, mientras que el modelo basado en NNAR puede ser más eficiente en condiciones de mercado más estables.

Esta investigación sugiere que una combinación de modelos tradicionales como GARCH y técnicas modernas de Machine Learning puede ofrecer una ventaja significativa en la previsión de la volatilidad y la optimización de las estrategias financieras.

Abstract

This paper compares two main approaches to forecasting USD/EUR exchange rate volatility: GARCH models and realised volatility. GARCH models, introduced by Engle and Bollerslev, are widely used in finance to model and forecast volatility because of their ability to capture the temporal dynamics and clustering of volatility. However, these models are based on lower frequency data and may not accurately reflect intraday volatility.

On the other hand, realised volatility, proposed by Andersen et al. (2003), uses high-frequency data to provide a more accurate measure of volatility. In this study, Machine Learning techniques, specifically neural network autoregressive regression (NNAR), are applied to predict realised volatility by comparing it with GARCH models.

The analysis reveals that GARCH models, although conservative, can result in an overestimation of volatility and, therefore, in an excessive reservation of capital on many occasions. The NNAR methodology proves to be effective in capturing complex non-linear patterns in the data, providing accurate predictions of volatility. The comparison between the two approaches is made by assessing Value at Risk (VaR) at 5% and 1%, showing that the GARCH model is more conservative and provides greater security in high volatility situations, while the NNAR-based model may be more efficient in more stable market conditions.

This research suggests that a combination of traditional models such as GARCH and modern Machine Learning techniques can offer a significant advantage in forecasting volatility and optimising financial strategies.

ÍNDICE:

1-	Introducción.....	1
2-	Base teórica.....	3
	2.1 - GARCH.....	3
	2.2- Volatilidad Realizada.....	5
3-	Datos.....	6
	3.1- Presentación:	6
	3.2- Análisis descriptivo:	7
4-	Modelización GARCH	8
	4.1- Comprobaciones previas:	8
	4.2- Modelización de la media	9
	4.3- Modelización de varianza: GARCH	12
	4.4- Predicción.....	14
5-	Volatilidad realizada.	16
	5.1- Modelo de predicción: NNAR.	16
	5.2. Modelización:	17
	5.3- Predicción:.....	18
6-	Comparación entre dos modelos. Aplicación en VaR:	20
7-	Conclusión y futuras líneas de investigación.	22
8-	Bibliografía:.....	24
9-	Anexo:	26

1- Introducción

La volatilidad en los mercados de divisas mide la intensidad de las fluctuaciones en los tipos de cambio. Una mayor volatilidad puede aumentar el riesgo de pérdidas, pero también ofrece oportunidades de obtener mayores ganancias. Por lo tanto, comprender y predecir la volatilidad es esencial para gestionar el riesgo y optimizar las estrategias de inversión.

Los modelos GARCH (Generalized Autoregressive Conditional Heteroskedasticity) son herramientas estadísticas ampliamente utilizadas para modelar y pronosticar la volatilidad en los mercados financieros. Estos modelos capturan la agrupación de volatilidad, es decir, la tendencia de los períodos de alta volatilidad a seguirse de períodos de alta volatilidad, y viceversa.

La volatilidad realizada, propuesta por Andersen et al. (2003), es una medida alternativa de la volatilidad calculada a partir de datos de alta frecuencia de los precios de los activos. A diferencia de los modelos GARCH, que se basan en datos de menor frecuencia (como precios diarios o semanales), la volatilidad realizada utiliza datos de intervalos de tiempo muy cortos (como precios cada minuto o cada segundo) para estimar la volatilidad.

El Valor en Riesgo (VaR) es una medida estadística que cuantifica el nivel de riesgo financiero dentro de una cartera de inversiones durante un período específico. Este indicador estima la pérdida máxima potencial con un nivel de confianza dado (por ejemplo, 95% o 99%). El VaR es fundamental para la gestión del riesgo y la toma de decisiones estratégicas en finanzas.

Comparar el VaR estimado por modelos GARCH con el VaR basado en la volatilidad realizada es interesante porque permite evaluar la precisión y eficacia de los modelos GARCH en la captura de la dinámica de la volatilidad en los mercados de divisas. Si la volatilidad realizada, que se considera una medida más precisa de la volatilidad real, difiere significativamente de la volatilidad estimada por GARCH, esto podría indicar que los modelos GARCH no están capturando adecuadamente la verdadera volatilidad del mercado.

Además, la comparación de ambas medidas puede proporcionar información valiosa sobre la naturaleza de la volatilidad en los mercados de divisas. Por ejemplo, si la volatilidad realizada muestra una mayor persistencia o una dinámica diferente a la volatilidad estimada por GARCH, esto podría sugerir la presencia de factores adicionales que influyen en la volatilidad y que no son capturados por los modelos GARCH tradicionales.

A partir del S.XXI con los avances tecnológicos, la capacidad de cálculo de la tecnología crece de una forma exponencial, gracias a eso, están surgiendo metodologías basadas en Machine Learning para pronosticar las volatilidades. Estas nuevas técnicas ofrecen la posibilidad de capturar patrones más complejos y no lineales en los datos, lo que puede mejorar la precisión de las predicciones. Por ejemplo, las Redes Neuronales Artificiales (ANN) han sido ampliamente revisadas por Zhang (2003) en "Forecasting financial time series using neural networks: A comprehensive review", destacando su capacidad para capturar relaciones no lineales en datos financieros. Además, las Máquinas de Soporte Vectorial (SVM), como se discute en "Financial time series forecasting using support vector machines" por Kim (2003), muestran un rendimiento superior en precisión predictiva comparado con los modelos tradicionales.

Otra técnica importante es el uso de Bosques Aleatorios, analizada en "Random forest for time series forecasting" por Adhikari y Agrawal (2013), que destaca su robustez frente al sobreajuste y su capacidad para manejar datos de alta dimensionalidad. Las Redes Neuronales Recurrentes (RNN) y su variante LSTM, introducidas por Hochreiter y Schmidhuber (1997) en "Long short-term memory networks for machine learning", mejoran la capacidad de modelado de dependencias a largo plazo en series temporales. Finalmente, las Redes Neuronales Convolucionales (CNN), como se demuestra en "Convolutional neural networks for stock price prediction" por Bao, Yue, y Rao (2017), son eficaces en capturar patrones locales en datos financieros.

Sería interesante comparar estas metodologías de Machine Learning con los modelos tradicionales de GARCH para determinar su eficacia relativa y potencial de mejora en la gestión del riesgo. Esta comparación es crucial para mejorar los modelos de predicción de riesgo y ajustar las estrategias de inversión y gestión del riesgo de manera más precisa. Integrar técnicas modernas de Machine Learning con modelos tradicionales como GARCH podría proporcionar una ventaja significativa en la previsión de la volatilidad y la optimización de las estrategias financieras.

2- Base teórica

2.1 - GARCH.

La teoría del modelo ARCH fue presentada por Engle, que rápidamente se puso en la vanguardia para afrontar el problema de la no constancia de la varianza en las series temporales, que es un gran avance en la rama de la econometría. Este fue un gran logro que le hizo merecer el Premio Nobel de Economía en el año 2003. A diferencia de los demás modelos que buscan pronosticar los valores esperados, el modelo ARCH es un modelo que puede pronosticar la varianza de una variable, una medida bastante crucial de la volatilidad.

Su aplicación en el mundo de las finanzas ha sido muy importante para el análisis y la gestión de riesgos, sobre todo para la previsión de series temporales, como las tasas de cambio, las tasas de interés y los índices de precios de las acciones, que tienen en común las siguientes características:

- Varianza no constante: La variabilidad de la serie es no constante en el tiempo, habiendo puntos de alta y baja intensidad.
- Clustering de volatilidades: La propiedad de los datos que significa que los niveles altos (bajos) de volatilidad tienden a agruparse, sobre el período, con más niveles altos (bajos).
- Leptocúrtico: Las distribuciones de estos datos tienen colas más densas que la distribución normal, lo que da lugar a una mayor probabilidad de eventos extremos (tanto altos como bajos).

Según Engle (1982), podemos expresar un modelo ARCH(q) como:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^q \alpha_i r_{t-i}^2$$

σ_t^2 es la varianza condicional en el tiempo t , α_0 es una constante, α_i son los coeficientes del modelo, r_{t-i}^2 son los retornos pasados al cuadrado.

Este modelo asume que la varianza condicional es una función lineal de los retornos pasados al cuadrado. Sin embargo, uno de los principales problemas del modelo ARCH es que requiere un gran número de parámetros cuando q es grande, lo que puede hacer que el modelo sea demasiado complejo y difícil de estimar.

Para superar las limitaciones del modelo ARCH, Bollerslev (1986) propuso el modelo GARCH (Generalized Autoregressive Conditional Heteroskedasticity), que extiende el modelo ARCH al permitir que la varianza condicional dependa no solo de los retornos pasados, sino también de sus propias varianzas pasadas. La formulación básica del modelo GARCH (p, q) es:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^q \alpha_i r_{t-i}^2 + \sum_{j=1}^p \beta_j \sigma_{t-j}^2$$

donde la innovación es β_j son los coeficientes que multiplican las varianzas pasadas, y σ_{t-j}^2 son las varianzas pasadas.

Para el modelo GARCH (1,1), que es el más utilizado en aplicaciones financieras, se puede escribir como:

$$\sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

Este modelo introduce la autoregresividad en la varianza condicional, permitiendo que los choques pasados influyan en la varianza futura. La inclusión de términos autoregresivos y de media móvil en la varianza condicional ayuda a capturar la persistencia y la agrupación de volatilidad observadas en los datos financieros.

A partir de GARCH(p,q) se ha ido desarrollando nuevos variantes, la línea de innovación se ha centrado sobretodo en la incorporación de asimetría.

El modelo eGARCH, propuesto por Nelson (1990), se formula en términos del logaritmo de la varianza condicional. La especificación general del modelo eGARCH(1,1) es la siguiente:

$$\ln(\sigma_t^2) = \omega + \beta \ln(\sigma_{t-1}^2) + \gamma \left(\frac{r_{t-1}}{\sigma_{t-1}} \right) + \theta \left(\left| \frac{r_{t-1}}{\sigma_{t-1}} \right| - \sqrt{\frac{2}{\pi}} \right)$$

Donde la innovación está en la incorporación de γ que captura el efecto apalancamiento, permitiendo que los choques negativos y positivos tengan impactos diferentes en la volatilidad y θ que captura la reacción de la volatilidad a los choques.

El modelo gjrGARCH, desarrollado por Glosten, Jagannathan y Runkle (1993), introduce un término adicional para capturar la asimetría en la respuesta de la volatilidad a los choques negativos y positivos. La especificación general del modelo gjrGARCH(1,1) es la siguiente:

$$\sigma_t^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2 + \gamma r_{t-1}^2 I_{t-1}$$

Donde la innovación está en la incorporación de γ mide el efecto apalancamiento, donde I_{t-1} es una variable indicadora que toma el valor de 1 si $r_{t-1} < 0$ y 0 en caso contrario.

Estos modelos permiten una mejor captura de la asimetría y la dinámica de la volatilidad en los datos financieros, mejorando la precisión de las predicciones y el ajuste del modelo a las series temporales observadas.

Aunque cabe destacar que Hansen y Lunde (2005), realizaron una comparación de más de 300 modelos de volatilidad diferentes utilizando datos diarios de tasas de cambio. Los resultados mostraron que ninguno de estos modelos proporcionaba pronósticos

significativamente mejores que el modelo GARCH(1,1). Este hallazgo sugiere que, a pesar de la variedad y complejidad de los modelos probados, el modelo GARCH(1,1) sigue siendo altamente competitivo en términos de su capacidad predictiva para la volatilidad.

2.2- Volatilidad Realizada

La volatilidad realizada es una medida empírica de la variabilidad del retorno de un activo financiero, calculada utilizando datos intradías de alta frecuencia. A diferencia de las medidas tradicionales de volatilidad, que se basan en datos diarios o en modelos paramétricos, la volatilidad realizada se obtiene sumando los cuadrados de los retornos intradía sobre un periodo específico, generalmente un día.

