



INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Qual será o melhor modelo para estimar a volatilidade futura da Bitcoin?

Roberto Manuel Duarte Medronho

Mestrado Economia Monetária e Financeira

Orientador:

Professor Doutor Diptes Chandrakante Prabhudas Bhimjee,
Professor Auxiliar Convidado, ISCTE-IUL Business School, ISCTE-IUL
– Instituto Universitário de Lisboa

Setembro, 2024



CIÊNCIAS SOCIAIS
E HUMANAS

Departamento de Economia Política

“Qual será o melhor modelo para estimar a volatilidade futura da Bitcoin?”.

Roberto Manuel Duarte Medronho

Mestrado Economia Monetária e Financeira

Orientador:

Professor Doutor Diptes Chandrakante Prabhudas Bhimjee,
Professor Auxiliar Convidado, ISCTE-IUL Business School, ISCTE-IUL
– Instituto Universitário de Lisboa

Setembro, 2024

Agradecimentos

Gostaria de agradecer a todos as pessoas presentes na minha vida ao longo do último ano, que me ajudaram ao longo do trabalho académico.

Primeiro de tudo gostaria de expressar com enorme gratidão ao meu orientador Professor Doutor Diptes pela sua experiência, paciência e conselhos que foram muito importantes ao longo do processo, ajudando-me a ultrapassar muitos obstáculos encontrados ao longo deste trabalho académico.

À minha família inteira, aos meus pais e mais especificamente à minha avó que partiu este ano, mas de onde estiver de certeza que estará sempre comigo.

A todos os meus amigos, que ao longo desta jornada estiveram sempre ao meu lado, mas muito especialmente ao meu amigo João Dias que muito me ajudou ao longo de todas as etapas da elaboração desta dissertação, tanto no que toca ao seu conhecimento técnico, académico e não menos importante à excelente amizade que temos tido ao longo destes últimos anos.

Ao ISCTE, por me receber como estudante, ajudando-me a abraçar os meus objetivos e proporcionando-me as ferramentas para continuar o meu desenvolvimento enquanto profissional e homem.

Muito obrigado a todos!

Resumo

Nos últimos anos o tema das criptomoedas tem gerado imenso debate tanto para os profissionais envolvidos com o meio financeiro como para muitos membros da sociedade como um todo. Atualmente a Bitcoin continua a ser a criptomoeda mais famosa do planeta. Como todos os instrumentos tradicionais ao conseguir prever a volatilidade será benéfico compreender qual será o melhor período para comprar ou vender por exemplo o ativo em si, tendo sempre em consideração sempre o nível de risco que é superior neste tipo de activos. O objetivo de qualquer investidor visa sempre maximizar o lucro e obter o retorno máximo sobre o investimento. Desta forma, a presente Dissertação utiliza modelos da família GARCH para conseguir prever a volatilidade da Bitcoin num período futuro, sendo a volatilidade um dos maiores fatores de decisão na aquisição de um novo ativo financeiro. Utilizemos a mesma metodologia para estudar um ETF que trabalha com mineração de criptomoedas com o intuito de compreender se a mesma estrutura de testes e modelos se adequa não só para a nossa criptomoeda em estudo, mas também para um conjunto de empresas que trabalha indiretamente com criptomoedas. Os resultados permitem concluir que dos modelos GARCH utilizados em ambas as séries estimadas o modelo EGARCH será o melhor para a sua estimação.

Abstract

In recent years, the topic of cryptocurrencies has sparked significant debate among financial professionals and many members of society as a whole. The question of whether to invest in this new financial asset has raised many doubts and convictions among investors. Should they invest in cryptocurrencies to diversify their portfolios or not. Currently, Bitcoin remains the most well-known cryptocurrency in the world, and like all traditional instruments, being able to predict its volatility makes it easier to understand the best times to buy or sell the asset, for instance. The goal of any investor is always to maximize profits and achieve the highest possible returns. Therefore, throughout our academic work, we decided to use models to predict Bitcoin's future volatility. Volatility is undoubtedly one of the key factors in the decision-making process for acquiring any new asset, and Bitcoin is no exception. We applied the same methodology to study an ETF focused on cryptocurrency mining, aiming to understand if the same testing structures and models are suitable not only for our cryptocurrency under study but also for a group of companies that work indirectly with cryptocurrencies. The results indicate that, among the GARCH models used in both estimated series, the EGARCH model is the most suitable for estimation.

Índice

Agradecimentos	i
Resumo	iii
Abstract	v
Capítulo 1: Introdução	13
Capítulo 2: Revisão de Literatura	15
Capítulo 3: Metodologia	19
3.1 Dados e Metodologia	19
3.1 Enquadramento Bitcoim	20
3.2 Enquadramento ETF WGMI	21
3.2.1 ADF Test	22
3.2.2 KPSS Test	22
3.2.3 PP Test	22
3.2.4 GARCH (1,1)	23
3.2.5 EGARCH (1,1)	23
3.2.6 GJR GARCH(1,1)	23
3.2.7 Box-Ljung Test	24
3.3 LR Test	24
3.3.1 RMSE	24
3.3.2 AIC e BIC	25
Capítulo 4: Resultados Empiricos	27
4.1 Bitcoin Resultados	27
4.2 WGMI Resultados	32
4.3 Resultados mais detalhados	38
4.3.1 Bitcoin	38
4.3.2 WGMI	39
4.4 Relação com pergunta de Investigação	40
Capítulo 5: Conclusão	42
Capítulo 6: Referencias Bibliográficas	44
Anexos	46

Lista de Tabelas

Tabela 1: ADF Test Bitcoin	27
Tabela 2: KPSS Test Bitcoin	27
Tabela 3: PP Test Bitcoin	27
Tabela 4: Box-Ljung Test Garch (1,1) Bitcoin	29
Tabela 5: Box-Ljung Test EGarch (1,1) Bitcoin	30
Tabela 6: Box-Ljung Test GJR-Garch (1,1) Bitcoin	30
Tabela 7: LR Test Bitcoin	31
Tabela 8: RMSE Bitcoin	31
Tabela 9: AIC e BIC Bitcoin	31
Tabela 10: ADF Test WGMI	32
Tabela 11: KPSS Test WGMI	32
Tabela 12: PPTest WGMI	32
Tabela 13: Box-Ljung Test Garch (1,1) WGMI	34
Tabela 14: Box-Ljung Test EGarch (1,1) WGMI	34
Tabela 15: Box-Ljung Test GJR-Garch (1,1) WGMI	35
Tabela 16: LR Test Bitcoin WGMI	36
Tabela 17: RMSE WGMI	36
Tabela 18: AIC e BIC WGMI	37

Lista de Figuras

Figure 1: Garch (1,1) Bitcoin	28
Figure 2: EGarch (1,1) Bitcoin	28
Figure 3: GJR-Garch (1,1) Bitcoin	29
Figure 4: Função de Autocorrelação do modelo GARCH (1,1) Bitcoin	29
Figure 5: Função de Autocorrelação do modelo EGARCH (1,1) Bitcoin	30
Figure 6: Função de Autocorrelação do modelo GARCH (1,1) Bitcoin	30
Figure 7: Garch (1,1) WGMI	33
Figure 8: EGarch (1,1) WGMI	33
Figure 9: GJR-Garch (1,1) WGMI	34
Figure 10: Função de Autocorrelação do modelo GARCH (1,1) WGMI	35
Figure 11: Função de Autocorrelação do modelo EGARCH (1,1) WGMI	35
Figure 12: Função de Autocorrelação do modelo GARCH (1,1) WGMI	36

CAPÍTULO 1

Introdução

Ao longo da última década, as criptomoedas chamaram a atenção do mercado financeiro e da sociedade geral como um todo. Muitos investidores com um perfil de mais propensão ao risco viram através deste ativo uma excelente oportunidade para obter altos retornos ou até mesmo para diversificação do seu próprio portfólio. Considerando as criptomoedas como um ativo financeiro, muitos investidores defendem que o ativo em si não possui qualquer valor intrínseco. Existe ainda a vasta disponibilidade de informações online, seja esta de fontes fidedignas ou não que atrai muitos investidores sejam estes mais experientes ou simplesmente novos investidores. Todos eles muito interessados neste novo ativo em detrimento das opções mais tradicionais como ações ou obrigações.

A mais conhecida entre as criptomoedas denominada por Bitcoin (BTC) é uma moeda virtual descentralizada, sem qualquer supervisão de uma autoridade central de regulação e supervisão como o Banco Central Europeu por exemplo. Diferentemente das moedas tradicionais, as transações em Bitcoin são monitorizadas por um software específico e operam em uma rede ponto a ponto (P2P), semelhante ao BitTorrent, onde todos os utilizadores possuem o mesmo nível de acesso. Embora existam outras criptomoedas, como Litecoin e Ethereum, o Bitcoin continua sendo a mais reconhecida e influente. Criada em 2009, sua origem é atribuída ao pseudônimo Satoshi Nakamoto, cuja verdadeira identidade permanece desconhecida¹.

Qual será o impacto da volatilidade no mercado de Criptomoedas?” foi a pergunta científica de partida para a elaboração da presente Dissertação.

O mais ambicioso objetivo de qualquer investidor é compreender possíveis alterações que irão ocorrer no mercado, sejam estas no curto, médio ou longo prazo. Frequentemente, modelos de previsão robustos são utilizados para esta mesma previsão e permitem aos investidores, tais como instituições financeiras, fundos de investimento ou até propriamente investidores em nome individual tomar as suas decisões na compra, venda ou até mesmo não manter criptomoedas no seu o portfólio de modo a maximizar o seu lucro.

No caso do mercado das criptomoedas a linha de raciocínio é semelhante ao de um qualquer outro investimento, pois pretende-se compreender se o valor da moeda (criptoativo) vai manter-se ou sofrer alterações, podendo o investidor através do horizonte temporal apresentado tomar a decisão mais benéfica para os retornos do seu próprio portfólio.

Por forma a responder à questão científica acima proposta, a presente Dissertação visa estimar qual será de facto o melhor modelo para prever a volatilidade da Bitcoin num período de 90 dias.

Para além disso, a Dissertação usa a mesma estrutura metodológica para um ETF (Exchange Trade Fund), que apresenta na sua estrutura empresas que trabalham com a mineração de criptomoedas (e não apenas a Bitcoin), mas também outras moedas conhecidas no mercado das Criptomoedas.

Bluteau et al. (2019) abordam a pluralidade e versatilidade no que toca à previsão da volatilidade através dos modelos GARCH. Por sua vez, Kim et al. (2021) abordam os mesmos modelos que foram utilizados

¹ Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton University Press.

e foi também mencionado o índice de volatilidade para as criptomoedas. Modelos como EGARCH e GJR-GARCH foram também estimados que serviram também de referência para a elaboração da presente Dissertação. Fakhfekh et al. (2020) utilizaram para além de modelos da família GARCH, os critérios de informação AIC e BIC que são excelentes parâmetros de robustez do modelo e que foram também utilizados ao longo da presente Dissertação para responder à nossa questão da dissertação.

No que diz respeito à metodologia utilizada, as séries foram testadas relativamente à sua estacionariedade, tendo igualmente sido testados diferentes modelos e efetuados os correspondentes testes de robustez. Os dados foram retirados da Yahoo Finance com uma periodicidade diária.

Os resultados obtidos demonstraram-nos que tanto para a Bitcoin como para a empresa de mineração existe um modelo que parece mais indicado para previsão da volatilidade, neste caso o modelo EGARCH (1,1). Através deste tipo de modelos conseguimos, conseguimos prever a volatilidade que será o excelente instrumento para qualquer empresas ou investidor.

A Dissertação está estruturada da seguinte forma: o capítulo II aborda a revisão de literatura, o cap. III descreve a metodologia empregue, bem como o desenvolvimento com os respetivos testes e modelos. o capítulo IV análise dos resultados e a relação com a pergunta de investigação e por fim o capítulo V a conclusão.

CAPÍTULO 2

Revisão de Literatura

Nos últimos anos, o tema das criptomoedas e seus diversos produtos derivados tem despertado um enorme interesse, não só entre investidores e profissionais do setor financeiro, mas também na sociedade como um todo. Os artigos que se seguem foram essenciais para compreender qual seriam de facto os melhores modelos a utilizar ao longo do trabalho académico.

Zulfqar et al. (2021) abordam a questão da “volatility smile” sobre as opções de Bitcoin, estimando a volatilidade implícita sobre as opções de Bitcoin através de métodos de numéricos de aproximação. Através das estimativas dos modelos escolhidos, as opções de Bitcoin demonstram as mesmas características de volatilidade “smile” como é demonstrado nos mercados tradicionais, como ações ou taxas de câmbio, por exemplo. Os resultados empíricos destes autores levam-nos a acreditar que a volatilidade implícita das opções de Bitcoin de curto prazo produz um alto nível de volatilidade à medida que se aproximam da sua data de expiração. Tudo isto se deve ao facto dos *traders* destas opções exigirem mais opções de curto prazo que resultam na pressão de compra do ativo subjacente, neste caso a Bitcoin. Os modelos e metodologias utilizadas correspondem aos objetivos da pesquisa, os resultados analíticos demonstram que através das técnicas *Newton Raphson* e do *Bisection method* são eficientes no cálculo do *pricing* das opções.

Bluteau et al. (2019) também investigaram a volatilidade, utilizando modelos GARCH para testar a presença de mudanças de regime na volatilidade do Bitcoin, por meio dos modelos MJGARCH baseados na metodologia das cadeias de Markov. Além disso, os autores compararam os modelos MJGARCH com os modelos GARCH tradicionais para prever o *Value at Risk* (VaR).

Os resultados das estimativas mostram que os retornos logarítmicos diários da Bitcoin realmente apresentam variações nas suas volatilidades dinâmicas. Uma conclusão importante é que, ao estimar o VAR com um dia de antecedência, as especificações baseadas em cadeias de Markov superam significativamente os regimes padrão dos modelos GARCH.

Conrad et al. (2018) utilizaram modelos semelhantes aos mencionados no artigo anterior ao longo do seu estudo. Estes autores examinaram os efeitos dos movimentos nos mercados financeiros, como o do famoso índice S&P 500, sobre a volatilidade da Bitcoin. Durante o período de 2013 a 2017 em que o estudo foi realizado, havia de facto uma maior dificuldade em compreender as flutuações do que nos dias atuais. Após as estimativas realizadas, concluiu-se que a volatilidade associada ao índice S&P 500 exerce um impacto altamente significativo e negativo sobre o Bitcoin no longo prazo, enquanto a volatilidade *premium*² por contrário durante o período de estimação mostra um efeito positivo. Além disso, os modelos GARCH-MIDAS mostraram-se superiores às previsões dos modelos GARCH mais simples, tornando-se uma opção valiosa para determinar o peso adequado de um determinado ativo, além do próprio Bitcoin, num portfólio que incluía outros ativos como por exemplo ações ou obrigações.

