

**Modelos score-driven  
não-Gaussianos para séries  
temporais com combinação não  
linear das componentes de  
tendência e sazonalidade**

**Matheus Carneiro Nogueira**

**Relatório de Projeto Final de Graduação**

**Centro Técnico Cienctífico - CTC**  
**Departamento Engenharia Elétrica e**  
**Departamento de Informática**  
Curso de Graduação em Engenharia de  
Computação

Rio de Janeiro  
Dezembro de 2023



**Matheus Carneiro Nogueira**

**Modelos score-driven não-Gaussianos para  
séries temporais com combinação não linear  
das componentes de tendência e sazonalidade**

**Relatório de Projeto Final de Graduação**

Relatório de Projeto Final de Graduação, apresentado ao Curso de Graduação em Engenharia de Computação, do Departamento de Informática da PUC-Rio.

Orientador: Prof. Cristiano Fernandes

Rio de Janeiro  
Dezembro de 2023

*Fazer previsões é muito difícil, especialmente  
se forem sobre o futuro.*

**Niels Bohr, .**

## **Agradecimentos**

É com grande carinho que gostaria de agradecer a todos aqueles envolvidos neste projeto final de graduação. Primeiramente, agradeço à PUC-Rio por toda a caminhada durante a graduação em Engenharia da Computação, em especial aos departamentos de Informática e de Engenharia Elétrica, responsáveis por promover um curso de excelência, que me fez crescer tanto pessoal quanto academicamente e que cuja caminhada da graduação se encerra com a apresentação deste projeto final.

Agradeço, em especial, ao meu orientador e professor, Cristiano Fernandes, por todo o conhecimento e extensa experiência repassados, primeiramente, durante suas aulas da graduação e, em além disso, nos inúmeros encontros e reuniões durante o desenvolvimento deste projeto. Sou eternamente grato pela disponibilidade e paciência que foram sempre presentes ao longo do último ano.

Também gostaria de destacar a importância de meus companheiros do LAMPS, agradecendo, principalmente, aos meu colega Matheus Alves, sem o qual este projeto não seria possível, tanto pelo pacote que faz parte de sua dissertação de mestrado, quanto pelos vários pedidos de socorro gentilmente respondidos. Além dele, agradeço à recorrente ajuda dos meus colegas Marina Dietze e Arthur Brigatto, que diversas vezes auxiliaram com ideias, sugestões, revisões de contas e apoio emocional.

Por fim, mas jamais menos importante, agradeço à minha família por todo o apoio incondicional prestado no decorrer de todos esses 6 anos de graduação. Ao meu pai, Reinaldo, pelas inúmeras trocas de experiências, à minha mãe, Valéria, pela recorrente preocupação de estar fazendo coisas demais ao mesmo tempo, ao meu irmão Pedro por sempre me ouvir contando casos do dia a dia, à minha namorada Fernanda, que ouviu mais que ninguém "não posso, estou fazendo o TCC" e aos meus avós, que são sempre a maior torcida e os que mais se orgulham de ver seus netos avançando na vida.

Embora individual, este trabalho é resultado de um esforço coletivo, no qual cada pessoa aqui citada, além de outras não mencionadas, possuiu um papel importante. A todos os envolvidos neste projeto, e em todas as etapas percorridas durante minha graduação, deixo meu mais sincero e profundo obrigado.

## **Resumo**

Nogueira, Matheus; Fernandes, Cristiano. **Modelos score-driven não-Gaussianos para séries temporais com combinação não linear das componentes de tendência e sazonalidade.** Rio de Janeiro, 2023. 85p. Projeto Final de Graduação – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Uma técnica comum em modelagem de séries temporais é a decomposição da série de interesse em componentes de tendência e sazonalidade. Dentro da classe de modelos Score-Driven, essa decomposição é usualmente realizada de forma aditiva, de tal modo que a série temporal de interesse é expressa como a componente de tendência somada à componente de sazonalidade. Entretanto, não é raro que, mesmo contabilizando a componente sazonal, os resíduos do modelo implementado ainda indiquem dependência sazonal não capturada pelo modelo. Dito isso, o principal objetivo desse projeto é estudar se diferentes combinações não lineares dessas componentes são capazes de gerar modelos *score-driven* adequados e bem especificados para séries temporais.

## **Palavras-chave**

Séries Temporais; Modelos Score Driven; Decomposição em Tendência e Sazonalidade.

## **Abstract**

Nogueira, Matheus; Fernandes, Cristiano (Advisor). **Non-Gaussian score driven models for time series with non-linear combination of trend and seasonality components.** Rio de Janeiro, 2023. 85p. Projeto Final de Graduação – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A common technique in time series modeling is to decompose the time series into its trend and seasonal components. Inside the Score-Driven Models class, this decomposition is usually made in an additive form, so that the series is expressed as the sum of its trend and seasonal components. However, it is not unusual that, even with the seasonal component being considered into the model, its residuals still show signs of seasonal dependency that was not captured by the model. With that said, the main objective of this project is to study if different non-linear combinations of those components are able to generate well specified score-driven models for time series.

## **Keywords**

Time Series; Score Driven Models; Trend and Seasonality Decomposition.

# **Sumário**

<b>1</b>	<b>Introdução</b>	<b>13</b>
<b>2</b>	<b>Situação Atual</b>	<b>15</b>
2.1	Implementação Computacional	16
<b>3</b>	<b>Objetivos</b>	<b>30</b>
<b>4</b>	<b>Metodologia</b>	<b>33</b>
4.1	Fundamentação Teórica	33
4.2	Implementação Computacional	41
<b>5</b>	<b>Resultados</b>	<b>52</b>
5.1	Série de carga	52
5.2	Série de ENA	62
<b>6</b>	<b>Conclusão</b>	<b>72</b>
<b>7</b>	<b>Referências bibliográficas</b>	<b>74</b>
<b>A</b>	<b>Desenvolvimento GAS</b>	<b>77</b>
A.1	GAS gama	77
<b>B</b>	<b>Sazonalidade em modelos de espaço de estados</b>	<b>79</b>

## **Lista de figuras**

Figura 2.1	Série temporal mensal de precipitação e seu histograma.	16
Figura 2.2	Tela inicial do TSL Score Edition.	18
Figura 2.3	Telas de especificação do modelo e análise de resíduos no TSL Score Edition.	18
Figura 2.4	Telas do TSL Score Edition com o <i>fit in sample</i> do modelo GAS estimado.	19
Figura 2.5	Tela do TSL-Score Edition de previsão do modelo GAS.	19
Figura 2.6	Especificação de modelo GAS estrutural no Time Series Lab.	20
Figura 2.7	<i>Fit in sample</i> do GAS estrutural estimado e gráfico das componentes estimadas.	20
Figura 2.8	Gráfico e FAC dos resíduos do modelo GAS estimado no TSL.	21
Figura 2.9	Previsão 12 passos à frente do modelo GAS estimado no TSL.	21
Figura 2.10	<i>Fit in sample</i> do modelo GAS para a série de precipitação utilizando ScoreDrivenModels.jl.	23
Figura 2.11	Resíduos e FAC dos resíduos do modelo GAS para a série de precipitação utilizando ScoreDrivenModels.jl.	23
Figura 2.12	Previsão pontual e simulações 12 passos à frente do modelo GAS utilizando ScoreDrivenModels.jl.	23
Figura 2.13	Fit in Sample do modelo GAS(1,1) gama estimado no pacote GAS R.	25
Figura 2.14	Resíduos e FAC dos resíduos do modelo GAS(1,1) gama estimado no pacote GAS R.	25
Figura 2.15	Previsão do modelos GAS(1,1) estimado no pacote GAS R.	25
Figura 2.16	Fit in Sample do modelo GAS-CNO normal estimado no pacote UnobservedComponentsGAS.jl.	28
Figura 2.17	Resíduos e FAC dos resíduos do modelo GAS-CNO normal estimado no pacote UnobservedComponentsGAS.jl.	28
Figura 2.18	Previsão do modelo GAS-CNO normal estimado no pacote UnobservedComponentsGAS.jl.	28
Figura 3.1	Família de modelos ETS.	31
Figura 3.2	<i>Fit in sample</i> de um modelo GAS-CNO log-normal com combinação aditiva das componentes de tendência e sazonalidade.	32
Figura 4.1	Série mensal de carga.	42
Figura 4.2	Série mensal de ENA.	43
Figura 5.1	Período de treino série de carga GAS-CNO aditivo.	53
Figura 5.2	Período de treino série de carga GAS-CNO multiplicativo 1.	53
Figura 5.3	FAC dos resíduos quantílicos série de carga GAS-CNO aditivo.	54
Figura 5.4	FAC dos resíduos quantílicos série de carga GAS-CNO multiplicativo 1.	54
Figura 5.5	Previsão 12 passos à frente para série de carga GAS-CNO aditivo.	56

Figura 5.6	Previsão 12 passos à frente para série de carga GAS-CNO multiplicativo 1.	56
Figura 5.7	Período de treino série de carga GAS-CNO aditivo.	58
Figura 5.8	Período de treino série de carga GAS-CNO multiplicativo 2.	58
Figura 5.9	FAC dos resíduos quantílicos série de carga GAS-CNO aditivo.	59
Figura 5.10	FAC dos resíduos quantílicos série de carga GAS-CNO multiplicativo 2.	59
Figura 5.11	Previsão 12 passos à frente para série de carga GAS-CNO aditivo.	61
Figura 5.12	Previsão 12 passos à frente para série de carga GAS-CNO multiplicativo 2.	61
Figura 5.13	Período de treino série de ENA GAS-CNO aditivo.	63
Figura 5.14	Período de treino série de ENA GAS-CNO multiplicativo 1.	63
Figura 5.15	FAC dos resíduos quantílicos série de ENA GAS-CNO aditivo.	64
Figura 5.16	FAC dos resíduos quantílicos série de ENA GAS-CNO multiplicativo 1.	64
Figura 5.17	Previsão 12 passos à frente para série de ENA GAS-CNO aditivo.	66
Figura 5.18	Previsão 12 passos à frente para série de ENA GAS-CNO multiplicativo 1.	66
Figura 5.19	Período de treino série de ENA GAS-CNO aditivo.	67
Figura 5.20	Período de treino série de ENA GAS-CNO multiplicativo 2.	67
Figura 5.21	FAC dos resíduos quantílicos série de ENA GAS-CNO aditivo.	68
Figura 5.22	FAC dos resíduos quantílicos série de ENA GAS-CNO multiplicativo 2.	68
Figura 5.23	Previsão 12 passos à frente para série de ENA GAS-CNO aditivo.	70
Figura 5.24	Previsão 12 passos à frente para série de ENA GAS-CNO multiplicativo 2.	70

## **Lista de tabelas**

Tabela 2.1 Tabela das distribuições e dinâmicas implementadas nos softwares apresentados.	29
Tabela 4.1 Divisão das séries temporais em períodos de treino e de teste.	43
Tabela 5.1 Resultados teste Jarque-Bera para série de carga GAS-CNO multiplicativo 1.	55
Tabela 5.2 Resultados teste ARCH para série de carga GAS-CNO multiplicativo 1.	55
Tabela 5.3 Resultados teste Ljung-Box para série de carga GAS-CNO multiplicativo 1.	55
Tabela 5.4 Resultados teste Jarque-Bera para série de carga GAS-CNO multiplicativo 2.	60
Tabela 5.5 Resultados teste ARCH para série de carga GAS-CNO multiplicativo 2.	60
Tabela 5.6 Resultados teste Ljung-Box para série de carga GAS-CNO multiplicativo 2.	60
Tabela 5.7 MAPEs dentro e fora da amostra para série de carga.	62
Tabela 5.8 Resultados teste de normalidade Jarque-Bera para série de ENA GAS-CNO multiplicativo 1.	65
Tabela 5.9 Resultados teste de heterocedasticidade ARCH para série de ENA GAS-CNO multiplicativo 1.	65
Tabela 5.10 Resultados teste Ljung-Box para série de ENA GAS-CNO multiplicativo 1.	65
Tabela 5.11 Resultados teste de normalidade Jarque-Bera para série de ENA GAS-CNO multiplicativo 2.	69
Tabela 5.12 Resultados teste de heterocedasticidade ARCH para série de ENA GAS-CNO multiplicativo 2.	69
Tabela 5.13 Resultados teste Ljung-Box para série de ENA GAS-CNO multiplicativo 2.	69
Tabela 5.14 MAPEs dentro e fora da amostra para série de ENA.	71

## **Lista de algoritmos**

Algoritmo 1      Previsão k passos à frente para modelos GAS(p,q).      37

## **Lista de Códigos**

Código 1	Script ScoreDrivenModels.jl	22
Código 2	Script GAS package R	24
Código 3	Script UnobservedComponentsGAS.jl	27

## **Lista de Abreviaturas e Notação**

GAS – *Generalized Autoregressive Score* ARMA – *Auto Regressive Moving*

*Average*

CNO – Componentes não observáveis

FAC – Função de Autocorrelação

QQPlot – Quantile-Quantile Plot

MV - Máxima Verossimilhança

TSL – *Time Series Lab*

OD – *Observation driven models*

PD – *Parameter driven models*

$\mathbf{y}$  ou  $\vec{y}$  – vetores

$y$  – Letras minúsculas representam números escalares

$s$  – *Score padronizado*

$S$  – Período sazonal de uma série

# 1

## Introdução

Modelos *score driven* são uma classe geral e flexível de modelos de série temporal desenvolvida simultaneamente por [Harvey 2013] e [Creal, Koopman e Lucas 2013]. Essa classe é reconhecida por diferentes nomenclaturas: *generalized autoregressive score (GAS)*, *dynamic conditional score models*, ou, simplesmente, modelos *score-driven*. Eles introduzem diversas vantagens de modelagem de séries temporais, descritas a seguir.