Para calcular la volatilidad realizada, se utiliza la siguiente fórmula:

$$RV_t = \sqrt{\sum_{i=1}^n r_{i,t}^2}$$

donde $r_{i,t}$ representa los retornos intradías del activo financiero en el intervalo i del día t , y n es el número de intervalos intradía en el día. Por ejemplo, si utilizamos retornos de 5 minutos, entonces nn será igual a 78, correspondiente a 390 minutos de negociación en un día de trading.

Esta medida se fundamenta en la teoría de la variación cuadrática, la cual establece que, bajo condiciones adecuadas, la volatilidad realizada es un estimador insesgado y eficiente de la volatilidad del retorno (Andersen et al., 2003). Según esta teoría, la volatilidad realizada captura la verdadera variabilidad de los retornos debido a la agregación de los movimientos del precio en intervalos cortos.

3- Datos

3.1- Presentación:

Gráfico 1. Evolución de USD/EUR



Para la modelización GARCH, utilizamos la serie USD/EUR que abarca desde el 01/01/2014 hasta el 01/04/2024 con datos diarios. Los datos fueron extraídos de Yahoo Finance utilizando la función del paquete Quandl de R.

Comentario sobre los cambios en las cotizaciones:

- Principios de 2015: El euro se debilitó drásticamente frente al dólar, alcanzando un mínimo de aproximadamente 1.05 dólares por euro. Esta gran caída se explica por la política monetaria expansiva adoptada por el Banco Central Europeo (BCE) y la divergencia en las políticas monetarias entre el BCE y la Reserva Federal de EE.UU.
- Finales de 2016 y principios de 2017: Se produjo otro declive del euro, aunque de una magnitud menor, alcanzando un mínimo de aproximadamente 1.03 dólares por euro. Este declive probablemente fue impulsado por la inestabilidad política en Europa, incluido el referéndum sobre el Brexit en el Reino Unido y las elecciones en otros países europeos.
- 2017-2018: El euro se fortaleció frente al dólar estadounidense, alcanzando un nivel récord de aproximadamente 1.25 dólares por euro. La razón detrás de esto fue la recuperación de la economía de la eurozona y las expectativas de que el BCE comenzaría a normalizar su política monetaria.

- Principios de 2021: El euro subió de valor nuevamente, alcanzando un máximo de alrededor de 1.23 dólares por euro. Esta apreciación reciente se debió en parte al debilitamiento del dólar como resultado de los agresivos estímulos fiscales y monetarios implementados en los Estados Unidos para enfrentar el impacto económico de la pandemia de COVID-19.
- 2022: El euro cayó nuevamente, alcanzando un nivel de aproximadamente 0.95 dólares por euro en septiembre de 2022. La principal razón detrás de esto fue la guerra en Ucrania, que llevó a una crisis energética y a una mayor incertidumbre económica en Europa.
- 2023 a 2024: El euro se devaluó inicialmente debido a la subida de tipos de interés en EE.UU., pero posteriormente se recuperó gradualmente en respuesta a las subidas de tipos de interés por parte de la Unión Europea.

3.2- Análisis descriptivo:

Tabla 1 Análisis descriptivo

Minimum	0.959619
Maximum	1.393088
Mean	1.143826
Median	1.126266
Variance	0.006648
Stdev	0.081535
Skewness	1.127388
Kurtosis	1.472627
J-B Normality Test	Asymptotic p Value: < 2.2e-16

El análisis descriptivo de la serie USD/EUR revela varias características importantes sobre la distribución y el comportamiento de los datos del tipo de cambio.

La serie muestra un rango de valores considerable, con un valor mínimo de 0.959619 y un valor máximo de 1.393088. La media del tipo de cambio es 1.143826, lo que sugiere que, en promedio, el euro se ha intercambiado a aproximadamente 1.14 dólares. La mediana, ligeramente inferior a la media con un valor de 1.126266, indica una distribución ligeramente sesgada.

La varianza del tipo de cambio es 0.006648 y la desviación estándar es 0.081535, reflejando la dispersión y volatilidad de los valores alrededor de la media. Una desviación estándar relativamente baja sugiere que los valores están bastante concentrados alrededor de la media.

La asimetría es 1.127388, indicando una distribución sesgada a la derecha, con más valores altos extremos que valores bajos extremos. La curtosis es 1.472627, sugiriendo que la distribución tiene colas más pesadas que una distribución normal, pero no excesivamente pronunciadas.

La prueba de normalidad de Jarque-Bera, con un valor p asintótico menor a 2.2e-16, rechaza la hipótesis nula de normalidad, confirmando que la distribución del tipo de cambio USD/EUR no es normal. Esto es consistente con la presencia de asimetría y curtosis observadas.

4- Modelización GARCH

4.1- Comprobaciones previas:

Para aplicar un modelo GARCH, es esencial que la serie temporal sea estacionaria. Esto se debe a que el modelo GARCH asume que la varianza condicional es una función lineal de los retornos y varianzas pasadas. Esto requiere que la varianza incondicional sea constante en el tiempo.

A continuación, tenemos algunos resultados de prueba de estacionariedad aplicado a la serie de precio.

Gráfico 2. ACF y PACF serie precio

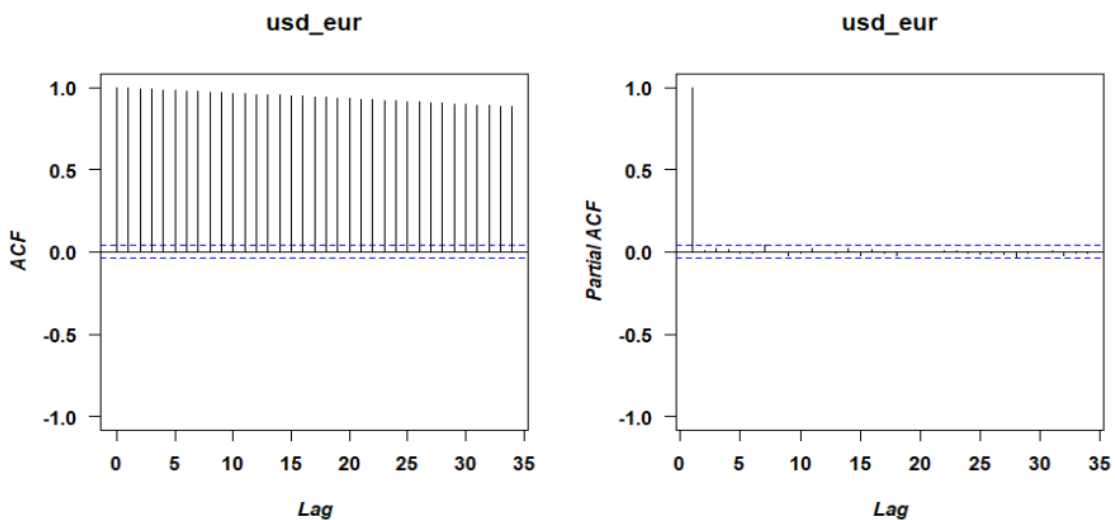


Tabla 2. Tests de estacionariedad serie precio

Test	Val. Crítico (5%)	Estadístico	H0	H1	Resultado
Ljung-Box (lag =5)	0.05	2.2e-16	Estacionaria	No estacionaria	H1
Ljung-Box (lag =10)	0.05	2.2e-16	Estacionaria	No estacionaria	H1
Ljung-Box (lag =20)	0.05	2.2e-16	Estacionaria	No estacionaria	H1
D-F test (5%)	-3.96	-2.6182	No estacionaria	Estacionaria	H0
P-P test (5%)	-3.4358	-2.8288	No estacionaria	Estacionaria	H0
KPSS test (5%)	0.463	6.0317	Estacionaria	No estacionaria	H1

La ACF no decae rápidamente, indicando una dependencia a largo plazo, y en cuando PACF muestra varias autocorrelaciones parciales significativas, lo que sugiere una fuerte influencia de los valores pasados.

Las pruebas de estacionariedad, incluyendo Ljung-Box, Dickey-Fuller, Phillips-Perron y KPSS, indican consistentemente que la serie temporal USD/EUR no es estacionaria. Para aplicar un modelo GARCH, será necesario transformar la serie para lograr la estacionariedad, posiblemente mediante la diferenciación o eliminación de tendencias y estacionalidades.

Aplicamos una diferenciación logarítmica a nuestra serie, obteniendo de esta forma el retorno. Realizamos las mismas pruebas. Para que los valores outliers no afecte en excesivo al ajuste, topamos que el rendimiento no exceda al percentil 1% y 99% de toda la serie.

Gráfico 3. Rendimientos de USD/EUR

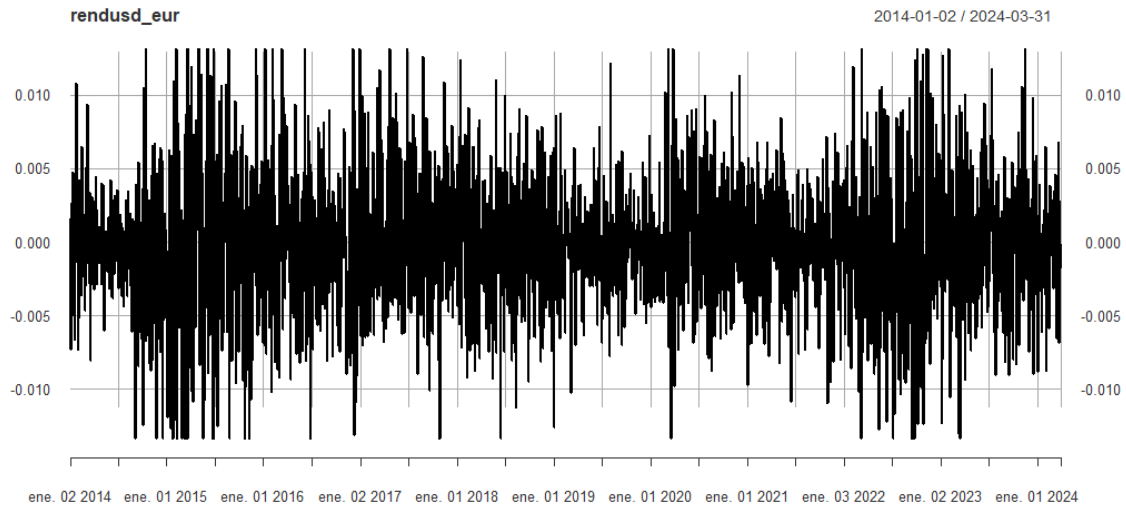


Tabla 3. Tests de estacionariedad serie rendimientos

Test	Val. Crítico	Estadístico	H0	H1	Resultado
Ljung-Box (lag =5)	0.05	0.8344	Estacionaria	No estacionaria	H0
Ljung-Box (lag =10)	0.05	0.01524	Estacionaria	No estacionaria	H1
Ljung-Box (lag =20)	0.05	0.1156	Estacionaria	No estacionaria	H0
D-F test (5%)	-1.95	-36.9599	No estacionaria	Estacionaria	H1
P-P test (5%)	-2.8631	-52.5827	No estacionaria	Estacionaria	H1
KPSS test (5%)	0.463	0.1666	Estacionaria	No estacionaria	H0

Los resultados de las pruebas Ljung-Box(excepto lag=10), Dickey-Fuller, Phillips-Perron y KPSS indican que la serie es estacionaria después de aplicar la diferenciación logarítmica. Esto significa que ahora la serie de retornos tiene propiedades estadísticas (media y varianza) constantes en el tiempo, lo que es un requisito esencial para aplicar un modelo GARCH.