² A volatilidade premium é a diferença entre a volatilidade implícita das opções e a volatilidade real do ativo subjacente. Representa o prêmio pago pelos investidores para se proteger contra movimentos inesperados no mercado.

Kim et al. (2021) desenvolvem uma medida e um modelo para capturar as expectativas do mercado de Criptomoedas e mapeá-las no índice referido (VCRIX). É importante referir que o VCRIX é especialmente focado no ecossistema das criptomoedas para conseguir medir o risco do mercado. Após terem sido selecionadas as *proxies*, o período de amostra e os vários modelos, o modelo selecionado foi o modelo HAR (Heterogeneous Auto-Regressive). Entre vários modelos GARCH, para ser mais preciso IGARCH, FGARCH, SGARCH, GJRGARCH, EGARCH, EWMA, HAR e neutral network-based Long short-term memory cell (LSTM). E por último os modelos multivariados de GARCH mais propriamente DCC & SGARCH e DCC & EWMA.

Ao estimar o modelo HAR, os resultados observados permitem aferir sobre a existência de um alto nível de correlação entre VIX e o CRIX, a rondar os 78 %, o que determina então a aplicabilidade do modelo. É também investigado o valor preditivo da volatilidade do índice ao criar estratégias de *trading* baseado nos sinais do índice que conduzem a 2 horizontes temporais um contra o outro em ETF do próprio índice. Os resultados demonstram que dezembro de 2017 representa o pico e que a melhor estratégia de *trading* que utiliza o VCRIX tem um desempenho superior à estratégia de referência em 99,8 % do período de teste e gera retornos acumulados adicionais de 164 %.

O artigo em si apresenta a ideia de que o VCRIX fornece uma funcionalidade de previsão que serve como uma *proxy* para as expectativas dos investidores na ausência de um desenvolvimento do mercado de produtos derivados de criptomoedas. Esta investigação fornece recursos aprimorados e capacidades de tomada de decisão para monitorizar o mercado, as próprias estratégias de negociação, e potencialmente o preço de opções.

Fakhfekh et al. (2020) pretende selecionar o modelo ou modelos que melhor conseguirão estimar 16 das mais famosas Criptomoedas associadas à volatilidade. O melhor modelo será escolhido para maximizar a probabilidade e minimizar os critérios de informação AIC e BIC. Dos modelos utilizados, a maioria das criptomoedas parece ter melhor resultados com o modelo TGARCH com dupla distribuição exponencial.

Assim sendo, depois de analisar todos os modelos em questão e determinar o modelo mais eficaz, compreendemos também que a volatilidade tende a aumentar em situações de choques positivos e não choques negativos. O efeito assimétrico do estudo irá refletir um efeito visivelmente diferente do presente no mercado bolsista. Trata-se de um artigo importante por falar de volatilidade relacionada com a questão principal da presente Dissertação, mas também por falar de modelos e critérios de informação que são relevantes para a estimação de modelos aqui apresentados.

Matic et al. (2023) visou estudar os comportamentos de *hedging* e efectividade dos modelos de difusão e de “*affine jumps*” e atividade infinita de processos Levy. Para cobrir o leque variado de dinâmicas de mercado, foram gerados os preços de Monte Carlo utilizando a volatilidade estocástica com saltos correlacionados, um Kernel filtrado pelos modelos GARCH próximo do mercado real numa estimativa de densidade de dados históricos.

Existem 2 cenários utilizados nas projeções de Monte Carlo. O processo de gerar os dados é designado pelo modelo SVCJ na primeira abordagem. Já na segunda abordagem o método utilizado é baseado em GARCH- filtrado pelo Kernel-Density Estimation (KDE) tendo em consideração uma simulação histórica e na volatilidade dinâmica mais aproximada à volatilidade de mercados.

Os mercados de Criptomoedas têm suscitado muito interesse, pois os preços são altamente voláteis e os seus derivados associados tem um elevado valor transacional por alguns anos. Com o modelo GARCH e as suas volatilidades e os estimadores KDE, podemos concluir que os caminhos são estatisticamente

semelhantes ao comportamento do mercado. No modelo SVCJ³ está incluindo a importante informação dos factores de risco para determinados “saltos” como operações Bullish ou a Covid 19.

Desta forma as opções de longo prazo são instrumentos de *hedging* que nos irão levar uma redução de risco considerável. Para uma opção de curto prazo com o mesmo método de instrumentos de *hedging*, o resultado já não será o mesmo pois já não haverá o mesmo nível de *performance* para um único de *hedging*. Todas estas ideias encontram-se alinhadas com o mercado que no curto prazo será menos impactado por instrumentos de *hedging* em comparação com o longo prazo. Assim sendo as opções em bolsa estão disponíveis para situações de hedging e diversificação de Portfolio e caso haja liquidez deverão ser utilizadas. Todas estas conclusões são suportadas pelos modelos de volatilidade estocástica. Este artigo retrata as opções de produtos derivados que estão bastante em foco na volatilidade implícita, um dos maiores focos da presente Dissertação.

Liu et al. (2019) abordam a evolução do preço da Bitcoin, exploram a evolução do preço do Bitcoin, sua volatilidade e possíveis interdependências com outros retornos, utilizando os modelos GARCH. Ao longo do artigo são apresentadas 2 visões diferentes. A primeira retrata o mecanismo de transmissão de criptomoedas através da correlação com os fundamentos económicos e a capitalização global. Já a segunda perspectiva defende a ineficiência e desta forma os investidores irão tentar estratégias de *hedging* entre as criptomoedas que pode causar um movimento de contágio nesse próprio mercado. O modelo utilizado foi o modelo trivariado BEKK GARCH-in-mean. Tudo isto devido ao preencher os requisitos necessários como a lacuna na modelagem GARCH-in-mean de retornos de criptomoedas que vai afetar o prémio de risco diretamente pela variação condicional em mudança. Este o estudo aborda a exposição das 3 principais criptomoedas e sua exposição a choques de mercados e, por último, as repercussões da volatilidade do mercado de criptomoedas para outros mercados financeiros.

A volatilidade dos modelos de retorno de criptomoedas em análise foram estimadas com os dados mais recentes até 2019 de forma univariada e multivariada. Houve então fortes evidências no progresso da integração do mercado de criptomoedas, O que pode trazer vantagens na diversificação de portfolios com ações e obrigações como foi já anteriormente referenciado noutros artigos, tendo maior relevo onde as criptomoedas já estarão a ser utilizadas. Este artigo, tal como os anteriormente apresentados, reforçou a enorme possibilidade da utilização dos modelos GARCH para estimação dos modelos a estimar para a dissertação.

Sapkota (2022) analisou o impacto dos sentimentos dos *media* na volatilidade do Bitcoin, utilizando três estimativas de volatilidade e quatro dicionários de sentimentos. Analisando 17.490 notícias de 91 jornais internacionais (2012-2021), os resultados alcançados mostraram que os sentimentos influenciam significativamente a volatilidade. O modelo HAR-RV, com sentimentos como variáveis explicativas, apresentou melhor ajuste e previsão. Observa-se que os sentimentos psicológicos impactam a volatilidade no médio prazo, enquanto os sentimentos financeiros afetam a longo prazo, com "antecipação" e "confiança" sendo os principais motores emocionais. Interessante como os meios de informação conseguem influenciar o preço de ativos financeiros, nomeadamente o preço da Bitcoin e de outras Criptomoedas.

Coulter (2022) examinou a relação entre eventos relatados em notícias internacionais por meio de discursos categóricos e o preço do Bitcoin. Utilizando processamento de linguagem natural, foram analisados artigos sobre criptomoedas (N = 4218) publicados em 60 países entre 2018 e 2020. A análise

³ Modelo financeiro que captura a volatilidade estocástica dos ativos, incorporando saltos simultâneos no preço e na volatilidade. Ele é usado para descrever movimentos bruscos nos mercados, refletindo melhor eventos extremos do que modelos tradicionais.

identificou 18 tópicos principais associados a crimes relacionados a criptos, governança financeira e mercados económicos. O estudo demonstra que esses discursos tiveram um impacto negativo na volatilidade do preço do Bitcoin no espaço de 24 horas após a publicação das notícias. Além disso, a fonte das notícias e a região geográfica amplificaram esse efeito em certos casos.

Bibi (2023) retrata historicamente a evolução das criptomoedas ao longo dos últimos anos e as possíveis ameaças ao sistema financeiro e monetário. É apresentado a visão de vários economistas sobre este ativo e o mais interessante o caso de El Salvador que já utiliza criptomoedas como método de pagamento oficial através de uma *app* móvel.

Seaman (2022) descreve a história da mais famosas das criptomoedas. O autor da obra para além de não possuir qualquer valor intrínseco e a sua utilização para a especulação o autor apresenta muitas outras críticas relevantes. Existe uma falha para a sua denominação de “good Money”, pois a sua volatilidade não é útil como reserva de valor, meio de pagamento ou unidade de conta. O facto de não ser emitida por qualquer Banco Central não possibilita o seu uso como pagamento oficial na maioria dos países. Falha também como ativo, o facto de não ter nenhum valor associado. O facto de ser comparada com o ouro e ser comparado como ouro digital é um conceito falacioso visto que o ouro nunca foi tão volátil com a Bitcoin.

Visto que não existe uma entidade que a emite, ninguém poderá ser responsável no caso de o sistema sucumbir, ao contrário do que acontece com as moedas tradicionais. Muitas mais razões são apresentadas por este autor para além das anteriormente apresentadas, mas uma das mais importantes pode ser o elevado consumo de eletricidade para todas as operações de produção e manutenção de criptomoedas, tais como os *chips* e equipamentos informático. Este autor documenta ainda vários conhecimentos sobre variadíssimos conceitos e noções relativas à Bitcoin e as restantes criptomoedas.

Estudos como os de Kim et al. (2021) abordaram a volatilidade implícita nas opções de Bitcoin, demonstrando que essas opções exibem um comportamento de "*volatility smile*" semelhante ao de mercados tradicionais. Outros autores, como Bluteau et al. (2019), utilizaram modelos MJGARCH para mostrar que os retornos diários do Bitcoin apresentam variações dinâmicas, destacando que esses modelos são superiores aos GARCH tradicionais na previsão do Value at Risk (VaR). Além disso, Conrad et al., (2018) exploraram os efeitos da volatilidade do S&P 500 sobre o Bitcoin, concluindo que modelos GARCH-MIDAS oferecem uma previsão mais eficaz comparada aos modelos GARCH convencionais.

Estudos mais recentes, como os de Kim et al. (2021), introduziram o índice VCRIX, específico para o mercado de criptomoedas, enquanto Fakhfekh et al. (2020) destacaram a eficácia dos modelos TGARCH na estimativa da volatilidade de diversas criptomoedas.

Da presente revisão da literatura académica conclui-se que a correta abordagem para prever a volatilidade futura da Bitcoin. Depois da extração dos dados, serão elaborados vários testes de modo a compreender se as séries em estudo serão ou não estacionárias. De seguida serão utilizados os modelos GARCH com o objetivo de compreender qual destes modelos será melhor para a respetiva estimação da série. Por últimos testes de robustez serão efetuados com o objetivo de compreender qual será exatamente o melhor modelo.

CAPÍTULO 3

Dados e Metodologia

Todos os dados foram extraídos através da plataforma Yahoo Finance, tanto para a Bitcoin quanto para o ETF de empresas de mineração WGMI.

Para a Bitcoin, os dados foram recolhidos no intervalo de 01/06/2018 a 30/06/2024. No caso da WGMI, os dados foram extraídos a partir de 08/02/2022, com o último registo em 30/06/2024. A escolha do período de análise para a Bitcoin incluiu uma data anterior ao início da pandemia da Covid-19 em 2019 e ao início da Guerra entre a Ucrânia e a Rússia. O objetivo foi garantir um período de análise que abrangesse datas anteriores aos vários eventos significativos, o período temporal em que os mesmos ocorreram, e por último um período posterior.

No caso da WGMI, os dados foram limitados à disponibilidade na base de dados, justificando a escolha do período. A periodicidade dos dados foi diária para ambas análises. O último dia de dados apresentados corresponde ao fim das respetivas estimações.

A estrutura da metodologia está constituída em três diferentes sub-seções. Primeiramente temos os testes de estacionariedade com o intuito de compreender se as séries serão ou não estacionárias para poderem ser estimadas pelos respetivos modelos. De seguida temos a própria estimação dos diferentes modelos GARCH, tendo em consideração a variável da Bitcoin e o nosso ETF WGMI. As componentes que diferenciam o modelo EGARCH do modelo GARCH SÃO a volatilidade condicional (é modelada em uma escala logarítmica). No caso do modelo GJR-GARCH tem em consideração o efeito de alavancagem. Serão também realizados testes de robustez com o objetivo de perceber qual será de facto o melhor modelo para estimação das séries em estudo. Para concluir teremos a interpretação dos resultados ao longo de toda a metodologia.

3.1 Enquadramento Bitcoin

Como o próprio nome indica, o grande foco da dissertação é encontrar o modelo com o melhor nível de robustez para estimar a volatilidade da Bitcoin para os próximos 90 dias. O objetivo será testar diferentes modelos GARCH que foram anteriormente estudados através de artigos previamente referidos. Para tal, é necessário efetuar alguns testes prévios para confirmar que cada série temporal é de facto estacionária, estimar os próprios modelo e de seguida realizar mais alguns testes de robustez para compreender qual será de facto o melhor modelo.

Começaremos desta forma por fazer uma breve explicação da criptomoeda mais conhecida, a Bitcoin.

A Bitcoin (BTC)⁴ é uma criptomoeda descentralizada que foi criada em 2009 por uma entidade sob o pseudónimo de Satoshi Nakamoto. Desde a sua criação, o Bitcoin destacou-se como a primeira criptomoeda do mundo, sendo a mais conhecida e amplamente utilizada.