Primeiramente, essa classe de modelo traz a flexibilidade de modelar séries temporais não-Gaussianas, isto é, séries cuja distribuição de probabilidade condicional não é a distribuição normal. Essa vantagem torna o modelo GAS uma poderosa ferramenta de modelagem, uma vez que há diversas situações nas quais a distribuição normal é, por definição, inadequada. Como exemplo, tome as séries de velocidade de vento, que assumem apenas valores positivos, fazendo com que a distribuição de probabilidade associada ao problema deva ser definida apenas para os reais positivos. As séries de retornos financeiros, por sua vez, possuem como fato estilizado excesso de curtose em relação à distribuição normal, tornando-a também inadequada. Além disso, séries temporais associadas à contagem, como por exemplo, quantidade de itens vendidos em uma determinada loja, necessitam de distribuições para variáveis aleatórias discretas. Sendo assim, a inexistência da restrição de normalidade da série a ser modelada torna os modelos GAS extremamente úteis em diversas aplicações.

Além dessa flexibilidade, os modelos GAS também permitem estimação de parâmetros variantes no tempo com diferentes processos de atualização. Isso os torna capazes de generalizar diferentes modelos clássicos de séries temporais, como, por exemplo, os modelos GARCH [Bollerslev 1986] e modelos de duração condicional [Engle e Russell 1998].

Dada a relevância dos modelos *score-driven* e a vasta aplicabilidade desse arcabouço de modelagem, diversas ferramentas computacionais vêm sendo desenvolvidas a fim de implementar essa classe de modelos. Podemos citar, por exemplo, o pacote *GAS* desenvolvido em R [Ardia, Boudt e Catania 2016], o pacote *PyFlux*, implementado em *Python* [Taylor 2016] e o pacote *ScoreDrivenModels* implementado em Julia [Bodin et al. 2020]. Somado a isso, existe o software livre *Time Series Lab* no qual também está implementada a classe de modelos GAS em sua versão *Time Series Lab - Score Edition* [Lit, Koopman e Harvey 2021]. No entanto, como será apresentado no decorrer deste relatório, todos esses *softwares* possuem a limitação de combinar

componentes não observáveis de forma aditiva, o que gera a relevância deste projeto.

Como resposta ao dinamismo com o qual os modelos GAS são estudados, foi criado um repositório dedicado exclusivamente a artigos e implementação de código sobre essa classe de modelos, que pode ser acessado em <http://www.gasmodel.com/> [Andre e Koopman].

Dito isso, para fins de organização, este relatório está estruturado da seguinte forma. No Capítulo 2 está apresentada a situação atual de modelos *score-driven*, tanto em termos de desenvolvimento teórico quanto aplicações práticas e desenvolvimento de ferramentas computacionais. No Capítulo 3 são apresentados os principais objetivos a serem alcançados com esse trabalho. O Capítulo 4 discorre sobre o desenvolvimento teórico realizado no projeto e apresenta toda a metodologia utilizada nos testes implementados. Por fim, o Capítulo 5 exibe os resultados obtidos a partir da implementação dos modelos propostos no capítulo 4 e o capítulo 6 as conclusões deste projeto.

## 2

### Situação Atual

Dada a flexibilidade trazida pela modelagem *score-driven* para séries temporais não Gaussianas, esses modelos estão sendo utilizados nas mais diversas áreas de aplicação, como finanças, varejo, energia e clima.

No contexto de finanças, eles têm se mostrado especialmente úteis para modelar a volatilidade de ativos financeiros. Por exemplo, um estudo recente de [Xu e Lien 2022] explorou o uso de modelos GAS na previsão da volatilidade de ativos de óleo e gás, comparando a qualidade de previsão com os modelos GARCH e EGARCH e constatando que, para os ativos de óleo, a modelagem GAS se mostrou superior em todos os cenários enquanto que, para ativos de gás natural, mostrou-se superior para previsões de 1 e 5 passos à frente, considerando séries diárias. Além disso, [Fuentes, Herrera e Clements 2023] utilizaram modelos *score-driven* para previsão de eventos extremos em séries de retornos financeiros, o que, dada a flexibilidade de adicionar medidas de volatilidade realizada, melhorou a acurácia da previsão quando comparada com outros modelos de *benchmark*.

Embora uma grande parte das publicações relacionadas a modelos *score driven* sejam dentro do contexto financeiro, há outras áreas que também estão utilizando essa classe de modelos de séries temporais. No varejo, [Hoeltgebaum et al. 2021] utilizaram modelos GAS com distribuição log-normal para realizar previsões de demanda diária em centros de distribuição, constatando que essa metodologia é competitiva quando comparada com *benchmarks* usuais, além de trazer a vantagem de produzir uma densidade preditiva de forma fechada.

No contexto de clima e energia, modelos *score driven* também são largamente utilizados. Por exemplo, [Blazsek e Escribano 2023] utilizaram modelos GAS para modelar e medir efeitos climáticos não lineares de longo prazo resultados de alterações em variáveis climáticas como volume de gelo global, nível de dióxido de carbono (CO<sub>2</sub>) na atmosfera e temperatura da superfície da Antártida. Além disso, [Kushwah e Wadhvani 2019] utilizaram modelos GAS, GAS com explicativas e GAS Não Lineares para previsão de séries de poder de vento, para monitorar a performance de turbinas eólicas, além de combinar a metodologia dos modelos GAS não lineares com explicativas com funções de ativação típicas de modelos de redes neurais (NLGASX).

Por fim, com o intuito de apresentar a versatilidade de modelos *score-driven*, [Koopman e Lit 2019] utilizou modelos baseados em *score* para prever

o resultado de partidas de futebol de campeonatos nacionais. Para isso, foram utilizadas as distribuições de poisson, skellam e *probit* ordenada. A primeira foi utilizada para modelar o resultado de uma partida como uma observação pareada, a segunda para modelar a diferença do número de gols dos times de uma partida e a terceira para modelar diretamente a probabilidade de um time vencer, empatar ou perder a partida. Essa metodologia foi testada simulando apostas esportivas a partir da previsão dos modelos e concluiu-se que modelos *score-driven* performaram melhor que diversos modelos *benchmarks*, tanto em termos de precisão de previsão quanto em termos de minimizar a perda nas apostas.

## 2.1 Implementação Computacional

Assim como foi comentado no Capítulo 1, já existem algumas ferramentas computacionais que implementam os modelos *score-driven* para séries temporais, sejam elas pacotes das principais linguagens utilizadas para ciência de dados e estatística, ou *softwares* específicos para modelagens de séries temporais. Nessa seção, estão apresentadas com maior profundidade essas ferramentas com o intuito de entender suas implementações, capacidades e limitações.

Somado a isso, nas seções seguintes será utilizada uma série temporal mensal de precipitação retirada do site do Centro de Previsão de Tempo e Estudos Climáticos <https://clima.cptec.inpe.br/> para exemplificar o funcionamento das ferramentas explicadas. A série temporal possui 269 observações, entre Junho de 2000 e Outubro de 2022. A Figura 2.1 exibe a série e o seu histograma. Note que modelar essa série a partir de uma distribuição normal não seria adequado devido a assimetria positiva do histograma.

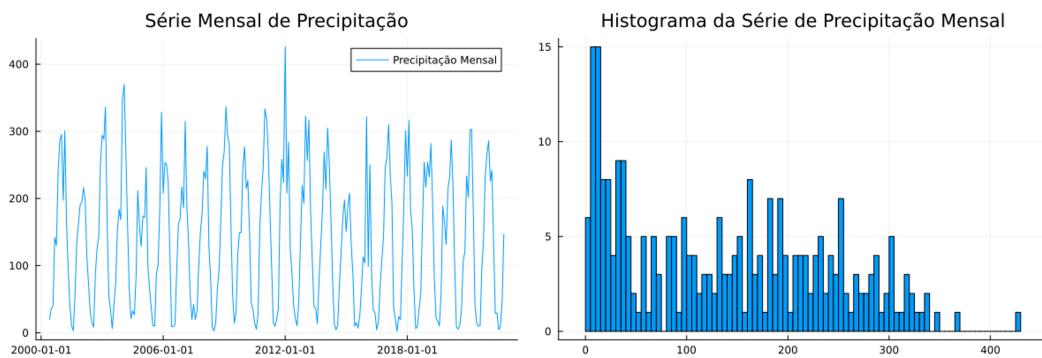


Figura 2.1: Série temporal mensal de precipitação e seu histograma.

### 2.1.1

#### TimeSeriesLab Score Edition

O TimeSeriesLab (TSL) é uma plataforma para análise, modelagem e previsão de séries temporais de forma interativa e repleta de suporte visual [Lit]. Essa ferramenta possui uma grande seleção de modelos de séries temporais, como modelos ARIMA de Box-Jenkins, modelos de suavização exponencial, modelos *score-driven* e modelos estruturais básicos. Sendo assim, tudo que o usuário precisa fazer para realizar a sua modelagem é escolher o tipo de modelo que melhor se adequa aos seus dados.

Além disso, o TSL também permite que o usuário crie seu próprio modelo de séries temporais com base em componentes dinâmicas como tendência, sazonalidade e ciclos.

Como o foco desse projeto é a modelagem *score-driven*, essa seção se limita a comentar sobre esse tipo de modelo e não traz um extenso resumo do funcionamento da ferramenta. Para fins de aprofundamento no TSL, recomenda-se a leitura do manual disponível no site da ferramenta <https://timeserieslab.com/> [Lit].

O TSL Score Edition fornece as seguintes distribuições de probabilidade para a implementação de seus modelos:

- Discretas: Poisson, Negativa Binomial, Bernoulli e Skellam
- Contínuas: Normal, t de Student, GED, Beta Generalizada Tipo 2, Exponencial e Weibull

O modelo *score driven* implementado no TSL *Score Edition* é o GAS-CNO que, assim como será visto no Capítulo 4, utiliza a formulação de um modelo de componentes não observáveis para a função de atualização dos seus parâmetros variantes no tempo. Esse *software* possui uma segunda versão denominada *Time Series Lab* que mistura o GAS com modelos estruturais. Essa versão está apresentada na seção 2.1.2.

Uma das principais qualidades do TSL é a sua usabilidade. Com interface simples e intuitiva, um usuário pode, rapidamente, estimar, avaliar e realizar previsões para diversos modelos de séries temporais sem a necessidade de conhecimento prévio sobre a teoria por detrás dos modelos ou o conhecimento de alguma linguagem de programação.

Por outro lado, uma vez sendo um *software* pronto, o TSL não possui a mesma flexibilidade de pacotes implementados para linguagens de programação como os outros que serão aqui apresentados.

A Figura 2.2 exibe a tela inicial do TSL, na qual o usuário pode visualizar a série temporal a ser modelada e realizar pré-processamentos como

transformações na escala da série. Em seguida, a Figura 2.3 exibe as telas de especificação do modelo, onde o usuário define a distribuição desejada e as características dos parâmetros do modelo, e de análise dos resíduos, onde é possível ver a série de resíduos do modelo estimado, bem como duas FAC e outras visualizações importantes. Junto da análise dos resíduos, é importante visualizar o *fit in sample* do modelo, que está exibido na Figura 2.4. Por fim, a Figura 2.5 exibe a tela de previsão do modelo estimado.

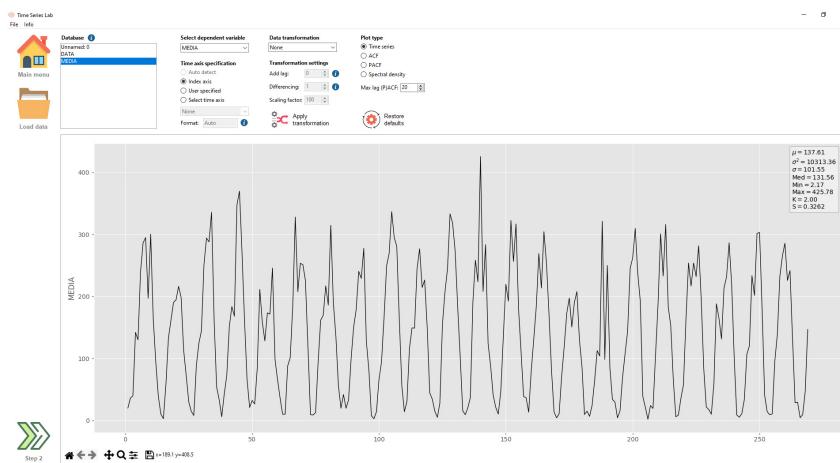


Figura 2.2: Tela inicial do TSL Score Edition.