4.2- Modelización de la media

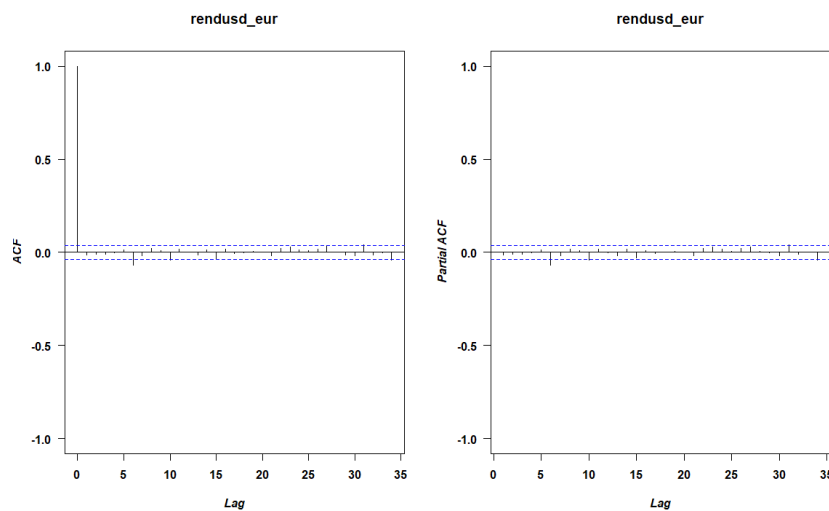
Modelizar la media es crucial porque los modelos GARCH están diseñados específicamente para capturar heterocedasticidad condicional, es decir, cambios en la varianza condicional a lo largo del tiempo. Sin embargo, estos modelos no están diseñados para capturar dependencias lineales en los datos. Si no se aborda primero la media, los residuos del modelo pueden contener estructuras autocorrelativas que no son

adecuadamente modeladas por GARCH, lo que puede llevar a resultados incorrectos y una mala especificación del modelo de volatilidad.

Al modelizar la media, se obtienen residuos que idealmente deben comportarse como ruido blanco, es decir, una serie de valores aleatorios con media cero y varianza constante. Esto es esencial porque los modelos GARCH asumen que los residuos son ruido blanco y cualquier cambio en la varianza condicional es lo que el modelo está tratando de capturar. Por lo tanto, al asegurarse de que los residuos del modelo de media son ruido blanco, se mejora la precisión en la estimación de los parámetros del modelo GARCH y se garantiza que el modelo captura correctamente la dinámica de la varianza condicional.

Para modelizar la media vamos a emplear un modelo tipo ARMA, porque este modelo es capaz de capturar y eliminar la dependencia lineal en los datos. Esto asegura que cualquier estructura autocorrelativa se maneje adecuadamente, proporcionando residuos ideales para la aplicación del modelo GARCH.

Gráfico 4. ACF y PACF serie rendimientos



A partir del gráfico de ACF y PACF es difícil sugerir algún tipo de modelo a ajustar. Podríamos decir que se ajusta mejor a un ARMA(1,1) por el hecho de que el comportamiento de ACF y PACF es muy similar, pero para obtener un resultado que nos ofrezca más confianza, vamos a modelizar la media en todos los modelos más comunes y ver cuál de estos nos proporciona un AIC más pequeño.

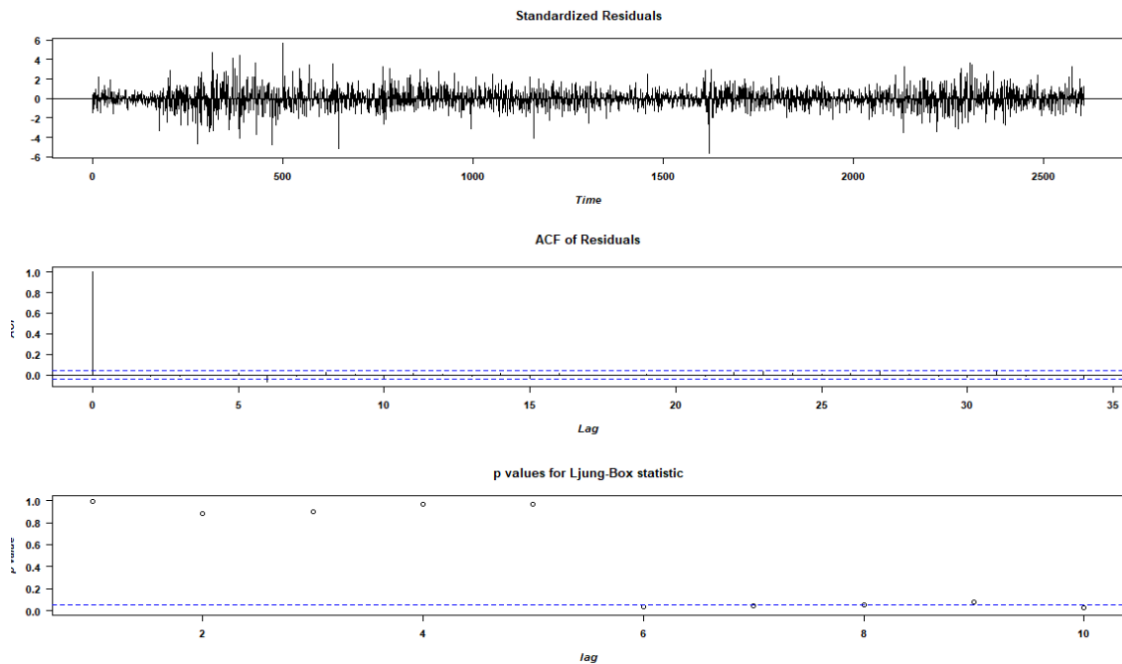
Observando la tabla, el modelo MA(1) parece ser el mejor modelo para esta serie temporal de retornos, ya que tiene los valores de AIC y BIC más bajos, lo que sugiere que proporciona el mejor ajuste.

Tabla 4. AIC y BIC de modelización de media propuesta

Modelo	AIC	BIC
AR(1)	-21073.16	-21055.48
MA(1)	-21073.17	-21055.49
ARMA(1,1)	-21071.16	-21047.6
AR(2)	-21071.4	-21047.84
MA(2)	-21071.43	-21047.86

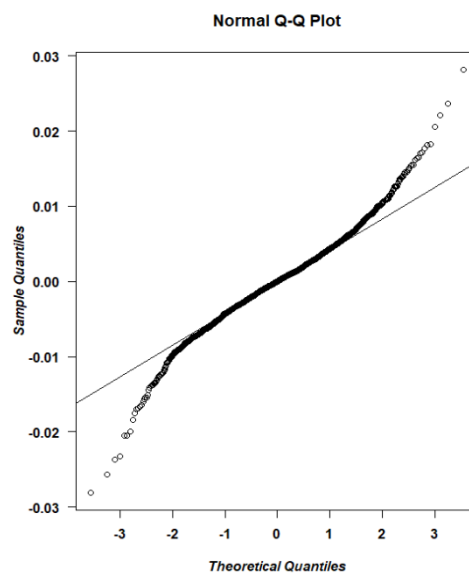
Obtenemos los residuos estandarizado del modelo MA(1)

Gráfico 5. Residuos estandarizado de MA(1)



Los residuos estandarizados no siguen ningún patrón, lo que indica que el modelo MA(1) ha capturado adecuadamente la estructura de la media de la serie temporal. El ACF de los residuos no presenta autocorrelación significativa, ya que todas las autocorrelaciones están dentro de los intervalos de confianza. Los resultados de la prueba de Ljung-Box muestran valores p mayores de 0.05 en casi todos los lags, lo que nos lleva a aceptar la hipótesis nula de no autocorrelación significativa. Por lo tanto, podemos afirmar que los residuos no presentan autocorrelación y que se comportan como ruido blanco.

Gráfico 6. Q-Q Plot de residuos

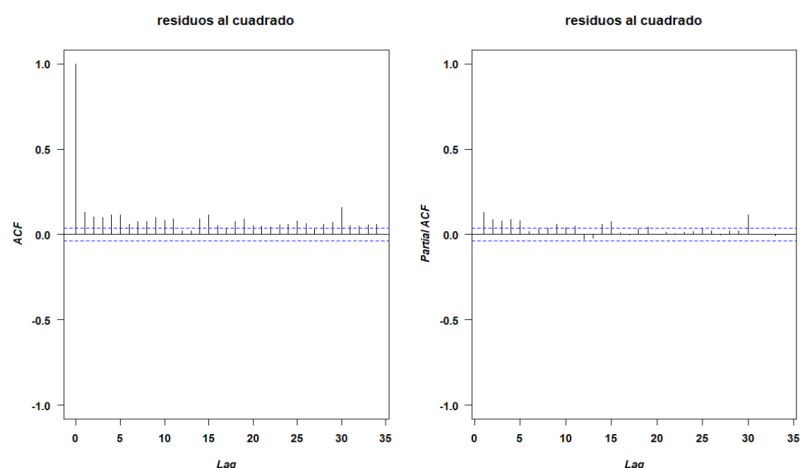


En el Q-Q plot vemos claramente que los residuos tienen una cola pesada, lo que nos indica que no siguen una distribución normal. Realizamos una prueba de normalidad de

Jarque-Bera y obtenemos un valor p asintótico menor a $2.2e-16$, lo que afirma nuestra conclusión.

Ahora vamos a comprobar si los residuos al cuadrado presentan autocorrelación o no. En el caso afirmativo, esto indicaría que la varianza varía a lo largo del tiempo, lo cual es necesario para aplicar una modelización GARCH.

Gráfico 7. ACF y PACF de residuos al cuadrado



Al observar los gráficos ACF y PACF de los residuos al cuadrado, podemos notar que hay autocorrelaciones significativas en el primer rezago y algunas autocorrelaciones en rezagos posteriores que también exceden el intervalo de confianza. Esto sugiere que existe autocorrelación en los residuos al cuadrado, indicando que la varianza de los retornos varía a lo largo del tiempo. Esta variación de la varianza a lo largo del tiempo es un requisito necesario para aplicar un modelo GARCH, sugiriendo que dicho modelo puede ser adecuado para capturar la dinámica de la volatilidad condicional en la serie temporal.

4.3- Modelización de varianza: GARCH

En este apartado, vamos a modelizar algunos de los modelos GARCH más comunes y seleccionaremos el modelo que mejor se ajuste a los datos utilizando AIC. Asumiremos una distribución t de Student en todos los casos, ya que, como se ha observado en apartados anteriores, la serie de retornos no sigue una distribución normal y presenta colas pesadas. Por lo tanto, la distribución t de Student es más adecuada para la serie debido a su capacidad para capturar estas colas pesadas.

Después de modelizar algunos modelos GARCH con diferentes parámetros, finalmente escogemos GARCH(1,1) porque tiene un AIC más pequeño y por su simplicidad.

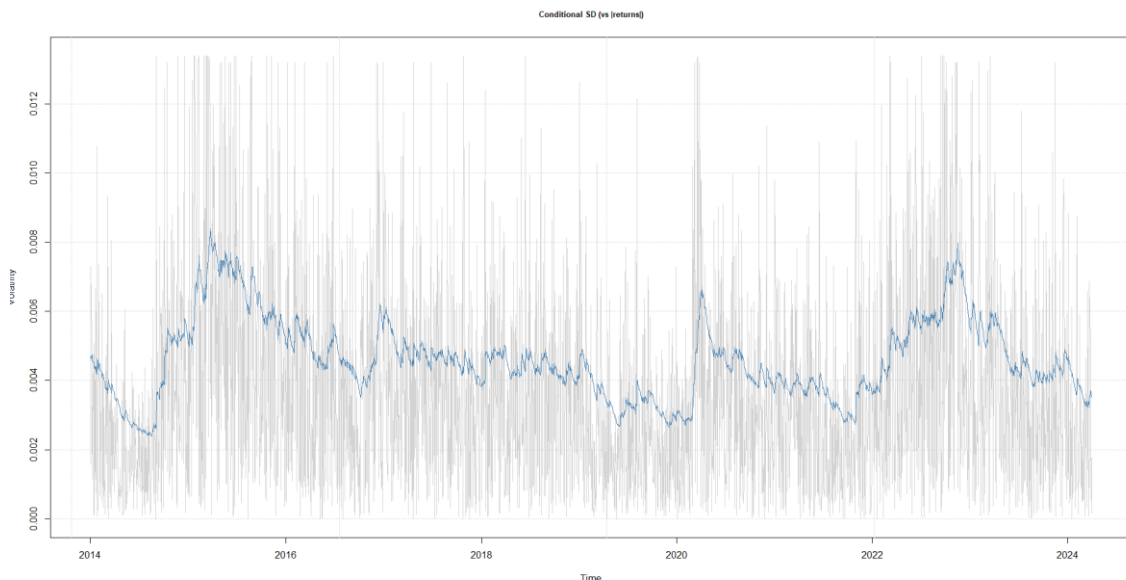
- GARCH(1,1); Akaike: -7.9932

Gráfico 8. Modelo GARCH(1,1)

```
Optimal Parameters
-----
      Estimate  Std. Error  t value  Pr(>|t|)
mu      -0.000111  0.000077  -1.44439  0.148629
ma1     -0.023485  0.019993  -1.17464  0.240137
omega   0.000000  0.000000  0.17223  0.863257
alpha1  0.039021  0.005206  7.49608  0.000000
beta1   0.958116  0.004697  203.97987  0.000000
shape  13.821435  4.086472  3.38224  0.000719
```

Los parámetros óptimos estimados del modelo GARCH(1,1) se muestran en la tabla. Se puede observar que los coeficientes de α_1 y β_1 son altamente significativos con p-valores muy bajos, lo que indica que estos parámetros son aptos para capturar la volatilidad condicional de la serie.