O Bitcoin, como ativo digital, pode ser negociado em várias plataformas de câmbio de criptomoedas, semelhante à forma como ações e outros instrumentos financeiros tradicionais são negociados em bolsas de valores. A sua relevância e adoção global têm atraído a atenção de investidores, instituições financeiras, e até governos. A sua composição de valor é influenciada por diversos fatores, tais como:

- **Mineradoras de Bitcoin:** Entidades e indivíduos que utilizam sistemas computacionais de alta performance para resolver problemas matemáticos complexos que validam as transações na rede Bitcoin. Essas operações consomem uma grande quantidade de energia (externalidade negativa) e têm um impacto significativo na oferta de novas unidades de Bitcoin.
- **Desenvolvedores de Bitcoin e tecnologia blockchain:** Profissionais e empresas que desenvolvem e mantêm a infraestrutura tecnológica do Bitcoin. Eles são responsáveis por melhorias no protocolo, implementação de novas funcionalidades, e a garantia da segurança da rede.
- **Infraestrutura de Bitcoin e exchanges:** Plataformas que fornecem serviços essenciais para a compra, venda, e armazenamento de Bitcoin. Isso inclui *exchanges* de criptomoedas, carteiras digitais, e soluções de custódia, que são cruciais para a operação e acessibilidade do Bitcoin a um público mais amplo.

⁴ Antonopoulos, A. M. (2014). *Mastering Bitcoin: unlocking digital cryptocurrencies*. " O'Reilly Media, Inc."

3.2 Enquadramento ETF WGMI

Após termos realizado o enquadramento à Bitcoin e alcançado os resultados anteriormente apresentados, irá ser efetuada uma outra análise subsequente para conseguirmos compreender as diferenças entre os resultados.

Escolhemos então estimar a variável do WGMI. O WGMI (*We are gonna make it*) é o *ticker* de um ETF (*Exchange Traded Fund*) denominado "*SonicShares Global Blockchain ETF*". Este fundo é negociado na bolsa de valores e investe em empresas que estão envolvidas na tecnologia blockchain, a mesma tecnologia que sustenta criptomoedas como o Bitcoin.

O WGMI pode ser perspectivado como um fundo de investimento, o que significa que podemos comprar e vender como se tratasse de uma qualquer outra ação. A sua composição passa por ações de várias empresas que trabalham com a tecnologia *blockchain*, tais como:

- **Mineradoras de criptomoedas:** Empresas que utilizam sistemas informáticos de extrema complexidade técnica para validar transações efetuadas através de redes de criptomoedas.
- **Desenvolvedoras de tecnologia *blockchain*:** Empresas que criam ou utilizam a tecnologia *blockchain* para diversos fins, como contratos inteligentes e segurança de dados.
- **Empresas de infraestrutura de *blockchain*:** Empresas que fornecem a base tecnológica necessária para que o *blockchain* funcione, como por exemplo *hardware* especializado.

Dado que esta empresa oferece diversos serviços relacionados a mineração de criptomoedas, será também interessante analisar uma série de dados deste ETF. Iremos aplicar a mesma estrutura de testes e modelos com o intuito de compreender se obteremos ou não resultados semelhantes com os anteriormente obtidos com a série da Bitcoin.

3.3 Testes de Estacionaridade

3.3.1 ADF test⁵

O teste de Dickey-Fuller Aumentado consiste num teste estatístico amplamente utilizado em econometria para verificar a presença de uma raiz unitária em uma série temporal. A presença de uma raiz unitária indica que a série temporal é não-estacionária, ou seja, suas propriedades estatísticas, como a média e a variância, mudam ao longo do tempo.

Hipóteses do Teste ADF:

- **Hipótese Nula (H₀):** A série temporal tem uma raiz unitária, ou seja, a série não é estacionária.

⁵ Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74(366a), 427-431.

- **Hipótese Alternativa (H1):** A série temporal não tem raiz unitária, ou seja, a série é estacionária.

3.3.2 KPSS Teste⁶

O teste KPSS (Kwiatkowski-Phillips-Schmidt-Shin) é outro método para testar a estacionariedade de uma série temporal. Enquanto o teste ADF testa a hipótese nula de que a série tem uma raiz unitária (ou seja, é não estacionária), o teste KPSS faz o oposto: sua hipótese nula é que a série é estacionária. Isso oferece uma perspectiva complementar na análise de séries temporais.

Hipóteses do Teste KPSS:

- **Hipótese Nula (H0):** A série temporal é estacionária em torno de uma média ou tendência determinística. Em outras palavras, a série não tem uma raiz unitária.
- **Hipótese Alternativa (H1):** A série temporal não é estacionária (ou seja, contém uma raiz unitária).

3.3.3 PP Teste⁷

O teste de Phillips-Perron (PP) pretende testar a presença de uma raiz unitária em uma série temporal, semelhante ao teste ADF (Dickey-Fuller Aumentado). O teste PP é uma alternativa ao ADF, mas é mais robusto a algumas formas de heterocedasticidade e autocorrelação nos resíduos.

Hipóteses do Teste PP:

- **Hipótese Nula (H0):** A série temporal tem uma raiz unitária, ou seja, a série não é estacionária.
- **Hipótese Alternativa (H1):** A série temporal não tem raiz unitária, ou seja, a série é estacionária.

⁶ Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?. *Journal of econometrics*, 54(1-3), 159-178.

⁷ Phillips, P. (1988). Testing for unit roots in time series regression. *Biometrika*, 71, 599-607.

3.4 Modelos Econométricos

3.4.1 GARCH (1,1)⁸

GARCH (Generalized Autoregressive Conditional Heteroskedasticity) é um modelo estatístico utilizado para modelar a volatilidade de séries temporais financeiras, como preços de ações ou taxas de câmbio por exemplo. O modelo GARCH (1,1) é uma das versões mais simples e amplamente utilizadas da família GARCH.

3.4.2 EGARCH (1,1)⁹

O modelo EGARCH (Exponential Generalized Autoregressive Conditional Heteroskedasticity) é uma extensão do modelo GARCH que permite capturar assimetrias na volatilidade e garantir que a variância condicional seja sempre positiva sem impor restrições nos parâmetros.

O modelo EGARCH (1,1) descreve a volatilidade condicional de uma série temporal em função de seus valores passados, similar ao GARCH, mas com uma formulação logarítmica para a variância condicional.

3.4.3 GJR-GARCH (1,1)¹⁰

O modelo GJR-GARCH (Glosten, Jagannathan, and Runkle Generalized Autoregressive Conditional Heteroskedasticity) é uma extensão do modelo GARCH que permite capturar assimetrias na volatilidade condicional.

O modelo GJR-GARCH (1,1) descreve a volatilidade condicional de uma série temporal considerando a possibilidade de que choques negativos e positivos tenham impactos diferentes na volatilidade futura.

⁸ Nelson, D. B. (1990). Stationarity and persistence in the GARCH (1, 1) model. *Econometric theory*, 6(3), 318-334.

⁹ Kat, H. M., & Heynen, R. C. (1994). Volatility Prediction: A Comparison of the Stochastic Volatility, GARCH (1, 1) and Egarch (1, 1) Models. *Journal of Derivatives*, 2(2).

¹⁰ Mostafa, F., Saha, P., Islam, M. R., & Nguyen, N. (2021). GJR-GARCH volatility modeling under NIG and ANN for predicting top cryptocurrencies. *Journal of Risk and Financial Management*, 14(9), 421.

3.5 Testes e Métricas de Pós-Estimação

3.5.1 Box-Ljung Test

O teste **Box-Ljung** é usado para verificar se existe autocorrelação significativa nos resíduos padronizados do modelo. No contexto de um modelo GARCH, ele ajuda a avaliar se o modelo capturou adequadamente a dependência temporal e a volatilidade condicional dos dados. A ausência de autocorrelação nos resíduos padronizados indica que o modelo é adequado.

- **Hipótese Nula (H0):** Não há autocorrelação significativa nos resíduos (os resíduos são independentes).
- **Hipótese Alternativa (H1):** Existe autocorrelação significativa nos resíduos (os resíduos não são independentes).

3.5.2 LR Test

O Teste de Razão de Verossimilhança (Likelihood Ratio Test) é um método estatístico utilizado para comparar dois modelos aninhados. Dois modelos são considerados aninhados se um dos modelos (o modelo restrito) neste caso o GARCH (1,1) for uma versão especial do outro modelo (o modelo irrestrito) e no outro caso o GARCH (1,2) com algumas restrições impostas.

- **Hipótese Nula (H0):** O modelo mais simples é suficiente para explicar os dados.
- **Hipótese Alternativa (H1):** O modelo mais complexo fornece um ajuste significativamente melhor aos dados.

3.5.3 RMSE

O RMSE (Root Mean Square Error) é uma métrica que mede o erro médio entre valores previstos e reais em um modelo de previsão. Ele é calculado tirando a raiz quadrada da média dos erros quadrados. Um RMSE menor indica um modelo mais preciso. É útil para avaliar o desempenho de modelos de regressão, comparando previsões com os dados observados.

3.5.4 AIC e BIC

Os critérios de informação AIC e BIC são usados para comparar a qualidade de ajuste de diferentes modelos estatísticos. Menores valores de AIC e BIC indicam um melhor equilíbrio entre a

complexidade do modelo e seu ajuste aos dados. Quanto menores forem os valores referentes aos critérios de informação melhor será a aderência do modelo.

CAPÍTULO 4

Resultados Empíricos

4.1 Bitcoin

ADF teste:

P-VALUE	0,01
---------	------

Tabela 1: ADF Test Bitcoin

Para a série de dados da Bitcoin, visto que o p value obtido é inferior a 0.05, nomeadamente 0.01 podemos rejeitar a hipótese nula e concluir que a série é estacionária.

KPSS Test:

P-VALUE	0,1
---------	-----

Tabela 2: KPSS Test Bitcoin

Para a série de dados da Bitcoin, visto que o p value obtido é superior a 0.05, nomeadamente 0.1 não podemos rejeitar a hipótese nula e concluir que a série é estacionária.

P-VALUE	0,01
---------	------

Tabela 3: PP Test Bitcoin

Para a série de dados da Bitcoin, visto que o p value obtido é inferior a 0.05, nomeadamente 0.01 podemos rejeitar a hipótese nula e concluir que a série é estacionária.

BITCOIN

GARCH (1,1)

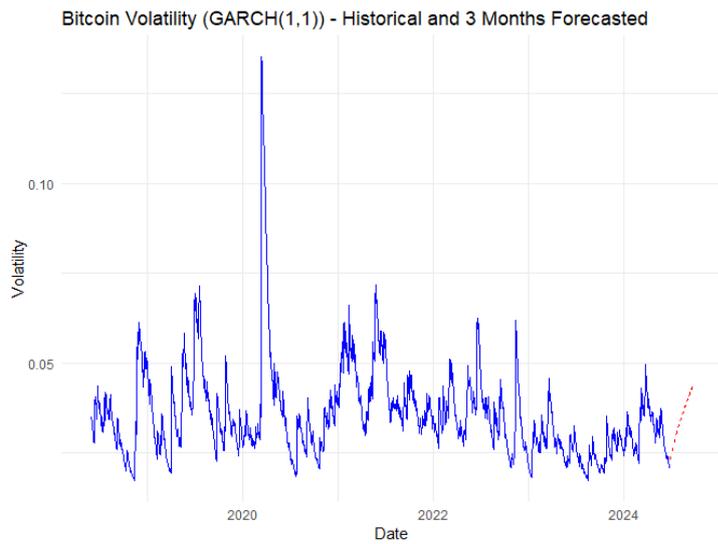


Figura 1: Modelo GARCH (1,1) aplicado à série de BITCOIN

EGARCH (1,1)

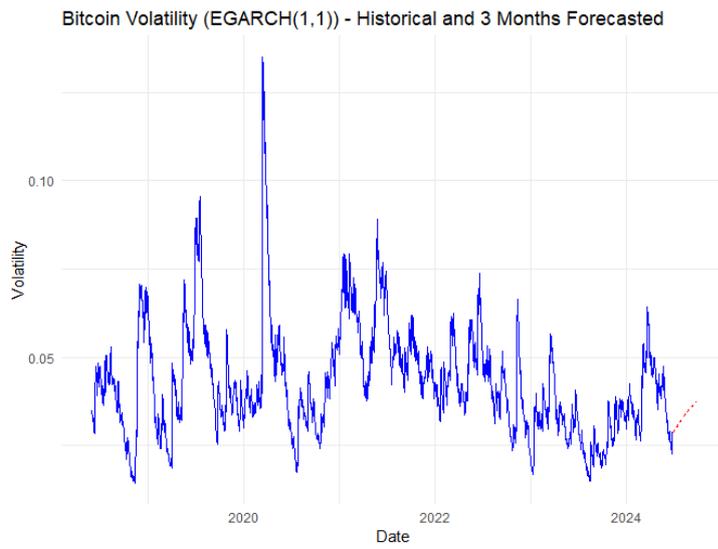


Figura 2: Modelo EGARCH (1,1) aplicado à série de BITCOIN

GJR-GARCH (1,1)

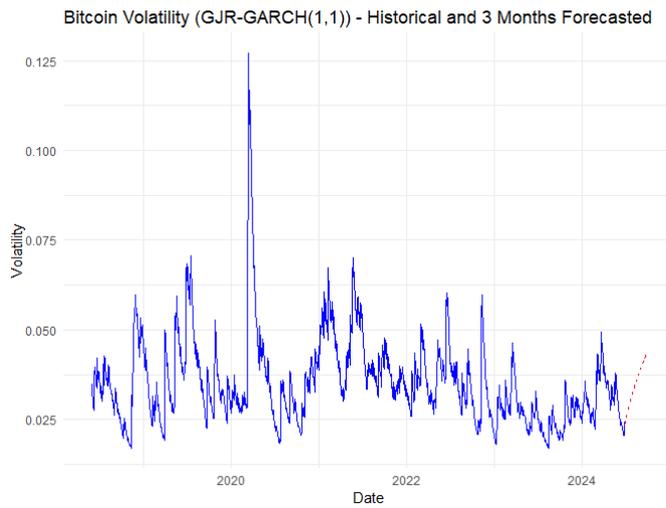


Figura 3: Modelo GJR-GARCH (1,1) aplicado à série de BITCOIN

Testes de Pós Estimação

Box-Ljung Test

GARCH (1,1)