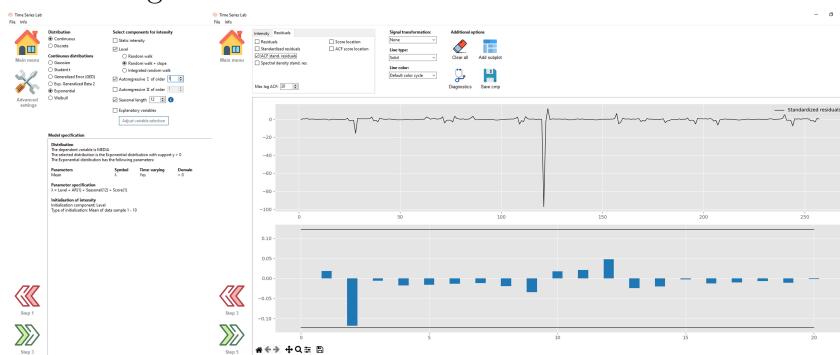


Figura 2.3: Telas de especificação do modelo e análise de resíduos no TSL Score Edition.

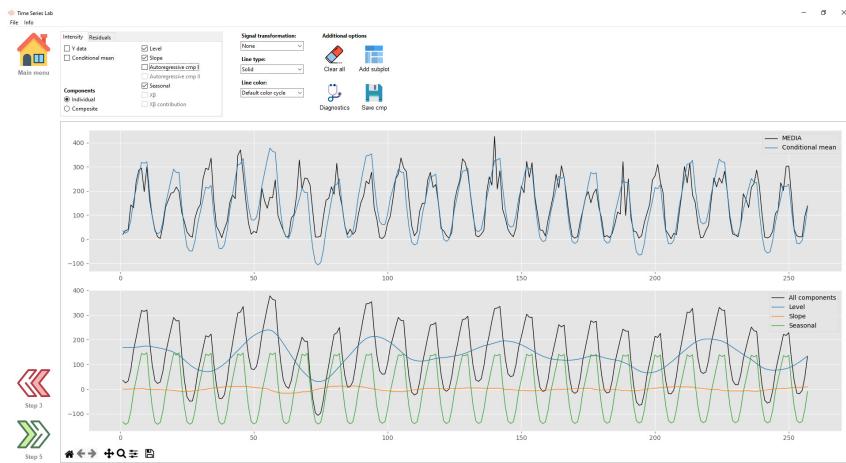


Figura 2.4: Telas do TSL Score Edition com o *fit in sample* do modelo GAS estimado.

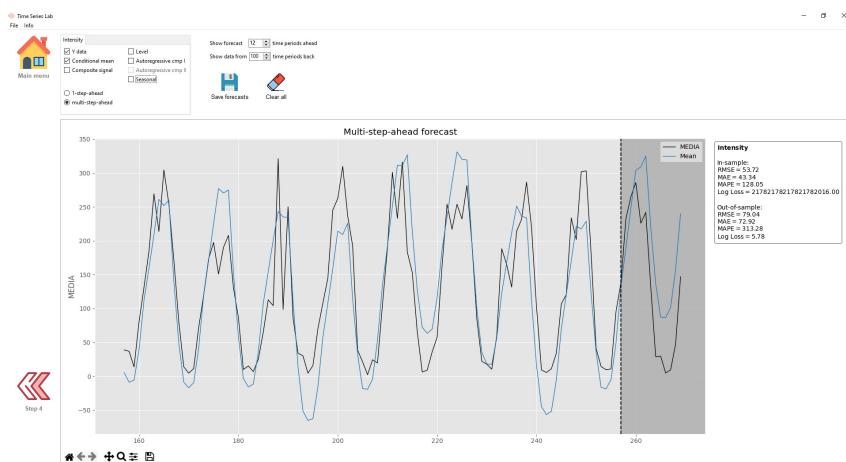


Figura 2.5: Tela do TSL-Score Edition de previsão do modelo GAS.

## 2.1.2

### TimeSeriesLab

Além do *software* apresentado na seção anterior, existe um *software*, mais geral, que, dentre outros tipos, implementa modelos *score-driven*. Ele é o *Time Series Lab*. Existiu uma versão denominada *Time Series Lab State Space Edition* que foi descontinuada e substituída por essa versão.

Nela, o usuário é capaz tanto de utilizar modelos pré construídos quanto criar seu próprio modelo, utilizando como base as componentes clássicas de um modelo estrutural: tendência, nível, sazonalidade e ciclos. Dentre os modelos pré construídos, estão:

- Métodos básicos: amortecimento exponencial simples (*EWMA*), *Holt Winters* e *Holt Winters Sazonal*;
- Modelos Estruturais: nível local (com tendência e com tendência e sazonalidade) e modelo estrutural básico;

- Modelos ARIMA;
- Modelos *score-driven*;
- Modelos de Machine Learning: *XGBoost Regressor*.

Importante ressaltar que essa versão engloba as funcionalidades do *Score Edition*. Ela possui modelos GAS(p,q) dentre as opções de modelos pré construídos e, além disso, permite ao usuário criar seu próprio modelo *score-driven* a partir de componentes não observáveis (ver seção 4.1.1). Essa é, justamente, a classe de modelos a ser focada neste projeto. A Figura 2.6 exibe a tela na qual o usuário especifica o modelo GAS estrutural desejado. Em seguida, a Figura 2.7 exibe o modelo estimado, com seu *fit in sample* e o gráfico das componentes não observáveis estimadas.

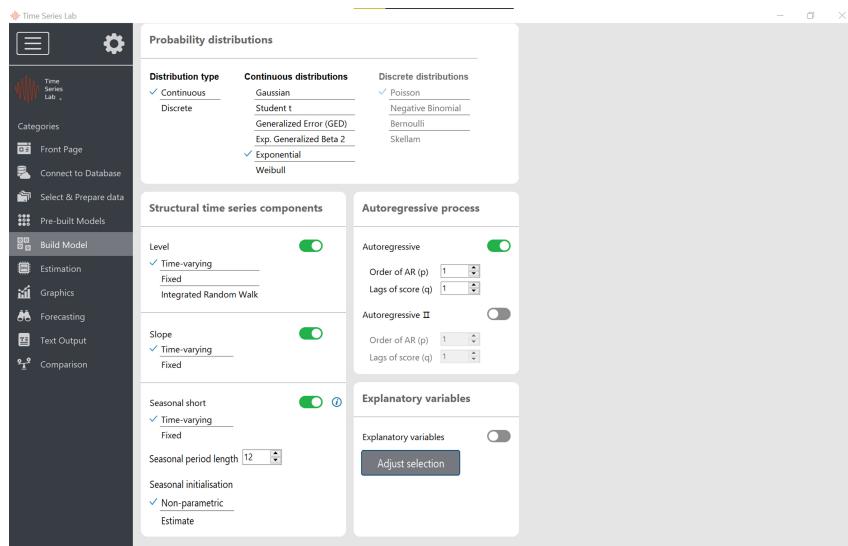


Figura 2.6: Especificação de modelo GAS estrutural no Time Series Lab.

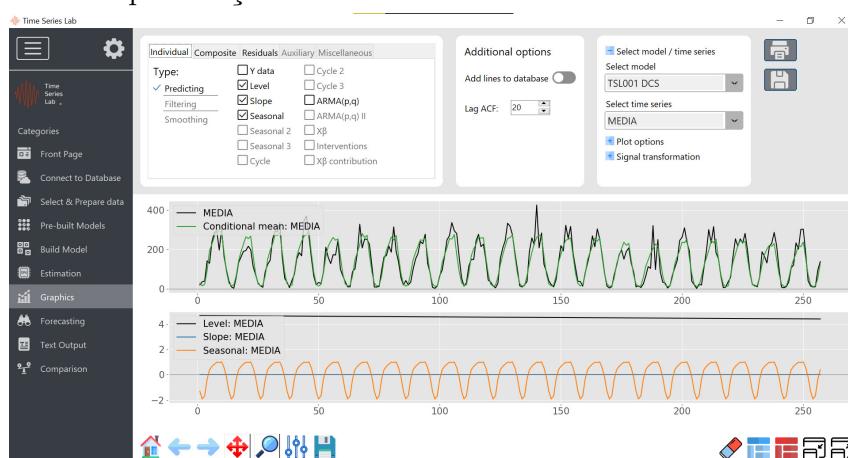


Figura 2.7: *Fit in sample* do GAS estrutural estimado e gráfico das componentes estimadas.

Como é usual, o usuário pode estudar os resíduos do modelo estimado, tanto com seu gráfico quanto com sua FAC, para avaliar a qualidade da

estimação e, em seguida, seguir com a previsão *out of sample* do modelo. As Figuras 2.8 e 2.9 ilustram as janelas nas quais o usuário realiza essas operações.



Figura 2.8: Gráfico e FAC dos resíduos do modelo GAS estimado no TSL.

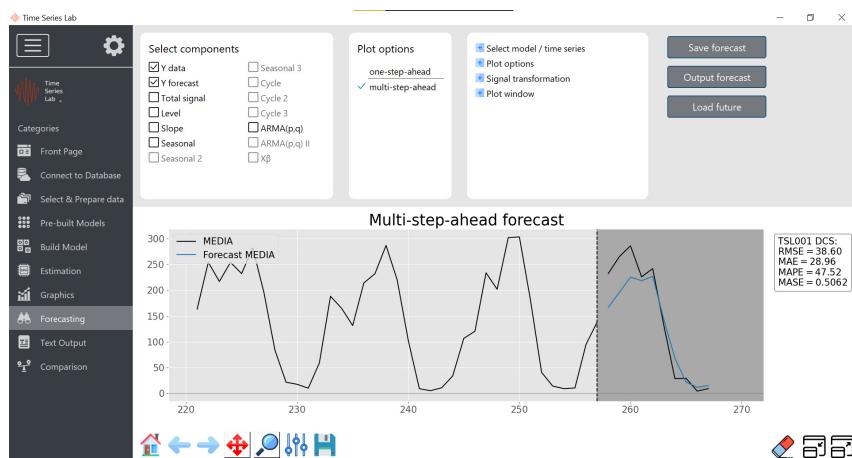


Figura 2.9: Previsão 12 passos à frente do modelo GAS estimado no TSL.

Certamente, dentre os *softwares* aqui apresentados, o *TimeSeriesLab* é aquele que melhor combina flexibilidade e usabilidade, fornecendo uma grande gama de modelos de séries temporais com uma interface simples e intuitiva, tornando-o de mais fácil acesso que pacotes implementados em linguagens de programação. Somado a isso, uma grande qualidade do *software*, especialmente para esse projeto, é a implementação de modelos *score-driven* tanto GAS(p,q) quanto estruturais. Embora muito completo, há uma limitação no TSL: as componentes dos modelos estruturais são combinadas apenas aditivamente. O principal objetivo deste projeto é, justamente, implementar modelos *score-driven* estruturais que supram essa limitação. Sobre isso, ver seção. 4.1.2.

### 2.1.3

#### ScoreDrivenModels.jl - Julia

ScoreDrivenModels.jl é um pacote desenvolvido na linguagem Julia para modelar, prever e simular séries temporais a partir de modelos *score-driven* [Bodin et al. 2020]. A API do pacote permite o usuário escolher entre diversas distribuições, estruturas de lag, métodos de otimização e valores de escala. Além disso, o pacote também traz a possibilidade do usuário adicionar novas distribuições de probabilidade caso sejam necessárias.

Para especificar um modelo, o usuário necessita especificar 4 informações: a distribuição desejada, a escala, a estrutura de *lag* (os valores de p e q) e especificar os parâmetros que serão variantes no tempo. Essas informações são passadas como parâmetros da função *ScoreDrivenModel()*. Em seguida, a função *fit!()* deve ser usada para estimar o modelo sendo por meio dela que o usuário indica a série temporal a ser modelada e qual o otimizador a ser utilizado. Por fim, uma vez identificado que o modelo estimado é satisfatório, o usuário pode realizar a previsão de sua série temporal por meio da função *forecast()* que recebe como parâmetros a série temporal usada para a estimação, o modelo estimado e a quantidade de passos à frente. O código 2.1.3 abaixo apresenta um script simples para a realização das etapas descritas.

---

#### Código 1: Script ScoreDrivenModels.jl

---

```

1 using ScoreDrivenModels
2
3 "Código para definição e estimação do modelo GAS(1,1) LogNormal"
4 gas = ScoreDrivenModel([1, 12], [1, 12], Gamma, 0.0) #definição do
    modelo
5 ScoreDrivenModels.fit!(gas, y_fit) #estimação do modelo
6 fit_in_sample = fitted_mean(gas, y_fit) #obtenção do fit in sample
7 residuals = y_fit .- fit_in_sample #obtenção dos resíduos
8 forecast = ScoreDrivenModels.forecast(y_fit, gas, 12) #obtenção das
    previsões

```

---

As Figuras 2.10 e 2.11 exibem o *fit in sample* e os resíduos do modelo estimado com sua FAC, enquanto que a Figura 2.12 exibe a previsão pontual 12 passos à frente e a simulação. Todos os resultados foram obtidos a partir da execução do script 2.1.3.

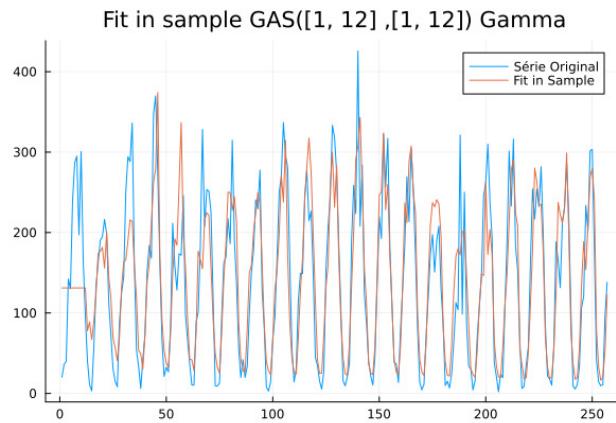


Figura 2.10: *Fit in sample* do modelo GAS para a série de precipitação utilizando ScoreDrivenModels.jl.