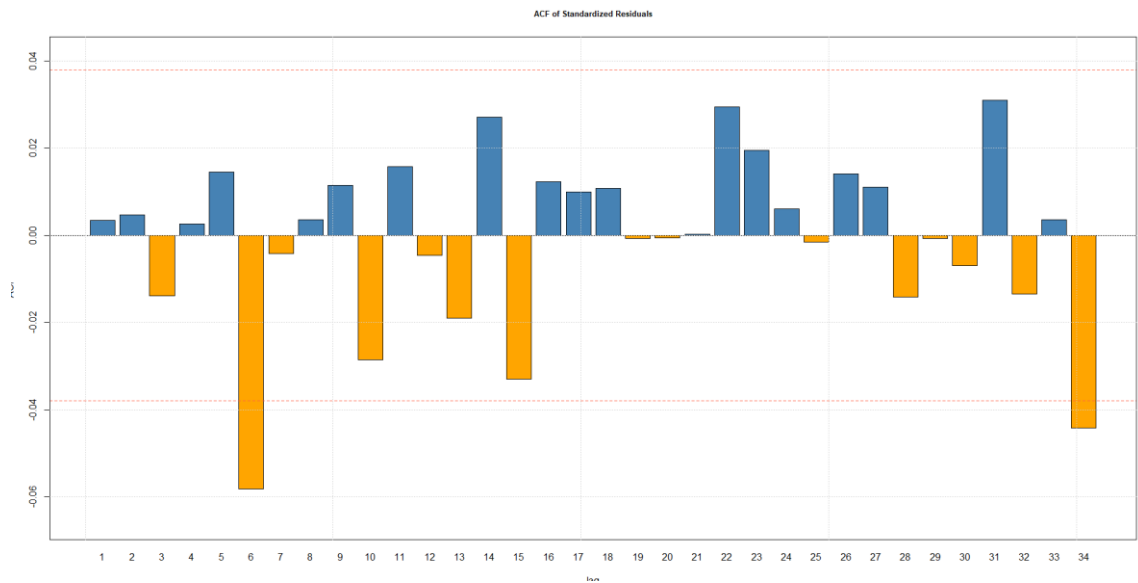
Gráfico 9. Volatilidad Condicional vs Retornos



El gráfico de la desviación estándar condicional muestra la volatilidad estimada a lo largo del tiempo. Se pueden identificar periodos de alta volatilidad, como en 2015 y 2020, que coinciden con eventos económicos y políticos importantes que afectaron la cotización del USD/EUR. Este patrón confirma la capacidad del modelo GARCH(1,1) para capturar los cambios en la volatilidad a lo largo del tiempo.

El gráfico de la función de autocorrelación (ACF) de los residuales estandarizados nos permite evaluar la calidad del ajuste del modelo. Idealmente, no debería haber autocorrelación significativa en los residuales estandarizados si el modelo GARCH ha capturado adecuadamente la estructura de la volatilidad. En este caso, aunque hay algunas autocorrelaciones significativas pero solo en lag 6 y lag 34, no afectaría a la interpretación de los resultados, además la mayoría de los residuos no presentan patrones claros de autocorrelación, lo que sugiere que el modelo GARCH(1,1) ha capturado bien la dinámica de la volatilidad de la serie.

Gráfico 10. ACF de los residuos estandarizados



4.4- Predicción

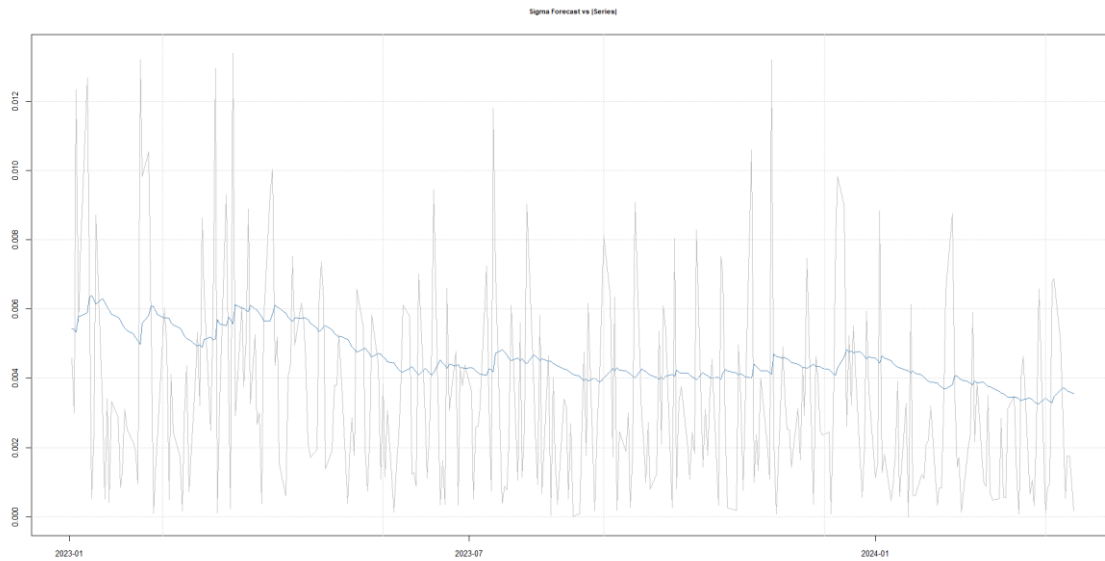
Para la predicción se emplea una metodología dinámica de ventana deslizante dejando 326 observaciones fuera (correspondiente de 2023 a 01/04/2024). Este método implica la reestimación continua del modelo utilizando una ventana deslizante de datos.

El proceso de predicción se realiza utilizando una ventana deslizante, reestimando el modelo en cada paso de predicción. En cada iteración, la ventana de datos se mueve hacia adelante, incluyendo las observaciones más recientes y excluyendo las más antiguas. Este enfoque asegura que, en cada paso, el modelo se reestima con la información más actualizada.

En este caso, hemos establecido que se reestimaré el modelo cada 10 pasos. Durante cada reestimación, se realiza una actualización completa del modelo con las observaciones dentro de la ventana deslizante. Por ejemplo, para la primera predicción fuera de muestra, se utilizan las observaciones desde el inicio hasta el punto donde comienza el período de validación, menos una observación. Para la siguiente reestimación, la ventana se mueve 10 observaciones hacia adelante, incluyendo las nuevas observaciones más recientes y excluyendo las más antiguas en el límite de la ventana. Este proceso continúa hasta que se hayan realizado todas las predicciones fuera de muestra.

Los resultados obtenidos son:

Gráfico 11. Volatilidad estimada por GARCH



La línea azul, que representa la volatilidad condicional predicha, sigue de cerca la volatilidad observada en la serie de retornos, capturando adecuadamente las fluctuaciones y picos de alta volatilidad.

Se observa una tendencia descendente en la volatilidad predicha, indicando una disminución en la incertidumbre del mercado a lo largo del tiempo. La predicción muestra menos variabilidad que la serie original, lo cual es típico en modelos GARCH que suavizan la volatilidad para proporcionar una medida más estable.

5- Volatilidad realizada.

Para obtener la volatilidad realizada, utilizaremos datos del tipo de cambio USD/EUR con una frecuencia de 5 minutos para el período de 2020 a 2024. Los datos han sido extraídos de la base de datos de Barchart. A partir de la serie de tipo de cambio, aplicaremos una diferenciación logarítmica para obtener los rendimientos a intervalos de 5 minutos. Luego, sumaremos los rendimientos de 5 minutos de cada día para obtener la varianza realizada, y al aplicar la raíz cuadrada, obtendremos finalmente la volatilidad realizada.

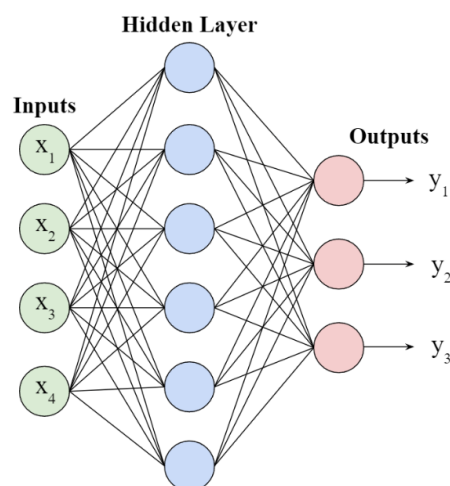
5.1- Modelo de predicción: NNAR.

Para la predicción utilizaremos el modelo NNAR. La elección de este modelo se debe a que la aplicación de Machine Learning en el ámbito econométrico está ganando cada vez más importancia, y en muchos casos, los resultados superan a los de los modelos tradicionales. Existen diversas variantes de modelos como MLP, LSTM, y NNAR. En este caso, utilizaremos NNAR debido a su relativa simplicidad en comparación con otros modelos.

La regresión autoregresiva con redes neuronales (NNAR) es una avanzada metodología de pronóstico que aprovecha la capacidad de las redes neuronales artificiales para modelar series temporales. A diferencia de los métodos tradicionales, NNAR puede capturar relaciones no lineales complejas entre una variable de respuesta y sus predictores, proporcionando una herramienta potente y flexible para el análisis de datos temporales.

Una red neuronal se visualiza como un conjunto de "neuronas" organizadas en capas. Los predictores constituyen la capa inferior, y las predicciones forman la capa superior. Puede haber capas intermedias que contienen "neuronas ocultas".

Gráfico 12. Ilustración de funcionamiento de NNAR



Un modelo NNAR(p,k) utiliza los últimos p valores observados como entradas para pronosticar la salida actual, y cuenta con k neuronas en la capa oculta. Un modelo NNAR(p,0) sería equivalente a un modelo ARIMA(p,0,0), pero sin las restricciones en los parámetros para asegurar la estacionariedad.

La predicción se realiza combinando técnicas de redes neuronales con autoregresión. Se preparan los datos seleccionando valores pasados (lagos) de la serie como entradas. La estructura del modelo incluye p capa de entrada para los datos de entrada, k capa oculta que permite aprender patrones no lineales mediante funciones de activación como la sigmoidea, y una capa de salida que produce la predicción. El modelo se entrena minimizando errores, ajustando pesos mediante algoritmos como el descenso del gradiente. Para pronósticos de un solo paso, se usa directamente la salida del modelo; para múltiples pasos, se reutilizan las predicciones como nuevas entradas. La efectividad del modelo se evalúa mediante pruebas en datos no vistos durante el entrenamiento.