P-VALUE	0.4894
---------	--------

Tabela 4: Box-Ljung GARCH (1,1) Test Bitcoin

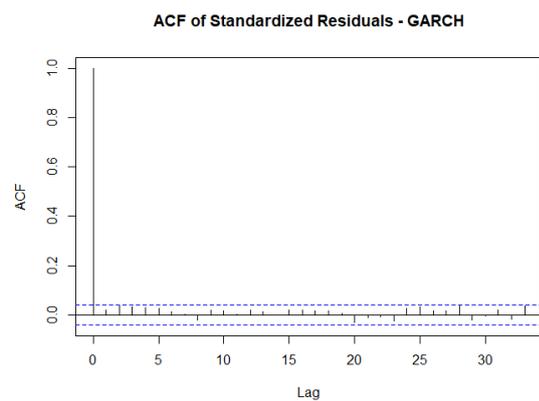


Figura 4: Função de Autocorrelação do modelo GARCH (1,1) BITCOIN

E-GARCH

P-VALUE	0.2862
---------	--------

Tabela 5: Box-Ljung EGARCH (1,1) Test Bitcoin

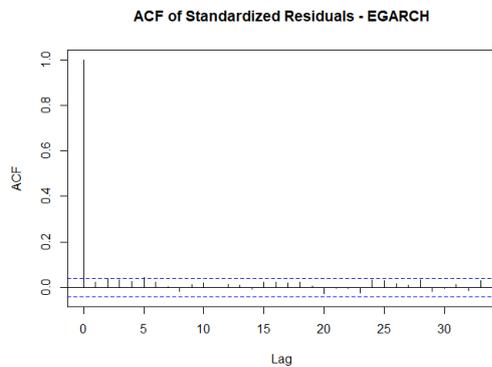


Figura 5: Função de Autocorrelação do modelo EGARCH (1,1) BITCOIN

GJR-GARCH

P-VALUE	0.5054
---------	--------

Tabela 6: Box-Ljung GJR-GARCH (1,1) Test Bitcoin

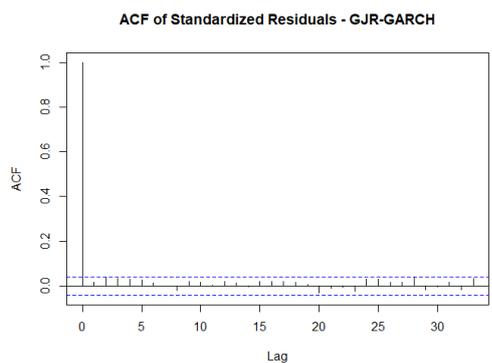


Figura 6: Função de Autocorrelação do modelo GJR-GARCH (1,1) BITCOIN

O teste Box-Ljung apresenta um p-value de 0.4894 para o modelo GARCH (1,1). Como este p-value é superior que 0.05, concluímos que não há evidência significativa de autocorrelação nos resíduos. O mesmo se sucede para o modelo EGARCH (1,1) com um p-value de 0.2862 e GJR-GARCH (1,1) e um p-value de 0.5054. Portanto, o modelo GARCH (1,1), EGARCH (1,1) e GJR-GARCH (1,1) parecem ser adequados para modelar a série temporal em análise.

LR Test

P-VALUE	0.1394
---------	--------

Output 7: LR Test Bitcoin

Como o p-valor de 0.1394 é superior a 0.05, não rejeitamos a hipótese nula de que o modelo GARCH (1,1) é suficiente. Portanto, o modelo GARCH (1,2) não proporciona um ajuste significativamente melhor aos dados em comparação com o GARCH (1,1).

De seguida iremos calcular o RMSE (Root Mean Squared Error). Três modelos foram ajustados aos dados correspondentes: GARCH, EGARCH e GJR-GARCH. Foram feitas previsões para o conjunto de teste, e o erro quadrático médio (RMSE) foi calculado para cada modelo. Quanto menor for o RMSE melhor será o modelo.

GARCH (1,1)	0,0490
EGARCH (1,1)	0,0753
GJR-GARCH (1,1)	0,0077

Tabela 8: RMSE Bitcoin

Os três modelos de previsão de volatilidade foram comparados usando o RMSE: EGARCH 0,0490 teve o menor erro, indicando maior precisão. O GJR-GARCH de 0,0753 apresentou desempenho intermediário, capturando melhor a volatilidade assimétrica. O GARCH de 0,0771 teve o maior erro. Portanto, o EGARCH é o mais eficaz com um valor de 0,04905, seguido por GJR-GARCH e GARCH.

AIC e BIC

Bitcoin	AIC	BIC
GARCH (1,1)	-4.2512	-4.2409
EGARCH (1,1)	-4.2631	-4.2502
GJR-GARCH (1,1)	-4.2506	-4.2377

Tabela 9: AIC e BIC Bitcoin

Tanto o AIC quanto o BIC indicam que o modelo EGARCH (1,1) é superior aos modelos GARCH (1,1) e GJR-GARCH (1,1) para os dados dos retornos de Bitcoin, dado que exibe os valores mais baixos em

ambos os parâmetros. Portanto, com base nesses critérios de informação, o EGARCH (1,1) é o modelo recomendado para capturar a volatilidade dos retornos de Bitcoin.

O modelo EGARCH é conhecido por capturar melhor a volatilidade assimétrica e as características de alavancagem dos retornos financeiros, o que pode explicar por que este modelo se ajusta mais eficazmente aos dados de Bitcoin do que os modelos GARCH e GJR-GARCH tradicionais.

4.2 ETF WGMI

Teste ADF

P-VALUE	0,01
---------	------

Tabela 10: ADF Test WGMI

Ao analisarmos o output em cima reparamos que P value obtido é inferior a 0,05, assim sendo podemos rejeitar a hipótese nula e concluir que a série é estacionária.

KPSS Test

P-VALUE	0,09
---------	------

Tabela11: KPSS Test WGMI

Neste caso o P value obtido é superior a 0,05 e desta forma não podemos rejeitar a hipótese nula e concluir que a série é estacionária.

PP Test

Por último seguindo a mesma lógica realizamos o teste PP.

P-VALUE	0,01
---------	------

Output 12: PP Test WGMI

O valor do p-value obtido é por sua vez superior a 0,05 e por esta razão podemos rejeitar a hipótese nula e concluir que a série é estacionária.

Modelos

GARCH (1,1)

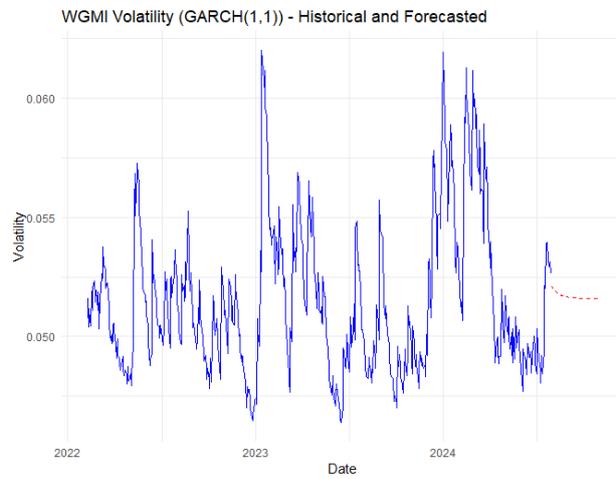


Figura 4: Modelo GARCH (1,1) aplicado à série de WGMI

EGARCH (1,1)

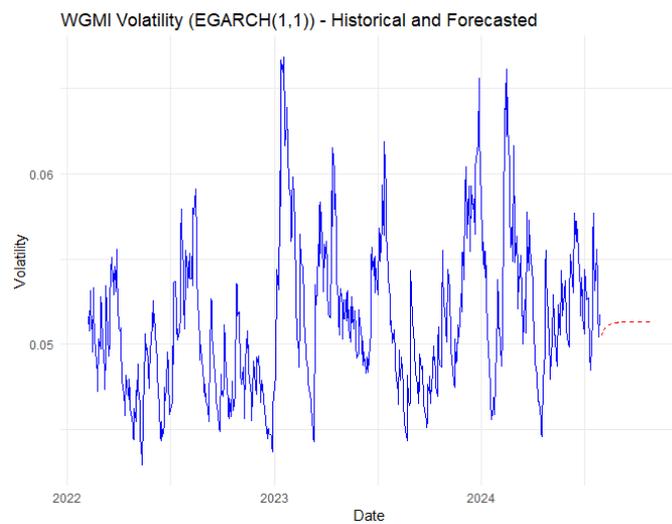


Figura 5: Modelo EGARCH (1,1) aplicado à série de WGMI

GJR-GARCH (1,1)

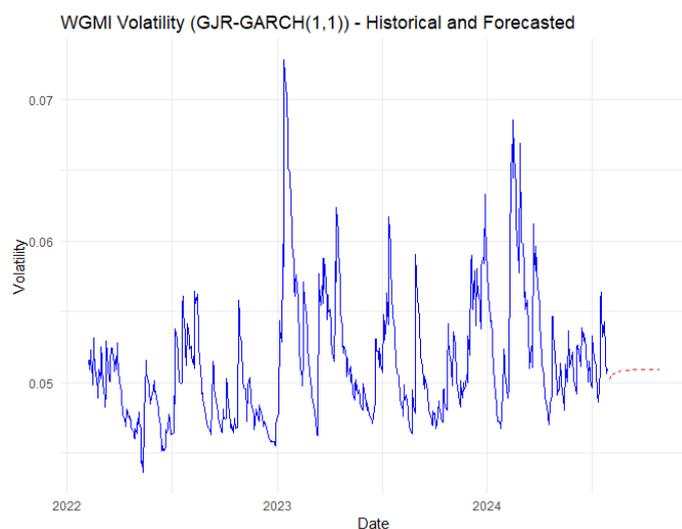


Figura 6: Modelo GARCH (1,1) aplicado à série de WGMI

Tendo em consideração todos os modelos apresentados é preciso compreender qual destes responderá adequadamente à pergunta proposta na presente Dissertação.

De forma a conseguirmos obter uma resposta precisa e com as respetivas justificações econométricas, iremos realizar uma série de testes para conseguir compreender qual será de facto o modelo mais adequado.

Teste Box-Ljung

GARCH (1,1)

P-VALUE	0.4399
---------	--------

Tabela 13: Box-Ljung GARCH (1,1) WGMI

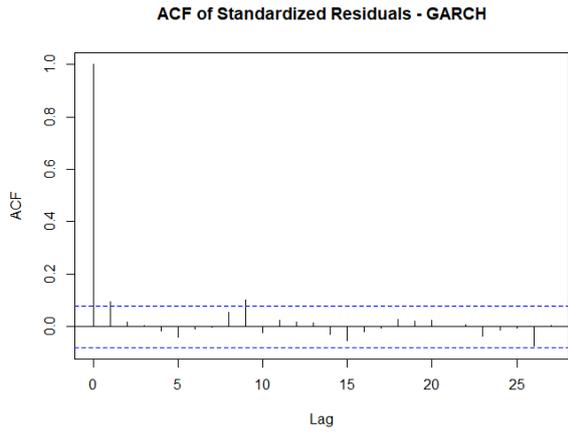


Figura 7: Função de Autocorrelação do modelo GARCH (1,1) à série do WGMI

P-VALUE	0.3506
---------	--------

Output 14: Box-Ljung EGARCH (1,1) WGMI

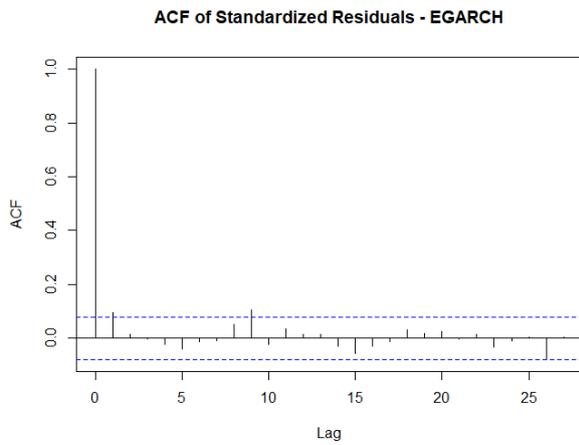


Figura 8: Função de Autocorrelação do modelo EGARCH (1,1) à série do WGMI

P-VALUE	0.3619
---------	--------

Figura 15: Box-Ljung GJR-GARCH (1,1) WGMI

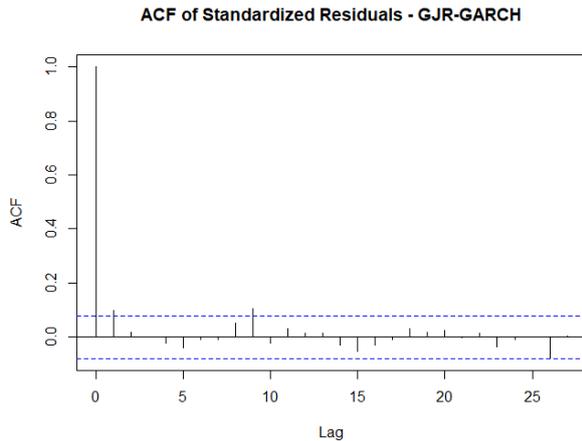


Figura 9: Função de Autocorrelação do modelo EGARCH (1,1) a série do WGMI

O teste Box-Ljung apresenta p-value de 0.4399; 0.3506 e 0.3619 para os respectivos modelos GARCH (1,1), EGARCH (1,1) e GJR-GARCH (1,1). Como este p-value é superior a 0.05, concluímos que não há evidência significativa de autocorrelação nos resíduos. Portanto, o modelo GARCH (1,1), EGARCH (1,1) e GJR-GARCH (1,1) parecem ser adequados para modelar a série temporal em análise.

Likelihood ratio test

Ao realizarmos o teste, o p value obtido foi 1 que é maior que 0.05, não rejeitamos a hipótese nula de que o modelo GARCH (1,1) é suficiente. Portanto, o modelo GARCH (1,2) não proporciona um ajuste significativamente melhor aos dados em comparação com o GARCH (1,1).

P-VALUE	1
---------	---

Output 16: LR Test Bitcoin WGMI

Iremos calcular os RMSE para cada modelo GARCH.

GARCH (1,1)	0,0724
EGARCH (1,1)	0,0722
GJR-GARCH (1,1)	0,0727

Figura 17: RMSE's WGMI

Os três modelos de previsão de volatilidade foram comparados pelo RMSE: o EGARCH (0,0722206) teve o menor erro, sendo o mais preciso. O GJR-GARCH (0,0727392) apresentou um desempenho

intermediário, capturando melhor a volatilidade assimétrica do que o GARCH (0,0724837). Portanto, o EGARCH é o modelo mais eficiente, seguido por GJR-GARCH e GARCH.