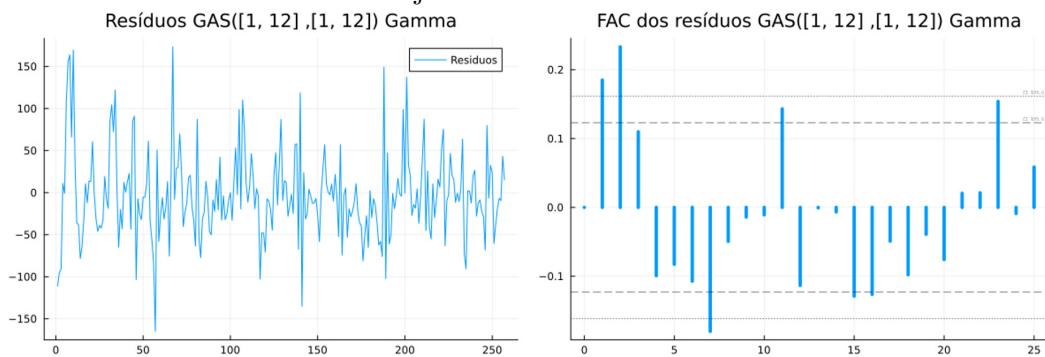


Figura 2.11: Resíduos e FAC dos resíduos do modelo GAS para a série de precipitação utilizando ScoreDrivenModels.jl.

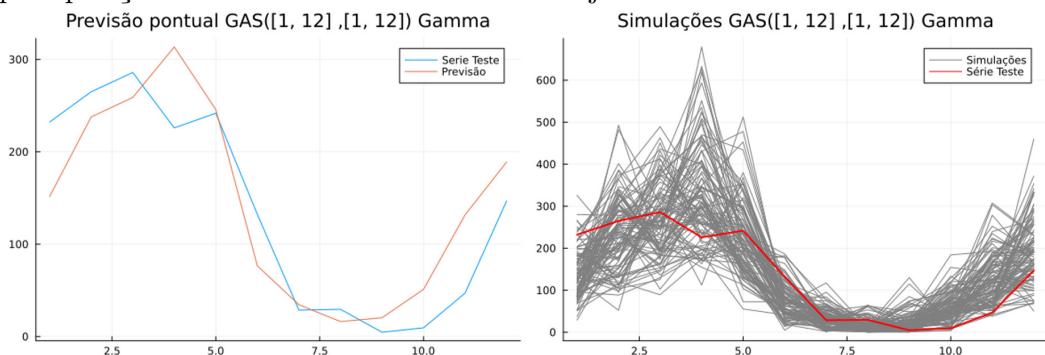


Figura 2.12: Previsão pontual e simulações 12 passos à frente do modelo GAS utilizando ScoreDrivenModels.jl.

Algumas das principais qualidades desse pacote é sua integração com a linguagem Julia, que vem sendo cada vez mais utilizada para modelagem de problemas que exigem ferramentas de estatística, otimização, ciência de dados e áreas correlatas para a sua solução. Somado a isso, o pacote traz grande flexibilidade de modelagem, uma vez que permite ao usuário adicionar qualquer distribuição desejada. Por fim, sua sintaxe é simples e muito similar à diversos outros pacotes.

Uma desvantagem, no entanto, é o fato desse pacote implementar apenas

a dinâmica GAS(p,q) para modelos *score-driven*, não sendo possível optar pela dinâmica via componentes não observáveis.. Essa limitação impossibilita modelar a série temporal a partir de suas componentes de tendência e sazonalidade, algo que será fundamental para este projeto.

### 2.1.4

#### GAS Package - R

O pacote GAS, desenvolvido por [Ardia, Boudt e Catania 2016] na linguagem R foi o primeiro a ser implementado e, por isso, foi utilizado também como referência para os pacotes apresentados. Ele permite ao usuário realizar simulação, estimação e previsão de modelos GAS(1,1) tanto univariados quanto multivariados, o que é uma grande qualidade. No entanto, o pacote apresenta limitações para modelos GAS multivariados com mais de 4 séries temporais, uma vez que ele não reporta a atualização exata dos parâmetros de atualização de correlação nesses casos. Para entender melhor essa limitação, ver [Ardia, Boudt e Catania 2016]. A distribuições implementadas no pacote são:

- Univariadas: Gaussiana, Skew-Gaussiana, t de Student, Skew t de Student, t de Student Assimétrica, Laplace Assimétrica, Poisson, Bernoulli, Gamma, Exponencial, Beta, Negativa Binomial e Skellam
- Multivariadas: Gaussiana e t de Student.

A funcionalidade desse pacote é muito similar aos demais apresentados, com funções usuais de um pacote de modelagem de séries temporais. O usuário possui a capacidade de especificar o modelo GAS a ser implementado. Para isso, é necessário definir a distribuição de probabilidade, o tipo de parametrização (identidade, inversa ou raiz quadrada da inversa) e quais parâmetros da distribuição definida serão variantes no tempo. A partir do modelo especificado, existem funções próprias para estimar o modelo, obter os resíduos do modelo e realizar a previsão do modelo. O script abaixo traz um exemplo de código R para realizar essas instruções. Em seguida, as Figuras 2.13 e 2.14 exibem o *fit in sample* e os resíduos do modelo com sua FAC, e a Figura 2.15 exibe a previsão 12 passos à frente.

---

#### Código 2: Script GAS package R

---

```

1 library(GAS)
2
3 # Especificação do modelo
4 GASSpec = UniGASSpec(Dist = "gamma", ScalingType = "Identity",
5                         GASPar = list(location = TRUE, scale = TRUE))
6 Fit      = UniGASFit(GASSpec, y) # estima o modelo

```

```
7 residuals = residuals(Fit) # obtm os resduos do modelo estimado
8 forecast  = UniGASFor(Fit, 12) #gera a previsao 12 passos a frente
```

---

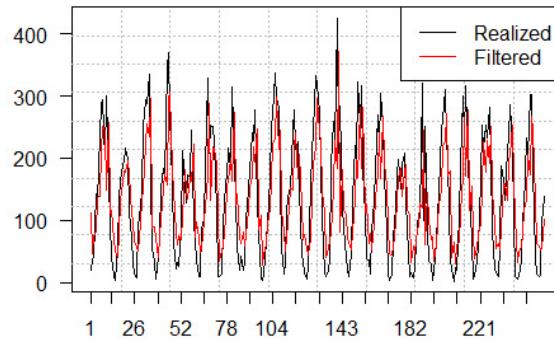


Figura 2.13: Fit in Sample do modelo GAS(1,1) gama estimado no pacote GAS R.

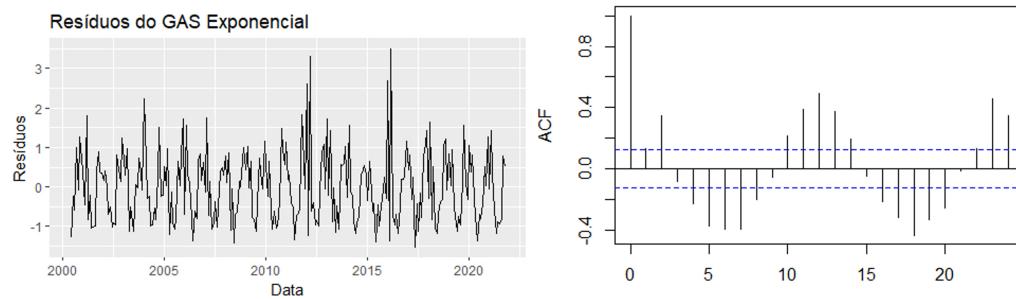


Figura 2.14: Resíduos e FAC dos resíduos do modelo GAS(1,1) gama estimado no pacote GAS R.

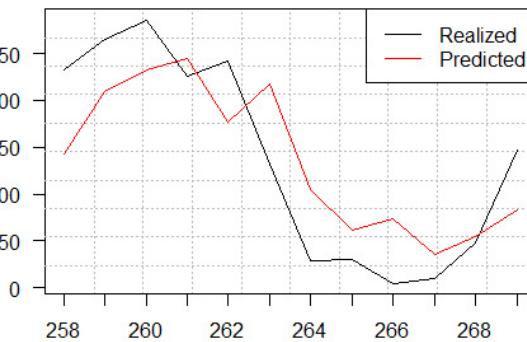


Figura 2.15: Previsão do modelos GAS(1,1) estimado no pacote GAS R.

Tal qual os outros pacotes, é possível escolher uma dentre diversas distribuições de probabilidade, o que é fundamental em termos de implementação para que o usuário possa desfrutar da flexibilidade de modelagem que a teoria dos modelos fornece. Ademais, ele também permite a escolha de diferentes métodos de otimização para a estimação do modelo.

Um diferencial deste pacote é a existência de funções prontas para realizar previsão de vários passos à frente a partir de uma janela rolante de reestimação do modelo após uma previsão de um passo à frente. Além disso, também

estão implementadas funções específicas para realização de *backtests* do modelo estimado e para simulações de séries a partir de um processo GAS.

Poderíamos citar apenas duas limitações do pacote. Primeiramente, ele implementa apenas a dinâmica GAS( $p,q$ ) (ver Capítulo 4), tal qual o pacote *ScoreDrivenModels.jl*. Somado a isso, o pacote está limitado a modelos com estrutura de lag (1,1), isto é, apenas implementa o GAS(1,1). Essa é uma grande desvantagem do pacote pois limita muito a modelagem das séries temporais sazonais, visto que utilizar mais *lags* da série captura melhor a estrutura de dependência linear e, inclusive, a inclusão de *lags* sazonais pode melhorar consideravelmente a estimação dos modelos. Vale comentar que essa limitação é notada no exemplo dessa seção, uma vez que a FAC dos resíduos exibida na Figura 2.14 indica que o modelo não capturou a estrutura sazonal da série.

### 2.1.5

#### **UnobservedComponentsGAS.jl - Julia**

Por último, o pacote *UnobservedComponentsGAS.jl*, que está sendo desenvolvido como uma dissertação de mestrado de [Alves], no laboratório LAMPS PUC-Rio, implementa modelos GAS cuja dinâmica dos parâmetros variantes no tempo segue um modelo de componentes não observáveis.

O grande diferencial deste pacote é o fato de sua implementação ter sido inteiramente desenvolvida a partir do JuMP, que é um poderoso *framework* de otimização dentro da linguagem Julia. Uma das vantagens dessa implementação é a possibilidade de tratar as restrições de parâmetros positivos das distribuições não com o ferramental de funções de ligação, mas como simples restrições do problema de otimização. Isso traz simplicidade e flexibilidade para a implementação. Somado a isso, o pacote traz funcionalidades de otimização robusta e penalização nos  $\kappa$ s das componentes.

Até o momento, as distribuições implementadas são a distribuição  $Normal(\mu, \sigma^2)$  e a distribuição  $tLocationScale(\mu, \sigma^2, \nu)$ , de média  $\mu$ , variância  $\sigma^2$  e  $\nu$  graus de liberdade.

As componentes que podem ser adicionadas à dinâmica dos parâmetros variantes no tempo até o momento são:

- Random walk
- Random walk with slope
- AR( $p$ )
- Sazonalidade por funções trigonométricas

Além disso, o pacote também fornece ao usuário a opção de estimar um *auto-gas*, de forma a deixar para o otimizador escolher os melhores valores do hiperparâmetro  $d$ , da matriz de informação de Fisher e do hiperparâmetro  $\alpha$ , que é uma penalização da otimização.

O *script* abaixo exemplifica a utilização deste pacote para estimar um modelo GAS-CNO Gaussiano, com média variante no tempo e variância fixa. Note que o modelo a ser estimado é um modelo cuja média segue uma dinâmica com *random walk with slope* e sazonalidade com período sazonal igual a 12. Além disso, o usuário pode utilizar a função *fit*, caso deseje especificar os valores de  $d$  e  $\alpha$ , ou a função *auto\_gas*, caso queira que o otimizador escolha os melhores valores para esses hiperparâmetros.

---

**Código 3:** Script UnobservedComponentsGAS.jl

---

```

1 using UnobservedComponentsGAS
2
3 "Código para definição e estimação do modelo GAS Normal"
4 dist = UnobservedComponentsGAS.NormalDistribution(missing, missing)
5 time_varying_params = [true, false]
6 random_walk = Dict(1=>false, 2=>false)
7 random_walk_slope = Dict(1=>true, 2=>false)
8 ar = Dict(1=>false, 2=>false)
9 seasonality = Dict(1=>12)
10 robust = false
11 stochastic = false
12 d = 1.0
13 num_scenarios = 500
14
15 gas_model = UnobservedComponentsGAS.GASModel(dist,
       time_varying_params, d, random_walk, random_walk_slope, ar,
       seasonality, robust, stochastic)
16
17 fitted_model = UnobservedComponentsGAS.fit(gas_model, y_fit)
18 fitted_model = UnobservedComponentsGAS.auto_gas(gas_model, y_fit,
       steps_ahead)
19
20 residuals = fitted_model.residuals
21 forecast = UnobservedComponentsGAS.predict(gas_model, fitted_model,
       y_fit, steps_ahead, num_scenarios)

```

---

As figuras a seguir exibem os mesmos gráficos exibidos para os demais pacotes: a Figura 2.16 exibe o *fit in sample* do modelo para a série de vazão, a Figura 2.17 exibe os resíduos padronizados e a FAC desses resíduos e, por fim, a Figura 2.18 exibe a previsão 12 passos à frente deste modelo, em comparação com os valores reais da série temporal.