5.2. Modelización:

Antes de comenzar con la modelización necesitamos comprobar la linealidad de la serie de volatilidad realizada, emplearemos el BDS Test y la prueba de Red Neuronal de Teraesvirta. Los resultados son:

Tabla 5. Resultado de BDS y TNN

```

Title:
  BDS Test

Test Results:
  PARAMETER:
    Max Embedding Dimension: 3
    eps [1]: 0.001
    eps [2]: 0.002
    eps [3]: 0.003
    eps [4]: 0.004
  STATISTIC:
    eps [1] m=2: 20.0803
    eps [1] m=3: 25.1689
    eps [2] m=2: 17.6
    eps [2] m=3: 18.9864
    eps [3] m=2: 17.3104
    eps [3] m=3: 18.2063
    eps [4] m=2: 14.6261
    eps [4] m=3: 15.3937
  P VALUE:
    eps [1] m=2: < 2.2e-16
    eps [1] m=3: < 2.2e-16
    eps [2] m=2: < 2.2e-16
    eps [2] m=3: < 2.2e-16
    eps [3] m=2: < 2.2e-16
    eps [3] m=3: < 2.2e-16
    eps [4] m=2: < 2.2e-16
    eps [4] m=3: < 2.2e-16
  
```

```

Title:
  Teraesvirta Neural Network Test

Test Results:
  PARAMETER:
    lag: 1
    m|df: 2
    t-lag-m|df: 828
  STATISTIC:
    Chi-squared: 16.7514
    F: 8.4301
  P VALUE:
    Chi-squared: 0.0002304
    F: 0.0002375
  
```

Todos los estadísticos BDS para diferentes dimensiones de embedding ($m=2$ y $m=3$) y diferentes valores de eps muestran valores muy altos con p-valores extremadamente bajos ($< 2.2e-16$), lo que sugiere un rechazo fuerte de la hipótesis nula de linealidad. Esto indica

que hay una estructura no lineal significativa en los datos analizados. La misma conclusión obtenemos con la prueba de Red Neuronal de Teraesvirta, con estadísticos de Chi-cuadrado y F significativos y p-valores muy bajos, esta prueba también rechaza la hipótesis de linealidad. A partir de las pruebas podemos afirmar que la serie de volatilidad realizada no presenta linealidad.

Separamos la serie en dos periodos, uno que va de 06/08/2021 a 31/12/2023 y otro que va de 01/01/2024 a 01/04/2024, el primero seria la serie de train y el segundo la serie de test.

Se ha modelizado NNAR con diferente número de k (neuronas ocultas) con diferentes números de iteraciones, para valorar cual es el modelo que ajusta mejor a la serie se ha utilizado MSE como criterio de valoración. El resultado es emplear 15 neuronas ocultas y haciendo 20 repeticiones con un MSE de 1.208022e-06 (la tabla con MSE correspondiente a cada modelo realizado se encuentra en anexo).

Tabla 6. NNAR resultante

```
Series: usd_train
Model: NNAR(25,15)
Call: nnetar(y = usd_train, size = 15, repeats = 20)

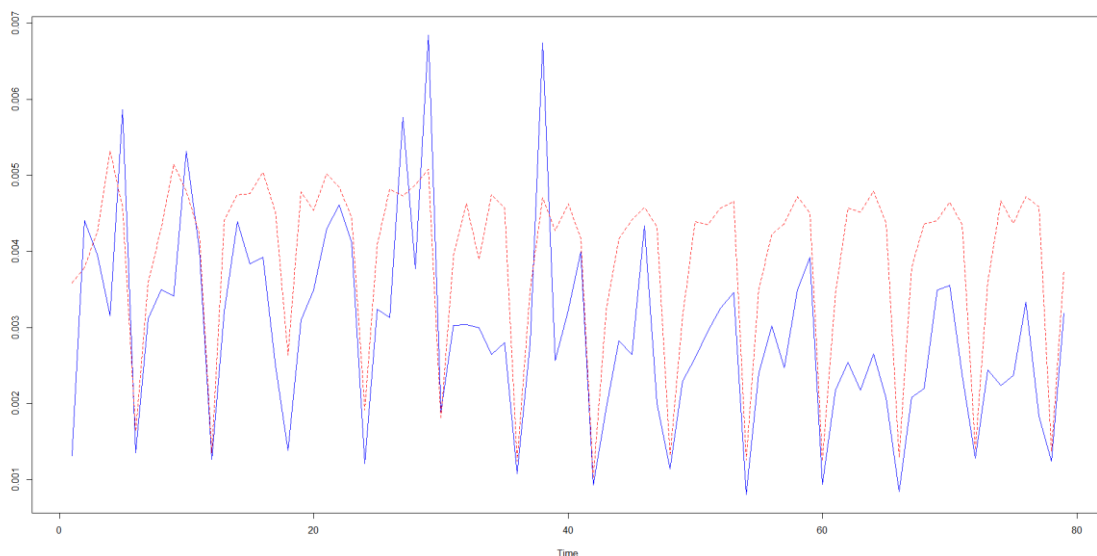
Average of 20 networks, each of which is
a 25-15-1 network with 406 weights
options were - linear output units

sigma^2 estimated as 9.963e-08
```

El modelo resultante es un NNAR (25,15), que utiliza 25 valores anteriores como entradas y tiene una capa oculta con 15 neuronas. Se ha entrenado 20 veces con diferentes inicializaciones de pesos, y las predicciones finales son el promedio de estos 20 modelos. La red tiene un total de 406 pesos ajustables y utiliza unidades de salida lineales.

5.3- Predicción:

Gráfico 13. Predicciones de NNAR de 01/01/2024 a 01/04/2024



El gráfico muestra las predicciones de volatilidad propuestas por nuestro modelo NNAR (25,15) para el período del 01/01/2024 al 01/04/2024. El modelo ha demostrado ser efectivo al capturar los movimientos de la volatilidad en la serie temporal del USD/EUR. La capacidad del modelo para predecir picos de alta volatilidad y su retorno a niveles normales sugiere que está capturando bien las dinámicas subyacentes del mercado.

6- Comparación entre dos modelos. Aplicación en VaR:

En este apartado, vamos a calcular el VaR al 5% y al 1% para el año 2024. Dado que las series de los dos modelos provienen de mercados diferentes, puede haber días en que un mercado esté operativo y el otro no. En caso de discrepancia, vamos a sustituir el valor ausente por su valor inmediatamente anterior.

El resultado del VaR pronosticado es el siguiente:

Gráfico 14. VaR estimado

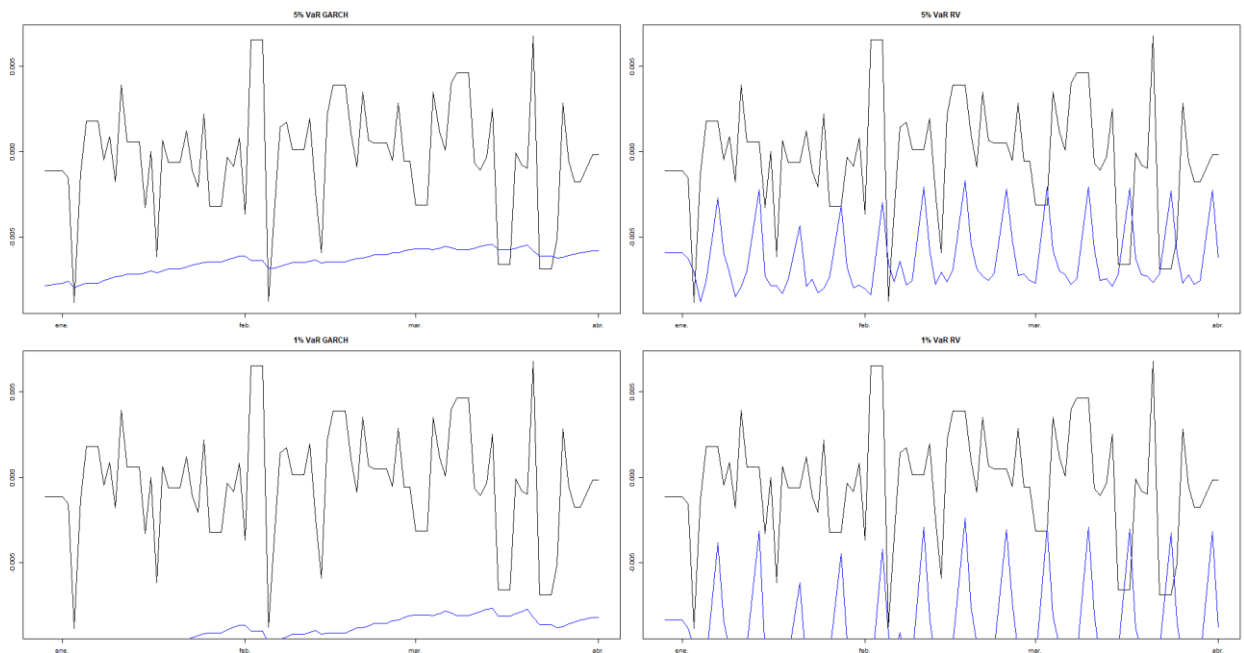


Tabla 7. % de casos que excede a VaR

	VaR(5%)	VaR(1%)
sGarch(1,1)	7.5%	0
RV	7.5%	3.75%

Los gráficos muestran las predicciones del VaR al 5% y al 1% utilizando los modelos GARCH y RV. En el primer conjunto de gráficos, se observa el VaR al 5% con las predicciones del modelo GARCH a la izquierda y del modelo RV a la derecha. De manera similar, en el segundo conjunto de gráficos, se muestra el VaR al 1% para ambos modelos.

Los resultados indican que el modelo GARCH predice de manera más conservadora en comparación con el modelo RV. Esto se evidencia en el hecho de que el VaR del 5% y del 1% predicho por GARCH es generalmente menor en magnitud y menos variable que el del modelo RV. Este comportamiento sugiere que el modelo GARCH es más efectivo para

capturar los picos de volatilidad, mientras que el modelo RV puede ser más sensible a las fluctuaciones diarias. Posiblemente es debido a que RV está modelizado a partir de datos de alta frecuencia, y en cambio el GARCH está modelizado por datos diarios.

La tabla proporciona una comparación cuantitativa del porcentaje de casos en los que la pérdida excede el VaR predicho. Para el VaR al 5%, ambos modelos tienen un 7.5% de casos que exceden el límite, lo que indica una precisión similar en este nivel de confianza. Sin embargo, para el VaR al 1%, el modelo GARCH no tiene ningún caso que exceda el límite, mientras que el modelo RV tiene un 3.75% de excedencias. Esto sugiere que el modelo GARCH es más adecuado para predecir eventos de cola más extrema, proporcionando una mayor seguridad en situaciones de alta volatilidad.

El hecho de que el modelo GARCH sea más conservador puede tener implicaciones significativas para la gestión del capital en una empresa. Si bien este conservadurismo ayuda a garantizar que la empresa esté adecuadamente protegida contra eventos de riesgo extremos, también puede resultar en la reserva excesiva de capital en muchas ocasiones. Esta sobreestimación de la volatilidad y del riesgo puede llevar a la empresa a mantener niveles más altos de capital de lo necesario, lo que podría ser subóptimo desde el punto de vista de la eficiencia del uso del capital.

Mantener capital excesivo puede limitar la capacidad de la empresa para invertir en otras oportunidades de crecimiento o para utilizar los fondos de manera más productiva. Por lo tanto, es importante encontrar un equilibrio entre la prudencia en la gestión del riesgo y la eficiencia del capital. Mientras que el modelo GARCH ofrece una mayor seguridad y puede ser más adecuado en periodos de alta volatilidad, el modelo RV, que es menos conservador, podría ser útil en entornos de mercado más estables, permitiendo a la empresa liberar capital para otras iniciativas estratégicas.

7- Conclusión y futuras líneas de investigación.

En este estudio se ha modelizado la volatilidad del tipo de cambio USD/EUR utilizando dos enfoques principales: el modelo GARCH(1,1) y un modelo de regresión autorregresiva con redes neuronales (NNAR) con estructura NNAR(25,15). Ambos modelos han mostrado su capacidad para predecir la volatilidad, proporcionando información valiosa para la gestión del riesgo.

El modelo GARCH(1,1), ha demostrado ser efectivo en capturar la dinámica temporal y la agrupación de la volatilidad. Sin embargo, su naturaleza conservadora puede llevar a una sobreestimación de la volatilidad y que en ocasiones se llega a una reserva excesiva de capital. Este enfoque es robusto y proporciona mayor seguridad en situaciones de alta volatilidad, aunque puede resultar en una menor eficiencia del capital en mercados más estables.