AIC e BIC

Ao analisar os critérios AIC e BIC, o modelo EGARCH (1,1) apresenta o menor AIC (-3,0933), sugerindo ser o melhor para previsão, enquanto o GARCH (1,1) tem o menor BIC (-3,0598), indicando simplicidade. Dado que o foco é prever a volatilidade do ETF WGMI em um período de 90 dias, o critério AIC, que favorece previsões, é mais relevante. Como o RMSE também apontou o EGARCH como o mais preciso, este modelo é a escolha preferida para estimar a volatilidade do WGMI.

Bitcoin	AIC	BIC
GARCH (1,1)	-4.2512	-4.2409
EGARCH (1,1)	-4.2631	-4.2502
GJR-GARCH (1,1)	-4.2506	-4.2377

Figura 18: AIC E BIC WGMI

4.3 RESULTADOS AO DETALHE

4.3.1 BITCOIN

Após verificarmos que a série é estacionária, realizamos diferentes teste GARCH para verificar a volatilidade desde o período inicial até à data em que os testes foram realizados e também prever a volatilidade num prazo de 90 dias. Os modelos realizados foram nomeadamente o GARCH (1,1), EGARCH (1,1) e o GJR-GARCH (1,1). Ao observarmos as figuras 1, 2 e 3 conseguimos observar picos e cavas em períodos semelhantes ao longo do período estimado. No modelo GARCH (1,1) o ponto mais alto é claramente no início de 2020, quando começou o período da Covid 19 atingido um valor aproximado de 0,17 unidades de desvio padrão de retornos *log*. Houve de facto pontos mínimos muito próximos como no meio de 2018, no meio de 2020 e recentemente em 2024 por exemplo com valores inferiores ao 0,05 unidades de desvio padrão de retornos *log*. A previsão é que no período de 90 dias previstos a nível de volatilidade iria subir.

No modelo EGARCH verificamos o ponto máximo no mesmo período do que no modelo anterior também a rondar os 0,17 unidades de desvio padrão de retornos *log*. Os pontos mínimos seguem a mesma lógica nos períodos do meio de 2018, no meio de 2020 e recentemente em 2024 com valores aos 0,025 unidades de desvio padrão de retornos *log*. Por sua vez também existe uma previsão de aumento de volatilidade para o próximo período de 90 dias.

Já o modelo GJR-GARCH (1,1) apesar de apresentar o ponto máximo e mínimo em períodos semelhantes aos modelos GARCH (1,1) e o EGARCH (1,1), apresenta uma escala diferente. O seu ponto máximo ultrapassa as 0,125 05 unidades de desvio padrão de retornos *log*. Por sua vez os seus períodos mínimos apresentam valores menores que os 0,025 05 unidades de desvio padrão de retornos *log*. O que mais tem em comum com os modelos inferiores é sua previsão de aumento de volatilidade para os próximos 90 dias.

Desta forma e com os valores obtidos tão semelhantes foi necessário realizar vários para compreender qual dos modelos da família GARCH (1,1) será mais adequado para estimar a nossa variável em estudo. Em primeiro lugar, foi realizado o teste Box-Ljung que é utilizado para verificar se há autocorrelação significativa nos resíduos padronizados do modelo. Os valores obtidos foram sempre superiores a 0,05. Para o modelo GARCH (1,1) obtivemos 0,4894, para o modelo EGARCH (1,1) 0,2862 e para o modelo GJR 0,5054 ao verificarmos os outputs 4,5 e 6. verificando os outputs Visto que os valores foram sempre superiores todos os modelos a 0,05 parecem adequados para a estimação da volatilidade da Bitcoin. Para além disso ao verificarmos as figuras de função de autocorrelação (ACF) dos resíduos padronizados de cada modelo GARCH utilizado, as barras encontram-se contidas nos limites de confiança, o que indica que os resíduos dos modelos GARCH, EGARCH e GJR-GARCH são aproximadamente independentes ou seja o que os modelos se encontram bem ajustados.

De seguida realizamos o teste LR com o p-value de 0.1394. O valor obtido é maior que 0.05, não rejeitamos a hipótese nula de que o modelo GARCH (1,1) será suficiente para a estimação. Portanto, o modelo GARCH (1,2) não proporciona um ajuste significativamente melhor aos dados em comparação com o GARCH (1,1).

Posteriormente foi estimado o RME (*Root Mean Squared Error*) de cada modelo. Todos os modelos obtiveram diferentes resultados que nos permitiram chegar a uma conclusão. O RMSE GARCH foi 0.0771. para o modelo EGARCH obtivemos 0.0490 e por último para o GJR-GARCH o valor obtido foi 0.0753 ao verificarmos o output8. Entre os resultados obtidos ao estimarmos o RMSE para decidir o melhor modelo para estimação da nossa variável precisamos de verificar de facto qual o modelo que

apresenta o menor valor entre os 3 apresentados. No caso em questão trata-se do modelo EGARCH (1,1) com um valor de 0.0490 e desta forma podemos concluir que este será de facto o melhor modelo a utilizar para variável da Bitcoin.

Para fortalecer ainda mais a nossa análise econométrica, utilizamos também os critérios de informação AIC e BIC para ver se a condição apresentada anteriormente se mantém.

Os valores obtidos foram os seguintes: para o modelo GARCH (1,1) AIC de -4.25120165001075 e BIC de -4.24092189780378, para o modelo EGARCH (1,1) AIC de -4.26305623313004 e BIC de -4.25020654287133 e por último para o modelo GJR-GARCH (1,1) AIC de -4.25055544936486 e BIC de -4.23770575910615 ao verificarmos o output 9. Ao observarmos os valores conseguimos verificar quem apresenta os menores valores em ambos os critérios é o modelo EGARCH. Desta forma e através do cálculo do RMSE, podemos concluir que o modelo EGARCH (1,1) será de facto o melhor modelo para prever a volatilidade da Bitcoin.

4.3.2 ETF WGMI

No caso do ETF WGMI, seguimos a mesma estrutura que na Bitcoin com intuito de compreender os resultados. Após verificarmos a estacionaridade da série, partimos para análise dos modelos.

Ao analisarmos as figuras obtidas, nomeadamente figura 4,5 e 6 deparamo-nos que as volatilidades seguem o mesmo padrão, no sentido em que apresentam pontos máximos e mínimos semelhantes. No modelo GARCH (1,1) o ponto máximo encontra-se no início de 2020, no entanto ao longo de 2024 existem valores a rondar o máximo anteriormente apresentado com valores a ultrapassar os 0,060 unidades de desvio padrão de retornos log. Os seus pontos mínimos encontram-se no início do ano de 2023 e ao longo do próprio com valores inferiores 0,050 unidades de desvio padrão de retornos log. A previsão é que a volatilidade para os próximos 90 dias terá uma ligeira subida tendo em conta último valor apresentado.

No modelo EGARCH (1,1) os pontos máximos apresentam os mesmos valores e nos mesmo períodos, mas o ponto mínimo ocorre a meio de 2022 com valores inferiores aos 0,045 unidades de desvio padrão de retornos log. Como aconteceu com o modelo anterior a previsão é que a volatilidade para os próximos 90 dias irá ter uma ligeira subida comparando com o último valor apresentado.

Já no modelo GJR-GARCH (1,1) apresenta o mesmo ponto mínimo que o modelo anterior no meio de 2022 a rondar os 0,044 unidades de desvio padrão de retornos log. Já o ponto máximo ultrapassa os 0,07 unidades de desvio padrão de retornos log no conhecido período do início de 2023. Tal como os seus antecessores é apresentada uma ligeira subida para a previsão de volatilidade comparando com o último valor apresentado.

Viste que os modelos apresentam resultados semelhantes iremos realizar os teste de pós estimação. Ao realizarmos o teste Box-Ljung para cada um dos modelos obtemos os seguintes valores: modelo GARCH (1,1) 0.4399, EGARCH(1,1) 0.3506 e GJR-GARCH(1,1) 0.3619. Como todos os modelos apresentam valores superior a 0,05 concluímos que não há evidência significativa de autocorrelação nos resíduos. Como aconteceu anteriormente ao observarmos as figuras das funções de autocorrelação, as barras desta figura encontram-se dentro dos limites de confiança o que indica que os resíduos dos modelos GARCH, EGARCH e GJR-GARCH são aproximadamente independentes ou seja o que os modelos se encontram bem ajustados.

De seguida efetuamos o teste LR, e mais uma vez como o p-valor é 1 e maior que 0.05, não rejeitamos a hipótese nula de que o modelo GARCH (1,1) é suficiente. Portanto, o modelo GARCH (1,2) não proporciona um ajuste significativamente melhor aos dados em comparação com o GARCH (1,1). Tal como sucedeu quando realizamos este teste para Bitcoin.

Assim sendo fomos calcular o RMSE para cada um dos modelos. Os valores apresentados foram os seguintes: modelo GARCH (1,1) 0.0724837036390901, EGARCH (1,1) 0.0722061963955763 e GJR-GARCH (1,1) 0.0727392511766621. Podemos verificar que o valor mais baixo do RMSE é o do modelo EGARCH que mais uma vez nos parece o modelo mais adequado para estimação da volatilidade no caso do ETF WGMI.

Ainda assim vamos utilizar os critérios de informação BIC e AIC. Os resultados obtidos foram os seguintes: para o modelo GARCH (1,1) AIC -3.08830884357657 e BIC -3.0597656482018, para o modelo EGARCH (1,1) AIC -3.09326846021364 e BIC 3.05758946599517 e por último para o modelo GJR-GARCH (1,1) AIC: -3.09254851245305 e BIC: -3.05686951823458.

Ao analisarmos os resultados deparamos que o modelo com o menor AIC é o EGARCH (1,1) e com o menor BIC é o GARCH (1,1).

É importante referir que o critério de informação AIC é melhor para a previsão e o BIC para interpretação e simplicidade do modelo¹¹. Visto que queremos prever a volatilidade num período de 90 dias, iremos ter mais atenção o critério AIC e desta forma escolher o modelo EGARCH também para estimar a volatilidade do ETF WGMI.

4.4 Relação com a pergunta de Investigação

A análise dos dados realizada da Bitcoin focou-se essencialmente em identificar o melhor modelo para estimar a volatilidade futura da criptomoeda. Esta questão é fundamental para entender as flutuações de preços, conseguir mitigar riscos e planejar estratégias de investimento que se adequem aos portfólios. Ao longo do processo, utilizamos diversos modelos da família GARCH, incluindo o GARCH (1,1), EGARCH (1,1) e GJR-GARCH (1,1), juntamente com testes de pós estimação para determinar qual modelo oferece a melhor precisão para prever a volatilidade futura da Bitcoin.

Após verificarmos que a série de dados da Bitcoin era estacionária, aplicamos os três modelos GARCH mencionados e comparamos os resultados obtidos. Observamos que, apesar dos três modelos identificarem padrões semelhantes de volatilidade ao longo do tempo, o modelo EGARCH (1,1) destacou-se em termos de desempenho. Este modelo não só apresentou o menor erro quadrático médio (RMSE), indicando maior precisão nas previsões, mas também obteve os menores valores no critério de informação AIC, o que reforça sua capacidade de prever volatilidade com maior exatidão em comparação aos outros modelos.

Além disso, o modelo EGARCH é conhecido por capturar melhor a volatilidade assimétrica e os efeitos de alavancagem presentes nos retornos financeiros, características frequentemente observadas no mercado de criptomoedas como o da Bitcoin. Esses fatores tornam o EGARCH (1,1) o modelo mais apropriado para a previsão da volatilidade futura da Bitcoin, conforme os resultados da nossa análise.

¹¹ Burnham, K. P., & Anderson, D. R. (2004). Multimodel inference: Understanding AIC and BIC in model selection. *Sociological Methods & Research*, 33(2), 261-304.

No caso do ETF WGMI como foi referido na secção anterior o modelo EGARCH parece-nos também mais adequado.

Portanto, ao responder à pergunta da Dissertação, concluímos que o modelo EGARCH (1,1) é o mais adequado para estimar a volatilidade futura da Bitcoin, pois combina precisão estatística e a capacidade de capturar dinâmicas complexas de volatilidade que são cruciais para previsões financeiras robustas.

É importante referir que com um tema tão complexo como a volatilidade de uma criptomoeda, cuja trajetória é muitas vezes tão difícil de prever, devem ser utilizados modelos mais sofisticados de modo a conseguir captar a volatilidade apresentada. Apesar dos modelos GARCH serem uma boa base de referência, os modelos GARCH podem servir como ponto de partida para uma análise mais profunda.

CAPÍTULO 5

Conclusão

A presente Dissertação tem como principal objetivo encontrar o melhor modelo para a previsão da volatilidade a 90 dias para a Bitcoin. Esta análise é alargada por forma a incluir um ETF constituído por empresas que trabalham com a mineração de criptomoedas, e não exclusivamente a Bitcoin. A nossa pergunta de investigação pretende então aferir qual será o melhor modelo para estimar a volatilidade futura da Bitcoin?”. Para responder a esta questão científica, serão utilizados modelos da família GARCH, para compreender com base nas nossas séries de dados com um intervalo de tempo determinado compreender qual seria o melhor modelo para prever a volatilidade. É importante referir que os modelos usam também a volatilidade histórica das séries, a partir dos dados históricos disponíveis.

Os dados de ambas as séries foram retirados da YAHOO Finance e o respetivo código no R-Studio permitiu implementar a metodologia adotada. A metodologia foi constituída por vários testes de modo a validar a possível estacionariedade das séries da Bitcoin e do ETF WGMI, nomeadamente o teste ADF, KPSS e PP. Para além disso da família dos modelos GARCH foram ainda utilizados os modelos GARCH (1,1), EGARCH (1,1) e GJR-GARCH (1,1) para obtermos os resultados. E por últimos testes de pós estimação como caso do teste Box-Ljung, teste LR com os RMSE e os critérios de informação AIC e BIC.

A Dissertação efetua então uma análise mais detalhada da volatilidade da Bitcoin e do ETF WGMI, utilizando diferentes modelos da família GARCH, como o GARCH(1,1), EGARCH(1,1) e GJR-GARCH(1,1). A análise começa pela verificação da estacionariedade das séries temporais, com testes ADF, KPSS e PP, confirmando que ambas as séries são estacionárias. Após a validação da estacionariedade, aplicaram-se modelos GARCH para estimar a volatilidade histórica e prever a volatilidade futura para um período de 90 dias.