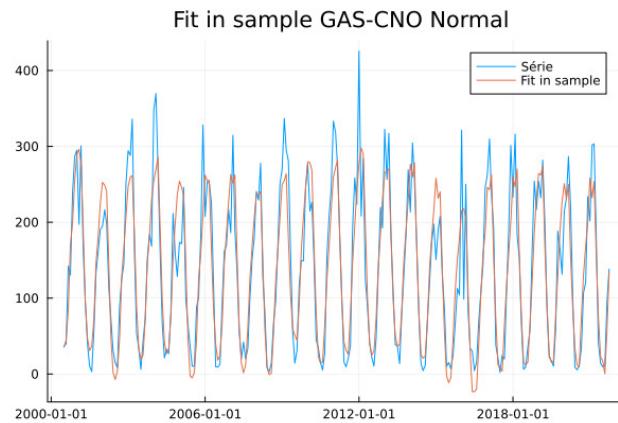


Figura 2.16: Fit in Sample do modelo GAS-CNO normal estimado no pacote UnobservedComponentsGAS.jl.

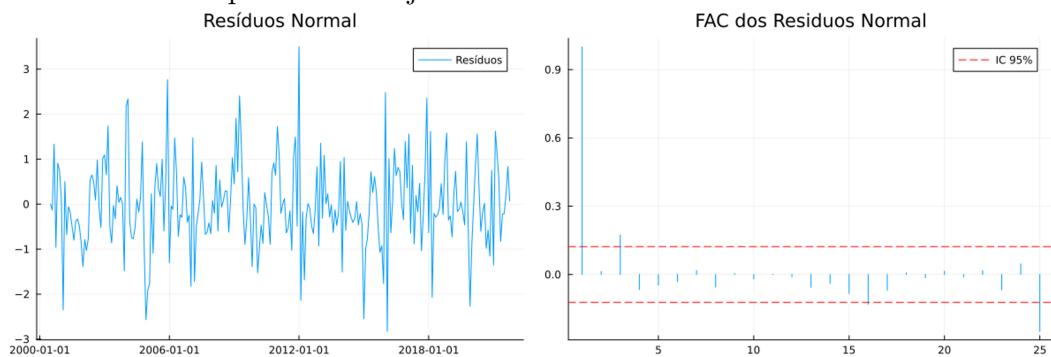


Figura 2.17: Resíduos e FAC dos resíduos do modelo GAS-CNO normal estimado no pacote UnobservedComponentsGAS.jl.

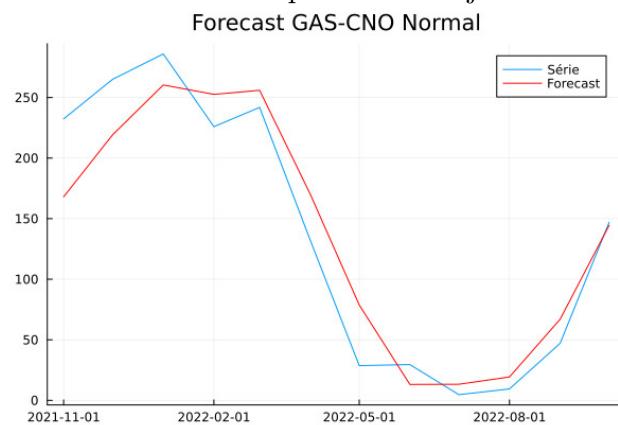


Figura 2.18: Previsão do modelo GAS-CNO normal estimado no pacote UnobservedComponentsGAS.jl.

## 2.1.6

### Comparação dos softwares

Nesta sessão, apresento uma tabela que resume a comparação dos diferentes *softwares* apresentados anteriormente. A Tabela 2.1 apresenta tanto se o *software* implementa GAS-CNO ou GAS( $p,q$ ), quanto as distribuições disponíveis.

Software/Pacote e Dinâmica	Distribuições Disponíveis
TSL Score GAS-CNO	Poisson, Negativa Binomial, Bernoulli e Skellam. Normal, t de Student, GED, Beta Generalizada Tipo 2, Exponencial e Weibull
TSL GAS-CNO GAS(p,q)	Poisson, Negativa Binomial, Bernoulli e Skellam. Normal, t de Student, GED, Beta Generalizada Tipo 2, Exponencial e Weibull
GAS R GAS(1,1)	Gaussiana, Skew-Gaussiana, t de Student, Skew t de Student, t de Student Assimétrica, Laplace Assimétrica, Poisson, Bernoulli, Gamma, Exponencial, Beta, Negativa Binomial e Skellam
ScoreDrivenModels Julia GAS(p,q)	Todas do Distributions.jl
UnobservedComponentsGAS Julia GAS-CNO	Normal e t de Student

Tabela 2.1: Tabela das distribuições e dinâmicas implementadas nos *softwares* apresentados.

### 3 Objetivos

O foco principal desse projeto são os modelos *score-driven* para séries temporais. Dentro deste escopo, a proposta consiste em aprofundar os estudos em modelos GAS cuja distribuição de probabilidade não é Gaussiana, e cuja dinâmica dos parâmetros variantes no tempo segue um modelo de componentes não observáveis (CNO), com o intuito de modelar séries temporais com restrições em seu domínio. Dois exemplos clássicos são séries ENA e séries de carga, ambas assumindo apenas valores positivos.

A partir desses modelos, um dos objetivos é estudar a decomposição das séries em componentes de tendência e sazonalidade, com parâmetros variantes no tempo e com combinações não lineares dessas componentes. A principal decomposição, gerada a partir do modelo de espaço de estados [Harvey 2013] e [Caivano, Harvey e Luati 2016], na qual a série é decomposta como a soma de uma componente de tendência  $\mu_t$  com uma componente sazonal  $\gamma_t$ , acrescida de um choque aleatório normal  $\epsilon_t$  pode ser expressa como:

$$y_t = \mu_t + \gamma_t + \epsilon_t \quad (3-1)$$

Partindo para a formulação de modelos *score-driven* seja  $y_t$  uma série temporal que seja uma distribuição *gamma* com a seguinte parametrização:

$$y_t \sim \text{gamma}(\alpha, \lambda_t) \quad (3-2)$$

onde  $\alpha > 0$ ,  $\lambda_t > 0$ ,  $E[y_t] = \lambda_t$  e  $Var[y_t] = \lambda_t^2/\alpha$ .

Assuma, por exemplo, que o parâmetro da média seja uma dinâmica CNO com uma componente de tendência  $m_t$  e uma de sazonalidade  $\gamma_t$ . Sendo assim, a combinação clássica dessas componentes é tal que:

$$\lambda_{t+1|t} = m_{t+t|t} + \gamma_{t+t|t} \quad (3-3)$$

Serão estudadas duas possibilidades de combinações não lineares dessas componentes. Essas combinações serão avaliadas de acordo com a qualidade do modelo estimado, tanto a partir de uma análise de resíduos quanto de poder preditivo fora da amostra. São elas:

$$\lambda_{t+1|t} = m_{t+t|t} \times \gamma_{t+t|t} \quad (3-4)$$

$$\lambda_{t+1|t} = m_{t+t|t} \times (1 + \gamma_{t+t|t}) \quad (3-5)$$

No entanto, ao entrar no escopo de combinações não lineares, a quantidade de diferentes maneiras de combinar as componentes de tendência e

sazonalidade torna-se muito maior que a quantidade possível com apenas combinações lineares. Também poderiam ser estudados componentes ao quadrado, multiplicação dessas componentes ao quadrado, dentre muitas outras.

A inspiração para essas combinações são os modelos ETS de [Hyndman et al. 2008], que são uma classe de modelos baseada em componentes de tendência e sazonalidade que contam com diversas combinações possíveis, tanto dessas componentes quanto do termo de erro. A Figura 3.1, retirada de [Hyndman e Athanasopoulos 2021] exibe todas as combinações possíveis dentro dessa classe de modelo. Na Figura, **N** significa a ausência da componente, **A** significa combinação aditiva, **M** significa combinação multiplicativa e **A<sub>d</sub>** significa tendência aditiva e amortecida.

Trend	Seasonal		
	N	A	M
<b>N</b>	$\hat{y}_{t+h t} = \ell_t$ $\ell_t = \alpha y_t + (1 - \alpha)\ell_{t-1}$	$\hat{y}_{t+h t} = \ell_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1 - \gamma)s_{t-m}$	$\hat{y}_{t+h t} = \ell_t s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1 - \alpha)\ell_{t-1}$ $s_t = \gamma(y_t/\ell_{t-1}) + (1 - \gamma)s_{t-m}$
<b>A</b>	$\hat{y}_{t+h t} = \ell_t + hb_t$ $\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$	$\hat{y}_{t+h t} = \ell_t + hb_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m}$	$\hat{y}_{t+h t} = (\ell_t + hb_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + b_{t-1})) + (1 - \gamma)s_{t-m}$
<b>A<sub>d</sub></b>	$\hat{y}_{t+h t} = \ell_t + \phi_h b_t$ $\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$	$\hat{y}_{t+h t} = \ell_t + \phi_h b_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1 - \gamma)s_{t-m}$	$\hat{y}_{t+h t} = (\ell_t + \phi_h b_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + \phi b_{t-1})) + (1 - \gamma)s_{t-m}$

Figura 3.1: Família de modelos ETS.

A motivação da implementação das combinações multiplicativas pode ser ilustrada pela Figura 3.2. Note que os valores estimados pelo modelo não conseguem alcançar os picos e vales da série temporal, o que indica que a componente sazonal não foi estimada com toda a amplitude necessária. A combinação multiplicativa das componentes, normalmente, se apresenta como uma solução para este problema de estimação.

A fim de realizar esses experimentos, após o estudo e desenvolvimento teórico necessário, a proposta é implementar essas extensões do arcabouço de modelos GAS a partir do pacote *UnobservedComponentsGAS.jl*, escrito em Julia por [Alves]. Apenas com desenvolvimento dessa ferramenta computacional poderemos averiguar se essas alterações propostas conseguem resolver o problema da dependência sazonal não capturada pelo modelo, evidenciada pela FAC dos resíduos do modelo estimado.

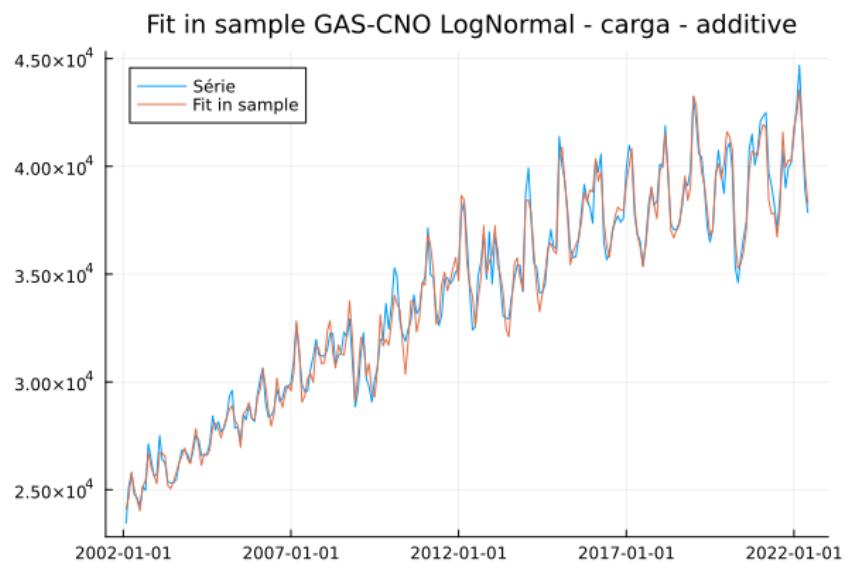


Figura 3.2: *Fit in sample* de um modelo GAS-CNO log-normal com combinação aditiva das componentes de tendência e sazonalidade.

## 4

# Metodologia

### 4.1

#### Fundamentação Teórica

Modelos para séries temporais podem ser classificados em duas classes: *observation driven* (OD) e *parameter driven* (PD), tal qual proposto em [Cox et al. 1981]. Modelos PD são aqueles cujos parâmetros variantes no tempo são função de seus valores passados e de inovações aleatórias. Ou seja, se  $f_t$  é o parâmetro variante no tempo e  $\eta_t$  é um ruído aleatório puro, temos que:

$$f_t = g(f_{t-1}, \eta_t) \quad (4-1)$$

onde  $g$  é uma função contínua e diferenciável.