Por otro lado, el modelo NNAR(25,15) ha mostrado una notable capacidad para capturar patrones no lineales complejos en los datos, ofreciendo predicciones precisas de la volatilidad realizada. Este modelo se basa en datos de alta frecuencia, lo que le permite reflejar de manera más precisa los movimientos intradías del mercado. La flexibilidad del NNAR lo hace especialmente útil en condiciones de mercado dinámicas y cambiantes.

Al evaluar el Valor en Riesgo (VaR) al 5% y al 1%, se encontró que ambos modelos tienen sus propias ventajas y limitaciones. Para el VaR al 5%, ambos modelos tuvieron un 7.5% de casos que excedieron el límite, lo que indica una precisión similar en este nivel de confianza. Sin embargo, para el VaR al 1%, el modelo GARCH no tuvo ningún caso que excediera el límite, mientras que el modelo NNAR tuvo un 3.75% de excedencias. Esto sugiere que el modelo GARCH es más adecuado para predecir eventos extremos, proporcionando una mayor seguridad en condiciones de alta volatilidad, mientras que el modelo NNAR puede ofrecer una mayor eficiencia en condiciones de mercado más estables.

Este trabajo presenta ciertas limitaciones. Hubo discrepancias en el formato de los datos debido a que se extrajeron de diferentes fuentes. Hubo cierta dificultad en la búsqueda de datos, no se pudo encontrar datos de cotizaciones a frecuencia 10 min para un periodo más allá de 2021 a 2024, lo que llevó a la necesidad de integrar y armonizar datos de distintos formatos y frecuencias, que podría causar cierta pérdida de información y pudo introducir inconsistencias que afectan la precisión de los modelos.

Para la futura línea de investigación, se podrían explorar varias líneas de investigación adicionales. En primer lugar, sería interesante realizar un análisis comparativo que incluya la volatilidad implícita derivada del modelo Black-Scholes (BS). Comparar la volatilidad implícita con las predicciones de los modelos GARCH y NNAR podría proporcionar una perspectiva más completa sobre la precisión y la utilidad práctica de estos enfoques. La volatilidad implícita es una medida anticipada de la volatilidad futura basada en los precios de las opciones, y su inclusión en el análisis podría mejorar significativamente la comprensión del riesgo.

Además, la investigación podría expandirse para incluir modelos multivariantes como VAR (Vector Autoregression) y GARCH multivariante. Estos modelos permitirían capturar las interdependencias entre múltiples series temporales financieras, proporcionando una visión más holística de la volatilidad en los mercados de divisas y mejorando la capacidad predictiva al considerar la influencia de otros factores económicos.

Otra línea de investigación prometedora sería la exploración de modelos alternativos de Machine Learning. Modelos como XGBoost, redes neuronales convolucionales (CNN) y modelos de transformers podrían ofrecer mejoras adicionales en la precisión de las predicciones al capturar patrones complejos y no lineales en los datos financieros. Comparar estos modelos con los enfoques GARCH y NNAR actuales podría revelar nuevas oportunidades para optimizar las estrategias de gestión del riesgo.

Trabajar con diferentes frecuencias de datos para los modelos GARCH también podría ser interesante. Analizar cómo se comportan los modelos GARCH cuando se aplican a datos de alta frecuencia, como cotizaciones cada minuto, en comparación con datos diarios o semanales, podría proporcionar información valiosa sobre la sensibilidad de estos modelos a la granularidad de los datos y su capacidad para adaptarse a diferentes horizontes temporales.

8- Bibliografia:

- Adhikari, R., & Agrawal, R.K. (2013). "An Introductory Study on Time Series Modeling and Forecasting Using ARIMA Model". arXiv preprint arXiv:1302.6613.
- Andersen, T. G., & Bollerslev, T. "Deutsche Mark–Dollar Volatility: Intraday Activity Patterns, Macroeconomic Announcements, and Longer Run Dependencies." DOI: 10.2307/2330957.
- Andersen, T. G., Bollerslev, T., Diebold, F. X., & Labys, P. (2003). Modeling and Forecasting Realized Volatility. *Econometrica*, 71(2), 579–625. <https://doi.org/10.1111/1468-0262.00418>.
- Bao, W., Yue, J., & Rao, Y. (2017). "A deep learning framework for financial time series using stacked autoencoders and long short-term memory". *PLoS One*, 12(7), e0180944.
- Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31(3), 307-327. [https://doi.org/10.1016/0304-4076\(86\)90063-1](https://doi.org/10.1016/0304-4076(86)90063-1).
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control*. John Wiley & Sons.
- Doucoure, B., Tkacz, G., et al. (2022). Modelling the GDP of KSA using linear and non-linear NNAR and hybrid stochastic time series models. *PLOS ONE*. <https://doi.org/10.1371/journal.pone.0297180>.
- Duan, J. C. "THE GARCH OPTION PRICING MODEL." DOI: 10.2139/ssrn.987654.
- Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50(4), 987-1007. <https://doi.org/10.2307/1912773>.
- Hansen, P. R., & Lunde, A. (2005). A Comparison of Volatility Models: Does Anything Beat a GARCH(1,1)? *Journal of Applied Econometrics*, 20(7), 873-889. <https://doi.org/10.1002/jae.800>.
- Hochreiter, S., & Schmidhuber, J. (1997). "Long short-term memory". *Neural Computation*, 9(8), 1735-1780.
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice* (3rd ed.). OTexts. Disponible en: <https://otexts.com/fpp3/>.
- Kim, K.J. (2003). "Financial time series forecasting using support vector machines". *Neurocomputing*, 55(1-2), 307-319.
- Kourtis, A., et al. "An International Comparison of Implied, Realized, and GARCH Volatility Forecasts." DOI: 10.2139/ssrn.2171325.
- Lee. "Are the GARCH models best in out-of-sample performance?" DOI: 10.2139/ssrn.1234567.
- Sheraz, M., & Preda, A. "Implied Volatility in Black-Scholes Model with Garch Volatility." DOI: 10.2139/ssrn.345678.
- Tsay, R. S. (2010). *Analysis of Financial Time Series* (3rd ed.). Wiley.

Yin, X., & Han, Y. "Hedging International Foreign Exchange Risks via Option Based Portfolio Insurance." DOI: 10.2139/ssrn.567890.

Zhang, G.P. (2003). "Forecasting financial time series using neural networks: A comprehensive review". *Neural Networks*, 14(1), 45-76.

Zhou, J., & Wang, Y. "Managing foreign exchange risk with derivatives in UK non-financial firms." DOI: 10.2139/ssrn.234567.

Liu, Y., & Morley, B. "Volatility Forecasting in the Hang Seng Index using the GARCH Approach." DOI: 10.2139/ssrn.765432.

9- Anexo:

Diferentes GARCH modelizados:

```
*-----*
*          GARCH Model Fit          *
*-----*

Conditional Variance Dynamics
-----
GARCH Model      : gjrGARCH(1,1)
Mean Model       : ARFIMA(0,0,1)
Distribution      : std

Optimal Parameters
-----
      Estimate  Std. Error   t value  Pr(>|t|)
mu        -0.000127  0.000078  -1.64121  0.100755
ma1       -0.022479  0.019992  -1.12443  0.260830
omega     0.000000  0.000000   0.14865  0.881830
alpha1    0.030092  0.006122   4.91510  0.000001
beta1     0.960817  0.003673 261.55653  0.000000
gamma1    0.014446  0.008845   1.63331  0.102404
shape    13.925237  3.930796   3.54260  0.000396

Robust Standard Errors:
      Estimate  Std. Error   t value  Pr(>|t|)
mu        -0.000127  0.000075  -1.703483  0.088478
ma1       -0.022479  0.022057  -1.019161  0.308127
omega     0.000000  0.000009   0.006487  0.994824
alpha1    0.030092  0.079477   0.378623  0.704967
beta1     0.960817  0.065851  14.590820  0.000000
gamma1    0.014446  0.017844   0.809574  0.418185
shape    13.925237  50.923217   0.273456  0.784503

LogLikelihood : 10689.93

Information Criteria
-----
Akaike          -7.9932
Bayes           -7.9778
shibata         -7.9932
Hannan-Quinn   -7.9876
```

```

*-----*
*           GARCH Model Fit           *
*-----*

Conditional Variance Dynamics
-----
GARCH Model      : eGARCH(1,1)
Mean Model       : ARFIMA(0,0,1)
Distribution      : std

Optimal Parameters
-----
      Estimate  Std. Error   t value Pr(>|t|)
mu      -0.000130   0.000077   -1.6809 0.092790
ma1     -0.020763   0.019837   -1.0467 0.295252
omega   -0.066574   0.001157  -57.5373 0.000000
alpha1  -0.016652   0.007853   -2.1204 0.033972
beta1    0.993821   0.000042 23667.6152 0.000000
gamma1   0.082869   0.006006   13.7975 0.000000
shape   14.933495   3.086919    4.8377 0.000001

Robust Standard Errors:
      Estimate  Std. Error   t value Pr(>|t|)
mu      -0.000130   0.000074   -1.75306 0.079591
ma1     -0.020763   0.021117   -0.98321 0.325503
omega   -0.066574   0.001270  -52.42214 0.000000
alpha1  -0.016652   0.008524   -1.95363 0.050745
beta1    0.993821   0.000054 18556.35133 0.000000
gamma1   0.082869   0.007198   11.51338 0.000000
shape   14.933495   3.464566    4.31035 0.000016

LogLikelihood : 10686.25

Information Criteria
-----
Akaike          -7.9905
Bayes           -7.9750
Shibata         -7.9905
Hannan-Quinn   -7.9849

```

```

*-----*
*           GARCH Model Fit           *
*-----*

Conditional Variance Dynamics
-----
GARCH Model      : sGARCH(1,2)
Mean Model       : ARFIMA(1,0,0)
Distribution      : std

Optimal Parameters
-----
      Estimate  Std. Error   t value  Pr(>|t|)
mu      -0.000110   0.000077  -1.42270  0.154824
ar1     -0.019639   0.019686  -0.99761  0.318468
omega    0.000000   0.000000   0.19581  0.844755
alpha1   0.039452   0.003098  12.73397  0.000000
beta1    0.954391   0.000833 1145.62418  0.000000
beta2    0.004169   0.000681   6.12172  0.000000
shape   14.376906   4.219346   3.40738  0.000656

Robust Standard Errors:
      Estimate  Std. Error   t value  Pr(>|t|)
mu      -0.000110   0.000074  -1.479591  0.13898
ar1     -0.019639   0.021128  -0.929536  0.35261
omega    0.000000   0.000006   0.010932  0.99128
alpha1   0.039452   0.033791   1.167519  0.24300
beta1    0.954391   0.017137  55.691864  0.00000
beta2    0.004169   0.012113   0.344143  0.73074
shape   14.376906  37.886276   0.379475  0.70434

LogLikelihood : 10688.76

Information Criteria
-----
Akaike          -7.9923
Bayes           -7.9769
shibata        -7.9924
Hannan-Quinn  -7.9868

```


Rendimiento según MSE de NNAR modelizados:

```
> print(results)
  Size Repeats      MSE
19  15      20 1.208022e-06
16  10      20 1.242683e-06
 8   4      50 1.266699e-06
18  10     100 1.297929e-06
17  10      50 1.301860e-06
13   8      20 1.307156e-06
 9   4     100 1.308093e-06
20  15      50 1.355522e-06
15   8     100 1.367964e-06
12   6     100 1.374809e-06
11   6      50 1.376415e-06
21  15     100 1.388775e-06
23  20      50 1.401037e-06
14   8      50 1.406267e-06
 7   4      20 1.414502e-06
10   6      20 1.423953e-06
 4   2      20 1.462515e-06
22  20      20 1.463230e-06
26  30      50 1.508351e-06
24  20     100 1.508712e-06
 6   2     100 1.518545e-06
 5   2      50 1.523551e-06
25  30      20 1.566740e-06
27  30     100 1.581114e-06
 3   1     100 2.209004e-06
 1   1      20 2.253782e-06
 2   1      50 2.261768e-06
```