Os resultados alcançados indicam que o modelo EGARCH (1,1) apresenta o melhor desempenho para prever a volatilidade tanto da Bitcoin quanto do ETF WGMI, como indicado pelos menores valores de RMSE e pelo critério de informação AIC. O teste Box-Ljung e a análise das funções de autocorrelação (ACF) confirmaram que os resíduos dos modelos são aproximadamente independentes, sugerindo que os modelos estão bem ajustados. O teste LR não mostrou melhorias significativas ao expandir o modelo GARCH, reforçando a adequação do GARCH (1,1).

O valor acrescentado desta Dissertação está na aplicação comparativa de diferentes modelos de volatilidade em dois ativos distintos, Bitcoin e ETF WGMI, fornecendo uma perspetiva crítica sobre qual modelo é o mais adequado para prever a volatilidade. Além disso, ao combinar critérios estatísticos e económicos, a Dissertação oferece uma abordagem consistente para a escolha do modelo, contribuindo para a literatura académica com uma análise detalhada de ativos financeiros em mercados voláteis e emergentes.

Para o caso de possíveis investigações futuras haverá a possibilidade de explorar a aplicação de modelos GARCH multivariados para capturar interações entre diferentes ativos financeiros. Além disso, o uso de redes neurais ou modelos híbridos pode de facto aprimorar previsões de volatilidade. Estudos que integram impactos de eventos macroeconómicos globais também seriam valiosos para

ampliar a compreensão da volatilidade em mercados emergentes como o de criptomoedas, onde existem muito ainda para explorar.

Concluindo, esta Dissertação contribui para uma compreensão mais profunda da volatilidade no mercado de criptomoedas e ETFs, utilizando modelos avançados da família GARCH. Os resultados demonstram a importância de uma modelagem robusta para previsões mais precisas, fornecendo uma base sólida para futuras pesquisas neste campo dinâmico e em constante evolução.

As maiores dificuldades da dissertação encontraram-se a encontrar os dados para elaboração dos testes e modelos. Para além disso acredito que modelos de machine learning com o suporte de Inteligência Artificial conseguirão prever níveis de volatilidade mais precisos.

Referências Bibliográficas

- Antonopoulos, A. M. (2014). *Mastering Bitcoin: Unlocking digital cryptocurrencies*. O'Reilly Media, Inc.
- Ardia, D., Bluteau, K., & Rüede, M. (2019). Regime changes in Bitcoin GARCH volatility dynamics. *Finance Research Letters*, 29, 266-271.
- Bibi, S. (2023). Money in the time of crypto. *Research in International Business and Finance*, 65, 101964.
- Burnham, K. P., & Anderson, D. R. (2004). Multimodel inference: Understanding AIC and BIC in model selection. *Sociological Methods & Research*, 33(2), 261-304.
- Conrad, C., Custovic, A., & Ghysels, E. (2018). Long-and short-term cryptocurrency volatility components: A GARCH-MIDAS analysis. *Journal of Risk and Financial Management*, 11(2), 23.
- Coulter, K. A. (2022). The impact of news media on Bitcoin prices: Modelling data driven discourses in the crypto economy with natural language processing. *Royal Society Open Science*, 9(4), 220276.
- Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366a), 427-431.
- Fakhfekh, M., & Jeribi, A. (2020). Volatility dynamics of crypto-currencies' returns: Evidence from asymmetric and long memory GARCH models. *Research in International Business and Finance*, 51, 101075.
- Kat, H. M., & Heynen, R. C. (1994). Volatility Prediction: A Comparison of the Stochastic Volatility, GARCH (1, 1) and Egarch (1, 1) Models. *Journal of Derivatives*, 2(2).
- Kim, A., Trimborn, S., & Härdle, W. K. (2021). VCRIX—A volatility index for cryptocurrencies. *International Review of Financial Analysis*, 78, 101915.
- Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics*, 54(1-3), 159-178.
- Liu, J., & Serletis, A. (2019). Volatility in the cryptocurrency market. *Open Economies Review*, 30(4), 779-811.
- Matic, J. L., Packham, N., & Härdle, W. K. (2023). Hedging cryptocurrency options. *Review of Derivatives Research*, 26(1), 91-133.
- Mostafa, F., Saha, P., Islam, M. R., & Nguyen, N. (2021). GJR-GARCH volatility modeling under NIG and ANN for predicting top cryptocurrencies. *Journal of Risk and Financial Management*, 14(9), 421.

- Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press.
- Nasdaq. (2022, February 8). WGMI ETF: 7 things to know as the Valkyrie Bitcoin Miners ETF starts trading today. Retrieved August 10, 2024, from <https://www.nasdaq.com/articles/wgmi-etf:-7-things-to-know-as-the-valkyrie-bitcoin-miners-etf-starts-trading-today>
- Nelson, D. B. (1990). Stationarity and persistence in the GARCH (1, 1) model. *Econometric Theory*, 6(3), 318-334.
- Phillips, P. (1988). Testing for unit roots in time series regression. *Biometrika*, 71, 599-607.
- Sapkota, N. (2022). News-based sentiment and bitcoin volatility. *International Review of Financial Analysis*, 82, 102183.
- Seaman, B. S. (2022). *The Coinmen* (Fourth Edition). CyberCurb.
- US News. (n.d.). WGMI - Valkyrie Bitcoin Miners ETF Stock Price, Holdings, Quote & News. Retrieved August 10, 2024, from <https://money.usnews.com/funds/etfs/large-blend/valkyrie-bitcoin-miners-etf/wgmi>
- Valkyrie Funds. (n.d.). WGMI - Valkyrie Bitcoin Miners ETF. Retrieved August 10, 2024, from <https://valkyrie-funds.com/wgmi/>
- Zulfiqar, N., & Gulzar, S. (2021). Implied volatility estimation of bitcoin options and the stylized facts of option pricing. *Financial Innovation*, 7(1), 67.

Anexos 1:

Formulas matemáticas dos testes e modelos

$$\Delta Y_t = \alpha + \beta t + \gamma Y_{t-1} + \sum_{i=1}^p \delta_i \Delta Y_{t-i} + \epsilon_t$$

ΔY_t : A primeira diferença da série temporal no tempo t .

α : Termo constante (intercepto).

βt : Termo de tendência temporal (se incluído).

γ : Coeficiente do nível defasado da série temporal $Y_{\{t-1\}}$, que testa a presença de uma raiz unitária.

$\sum_{i=1}^p \delta_i \Delta Y_{t-i}$: Soma das diferenças defasadas da variável dependente para contabilizar autocorrelação de ordem superior.

ϵ_t : Termo de erro (ruído branco).

$$KPSS = \frac{1}{T^2} \sum_{t=1}^T S_t^2 / \hat{\sigma}^2$$

T : O número total de observações na série temporal. Isso representa o tamanho da amostra.

S_t : A soma cumulativa dos resíduos até o tempo t . Ele é calculado como:

$$S_t = \sum_{i=1}^t \hat{\epsilon}_i$$

onde $\hat{\epsilon}_i$ são os resíduos estimados do modelo de regressão. Esses resíduos representam as diferenças entre os valores observados e os valores ajustados pela regressão.

$\hat{\sigma}^2$: A estimativa da variância dos resíduos. Essa variância é usada para padronizar a soma cumulativa dos resíduos e garantir que a estatística KPSS seja dimensionada adequadamente.

PP Test

$$\Delta Y_t = \alpha + \beta t + \gamma Y_{t-1} + \epsilon_t$$

ΔY_t : Primeira diferença da série temporal Y no tempo t .

α : Termo constante (intercepto).

βt : Termo de tendência temporal (se incluído).

γY_{t-1} : Coeficiente do nível defasado da série temporal YYY, que testa a presença de uma raiz unitária.

ϵ_t : Termo de erro (ruído branco).

Modelo GARCH (1,1)

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

σ_t^2 é a variância condicional no tempo ttt (volatilidade),

ω é uma constante,

$\alpha \epsilon_{t-1}^2$ é o impacto dos choques passados (retornos anteriores),

$\beta \sigma_{t-1}^2$ é a volatilidade passada.

Modelo EGARCH (1,1)

$$\log(\sigma_t^2) = \omega + \beta \log(\sigma_{t-1}^2) + \alpha \left(\frac{\epsilon_{t-1}}{\sigma_{t-1}} \right) + \gamma \left(\left| \frac{\epsilon_{t-1}}{\sigma_{t-1}} \right| - E \left(\left| \frac{\epsilon_{t-1}}{\sigma_{t-1}} \right| \right) \right)$$

$\log(\sigma_t^2)$ O logaritmo natural da variância condicional (σ_t^2) no tempo t. Usar o logaritmo garante que a variância condicional seja sempre positiva, sem a necessidade de impor restrições nos parâmetros.

ω É uma constante que controla o nível médio da volatilidade. Representa o termo base que não depende de choques anteriores ou da volatilidade passada.

$\beta \log(\sigma_{t-1}^2)$ Captura a persistência da volatilidade. Ou seja, se β for alto, a volatilidade passada (σ_{t-1}^2) tem um grande efeito na volatilidade atual. Esse termo modela o impacto da volatilidade anterior na volatilidade futura.

$\alpha \left(\frac{\epsilon_{t-1}}{\sigma_{t-1}} \right)$ Este termo representa o impacto dos retornos passados (ϵ_{t-1}) padronizados pela volatilidade anterior (σ_{t-1}) . A magnitude e o sinal de α determinam como os choques anteriores afetam a volatilidade futura. Se $\alpha > 0$, grandes retornos passados aumentam a volatilidade.

$\gamma \left(\left| \frac{\epsilon_{t-1}}{\sigma_{t-1}} \right| - E \left(\left| \frac{\epsilon_{t-1}}{\sigma_{t-1}} \right| \right) \right)$ Este termo captura o efeito assimétrico na volatilidade. Aqui, mede o impacto de choques positivos e negativos de forma diferente. Se $\gamma < 0$, choques negativos $(\epsilon_{t-1} < 0)$ aumentam mais a volatilidade do que choques positivos de mesma magnitude. O termo absoluto $\left(\left| \frac{\epsilon_{t-1}}{\sigma_{t-1}} \right| \right)$ é usado para medir o tamanho do choque, enquanto o esperado (E) serve como ajuste para capturar desvios da média.

Modelo GJR-GARCH (1,1)

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \gamma \epsilon_{t-1}^2 I(\epsilon_{t-1} < 0) + \beta \sigma_{t-1}^2$$

σ_t^2 Variância condicional no tempo t, ou seja, a volatilidade atual.

ω Constante que captura o nível base da volatilidade. Funciona como o termo constante no modelo, que não depende de choques passados ou volatilidade passada.

$\alpha \epsilon_{t-1}^2$ Impacto dos choques passados. A magnitude de ϵ_{t-1}^2 reflete a influência do retorno no período anterior sobre a volatilidade atual. Quanto maior α , maior o impacto dos retornos passados na volatilidade.

$\gamma \epsilon_{t-1}^2 I(\epsilon_{t-1} < 0)$ Este é o termo assimétrico do modelo GJR-GARCH. Este usa uma **função indicadora** $I(\epsilon_{t-1} < 0)$ que vale 1 quando ϵ_{t-1} é negativo e 0 quando ϵ_{t-1} é positivo. Isso significa que quando o retorno anterior foi negativo $(\epsilon_{t-1} < 0)$, o efeito desse retorno na volatilidade atual é aumentado em $\gamma \epsilon_{t-1}^2$. Se $\gamma > 0$ choques negativos aumentam mais a volatilidade do que choques positivos, capturando o efeito de alavancagem.

$\beta \sigma_{t-1}^2$ Persistência da volatilidade. Assim como no GARCH (1,1), este termo captura o efeito da volatilidade passada (σ_{t-1}^2) na volatilidade atual. Quanto maior β , mais persistente é a volatilidade.

Teste de Ljung-Box

$$Q = n(n + 2) \sum_{k=1}^h \frac{\hat{\rho}_k^2}{n - k}$$

Q : Estatística do teste de Ljung-Box.

n : Tamanho da amostra (número de observações).

h : Número de defasagens (lags) a serem testadas.

$\hat{\rho}_k$: Autocorrelação estimada no lag k .

LR Test

$$LR = -2 (\log(L_{restrito}) - \log(L_{nãorestrito}))$$

$L_{restrito}$: Log-verossimilhança do modelo restrito (mais simples).

$L_{nãorestrito}$: Log-verossimilhança do modelo não restrito (mais complexo).