Um exemplo clássico de modelos PD são os modelos de espaço de estados ou modelos de componentes não observáveis (CNO), onde, por exemplo

$$f_t = \phi f_{t-1} + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2), \quad |\phi| < 1 \quad (4-2)$$

Os modelos score-driven, por sua vez, pertencem à classe OD, o que significa que eles são definidos a partir de uma distribuição condicional de probabilidade que depende dos valores passados da própria série temporal [Harvey 2022]. Em outras palavras,

$$f_{t|t-1} = g(f_{t-1|t-2}, s_t), \quad s_t = s_t(\mathbf{Y}_{t-1}) \quad (4-3)$$

onde  $s_t$  é chamado de *score* e  $\mathbf{Y}_{t-1} = (y_{t-1}, \dots, y_2, y_1)$ .

Sendo assim, o primeiro passo da modelagem é definir qual é a distribuição de probabilidade seguida pela série a ser modelada. Ou seja, dada uma série  $\mathbf{y}_t$  e o seu passado  $\mathbf{Y}_{t-1} = (y_{t-1}, y_{t-2}, \dots, y_2, y_1)$ , preciso definir uma distribuição condicional tal que

$$y_t \sim p(y_t | f_{t|t-1}; \mathbf{Y}_{t-1}, \theta) \quad (4-4)$$

onde  $f_{t|t-1}$  é o parâmetro variante no tempo estimado em  $t$  e  $\theta$  é o vetor de parâmetros fixos a serem estimados.

Vale ressaltar que a distribuição pode ser de qualquer tipo: contínua, discreta, univariada, multivariada, etc.

Uma vez definida a distribuição condicional de probabilidade, o próximo passo é definir a função de atualização dos parâmetros variantes no tempo, ou *updating mechanism*. Existem duas possibilidades: a primeira é a função de atualização seguir um processo ARMA(p,q) e a segunda é desenvolver essa

função a partir do arcabouço dos modelos observáveis (CNO). A sessão 4.1.1 desenvolve a abordagem via CNO, que é a dinâmica utilizada neste projeto.

É importante definirmos alguns conceitos essenciais para modelos *score-driven*, sendo o primeiro deles o *score*. É esse conceito que conduz a dinâmica do modelo de série temporal a cada instante de tempo, sendo definido como:

$$\nabla_t = \frac{\partial \ln p(y_t | f_t; \mathbf{Y}_{t-1}, \theta)}{\partial f_t} \quad (4-5)$$

Quando combinado com a matriz de informação de Fisher,  $I_{t|t-1}$ , obtemos o chamado *score* padronizado  $s_t$  que é quem irá guiar o desenvolvimento do modelo. Suas definições são:

$$I_{t|t-1} = E_{t-1}[\nabla_t \nabla_t'] = -E_{t-1}\left[\frac{\partial^2 \ln p(y_t | f_t; \theta, \mathbf{Y}_{t-1})}{\partial f_t \partial f_t'}\right] \quad (4-6)$$

$$s_t = I_{t|t-1}^{-d} \nabla_t, d = 0, 1, 0.5 \quad (4-7)$$

Vale comentar que, quando  $d = 0.5$ , a definição de  $s_t$  é ligeiramente diferente, como mostra [Sarlo, Fernandes e Borenstein 2023].

Usualmente, modelos *score-driven* utilizam funções de ligação  $h(\cdot)$  para restringir os parâmetros variantes no tempo ao seu domínio.

Assuma, por exemplo, uma distribuição Gamma tal qual a apresentada na seção anterior:  $X \sim Gama(\alpha_{t|t-1}, \lambda_{t|t-1})$ , tal que  $E[X] = \lambda_{t|t-1}$  e  $V[X] = \lambda_{t|t-1}^2 / \alpha_{t|t-1}$ .

Por definição, ambos os parâmetros  $\alpha_{t|t-1}$  e  $\lambda_{t|t-1}$  devem ser positivos. Para garantir que essas restrições sejam atendidas, o usual seria utilizar a função logarítmica de tal modo que:

$$\begin{aligned} h_1(\alpha_t) &= \tilde{f}_{1,t} \longrightarrow \ln(\alpha_t) = \tilde{f}_{1,t} \longrightarrow \alpha_t = e^{\tilde{f}_{1,t}}, & \alpha_t &= f_{1,t} \\ h_2(\lambda_t) &= \tilde{f}_{2,t} \longrightarrow \ln(\lambda_t) = \tilde{f}_{2,t} \longrightarrow \lambda_t = e^{\tilde{f}_{2,t}}, & \lambda_t &= f_{2,t} \end{aligned}$$

A partir dessas novas expressões para os parâmetros variantes no tempo com função de ligação, seria necessário obter todas as novas expressões para  $\tilde{\nabla}_t$ ,  $\tilde{s}_t$  e  $\tilde{I}_{t|t-1}$ .

Uma propriedade conveniente de modelos *observation driven* é que a estimativa dos parâmetros pode ser realizada via máxima verossimilhança (MV) [Creal, Koopman e Lucas 2013]. A função verossimilhança, denotada por  $L(\theta)$  é, simplesmente, a distribuição conjunta condicional de probabilidade escrita como função de um parâmetro  $\theta$  da distribuição. É comum utilizar a função log da verossimilhança, representada por  $l(\theta) = \ln L(\theta)$ . Dito isso, seja  $\mathbf{y}_t$  uma série temporal de observações  $y_1, \dots, y_T$ . O problema de maximização

da verossimilhança pode ser escrito como

$$\hat{\theta} = \arg \max_{\theta} \sum_{t=1}^T l_t$$

onde  $l_t = \ln p(y_t|f_t, \mathbf{Y}_{t-1}; \theta)$  é a função log da verossimilhança para uma realização de  $y_t$ . Segundo [Creal, Koopman e Lucas 2013], estimar via MV para modelos GAS é particularmente simples, bastando implementar a função de atualização e avaliar a função  $l_t$  em um valor particular  $\theta^*$  de  $\theta$ .

Como é usual para modelos de séries temporais, a qualidade do modelo é avaliada a partir de diagnósticos dos resíduos do modelo estimado. No caso de GAS, há duas classes possíveis de resíduos: os resíduos de Pearson e os resíduos quantílicos.

Os resíduos de Pearson são definidos como:

$$r_t^P = \frac{y_t - E(y_t|f_{t|t-1}, \mathbf{Y}_{t-1}, \theta)}{\sqrt{Var(y_t|f_{t|t-1}, \mathbf{Y}_{t-1}, \theta)}} \quad \forall t = 1, \dots, T \quad (4-8)$$

onde  $E(y_t|f_t, \mathbf{Y}_{t-1}, \theta)$  e  $Var(y_t|f_t, \mathbf{Y}_{t-1}, \theta)$  são, respectivamente, a média e a variância do modelo probabilístico  $p(y_t|f_t, \mathbf{Y}_{t-1}, \theta)$  [Fernandes 2020].

Os resíduos quantílicos, por sua vez, são definidos como:

$$r_t^q = \Phi^{-1}(F(y_t|f_t, \mathbf{Y}_{t-1}, \hat{\theta})) \quad \forall t = 1, \dots, T \quad (4-9)$$

onde  $\Phi^{-1}(\cdot)$  é o quantil de uma distribuição normal padrão  $Z = N(0, 1)$  e  $F(y_t|f_t, \mathbf{Y}_{t-1}, \hat{\theta})$  é a função de distribuição acumulada associada a  $p(y_t|f_t, \mathbf{Y}_{t-1}, \theta)$ .

Ambos os resíduos devem ser submetidos aos testes usuais de diagnóstico para avaliar normalidade, autocorrelação nula e homocedasticidade [Jonathan e Kung-Sik 2008]. É importante comentar que, assim como mostrou [Kalliovirta 2012], se o modelo GAS foi bem especificado, os resíduos quantílicos são assintoticamente normais e independentemente distribuídos, mesmo que o modelo probabilístico não tenha sido Gaussiano. Com isso, os testes de normalidade dos resíduos podem ser mantidos como diagnóstico.

É importante ressaltar que os resíduos de Pearson são apropriados apenas para modelos aditivos e gaussianos. Assuma, por exemplo, que a estrutura de um modelo de série temporal aditivo segue a seguinte equação:

$$y_t = \mu_{t|t-1} + \gamma_{t|t-1} + \epsilon_t \quad (4-10)$$

onde  $\mu_{t|t-1}$  é a componente de tendência e  $\gamma_{t|t-1}$  a de sazonalidade, além de que  $\epsilon_t \sim N(0, 1)$ .

Podemos reescrever essa equação para isolar o termo de erro, dando

origem ao resíduo de Pearson. Note que, por construção, o erro é homocedástico e normal.

$$\frac{\epsilon_t}{\sigma} = \frac{y_t - (\mu_{t|t-1} + \gamma_{t|t-1})}{\sigma} \sim N(0, 1) \quad (4-11)$$

O mesmo seria verdade para um outro modelo com combinação multiplicativa das componentes mas ainda com o termo de erro aditivo.

$$y_t = \mu_{t|t-1} \times \gamma_{t|t-1} + \epsilon_t \quad (4-12)$$

$$\frac{\epsilon_t}{\sigma} = \frac{y_t - (\mu_{t|t-1} \times \gamma_{t|t-1})}{\sigma} \sim N(0, 1) \quad (4-13)$$

Por outro lado, caso fosse definido um modelo cujo termo de erro se combina de forma multiplicativa, esse tipo de resíduo deixaria de ser adequado. Tome, por exemplo, o modelo de equação

$$y_t = \mu_{t|t-1}(1 + \epsilon_t) \quad (4-14)$$

Desenvolvendo essa equação para obter a expressão para o termo de erro não obteríamos a expressão equivalente ao resíduo de Pearson, como mostrado a seguir

$$\epsilon_t = \frac{y_t - \mu_{t|t-1}}{\mu_{t|t-1}} \quad (4-15)$$

Além disso, caso o modelo escolhido seja baseado em uma distribuição não Gaussiana, como a gama, por exemplo, não seria possível reescrever a equação em função do termo de erro, de tal modo que a definição do resíduo do modelo seria ambígua.

Devido a esses fatos, é aconselhável utilizar os resíduos quantílicos para modelos *score-driven*, dado que eles são bem definidos para todas as distribuições e combinações de componentes.

Por fim, uma vez constatado que o modelo foi bem estimado, o último passo é realizar a previsão do modelo para  $k$  passos à frente. Note que, se  $k = 1$ , a previsão é obtida de forma imediata pela distribuição condicional de probabilidade. No entanto, para  $k > 1$ , não existe forma analítica e deve ser realizada através de simulações de Monte Carlo. O algoritmo 1 descreve o pseudo-código para realizar essas simulações.

---

**Algoritmo 1:** Previsão k passos à frente para modelos GAS(p,q).

---

**Entrada:**  $f_{t,q} = \{f_t, f_{t-1}, \dots, f_{t-q}\}$ ,  $s_{t,p} = \{s_t, s_{t-1}, \dots, s_{t-p}\}$

**Saída:**  $\hat{y}_{t,k} = \{y_{t+1}, \dots, y_{t+k}\}$

```

1    $k \leftarrow 1;$ 
2    $m \leftarrow 1;$ 
3   para  $m \leftarrow 1$  até  $m \leftarrow 2000$  faça
4      $k \leftarrow 1;$ 
5     para  $k \leftarrow 1$  até  $k = K$  faça
6       Com base em  $\hat{\theta}$ , gere 1 valor da densidade preditiva de  $\hat{y}_{t+k|t}^{(m)}$ ;
7       Calcule  $f_{t+k-1,q}$  e  $s_{t+k-1,p}$ ;
8       Atualize  $f_{t+k,q}$  e  $s_{t+k,p}$ ;
9     se  $m = 2000$  e  $k = K$  então
10       $\hat{y}_{t+k|t}$  é obtido pela média das m trajetórias calculadas a cada
           instante de tempo  $k$ ;
11      Os intervalos de confiança são obtidos pelos quantis  $(\alpha/2)$  e
            $(1 - \alpha/2)$  das m trajetórias;
12    senão
13       $m = m + 1;$ 

```

---

#### 4.1.1

##### Modelos score-driven - CNO

Podemos especificar os modelos GAS a partir do arcabouço dos modelos de componentes não observáveis para a função de atualização de  $f_{t|t-1}$ . Essa formulação foi apresentada em [Harvey 2013] e [Caivano, Harvey e Luati 2016] e é especialmente útil para o problema que este trabalho vai endereçar, uma vez que, dentro desse arcabouço, as componentes de tendência e sazonalidade ficam explícitas.

Para explicitar essa formulação, devemos partir de um modelo estrutural básico onde a série temporal é decomposta em componentes de tendência linear estocástica  $\mu_t$  e sazonalidade estocástica  $\gamma_t$  por *dummies*, como dado a seguir:

$$\begin{aligned}
 y_t &= \mu_t + \gamma_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma^2) \\
 \mu_{t+1} &= \mu_t + \beta_t + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2) \\
 \beta_{t+1} &= \beta_t + \xi_t, \quad \xi_t \sim N(0, \sigma_\xi^2) \\
 \gamma_{t+1} &= \sum_{j=0}^{S-2} \gamma_{t-j} + \omega_t, \quad \omega_t \sim N(0, \sigma_\omega^2)
 \end{aligned} \tag{4-16}$$

Onde  $S$  é o período sazonal da série e  $\epsilon_t, \eta_t, \xi_t, \omega_t$  são choques aleatórios

normais de média zero e variância constante no tempo. Essa formulação está melhor desenvolvida no Apêndice B.