Código R:

```
##HAODONG DU##
```

```
#####Comparación de Modelos GARCH y Volatilidad Realizada en la Predicción de la  
Volatilidad del Tipo de Cambio USD/EUR.
```

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

```
install.packages("tseries")
```

```
install.packages("fBasics")
```

```
install.packages("car")
```

```
install.packages("urca")
```

```
install.packages("forecast")
```

```
install.packages("rugarch")
```

```
install.packages("quantmod")
```

```
install.packages("fTrading")
```

```
install.packages("ggplot2")
```

```
library(tseries)
```

```
library(fBasics)
```

```
library(car)
```

```
library(urca)
```

```
library(forecast)
```

```
library(rugarch)
```

```
library(quantmod)
```

```
library(fTrading)
```

```
library(ggplot2)
```

```
getSymbols("EURUSD=X",from="2014-01-01", to="2024-04-01")
usd_eur=`EURUSD=X`[,6]
usd_eur<-na.locf(usd_eur, fromLast = TRUE)
summary(usd_eur)
basicStats(usd_eur)
normalTest(usd_eur,method="jb")
```

###ANALISIS DE ESTACIONARIEDAD DE LA SERIE PRECIO:

##HERRAMIENTAS NO FORMALES

#GRAFICO

```
plot(usd_eur)
```

#EL GRAFICO DEL PRECIO NO PRESENTA ESTACIONARIEDAD

#ACF

```
acf(usd_eur,ylim=c(-1,1),main="usd_eur")
```

```
pacf(usd_eur,ylim=c(-1,1),main="usd_eur")
```

#EL GRAFICO MUESTRA AUTOCORRELACIÓN (NO TIENDE A 0 RAPIDAMENTE),

#LA SERIE NO ES ESTACIONARIA

#LJUNG-BOX TEST

```
Box.test(usd_eur, lag = 5, type = c("Ljung-Box"))
```

```
Box.test(usd_eur, lag = 10, type = c("Ljung-Box"))
```

```
Box.test(usd_eur, lag = 20, type = c("Ljung-Box"))
```

#P-VALUE MENOR QUE 0.05 SE RECHAZA LA h_0 , LA SERIE NO ES ESTACIONARIA

```
##HERAMIENTAS FORMALES
```

```
#DICKY-FULLER TEST
```

```
usd_eur.df<-ur.df(usd_eur, type = c("drift"), lags=2)
```

```
summary(usd_eur.df)
```

```
#NO RECHAZO LA h0(LA SERIE ES NO ESTACIONARIA)
```

```
#PHILLIPS-PERRON TEST
```

```
usd_eur.pp<-ur.pp(usd_eur, type = c("Z-tau"), model = c("constant"), lags = c("long"))
```

```
#0.26 no rechazo h0 i(1).
```

```
summary(usd_eur.pp)
```

```
#NO RECHAZO LA h0(LA SERIE ES NO ESTACIONARIA)
```

```
#KPSS TEST
```

```
usd_eur.kpss<-ur.kpss(usd_eur, type = c("mu"), lags = c("short"))
```

```
summary(usd_eur.kpss)
```

```
#RECHAZO LA h0(i(0) LA SERIE ES ESTACIONARIA)
```

```
###ANALISIS DE LA SERIE DE RENDIMIENTOS
```

```
#####
```

```
rendusd_eur=diff(log(usd_eur))
```

```
rendusd_eur<-rendusd_eur[-1]
```

```
if (!require("DescTools")) {
```

```
  install.packages("DescTools", dependencies = TRUE)
```

```
  library(DescTools)
```

```
}
```

```

# Aplicar Winsorizing con DescTools
rendusd_eur_winsorized <- Winsorize(rendusd_eur, probs = c(0.01, 0.99))

# Graficar original vs. winsorized para comparar
plot(index(rendusd_eur), coredata(rendusd_eur), type = "l", col = "blue", main =
"Original vs Winsorized", ylab = "Values")

lines(index(rendusd_eur_winsorized), coredata(rendusd_eur_winsorized), col = "red")

legend("topright", legend = c("Original", "Winsorized"), col = c("blue", "red"), lty = 1)

rendusd_eur<-Winsorize(rendusd_eur, probs = c(0.01, 0.99))

###HERRAMIENTAS NO FORMALES

#GRAFICO
plot(rendusd_eur)

#ACF
acf(rendusd_eur,ylim=c(-1,1),main="usd_eur")

#NO OBSERVAMOS AUTOCORRELACIÓN, LOS VALORES TIENDEN A 0 RAPIDAMENTE.

#LJUNG-BOX TEST
Box.test(rendusd_eur, lag = 5, type = c("Ljung-Box"))
Box.test(rendusd_eur, lag = 10, type = c("Ljung-Box"))
Box.test(rendusd_eur, lag = 20, type = c("Ljung-Box"))

#SE ACEPTA LA h0 LA SERIE ES ESTACIONARIA

```

##HERRAMIENTAS FORMALES

#DICKY-FULLER TEST

```
rendusd_eur.df<-ur.df(rendusd_eur, type = c("none"), lags=20, selectlags = c("BIC"))  
summary(rendusd_eur.df)
```

#RECHAZO LA H_0 (LA SERIE ES NO ESTACIONARIA), LA SERIE ES ESTACIONARIA.

#PHILLIPS PERRON TEST

```
rendusd_eur.pp<-ur.pp(rendusd_eur, type = c("Z-tau"), model = c("constant"), lags =  
c("long"))  
summary(rendusd_eur.pp)
```

#RECHAZO LA H_0 (LA SERIE ES NO ESTACIONARIA), LA SERIE ES ESTACIONARIA.

#KPSS TEST

```
rendusd_eur.kpss<-ur.kpss(rendusd_eur, type = c("mu"), lags = c("short"))  
summary(rendusd_eur.kpss)
```

#RECHAZO LA $H_0(i(0))$ LA SERIE ES ESTACIONARIA).

##ESTADISTICOS BASICOS:

```
sd(rendusd_eur)
```

```
min(rendusd_eur)
```

```
max(rendusd_eur)
```

```
skewness(rendusd_eur)
```

#-0.5979001 nuestra serie es ligeramente asimétrica hacia izquierda

```
kurtosis(rendusd_eur)
```

#6.168873 la muestra de datos es leptocúrtica, lo que indica una mayor

#concentración de datos en el centro de la distribución y colas más pesada

```

normalTest(rendusd_eur,method="jb")

#h0<-normalidad, p-value 0.00000... no aceptamos h0, la serie de rendimiento no sigue
una DIST.NORMAL

basicStats(rendusd_eur)

hist(rendusd_eur,breaks=20,freq=F, main = 'Histograma de los rendimientos')
curve(dnorm(x, mean=mean(rendusd_eur), sd=sd(rendusd_eur)), col=2, add=T)

####PROPUESTA DE MODELO PARA LA MEDIA

##ANALISIS GRAFICO

par(mfrow=c(1,2),font=2,font.lab=4,font.axis=2,las=1)
acf(rendusd_eur,ylim=c(-1,1),main="rendusd_eur")
pacf(rendusd_eur,ylim=c(-1,1),main="rendusd_eur")

##A NIVEL GRAFICO VEMOS QUE ES MEJOR MODELIZAR UN ARMA

##ANALISIS EMPIRICO

#ar(1) AIC=-21073.16 AICc=-21073.15 BIC=-21055.48
model1 = Arima(rendusd_eur, order = c(1,0,0),include.mean = TRUE)
model1
pnorm(c(abs(model1$coef)/sqrt(diag(model1$var.coef))), mean=0, sd=1,
lower.tail=FALSE)
class(model1)

#ma(1) AIC=-21073.17 AICc=-21073.16 BIC=-21055.49
model2 = Arima(rendusd_eur, order = c(0,0,1),include.mean = TRUE)
model2

```

```
pnorm(c(abs(model2$coef)/sqrt(diag(model2$var.coef))), mean=0, sd=1,  
lower.tail=FALSE)
```

```
#arma(1,1) AIC=-21071.16 AICc=-21071.15 BIC=-21047.6
```

```
model3 = Arima(rendusd_eur, order = c(1,0,1),include.mean = TRUE)
```

```
model3
```

```
pnorm(c(abs(model3$coef)/sqrt(diag(model3$var.coef))), mean=0, sd=1,  
lower.tail=FALSE)
```

```
#ar(2) AIC=-21071.4 AICc=-21071.39 BIC=-21047.84
```

```
model4 = Arima(rendusd_eur, order = c(2,0,0),include.mean = TRUE)
```

```
model4
```

```
pnorm(c(abs(model4$coef)/sqrt(diag(model4$var.coef))), mean=0, sd=1,  
lower.tail=FALSE)
```

```
#MA(2) AIC=-21071.43 AICc=-21071.41 BIC=-21047.86
```

```
model5 = Arima(rendusd_eur, order = c(0,0,2),include.mean = TRUE)
```

```
model5
```

```
pnorm(c(abs(model5$coef)/sqrt(diag(model5$var.coef))), mean=0, sd=1,  
lower.tail=FALSE)
```

```
#arma(2,1) AIC=-21071.32 AICc=-21071.3 BIC=-21041.87
```

```
model6 = Arima(rendusd_eur, order = c(2,0,1),include.mean = TRUE)
```

```
model6
```

```
###ELEGIMOS A MA(1) POR AIC Y BIC
```

```
autoplot(model2)
```



```
##cae dentro de la banda de confianza
```

```
# Análisis de los residuos
```

```
tsdiag(model2)
```

```
#los residuos se comporta como un ruido blanco
```

```
tsdiag(model3)
```

```
par(mfrow=c(1,2),font=2,font.lab=4,font.axis=2,las=1)
```

```
acf(model2$residuals,ylim=c(-1,1),main="residuosma1")
```

```
pacf(model2$residuals,ylim=c(-1,1),main="residuosma1")
```

```
acf(model3$residuals,ylim=c(-1,1),main="residuosarma")
```

```
pacf(model3$residuals,ylim=c(-1,1),main="residuosarma")
```

```
# vemos si tenemos heavy tails
```

```
qqnorm(model2$residuals)
```

```
qqline(model2$residuals, datax = FALSE)
```

```
##En el Q-Q plot vemos que la distribución de nuestra serie no se ajusta a una normal
```

```
normalTest(model2$residuals,method="jb")
```

```
##P-VALUE MENOR DE 0.05 LA SERIE NO SIGUE UNA NORMAL
```

```
# TESTS GARCH Y FASE1 DE IDENTIFICACION DEL MODELO GARCH
```

```
# ACF y PACF de los residuos al cuadrado
```

```

residuos=model2$residuals
residuos2=residuos^2

par(mfrow=c(1,2),font=2,font.lab=4,font.axis=2,las=1)
acf(residuos2,ylim=c(-1,1),main="residuos al cuadrado")
pacf(residuos2,ylim=c(-1,1),main="residuos al cuadrado")

##presenta autocorrelacion,tenemos una varianza que va cambiando a lo largo de
tiempo,

##podemos modelizar la varianza con un modelo ARCH o GARCH

Box.test(residuos2,lag=1,type='Ljung')
Box.test(residuos2,lag=5,type='Ljung')
Box.test(residuos2,lag=15,type='Ljung')
Box.test(residuos2,lag=20,type='Ljung')

#todos los valores menores que 0.05, los residuos presentan autocorrelación.

###PROPUESTA DE MODELO DE VOLATILIDAD (GARCH)

##EN TODOS LOS CASOS UTILIZAREMOS LA DISTRIBUCIÓN DE t-student,

##la razón es porque estamos analizando una serie financiera y este tipo de serie se
caracteriza por tener

##una cola pesado por presencia de valores extremos. En los test de normalidad, Q-Q
plot... también justifica la decisión.

##STANDARD GARCH

std_g=ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
mean.model=list(armaOrder=c(0,1)), distribution.model = "std")

```

```
m_std=ugarchfit(spec=std_g,data=rendusd_eur)
```

```
m_std
```

```
plot(m_std) #plot 3, 10, 11
```

```
##GJR-GARCH
```

```
gjr_g=ugarchspec(variance.model = list(model = "gjrGARCH", garchOrder = c(1,1)),  
mean.model=list(armaOrder=c(0,1)), distribution.model = "std")
```

```
m_gjr=ugarchfit(spec=gjr_g,data=rendusd_eur)
```

```
m_gjr
```

```
plot(m_gjr)
```

#esperamos que gamma sea positivo y significativo, sale positivo pero no significativo

```
##E-GARCH
```

```
e_g=ugarchspec(variance.model = list(model = "eGARCH", garchOrder = c(1,1)),  
mean.model=list(armaOrder=c(0,1)), distribution.model = "std")
```

```
m_e=ugarchfit(spec=e_g,data=rendusd_eur)
```

```
m_e
```

```
plot(m_e)
```

#alpha1 es la gamma, esperamos que sea significativo y negativo. Sale positivo y no significativo...