RMSE

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2}$$

y_t : Valor observado no tempo t

\hat{y}_t : Valor previsto pelo modelo no tempo t

n : Número de observações

AIC

$$AIC = 2k - 2 \log(L)$$

k : Número de parâmetros no modelo

L : Log-verossimilhança do modelo

BIC

$$BIC = \log(n)k - 2 \log(L)$$

n : Número de observações

k : Número de parâmetros no modelo

L : Log-verossimilhança do modelo

Anexos 2:

Código utilizado

GARCH

```
# Load necessary libraries
```

```
library(quantmod)
```

```
library(rugarch)
```

```
library(ggplot2)
```

```
# Set the start and end dates for historical data
```

```
start_date <- as.Date("2018-06-01")
```

```
end_date <- Sys.Date()
```

```
# Get the Bitcoin data from Yahoo Finance
```

```
getSymbols("BTC-USD", src = "yahoo", from = start_date, to = end_date)
```

```
# Extract the adjusted closing prices
```

```
btc <- Cl(`BTC-USD`)
```

```
# Calculate daily returns
```

```
btc_returns <- dailyReturn(btc, type = "log")
```

```

# Specify the GARCH(1,1) model
spec <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
  distribution.model = "std"
)

# Fit the GARCH model
fit <- ugarchfit(spec = spec, data = btc_returns)

# Forecast the volatility for the next few months
months_ahead <- 3 # Change this value to the number of months you want to forecast
days_per_month <- 30
forecast_horizon <- months_ahead * days_per_month
forecast <- ugarchforecast(fit, n.ahead = forecast_horizon)

# Extract the forecasted volatility
forecast_volatility <- sigma(forecast)

# Create a data frame with the forecasted volatility
forecast_dates <- seq(end_date + 1, by = "days", length.out = forecast_horizon)
forecast_df <- data.frame(Date = forecast_dates, Forecasted_Volatility =
as.numeric(forecast_volatility))

# Print the first few rows of the forecasted volatility data frame
head(forecast_df)

```

```

# Plot the historical and forecasted volatility
volatility_df <- data.frame(Date = index(sigma(fit)), Volatility = coredata(sigma(fit)))
combined_dates = c(volatility_df$Date, forecast_df$Date)
comb_df <- data.frame(Date = combined_dates)

library(dplyr)

df_merged <- left_join(comb_df, volatility_df, by = "Date")
df_merged <- left_join(df_merged, forecast_df, by = "Date")
tail(df_merged)

```

Testes para ver se é estacionária

```

result_adf <- adf.test(btc_returns)
print(paste("ADF Statistic:", result_adf$statistic))
print(paste("p-value:", result_adf$p.value))
print("Critical Values:")
print(result_adf$critical)

result_kpss <- kpss.test(btc_returns)
print(paste("KPSS Statistic:", result_kpss$statistic))
print(paste("p-value:", result_kpss$p.value))
print("Critical Values:")
print(result_kpss$critical)

library(tseries)

```

```

# Perform PP test
result_pp <- pp.test(btc_returns)

# Print PP statistic and p-value
print(paste("PP Statistic:", result_pp$statistic))
print(paste("p-value:", result_pp$p.value))

# Print critical values
print("Critical Values:")
print(result_pp$critical)

#Modelo EGARCH

# Set the start and end dates for historical data
start_date <- as.Date("2018-06-01")
end_date <- Sys.Date()

# Get the Bitcoin data from Yahoo Finance
getSymbols("BTC-USD", src = "yahoo", from = start_date, to = end_date)

# Extract the adjusted closing prices
btc <- Cl(`BTC-USD`)

# Calculate daily returns
btc_returns <- dailyReturn(btc, type = "log")

# Specify the EGARCH(1,1) model

```

```
egarch_spec <- ugarchspec(  
  variance.model = list(model = "eGARCH", garchOrder = c(1, 1)),  
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),  
  distribution.model = "std"  
)
```

Fit the EGARCH model

```
egarch_fit <- ugarchfit(spec = egarch_spec, data = btc_returns)
```

```
# Forecast the volatility for the next few months
```

```
months_ahead <- 3 # Change this value to the number of months you want to forecast
```

```
days_per_month <- 30
```

```
forecast_horizon <- months_ahead * days_per_month
```

```
forecast_egarch <- ugarchforecast(egarch_fit, n.ahead = forecast_horizon)
```

```
# Extract the forecasted volatility
```

```
forecast_volatility_egarch <- sigma(forecast_egarch)
```

```
# Create a data frame with the forecasted volatility
```

```
forecast_dates_egarch <- seq(end_date + 1, by = "days", length.out = forecast_horizon)
```

```
forecast_df_egarch <- data.frame(Date = forecast_dates_egarch, Forecasted_Volatility =  
as.numeric(forecast_volatility_egarch))
```

```
# Print the first few rows of the forecasted volatility data frame
```

```
head(forecast_df_egarch)
```

```
# Plot the historical and forecasted volatility
```

```

volatility_df_egarch <- data.frame(Date = index(sigma(egarch_fit)), Volatility =
coredata(sigma(egarch_fit)))

combined_dates_egarch = c(volatility_df_egarch$Date, forecast_df_egarch$Date)

comb_df_egarch <- data.frame(Date = combined_dates_egarch)

df_merged_egarch <- left_join(comb_df_egarch, volatility_df_egarch, by = "Date")
df_merged_egarch <- left_join(df_merged_egarch, forecast_df_egarch, by = "Date")

ggplot(df_merged_egarch, aes(x = Date, y = Volatility)) +
  geom_line(data = volatility_df_egarch, aes(x = Date, y = Volatility), color = "blue") +
  geom_line(data = forecast_df_egarch, aes(x = Date, y = Forecasted_Volatility), color = "red",
linetype = "dashed") +
  labs(title = paste("Bitcoin Volatility (EGARCH(1,1)) - Historical and", months_ahead, "Months
Forecasted"),
  x = "Date", y = "Volatility") +
  theme_minimal()

```

#Modelo de GJR-GARCH

```

# Load necessary libraries

library(quantmod)

library(rugarch)

library(ggplot2)

# Set the start and end dates for historical data

start_date <- as.Date("2018-06-01")

end_date <- Sys.Date()

# Get the Bitcoin data from Yahoo Finance

```

```

getSymbols("BTC-USD", src = "yahoo", from = start_date, to = end_date)

# Extract the adjusted closing prices
btc <- Cl(`BTC-USD`)

# Calculate daily returns
btc_returns <- dailyReturn(btc, type = "log")

# Specify the GJR-GARCH(1,1) model
gjr_garch_spec <- ugarchspec(
  variance.model = list(model = "gjrGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
  distribution.model = "std"
)

# Fit the GJR-GARCH model
gjr_garch_fit <- ugarchfit(spec = gjr_garch_spec, data = btc_returns)

# Forecast the volatility for the next few months
months_ahead <- 3 # Change this value to the number of months you want to forecast
days_per_month <- 30
forecast_horizon <- months_ahead * days_per_month
forecast_gjr_garch <- ugarchforecast(gjr_garch_fit, n.ahead = forecast_horizon)

# Extract the forecasted volatility
forecast_volatility_gjr_garch <- sigma(forecast_gjr_garch)

```

```

# Create a data frame with the forecasted volatility

forecast_dates_gjr_garch <- seq(end_date + 1, by = "days", length.out = forecast_horizon)

forecast_df_gjr_garch <- data.frame(Date = forecast_dates_gjr_garch, Forecasted_Volatility =
as.numeric(forecast_volatility_gjr_garch))

# Print the first few rows of the forecasted volatility data frame

head(forecast_df_gjr_garch)

# Plot the historical and forecasted volatility

volatility_df_gjr_garch <- data.frame(Date = index(sigma(gjr_garch_fit)), Volatility =
coredata(sigma(gjr_garch_fit)))

combined_dates_gjr_garch = c(volatility_df_gjr_garch$Date, forecast_df_gjr_garch$Date)

comb_df_gjr_garch <- data.frame(Date = combined_dates_gjr_garch)

df_merged_gjr_garch <- left_join(comb_df_gjr_garch, volatility_df_gjr_garch, by = "Date")

df_merged_gjr_garch <- left_join(df_merged_gjr_garch, forecast_df_gjr_garch, by = "Date")

ggplot(df_merged_gjr_garch, aes(x = Date, y = Volatility)) +

  geom_line(data = volatility_df_gjr_garch, aes(x = Date, y = Volatility), color = "blue") +

  geom_line(data = forecast_df_gjr_garch, aes(x = Date, y = Forecasted_Volatility), color = "red",
linetype = "dashed") +

  labs(title = paste("Bitcoin Volatility (GJR-GARCH(1,1)) - Historical and", months_ahead, "Months
Forecasted"),

  x = "Date", y = "Volatility") +

  theme_minimal()

# Testes de para ver os resíduos

# Residual diagnostics for GARCH(1,1)

```

```

garch_residuals <- residuals(fit, standardize = TRUE)
acf(garch_residuals, main = "ACF of Standardized Residuals - GARCH")
Box.test(garch_residuals, lag = 20, type = "Ljung-Box")

# Residual diagnostics for EGARCH(1,1)
egarch_residuals <- residuals(egarch_fit, standardize = TRUE)
acf(egarch_residuals, main = "ACF of Standardized Residuals - EGARCH")
Box.test(egarch_residuals, lag = 20, type = "Ljung-Box")

# Residual diagnostics for GJR-GARCH(1,1)
gjr_garch_residuals <- residuals(gjr_garch_fit, standardize = TRUE)
acf(gjr_garch_residuals, main = "ACF of Standardized Residuals - GJR-GARCH")
Box.test(gjr_garch_residuals, lag = 20, type = "Ljung-Box")

```

#Likelihood test

```

# Fit a more complex GARCH model, e.g., GARCH(1,2)
spec_garch_12 <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 2)),
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
  distribution.model = "std"
)
fit_garch_12 <- ugarchfit(spec = spec_garch_12, data = btc_returns)

# Likelihood Ratio Test (veremos se vou usar)
ll_garch_11 <- fit@fit$LLH
ll_garch_12 <- fit_garch_12@fit$LLH

```

```

LR_statistic <- -2 * (ll_garch_11 - ll_garch_12)
p_value <- pchisq(LR_statistic, df = 1, lower.tail = FALSE)
print(paste("LR Statistic:", LR_statistic))
print(paste("p-value:", p_value))

# Similar process for EGARCH and GJR-GARCH models can be done

# Split data into training and test sets
train_end <- index(btc)[floor(0.8 * length(btc))]
btc_train <- window(btc_returns, end = train_end)
btc_test <- window(btc_returns, start = train_end + 1)

# Fit the models on the training set
fit_garch_train <- ugarchfit(spec = spec, data = btc_train)
fit_egarch_train <- ugarchfit(spec = egarch_spec, data = btc_train)
fit_gjr_garch_train <- ugarchfit(spec = gjr_garch_spec, data = btc_train)

# Forecast on the test set
forecast_garch <- ugarchforecast(fit_garch_train, n.ahead = length(btc_test))
forecast_egarch <- ugarchforecast(fit_egarch_train, n.ahead = length(btc_test))
forecast_gjr_garch <- ugarchforecast(fit_gjr_garch_train, n.ahead = length(btc_test))

# Calculate RMSE for the forecasts
rmse <- function(actual, forecast) {
  sqrt(mean((actual - forecast)^2))
}

```

```

rmse_garch <- rmse(btc_test, sigma(forecast_garch))
rmse_egarch <- rmse(btc_test, sigma(forecast_egarch))
rmse_gjr_garch <- rmse(btc_test, sigma(forecast_gjr_garch))

print(paste("RMSE GARCH:", rmse_garch))
print(paste("RMSE EGARCH:", rmse_egarch))
print(paste("RMSE GJR-GARCH:", rmse_gjr_garch))

```

#AIC and BIC

```

# Load necessary libraries
library(quantmod)
library(rugarch)
library(ggplot2)

# Set the start and end dates for historical data
start_date <- as.Date("2018-06-01")
end_date <- Sys.Date()

# Get the Bitcoin data from Yahoo Finance
getSymbols("BTC-USD", src = "yahoo", from = start_date, to = end_date)

# Extract the adjusted closing prices
btc <- Cl(`BTC-USD`)

# Calculate daily returns
btc_returns <- dailyReturn(btc, type = "log")

```

```

# Specify the GARCH(1,1) model
spec_garch <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
  distribution.model = "std"
)

# Specify the EGARCH(1,1) model
spec_egarch <- ugarchspec(
  variance.model = list(model = "eGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
  distribution.model = "std"
)

# Specify the GJR-GARCH(1,1) model
spec_gjr_garch <- ugarchspec(
  variance.model = list(model = "gjrGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
  distribution.model = "std"
)

# Fit the models
fit_garch <- ugarchfit(spec = spec_garch, data = btc_returns)
fit_egarch <- ugarchfit(spec = spec_egarch, data = btc_returns)
fit_gjr_garch <- ugarchfit(spec = spec_gjr_garch, data = btc_returns)

```

```

# Extract AIC and BIC for each model

aic_garch <- infocriteria(fit_garch)[1]
bic_garch <- infocriteria(fit_garch)[2]
aic_egarch <- infocriteria(fit_egarch)[1]
bic_egarch <- infocriteria(fit_egarch)[2]
aic_gjr_garch <- infocriteria(fit_gjr_garch)[1]
bic_gjr_garch <- infocriteria(fit_gjr_garch)[2]

# Print the AIC and BIC values

print(paste("GARCH(1,1) - AIC:", aic_garch, "BIC:", bic_garch))
print(paste("EGARCH(1,1) - AIC:", aic_egarch, "BIC:", bic_egarch))
print(paste("GJR-GARCH(1,1) - AIC:", aic_gjr_garch, "BIC:", bic_gjr_garch))

# Determine the model with the lowest AIC and BIC

if(aic_garch < aic_egarch & aic_garch < aic_gjr_garch) {
  print("GARCH(1,1) has the lowest AIC")
} else if(aic_egarch < aic_garch & aic_egarch < aic_gjr_garch) {
  print("EGARCH(1,1) has the lowest AIC")
} else {
  print("GJR-GARCH(1,1) has the lowest AIC")
}

if(bic_garch < bic_egarch & bic_garch < bic_gjr_garch) {
  print("GARCH(1,1) has the lowest BIC")
} else if(bic_egarch < bic_garch & bic_egarch < bic_gjr_garch) {
  print("EGARCH(1,1) has the lowest BIC")
} else {

```

```
print("GJR-GARCH(1,1) has the lowest BIC")
}
```

#Mesma análise para a WGMI

```
# Load necessary libraries
```

```
library(quantmod)
```

```
library(tseries)
```

```
# Set the start and end dates for historical data
```

```
start_date <- as.Date("2022-02-08") # WGMI was launched on February 8, 2022
```

```
end_date <- Sys.Date()
```

```
end_date <- Sys.Date()
```

```
# Get the WGMI data from Yahoo Finance
```

```
getSymbols("WGMI", src = "yahoo", from = start_date, to = end_date)
```

```
# Extract the adjusted closing prices
```

```
wgmi <- Cl(WGMI)
```

```
# Calculate daily returns
```

```
wgmi_returns <- dailyReturn(wgmi, type = "log")
```

```
# Perform Augmented Dickey-Fuller (ADF) Test
```

```
result_adf <- adf.test(wgmi_returns)
```

```
# Print ADF results
```

```

cat("ADF Test Results:\n")
cat(paste("ADF Statistic:", result_adf$statistic), "\n")
cat(paste("p-value:", result_adf$p.value), "\n")
cat("Critical Values:\n")
print(result_adf$critical)

# Perform Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test
result_kpss <- kpss.test(wgmi_returns, null = "Level")

# Print KPSS results
cat("\nKPSS Test Results:\n")
cat(paste("KPSS Statistic:", result_kpss$statistic), "\n")
cat(paste("p-value:", result_kpss$p.value), "\n")
cat("Critical Values:\n")
print(result_kpss$critical)

# Perform Phillips-Perron (PP) Test
result_pp <- pp.test(wgmi_returns)

# Print PP results
cat("\nPP Test Results:\n")
cat(paste("PP Statistic:", result_pp$statistic), "\n")
cat(paste("p-value:", result_pp$p.value), "\n")
cat("Critical Values:\n")
print(result_pp$critical)

#GARCH(1,1) for WGMI for the next 90 days