É necessário, a partir dessa formulação, alterar os choques aleatórios de cada componente, isto é,  $\epsilon_t, \eta_t, \xi_t$  e  $\omega_t$  para transformar esse modelo em um modelo GAS. A seguir estão os passos necessários para obter um modelo GAS estrutural, supondo que a distribuição de  $y_t$  é uma  $gama(\alpha, \lambda_t/\alpha)$ , com  $\lambda_t > 0, \alpha > 0$ , sendo o primeiro o parâmetro forma e o segundo escala.

Sabe-se que, utilizando essa parametrização,  $E(y_t|f_{t|t-1}, \mathbf{Y}_{t-1}, \theta) = \lambda_t$  e  $V(y_t|f_{t|t-1}, \mathbf{Y}_{t-1}, \theta) = \lambda_t^2/\alpha$ . Note que decidi por apenas um parâmetro variante no tempo, o *shape*. Além disso, sabe-se que o *score* padronizado associado ao parâmetro *shape*, com  $d = 1$  e *link function* logarítmica, é  $\nabla_t^\lambda = \frac{y_t}{\lambda_t - 1}$  (ver Apêndice A.1). Chamarei, a partir desse momento, esse *score* de  $u_t$  e ele fará o papel dos choques aleatórios existentes em um modelo CNO ao ser transscrito para a modelagem *score-driven*.

Agora vamos apresentar a especificação da componente sazonal  $\gamma_t$  através de um mecanismo um pouco mais geral do que aquele apresentado em 4-16. Diferente do que está descrito no Apêndice B, vamos começar assumindo que o fator sazonal  $\gamma_{j,t}$  da estação  $j$  no mês  $t$ , assumindo uma série mensal, possui a dinâmica de um passeio aleatório sem *drift*. Ou seja:

$$\gamma_{j,t+1} = \gamma_{j,t} + \omega_{j,t} \quad \omega_{j,t} \sim N(0, \sigma_\omega^2) \quad j = 1, \dots, s \quad (4-17)$$

Estação pode significar a própria frequência mensal da série, ou outra divisão periódica qualquer, como as próprias estações do ano.

É possível colocar a equação anterior em forma vetorial, tal que:

$$\boldsymbol{\gamma}_{t+1} = \boldsymbol{\gamma}_t + \boldsymbol{\omega}_t \quad (4-18)$$

onde  $\boldsymbol{\gamma}_t = [\gamma_{1,t}, \dots, \gamma_{s,t}]^T$  e  $\boldsymbol{\omega}_t = [\omega_{1,t}, \dots, \omega_{s,t}]^T$ . Assim como foi feito na parametrização 3 apresentada no Apêndice B, é necessário adicionar a restrição de que a soma dos componentes sazonais de todos os  $j$  é igual a zero.

Até o momento, ainda estamos com a formulação dentro do escopo dos modelos de espaço de estados. Para alterar essa formulação para modelos *score-driven*, o primeiro passo será substituir os choque aleatórios  $\boldsymbol{\omega}_t$  por  $\kappa_t u_t$ , obtendo uma nova equação vetorial. Note que  $u_t$  não está em negrito por não representar um vetor, e sim um escalar. A nova equação é que descreve a componente sazonal em um modelo GAS-CNO é:

$$\boldsymbol{\gamma}_{t+1|t} = \boldsymbol{\gamma}_{t|t-1} + \kappa_t u_t \quad (4-19)$$

onde  $u_t$  é o *score* e  $\kappa_t = [\kappa_{1,t}, \dots, \kappa_{s,t}]$ .

É necessário definir um conjunto de variáveis binárias  $z_t$  para indicar se

o fator sazonal  $\gamma_{j,t}$  está ou não ativo em um dado mês  $t$ . Sendo assim,

$$z_t = \begin{cases} 1, & \text{se } t = j \\ 0, & \text{se } t \neq j \end{cases} \quad (4-20)$$

A partir da definição dessas variáveis, temos que o fator sazonal de cada mês  $t$  será dado por

$$s_{t+1|t} = z_t \gamma_{t+1|t} \quad (4-21)$$

Tal como apresentado para as componentes sazonais de modelos CNO, a soma dos fatores sazonais do modelo *score-driven* - CNO deve ser igual a zero, o que implica restrições nos  $\kappa_t$ 's ver [Harvey 2013]. Essa restrição, embora valha apenas para modelos aditivos, pode ser utilizada para modelos multiplicativos caso não seja do interesse analisar as componentes estimadas, uma vez que a previsão do modelo não é impactada [Hyndman et al. 2008].

Com isso, utilizando a equação 4-21 junto das equações para descrever a dinâmica da tendência ao longo do tempo, obtemos o seguinte modelo GAS-CNO com distribuição *gama*.

$$\hat{y}_{t+1|t} = E[y_t|f_t, \mathbf{Y}_t, \theta] = \lambda_t = \exp(f_t) \quad (4-22)$$

$$f_{t+1|t} = \mu_{t+1|t} + s_{t+1|t} \quad (4-23)$$

$$\mu_{t+1|t} = \mu_{t|t-1} + \beta_{t|t-1} + k_1 u_t \quad (4-24)$$

$$\beta_{t+1|t} = \beta_{t|t-1} + k_2 u_t \quad (4-25)$$

$$s_{t+1|t} = \mathbf{z}_t \boldsymbol{\gamma}_{t+1|t} \quad (4-26)$$

$$\boldsymbol{\gamma}_{t+1|t} = \boldsymbol{\gamma}_{t|t-1} + \mathbf{k}_t u_t \quad (4-27)$$

sendo  $u_t = \nabla_t^\lambda$ , o *score*,  $\mu_t$  a componente de tendência,  $\beta_t$  a componente de inclinação e  $s_t$  a componente sazonal.

A principal diferença entre a formulação de um modelo de espaço de estados descrito como 4-16 para um modelo *score driven* descrito como 4-22 é o fato do primeiro possuir choques aleatórios em suas equações de componentes enquanto que o segundo substitui esses choque por uma função do *score*, tornando o modelo dependente das observações passadas da série temporal. Com isso, agora temos a formulação de um modelo *score-driven* cujos parâmetros variantes no tempo seguem um processo CNO.

#### 4.1.2

#### Combinação não linear de tendência e sazonalidade

A maneira mais usual de combinar as componentes de tendência e sazonalidade, tanto em modelos estruturais, quanto em modelos GAS-CNO,

quando utilizadas distribuições Gaussianas e *t-student* é a combinação aditiva:

$$y_t = \mu_t + \gamma_t + \epsilon_t$$

No entanto, para distribuições não Gaussianas, é comum que utilizemos alguma forma multiplicativa de combinação de tendência e sazonalidade. Suponha, por exemplo, a modelagem de uma série  $y_t$  utilizando uma distribuição gama:

$$\begin{aligned} y_t &\sim Gama(\mu/\theta, \theta) \quad \mu > 0, \theta > 0 \\ E(y_t) &= \frac{\mu}{\theta} \cdot \theta = \mu \end{aligned}$$

Como foi visto, é preciso definir a distribuição condicional associada ao modelo:

$$\begin{aligned} p(y_t | \mathbf{Y}_{t-1}) &\sim Gama\left(\frac{\mu_{t|t-1}}{\theta_{t|t-1}}, \theta_{t|t-1}\right) \\ E(y_t | \mathbf{Y}_{t-1}) &= \mu_{t|t-1} \end{aligned}$$

Como o objetivo é estimar a média dessa distribuição ao longo do tempo, podemos decompor essa média em componentes de tendência e sazonalidade. Ao mesmo tempo, para garantir que a média seja sempre positiva, tal como manda a restrição desse parâmetro, podemos utilizar uma função de ligação exponencial. Com isso, obtemos a seguinte expressão para a média:

$$\begin{aligned} \mu_{t|t-1} &= \exp(m_{t|t-1} + \gamma_{t|t-1}) \\ &= \exp(m_{t|t-1}) \cdot \exp(\gamma_{t|t-1}) \\ &= m_{t|t-1}^* \cdot \gamma_{t|t-1}^* \end{aligned}$$

Note que, com essa formulação, o modelo final obtido foi um modelo multiplicativo entre as componentes.

Como este projeto não utilizou o ferramental de funções de ligação, essa formulação não foi a implementada.

As alterações propostas com o objetivo de generalizar a combinação de tendência e sazonalidade foi inspirada nas variações de modelos ETS, um outro modelo muito importante para séries temporais. A Figura 3.1, obtida de [Hyndman e Athanasopoulos 2021], apresentou diversas combinações possíveis em modelos ETS.

Como as possibilidades são inúmeras, a primeira a ser testada é a combinação multiplicativa entre tendência e sazonalidade, expressa pelo modelo

a seguir. Note que esta combinação independe da dinâmica de cada uma das componentes, que não foram especificadas neste momento.

$$\mu_t = m_t \times \gamma_t$$

onde  $\mu_t$  é um parâmetro variante no tempo de uma distribuição de probabilidade,  $m_t$  é a componente de tendência e  $\gamma_t$  é a componente de sazonalidade. A essa combinação foi dado o nome de "multiplicativa 1"

A segunda combinação não linear implementada e testada foi similar à multiplicativa 1:

$$\mu_t = m_t \times (1 + \gamma_t)$$

Essa combinação foi chamada de "multiplicativa 2". Ambas foram comparadas com a combinação aditiva usual.

## 4.2 Implementação Computacional

Nesta sessão, estão descritas as implementações e os testes realizados neste projeto. Na sessão 4.2.1 estão definidas as séries temporais que foram utilizadas. Em seguida, na sessão 4.2.2, apresento as distribuições de probabilidade escolhidas para modelar as séries temporais definidas. Nas sessões 4.2.3 e 4.2.4 apresento algumas questões de implementação dos modelo. Já na sessão 4.2.5 explicito o modelo GAS utilizado nos testes, definindo quais foram as componentes não observáveis consideradas. As sessões 4.2.6 e 4.2.7 discorrem sobre os diagnósticos e avaliações do modelo estimado. Por fim, a sessão 4.2.8 apresenta as formulações dos modelos testados para cada uma dessas alterações propostas.

É importante comentar que nenhum resultado é apresentado neste capítulo, dado que o objetivo seu objetivo é apenas formalizar a metodologia utilizada para os testes. Todos os resultados dos testes desenvolvidos encontram-se no capítulo 5.

### 4.2.1 Definição das séries temporais

Nesta seção está descrita a metodologia de testes utilizada neste projeto. Com o intuito de implementar e testar as alterações apresentadas para modelos GAS-CNO na seção 4.1.2 foram escolhidas 2 séries temporais mensais dentro do contexto de clima e energia.

A primeira é uma série temporal de carga mensal média, em MW, das regiões Sudeste e Centro Oeste do Brasil, obtida de <https://www.snirh.gov.br>.

[br/hidroweb/serieshistoricas](https://www.hidroweb.mcti.gov.br/hidroweb/serieshistoricas). A Figura 4.1 exibe o gráfico dessa série temporal. Optou-se por utilizar dados dessa série a partir de janeiro de 2002, até Junho de 2023 para evitar os *outliers* gerados pelo racionamento de junho e julho de 2001. Com isso, a série temporal possui 258 observações. Algumas características a se destacar sobre essa série é sua não estacionariedade, dada a existência de tendência, um visível aumento da variabilidade da série com o passar do tempo e, por fim, uma possível diminuição da inclinação da tendência nos últimos anos da série.

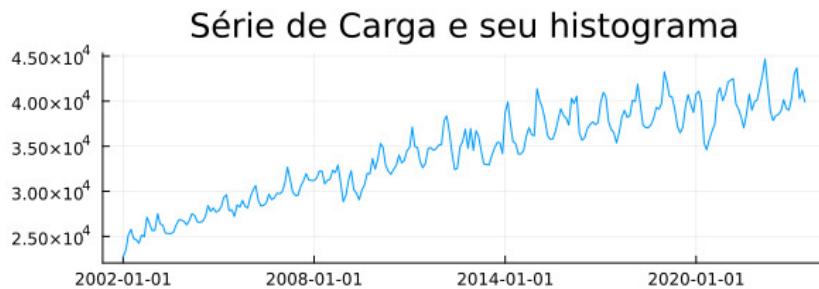


Figura 4.1: Série mensal de carga.

Em seguida, foi escolhida uma série mensal de ENA, isto é de Energia Natural Afluente, que corresponde à quantidade de água que chega a uma certa usina ou reservatório. A série escolhida foi obtida de [https://www.ons.org.br/Paginas/resultados-da-operacao/historico-da-operacao/energia\\_afluentes\\_subsystem.aspx](https://www.ons.org.br/Paginas/resultados-da-operacao/historico-da-operacao/energia_afluentes_subsystem.aspx) e corresponde a ENA Bruta do Sudeste e Centro Oeste do país, com 284 observações, entre Janeiro de 2000 e Agosto de 2023. A Figura 4.2 exibe o gráfico da série e seu histograma. A primeira característica que chama a atenção sobre essa série é seu histograma indicar que não se trata de uma distribuição normal. Outro fato importante é a série apresentar estacionariedade na média e um padrão sazonal visualmente marcante.