#aumentar el orden de garch

```

std_g=ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,2)),
mean.model=list(armaOrder=c(1,0)), distribution.model = "std")

m_std=ugarchfit(spec=std_g,data=rendusd_eur)

m_std

plot(m_std)

##no corrige el problema de residuo al cuadrado. Volvemos al GARCH original.

# Filtrar las observaciones de 2023 a 2024
rendusd_2023_2024 <- window(rendusd_eur, start = "2023-01-01", end = "2024-04-
01")

# Contar el número de observaciones en ese rango de fechas
num_observations <- nrow(rendusd_2023_2024)

# Imprimir el número de observaciones
print(num_observations)

spec1=ugarchspec(variance.model=list(model="sGARCH", garchOrder = c(1,1)),
mean.model=list(armaOrder=c(0,1)), distribution.model = "std")

m3=ugarchfit(spec=spec1,data=rendusd_eur, out.sample = 326)#out of sample son las
observaciones que deja fuera para comparar, no las utiliza para estimar

forc3 = ugarchroll(spec1, data = rendusd_eur, n.ahead = 1, forecast.length = 326,
refit.every = 10, refit.window = "rolling",
)

plot(forc3)

#####

```

```

#####

#####

###Volatilidad Realizada

#####

#####

install.packages("readxl")
install.packages("dplyr")
install.packages("xts")
install.packages("quantmod")

library(readxl)
library(dplyr)
library(xts)
library(quantmod)

file_path <- "C:/Users/Usuario/Desktop/Riesgo de cambio/10MIN SINCE2021.xlsx"
usd_eur_data <- read_excel(file_path)
print(head(usd_eur_data))
summary(usd_eur_data)

# Convertir las fechas y horas al formato adecuado
usd_eur_data <- usd_eur_data %>%
  mutate(Date = as.POSIXct(Date, format="%m/%d/%Y %l:%M %p"))
usd_eur_xts <- xts(usd_eur_data$Close, order.by = usd_eur_data$Date)
head(usd_eur_xts)

#Calcular el rendimiento mediante diferencia logaritmica
returns <- diff(log(usd_eur_xts))

```

```
head(returns)
```

```
rv_daily <- apply.daily(returns, function(x) sqrt(sum(x^2, na.rm = TRUE)))
```

```
head(rv_daily)
```

```
class(rv_daily)
```

```
library(xts)
```

```
# Cargar tus datos aquí, asumiendo que 'rv_daily' ya está cargado y es un objeto xts.
```

```
# Eliminar las horas del índice de tiempo
```

```
index(rv_daily) <- as.Date(index(rv_daily))
```

```
# Dividir los datos en dos conjuntos: entrenamiento y prueba
```

```
usd_train <- window(rv_daily, start = "2021-08-06", end = "2023-12-31")
```

```
usd_test <- window(rv_daily, start = "2024-01-01", end = "2024-04-01")
```

```
# Verificar los resultados
```

```
print(head(usd_train))
```

```
print(head(usd_test))
```

```
#NNAR
```

```
install.packages("forecast")
```

```
library(forecast)
```

```
install.packages("nnfor")
```

```
library(nnfor)
```

```

install.packages("tidyverse")
library(tidyverse)

install.packages("fNonlinear")
library(fNonlinear)

bdsTest(rv_daily, m=3)
tnnTest(rv_daily)
length(rv_daily)

## Guardar los resultados en archivos CSV
# write.csv(usd_train, file = "usd_train.csv", row.names = FALSE)
# write.csv(usd_test, file = "usd_test.csv", row.names = FALSE)

plot(usd_train)
plot(usd_test)

length(usd_test)
# Fitting nnetar model

usd.fit.nnetar <- nnetar(usd_train)
usd.fcst.nnetar <- forecast(usd.fit.nnetar, h = 79)
usd.fcst.nnetar$mean

usd.fit.nnetar_02 <- nnetar(usd_train, repeats = 20, size = 15)
print(usd.fit.nnetar_02)
usd.fcst.nnetar_2 <- forecast(usd.fit.nnetar, h = 79)

```

```

autoplot(usd.fcst.nnetar_2)

# Visualize model predictions
plot(usd_test)

# Visualize model predictions

start_time <- start(as.ts(usd_test)) # obteniendo el inicio de la serie real
names(usd.fcst.nnetar$mean) <- seq(start_time[1], length.out =
length(usd.fcst.nnetar$mean))

# Graficando las series
ts.plot(
  as.ts(usd_test),
  ts(usd.fcst.nnetar$mean, start = start_time),
  col = c("blue", "red"),
  lty = c(1, 2)
)

start_time <- start(as.ts(usd_test)) # obteniendo el inicio de la serie real
names(usd.fcst.nnetar_2$mean) <- seq(start_time[1], length.out =
length(usd.fcst.nnetar_2$mean))

# Graficando las series
ts.plot(
  as.ts(usd_test),
  ts(usd.fcst.nnetar_2$mean, start = start_time),
  col = c("blue", "red"),
  lty = c(1, 2)
)

```



```

library(forecast)

# Define los valores para 'size' y 'num_repeats' que quieres probar
sizes <- c(1, 2, 4, 6, 8, 10, 15, 20, 30) # Número de neuronas en la capa oculta
num_repeats <- c(20, 50, 100) # Número de redes entrenadas

# Prepara un data frame para almacenar los resultados del MSE
results <- data.frame(Size = integer(), Repeats = integer(), MSE = numeric())

# Bucle para probar diferentes combinaciones de 'size' y 'num_repeats'
for (size in sizes) {
  for (num_repeat in num_repeats) {
    # Ajustar el modelo con los parámetros actuales
    set.seed(123) # Para reproducibilidad
    model <- nnetar(usd_train, size = size, repeats = num_repeat)

    # Predecir los valores en el conjunto de prueba
    predictions <- forecast(model, h = length(usd_test))

    # Calcular el MSE
    mse <- mean((usd_test - predictions$mean)^2, na.rm = TRUE)

    # Almacenar los resultados en el data frame
    results <- rbind(results, data.frame(Size = size, Repeats = num_repeat, MSE = mse))
  }
}

```

```
# Ordenar los resultados por MSE para ver cuál configuración tiene mejor rendimiento
```

```
results <- results[order(results$MSE),]
```

```
# Imprimir los resultados ordenados
```

```
print(results)
```

```
####Value at Risk#####
```

```
#####
```

```
length(usd.fcst.nnetar_2$mean)
```

```
v_rv= usd.fcst.nnetar_2$mean
```

```
length(v_rv)
```

```
length(usd_test)
```

```
head(v_rv)
```

```
head(usd_test)
```

```
v_rv <- xts(v_rv, order.by=index(usd_test))
```

```
library(rugarch)
```

```
# Supongamos que rendusd_eur contiene los datos de rendimiento del USD/EUR desde  
01/01/2014 a 01/04/2024.
```

```
# Dividir los datos en entrenamiento y prueba
```

```
train_data <- window(rendusd_eur, end = as.Date("2023-12-31"))
```

```
test_data <- window(rendusd_eur, start = as.Date("2024-01-01"))
```

```

# Especificación del modelo GARCH
spec1 = ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
  mean.model = list(armaOrder = c(0,1)),
  distribution.model = "std")

# Ajuste del modelo con datos de entrenamiento
m1 = ugarchfit(spec = spec1, data = rendusd_eur, out.sample = length(test_data))

# Realizar el pronóstico con la ventana de pronóstico a partir del 01/01/2024
forc = ugarchforecast(m1, n.ahead = 1, n.roll = length(test_data))

var5.garch <- - qnorm(0.95) * sigma(forc) # equivalente a var5.gjrgarch <- qnorm(0.05)
* sigma(forc)

show(var5.garch)

var5.rv=qnorm(0.05) * v_rv

var5.garch_dates <- as.Date(colnames(var5.garch)) # Extraer las fechas de var5.garch
var5.garch_xts <- xts(t(var5.garch), order.by = var5.garch_dates)

# Renombrar columnas para diferenciarlas después de combinar
colnames(var5.garch_xts) <- "var5.garch"

colnames(var5.rv) <- "var5.rv"

colnames(test_data) <- "test_data"

# Combinar los tres objetos xts
combined_xts <- merge(var5.garch_xts, var5.rv, test_data, all = TRUE)

# Rellenar los valores NA con el valor más cercano

```

```

combined_xts <- na.locf(combined_xts, fromLast = FALSE)
combined_xts <- na.locf(combined_xts, fromLast = TRUE)

# Mostrar las primeras observaciones del objeto combinado
tail(combined_xts)

###VaR 5%

# Extraer las fechas y las series de tiempo
Fechas <- index(combined_xts)
var5_garch <- combined_xts$var5.garch
var5_rv <- combined_xts$var5.rv
test_data <- combined_xts$test_data

# Dibujar el VaR
par(mfrow=c(2,2), cex=0.6, mar=c(2,2,3,1))

# Gráfico 1: VaR usando var5_garch
plot(Fechas, test_data, type="l", main = "5% VaR GARCH")
lines(Fechas, var5_garch, col = "blue")

# Gráfico 2: VaR usando var_rv
plot(Fechas, test_data, type="l", main = "5% VaR RV")
lines(Fechas, var5_rv, col = "blue")

fraction_garch5_exceeds <- sum(test_data < var5_garch) / length(test_data)
fraction_rv5_exceeds <- sum(test_data < var5_rv) / length(test_data)

```

```

fraction_garch5_exceeds
fraction_rv5_exceeds

###VaR 1%
var1.garch <- - qnorm(0.99) * sigma(forc) # equivalente a var5.gjrgarch <- qnorm(0.05)
* sigma(forc)
show(var1.garch)
var1.rv=qnorm(0.01) * v_rv
var1.garch_dates <- as.Date(colnames(var1.garch)) # Extraer las fechas de var5.garch
var1.garch_xts <- xts(t(var1.garch), order.by = var1.garch_dates)

# Renombrar columnas para diferenciarlas después de combinar
colnames(var1.garch_xts) <- "var1.garch"
colnames(var1.rv) <- "var1.rv"
colnames(test_data) <- "test_data"

# Combinar los tres objetos xts
combined_xts <- merge(var1.garch_xts, var1.rv, test_data, all = TRUE)

# Rellenar los valores NA con el valor más cercano
combined_xts <- na.locf(combined_xts, fromLast = FALSE)
combined_xts <- na.locf(combined_xts, fromLast = TRUE)

# Mostrar las primeras observaciones del objeto combinado
tail(combined_xts)

###VaR

# Extraer las fechas y las series de tiempo

```

```
Fechas <- index(combined_xts)
var1_garch <- combined_xts$var1.garch
var1_rv <- combined_xts$var1.rv
test_data <- combined_xts$test_data
```

```
# Gráfico 1: VaR usando var5_garch
```

```
plot(Fechas, test_data, type="l", main = "1% VaR GARCH")
lines(Fechas, var1_garch, col = "blue")
```

```
# Gráfico 2: VaR usando var_rv
```

```
plot(Fechas, test_data, type="l", main = "1% VaR RV")
lines(Fechas, var1_rv, col = "blue")
```

```
fraction_garch1_exceeds <- sum(test_data < var1_garch) / length(test_data)
```

```
fraction_rv1_exceeds <- sum(test_data < var1_rv) / length(test_data)
```

```
fraction_garch1_exceeds
```

```
fraction_rv1_exceeds
```