```

```

# Load necessary libraries

library(quantmod)

library(rugarch)

library(ggplot2)

library(dplyr)

# Set the start and end dates for historical data

start_date <- as.Date("2022-02-08") # WGMI's inception date

end_date <- Sys.Date()

# Get the WGMI data from Yahoo Finance

getSymbols("WGMI", src = "yahoo", from = start_date, to = end_date)

# Extract the adjusted closing prices

wgmi <- Cl(WGMI)

# Calculate daily returns

wgmi_returns <- dailyReturn(wgmi, type = "log")

# Check for any NA values in returns and remove them if necessary

wgmi_returns <- na.omit(wgmi_returns)

# Specify the GARCH(1,1) model

spec <- ugarchspec(

  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),

  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),

```

```

distribution.model = "std"
)

# Fit the GARCH model
fit <- ugarchfit(spec = spec, data = wgmi_returns)

# Forecast the volatility for the next few months
months_ahead <- 3 # Change this value to the number of months you want to forecast
days_per_month <- 30
forecast_horizon <- months_ahead * days_per_month
forecast <- ugarchforecast(fit, n.ahead = forecast_horizon)

# Extract the forecasted volatility
forecast_volatility <- sigma(forecast)

# Create a data frame with the forecasted volatility
forecast_dates <- seq(end_date + 1, by = "days", length.out = forecast_horizon)
forecast_df <- data.frame(Date = forecast_dates, Forecasted_Volatility =
as.numeric(forecast_volatility))

# Print the first few rows of the forecasted volatility data frame
print("Forecasted Volatility:")
head(forecast_df)

# Extract historical volatility
historical_volatility <- sigma(fit)

# Create a data frame for historical volatility

```

```

volatility_df <- data.frame(Date = index(historical_volatility), Volatility =
coredata(historical_volatility))

# Combine historical and forecasted volatility for plotting
combined_dates <- c(volatility_df$Date, forecast_df$Date)
combined_df <- data.frame(Date = combined_dates)

# Merge historical and forecasted data frames
df_merged <- left_join(combined_df, volatility_df, by = "Date")
df_merged <- left_join(df_merged, forecast_df, by = "Date")

# Plot the historical and forecasted volatility
ggplot(df_merged, aes(x = Date)) +
  geom_line(aes(y = Volatility), color = "blue") +
  geom_line(aes(y = Forecasted_Volatility), color = "red", linetype = "dashed") +
  labs(title = "WGMI Volatility (GARCH(1,1)) - Historical and Forecasted",
       x = "Date", y = "Volatility") +
  theme_minimal()

# Print the last few rows of the merged data frame to see historical and forecasted volatility
print("Combined Historical and Forecasted Volatility:")
tail(df_merged)

```

#EGARCH(1,1) for WGMI for the next 90 days

```

# Load necessary libraries
library(quantmod)
library(rugarch)

```

```

library(ggplot2)

library(dplyr)

# Set the start and end dates for historical data
start_date <- as.Date("2022-02-08") # WGMI's inception date
end_date <- Sys.Date()

# Get the WGMI data from Yahoo Finance
getSymbols("WGMI", src = "yahoo", from = start_date, to = end_date)

# Extract the adjusted closing prices
wgmi <- Cl(WGMI)

# Calculate daily returns
wgmi_returns <- dailyReturn(wgmi, type = "log")

# Check for any NA values in returns and remove them if necessary
wgmi_returns <- na.omit(wgmi_returns)

# Specify the EGARCH(1,1) model
spec_egarch <- ugarchspec(
  variance.model = list(model = "eGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
  distribution.model = "std"
)

# Fit the EGARCH model

```

```

fit_egarch <- ugarchfit(spec = spec_egarch, data = wgmi_returns)

# Forecast the volatility for the next few months

months_ahead <- 3 # Change this value to the number of months you want to forecast

days_per_month <- 30

forecast_horizon <- months_ahead * days_per_month

forecast_egarch <- ugarchforecast(fit_egarch, n.ahead = forecast_horizon)

# Extract the forecasted volatility

forecast_volatility_egarch <- sigma(forecast_egarch)

# Create a data frame with the forecasted volatility

forecast_dates_egarch <- seq(end_date + 1, by = "days", length.out = forecast_horizon)

forecast_df_egarch <- data.frame(Date = forecast_dates_egarch, Forecasted_Volatility =
as.numeric(forecast_volatility_egarch))

# Print the first few rows of the forecasted volatility data frame

print("Forecasted Volatility using EGARCH:")

head(forecast_df_egarch)

# Extract historical volatility

historical_volatility_egarch <- sigma(fit_egarch)

# Create a data frame for historical volatility

volatility_df_egarch <- data.frame(Date = index(historical_volatility_egarch), Volatility =
coredata(historical_volatility_egarch))

# Combine historical and forecasted volatility for plotting

```

```

combined_dates_egarch <- c(volatility_df_egarch$Date, forecast_df_egarch$Date)
combined_df_egarch <- data.frame(Date = combined_dates_egarch)

# Merge historical and forecasted data frames
df_merged_egarch <- left_join(combined_df_egarch, volatility_df_egarch, by = "Date")
df_merged_egarch <- left_join(df_merged_egarch, forecast_df_egarch, by = "Date")

# Plot the historical and forecasted volatility
ggplot(df_merged_egarch, aes(x = Date)) +
  geom_line(aes(y = Volatility), color = "blue") +
  geom_line(aes(y = Forecasted_Volatility), color = "red", linetype = "dashed") +
  labs(title = "WGMI Volatility (EGARCH(1,1)) - Historical and Forecasted",
       x = "Date", y = "Volatility") +
  theme_minimal()

# Print the last few rows of the merged data frame to see historical and forecasted volatility
print("Combined Historical and Forecasted Volatility using EGARCH:")
tail(df_merged_egarch)

```

#GJR-GARCH(1,1) for WGMI for the next 90 days

```

# Load necessary libraries
library(quantmod)
library(rugarch)
library(ggplot2)
library(dplyr)

```

```

# Set the start and end dates for historical data
start_date <- as.Date("2022-02-08") # WGMI's inception date
end_date <- Sys.Date()

# Get the WGMI data from Yahoo Finance
getSymbols("WGMI", src = "yahoo", from = start_date, to = end_date)

# Extract the adjusted closing prices
wgmi <- Cl(WGMI)

# Calculate daily returns
wgmi_returns <- dailyReturn(wgmi, type = "log")

# Check for any NA values in returns and remove them if necessary
wgmi_returns <- na.omit(wgmi_returns)

# Specify the GJR-GARCH(1,1) model
spec_gjr <- ugarchspec(
  variance.model = list(model = "gjrGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
  distribution.model = "std"
)

# Fit the GJR-GARCH model
fit_gjr <- ugarchfit(spec = spec_gjr, data = wgmi_returns)

# Forecast the volatility for the next few months

```

```

months_ahead <- 3 # Change this value to the number of months you want to forecast
days_per_month <- 30
forecast_horizon <- months_ahead * days_per_month
forecast_gjr <- ugarchforecast(fit_gjr, n.ahead = forecast_horizon)

# Extract the forecasted volatility
forecast_volatility_gjr <- sigma(forecast_gjr)

# Create a data frame with the forecasted volatility
forecast_dates_gjr <- seq(end_date + 1, by = "days", length.out = forecast_horizon)
forecast_df_gjr <- data.frame(Date = forecast_dates_gjr, Forecasted_Volatility =
as.numeric(forecast_volatility_gjr))

# Print the first few rows of the forecasted volatility data frame
print("Forecasted Volatility using GJR-GARCH:")
head(forecast_df_gjr)

# Extract historical volatility
historical_volatility_gjr <- sigma(fit_gjr)

# Create a data frame for historical volatility
volatility_df_gjr <- data.frame(Date = index(historical_volatility_gjr), Volatility =
coredata(historical_volatility_gjr))

# Combine historical and forecasted volatility for plotting
combined_dates_gjr <- c(volatility_df_gjr$Date, forecast_df_gjr$Date)
combined_df_gjr <- data.frame(Date = combined_dates_gjr)

```

```

# Merge historical and forecasted data frames

df_merged_gjr <- left_join(combined_df_gjr, volatility_df_gjr, by = "Date")
df_merged_gjr <- left_join(df_merged_gjr, forecast_df_gjr, by = "Date")

# Plot the historical and forecasted volatility

ggplot(df_merged_gjr, aes(x = Date)) +
  geom_line(aes(y = Volatility), color = "blue") +
  geom_line(aes(y = Forecasted_Volatility), color = "red", linetype = "dashed") +
  labs(title = "WGMI Volatility (GJR-GARCH(1,1)) - Historical and Forecasted",
       x = "Date", y = "Volatility") +
  theme_minimal()

# Print the last few rows of the merged data frame to see historical and forecasted volatility

print("Combined Historical and Forecasted Volatility using GJR-GARCH:")
tail(df_merged_gjr)

```

#Testes de robustez para WGMI

```

# Fit the models

fit_garch <- ugarchfit(spec = spec_garch, data = wgmi_returns)
fit_egarch <- ugarchfit(spec = spec_egarch, data = wgmi_returns)
fit_gjr_garch <- ugarchfit(spec = spec_gjr_garch, data = wgmi_returns)

# Residual diagnostics for GARCH(1,1)

garch_residuals <- residuals(fit_garch, standardize = TRUE)
acf(garch_residuals, main = "ACF of Standardized Residuals - GARCH")
garch_lb_test <- Box.test(garch_residuals, lag = 20, type = "Ljung-Box")

```

```

print("Ljung-Box Test for GARCH(1,1):")
print(garch_lb_test)

# Residual diagnostics for EGARCH(1,1)
egarch_residuals <- residuals(fit_egarch, standardize = TRUE)
acf(egarch_residuals, main = "ACF of Standardized Residuals - EGARCH")
egarch_lb_test <- Box.test(egarch_residuals, lag = 20, type = "Ljung-Box")
print("Ljung-Box Test for EGARCH(1,1):")
print(egarch_lb_test)

# Residual diagnostics for GJR-GARCH(1,1)
gjr_garch_residuals <- residuals(fit_gjr_garch, standardize = TRUE)
acf(gjr_garch_residuals, main = "ACF of Standardized Residuals - GJR-GARCH")
gjr_garch_lb_test <- Box.test(gjr_garch_residuals, lag = 20, type = "Ljung-Box")
print("Ljung-Box Test for GJR-GARCH(1,1):")
print(gjr_garch_lb_test)

# Likelihood Ratio Test between GARCH(1,1) and GARCH(1,2)
spec_garch_12 <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 2)),
  mean.model = list(armaOrder = c(0, 0), include.mean = FALSE),
  distribution.model = "std"
)
fit_garch_12 <- ugarchfit(spec = spec_garch_12, data = wgmi_returns)

# Calculate likelihoods
ll_garch_11 <- fit_garch@fit$LLH

```

```
ll_garch_12 <- fit_garch_12@fit$LLH
```

Likelihood Ratio Test

```
LR_statistic <- -2 * (ll_garch_11 - ll_garch_12)
```

```
p_value <- pchisq(LR_statistic, df = 1, lower.tail = FALSE)
```

```
print(paste("Likelihood Ratio Statistic (GARCH(1,1) vs GARCH(1,2)):", LR_statistic))
```

```
print(paste("p-value:", p_value))
```

Train-test split for forecasting

```
train_end <- index(wgmi)[floor(0.8 * length(wgmi))]
```

```
wgmi_train <- window(wgmi_returns, end = train_end)
```

```
wgmi_test <- window(wgmi_returns, start = train_end + 1)
```

Fit the models on the training set

```
fit_garch_train <- ugarchfit(spec = spec_garch, data = wgmi_train)
```

```
fit_egarch_train <- ugarchfit(spec = spec_egarch, data = wgmi_train)
```

```
fit_gjr_garch_train <- ugarchfit(spec = spec_gjr_garch, data = wgmi_train)
```

Forecast on the test set

```
forecast_garch <- ugarchforecast(fit_garch_train, n.ahead = length(wgmi_test))
```

```
forecast_egarch <- ugarchforecast(fit_egarch_train, n.ahead = length(wgmi_test))
```

```
forecast_gjr_garch <- ugarchforecast(fit_gjr_garch_train, n.ahead = length(wgmi_test))
```

Calculate RMSE for the forecasts

```
rmse <- function(actual, forecast) {
```

```
  sqrt(mean((actual - forecast)^2))
```

```
}
```

```

rmse_garch <- rmse(wgmi_test, sigma(forecast_garch))
rmse_egarch <- rmse(wgmi_test, sigma(forecast_egarch))
rmse_gjr_garch <- rmse(wgmi_test, sigma(forecast_gjr_garch))

print(paste("RMSE GARCH:", rmse_garch))
print(paste("RMSE EGARCH:", rmse_egarch))
print(paste("RMSE GJR-GARCH:", rmse_gjr_garch))

# Extract AIC and BIC for each model

aic_garch <- infocriteria(fit_garch)[1]
bic_garch <- infocriteria(fit_garch)[2]
aic_egarch <- infocriteria(fit_egarch)[1]
bic_egarch <- infocriteria(fit_egarch)[2]
aic_gjr_garch <- infocriteria(fit_gjr_garch)[1]
bic_gjr_garch <- infocriteria(fit_gjr_garch)[2]

# Print the AIC and BIC values

print(paste("GARCH(1,1) - AIC:", aic_garch, "BIC:", bic_garch))
print(paste("EGARCH(1,1) - AIC:", aic_egarch, "BIC:", bic_egarch))
print(paste("GJR-GARCH(1,1) - AIC:", aic_gjr_garch, "BIC:", bic_gjr_garch))

# Determine the model with the lowest AIC and BIC

if(aic_garch < aic_egarch & aic_garch < aic_gjr_garch) {
  print("GARCH(1,1) has the lowest AIC")
} else if(aic_egarch < aic_garch & aic_egarch < aic_gjr_garch) {
  print("EGARCH(1,1) has the lowest AIC")
} else {

```

```
print("GJR-GARCH(1,1) has the lowest AIC")
}

if(bic_garch < bic_egarch & bic_garch < bic_gjr_garch) {
  print("GARCH(1,1) has the lowest BIC")
} else if(bic_egarch < bic_garch & bic_egarch < bic_gjr_garch) {
  print("EGARCH(1,1) has the lowest BIC")
} else {
  print("GJR-GARCH(1,1) has the lowest BIC")
}
```