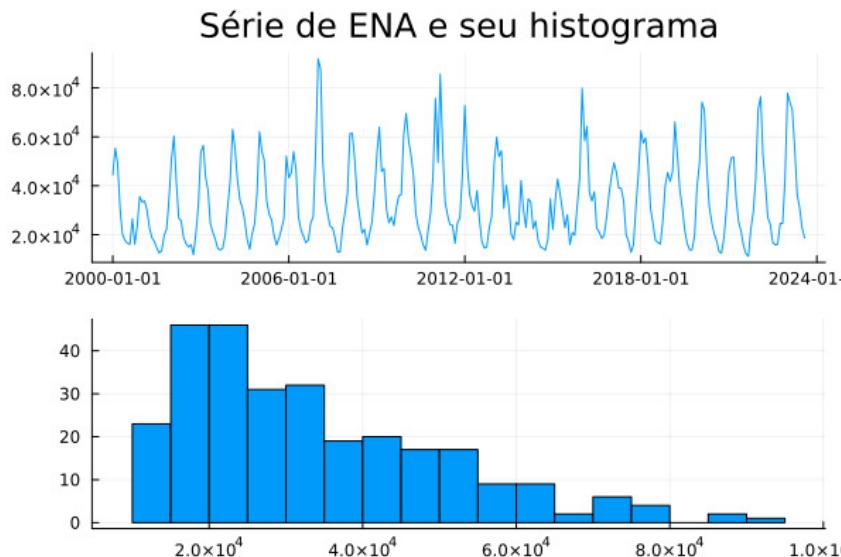


Figura 4.2: Série mensal de ENA.

Todas as séries foram separadas em períodos de treino, ou *in sample* e de teste, ou *out of sample*. A tabela 4.1 exibe, para cada série temporal, seu tamanho  $T$ , as datas separadas para a estimativa dos modelos (treino) e as datas separadas para validação do modelo (teste). Note que, para todas as séries, foram separadas as últimas 12 observações para servirem de dados *out of sample*.

Série	T	Treino	Teste
Carga	258	01/2002	07/2022
		a 06/2022	a 06/2023
ENA	284	01/2000	09/2022
		a 08/2022	a 08/2023

Tabela 4.1: Divisão das séries temporais em períodos de treino e de teste.

#### 4.2.2

#### Definição das distribuições de probabilidade a serem utilizadas

Após escolhidas as séries temporais a serem utilizadas nos experimentos, o próximo passo foi decidir quais distribuições seriam testadas. Foram escolhidas 2 distribuições não Gaussianas: lognormal e gama.

#### Distribuição LogNormal

Seja  $Z$  uma variável aleatória normal padrão,  $Z \sim N(0, 1)$  e  $\mu, \sigma$  dois números reais tal que  $\sigma > 0$ . Então, uma variável aleatória  $X = e^{\mu + \sigma Z}$  segue uma distribuição LogNormal, tal que  $E[\ln(X)] = \mu$  e  $V[\ln(X)] = \sigma^2$ .

Dado que o pacote *UnobservedComponentsGAS* já possui a distribuição normal implementada, ao invés de implementar a distribuição LogNormal, optou-se por estimar o GAS-CNO Normal no logaritmo da série temporal. Isto é, dado uma série temporal  $\mathbf{Y}_t$ , definiu-se:

$$p(\ln(y_t) | \mathbf{Y}_{t-1}) \sim \text{Normal}(\mu, \sigma^2)$$

É importante ressaltar que, após estimado o modelo na escala logarítmica, com o intuito de recuperar o *fit in sample* e a previsão para a escala original, não basta exponenciar os valores estimados. O correto é seguir a expressão:

$$\hat{y}_{original} = e^{\hat{y}_{log}} e^{\frac{1}{2}\hat{\sigma}^2}$$

onde  $\hat{y}_{log}$  são os valores estimados na escala logarítmica,  $SSR$  é a soma dos quadrados dos resíduos do modelo estimado (*sum of squared residuals*),  $T$  é o tamanho da série estimada,  $K$  são os graus de liberdade do modelo, que equivalhe à quantidade de parâmetros estimados e  $\hat{\sigma}^2 = \frac{SSR}{T-K}$  é um estimador da variância do modelo.

Por fim, vale definir o *score*, a matriz de informação de Fisher e o *score* padronizado para uma distribuição normal aplicada no log da série temporal. São eles:

$$\begin{aligned} \nabla_t &= \begin{pmatrix} \nabla_t^\mu \\ \nabla_t^{\sigma^2} \end{pmatrix} = \begin{pmatrix} \frac{\ln(y_t) - \mu_t}{\sigma_t^2} \\ \frac{-1}{2\sigma_t^2} + \frac{1}{2\sigma_t^4}(\ln(y_t) - \mu_t) \end{pmatrix} \\ I_{t|t-1} &= \begin{pmatrix} I_{t|t-1}^\mu & I_{t|t-1}^{\mu, \sigma^2} \\ I_{t|t-1}^{\mu, \sigma^2} & I_{t|t-1}^{\sigma^2} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{1}{2\sigma^4} \end{pmatrix} \\ s_t &= \begin{pmatrix} s_t^\mu \\ s_t^{\sigma^2} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{1}{2\sigma^4} \end{pmatrix}^{-d} \begin{pmatrix} \frac{\ln(y_t) - \mu_t}{\sigma_t^2} \\ \frac{-1}{2\sigma_t^2} + \frac{1}{2\sigma_t^4}(\ln(y_t) - \mu_t) \end{pmatrix} \end{aligned}$$

onde  $d$  é um parâmetro de escala que pode assumir valores 0, 0.5 ou 1.

## Distribuição Gama

Como a distribuição gama não estava originalmente implementada no pacote *UnobservedComponentsGAS*, sua implementação fez parte desse projeto. Para tal, escolheu-se a parametrização  $X \sim Gama(\alpha, \lambda)$ , tal que  $E[X] = \lambda$  e  $V[X] = \lambda^2/\alpha$ .

Dado que essa distribuição foi implementada dentro do escopo deste projeto, aqui estão expostos alguns resultados importantes para a distribuição gama em sua parametrização  $\alpha$  e  $\lambda$ . Somado a isso, o desenvolvimento completo dos resultados mostrados a seguir encontram-se no apêndice A.1

A função densidade de probabilidade e o logaritmo natural dessa função são:

$$f(y_t | \alpha_t, \lambda_t; \mathbf{Y}_{t-1}) = \frac{1}{\Gamma(\alpha_t)} \frac{1}{(\alpha_t^{-1} \lambda_t)^{\alpha_t}} y_t^{\alpha_t - 1} e^{-\frac{\alpha_t}{\lambda_t} y_t}$$

$$\begin{aligned} \ln f(y_t | \mathbf{Y}_{t-1}; \alpha_t, \lambda_t) &= -\ln(\Gamma(\alpha_t)) - \alpha_t \ln(1/\alpha_t) - \alpha_t \ln(\lambda_t) + \\ &\quad + (\alpha_t + 1) \ln(y_t) - (\alpha_t / \lambda_t) y_t \end{aligned}$$

O vetor score obtido para essa distribuição é:

$$\nabla_t = \begin{pmatrix} \nabla_t^\alpha \\ \nabla_t^\lambda \end{pmatrix} = \begin{pmatrix} \ln(y_t) - \frac{y_t}{\lambda_t} + \ln(\alpha_t) - \psi_1(\alpha_t) - \ln(\lambda_t) + 1 \\ \frac{\alpha_t}{\lambda_t} \left( \frac{y_t}{\lambda_t} - 1 \right) \end{pmatrix}$$

onde a função  $\psi_1(\alpha_t)$  é a função *digamma* definida como  $\psi_1(\alpha_t) = \frac{\Gamma'(\alpha_t)}{\Gamma(\alpha_t)}$ . Em seguida, é necessário definir a matrix de informação de Fisher  $I_{t|t-1}$ .

$$I_{t|t-1} = \begin{pmatrix} I_{t|t-1}^{\alpha, \alpha} & I_{t|t-1}^{\alpha, \lambda} \\ I_{t|t-1}^{\lambda, \alpha} & I_{t|t-1}^{\lambda, \lambda} \end{pmatrix} = \begin{pmatrix} \psi_2(\alpha_t) - \frac{1}{\alpha_t} & 0 \\ 0 & \frac{\alpha_t}{\lambda_t^2} \end{pmatrix}$$

onde a função  $\psi_2(\alpha_t)$  é a função *trigamma* definida como  $\psi_2(\alpha_t) = \frac{\psi'_1(\alpha_t)}{\psi_1(\alpha_t)}$ .

É essa matriz que, combinada com o *score*, dará origem ao *score* padronizado, que é a estrutura responsável por guiar as dinâmicas de um modelo *score-driven*. Sua expressão para a distribuição gama é:

$$\begin{aligned} s_t &= I_{t|t-1}^{-d} \nabla_t \\ s_t &= \begin{pmatrix} s_t^\alpha \\ s_t^\lambda \end{pmatrix} = \begin{pmatrix} \psi_2(\alpha_t) - \frac{1}{\alpha_t} & 0 \\ 0 & \frac{\alpha_t}{\lambda_t^2} \end{pmatrix}^{-d} \begin{pmatrix} \ln(y_t) - \frac{y_t}{\lambda_t} + \ln(\alpha_t) - \psi_1(\alpha_t) - \ln(\lambda_t) + 1 \\ \frac{\alpha_t}{\lambda_t} \left( \frac{y_t}{\lambda_t} - 1 \right) \end{pmatrix} \end{aligned}$$

onde  $d$  é um parâmetro de escala que pode assumir valores 0, 0.5 ou 1.

### 4.2.3

#### Restrições no domínio dos parâmetros variantes no tempo

Embora o ferramental de funções de ligação, apresentados na sessão 4.1 seja muito utilizado em modelos *score-driven*, o pacote *UnobservedComponentsGAS*, utilizado como base deste trabalho, não as utiliza para garantir as restrições de domínio dos parâmetros variantes no tempo. Ele, uma vez implementado inteiramente dentro do ambiente de otimização JuMP, da linguagem Julia, adiciona restrições de positividade diretamente no problema de otimização que estima o modelo GAS. Dito isso, está subentendido que toda a

formulação dos modelos GAS neste projeto está utilizando a função de ligação identidade, não sendo necessário especificar novos valores de  $\tilde{\nabla}_t$ ,  $\tilde{s}_t$  e  $\tilde{I}_{t|t-1}$ .

#### 4.2.4

#### Questões de implementação

Além de não utilizar funções de ligação, duas outras decisões foram tomadas com o intuito de facilitar a estimação dos modelos GAS, utilizando o pacote *UnobservedComponentsGAS*.

A primeira delas é uma funcionalidade do próprio pacote, que possibilita penalização nos valores dos  $\kappa$ s de todas as componentes do modelo. Essa penalização é feita por meio de um hiper-parâmetro  $\alpha \in [0, 1]$  que, quando assume o valor 0, indica que não há penalização nos  $\kappa$ s. Foram testados diversos valores desse hiper-parâmetro durante a implementação dos testes.

O segundo comentário a ser feito é que todos os modelos foram estimados com a série normalizada, a fim de evitar problemas numéricos que surgiram durante o desenvolvimento do projeto. A normalização inicialmente utilizada foi a denominada *MinMaxScale*, que possui o objetivo de colocar a série temporal original dentro do intervalo de zero e um. Essa normalização pode ser expressa como:

$$y_{norm} = \frac{y - min(y)}{max(y) - min(y)}$$

onde  $y$  é uma série temporal qualquer,  $min(y)$  é o valor mínimo da série,  $max(y)$  é seu valor máximo e  $y_{norm}$  é a série normalizada entre zero e um.

No entanto, foi observado que a escala da série no momento da estimação alterava significativamente os resultados, de tal modo que, ao invés de normalizar a série para a escala de 0 a 1, generalizei a transformação para qualquer escala. Para isso, basta fazer:

$$y_{norm} = min_{val} + \frac{(max_{val} - min_{val})(y - min(y))}{(max(y) - min(y))}$$

onde  $min_{val}$  e  $max_{val}$  são os limites da escala na qual se deseja colocar a série temporal.

Para evitar problemas durante a estimação de modelos com restrição de positividade nos parâmetros variantes no tempo, foi sempre escolhida uma escala cujo valor mínimo fosse maior que zero.

#### 4.2.5

##### Definição das componentes não observáveis

O próximo passo na definição da metodologia do projeto é especificar as componentes não observáveis dos modelos GAS a serem implementados e testados. Embora o pacote *UnobservedComponentsGAS* forneça diferentes possibilidades de especificação de componentes, por fins de simplicidade, o modelo GAS a ser utilizado será um modelo com tendência e sazonalidade estocásticas com apenas 1 parâmetro variando no tempo. Dito isso, para as distribuições listadas anteriormente, optei pelos seguintes modelos:

$$\begin{aligned} p(y_t | \mu_{t|t-1}, \sigma^2; \mathbf{Y}_{t-1}) &\sim \text{Lognormal}(\mu_{t|t-1}, \sigma^2) \\ p(y_t | \alpha, \lambda_{t|t-1}; \mathbf{Y}_{t-1}) &\sim \text{Gama}(\alpha, \lambda_{t|t-1}) \end{aligned}$$

A escolha de  $\mu$  e  $\lambda$  como parâmetros variantes no tempo se deu pelo fato deles representarem as médias de suas respectivas distribuições.

Uma vez definidos os parâmetros variantes no tempo, define-se a dinâmica de atualização desses parâmetros. Essa escolha depende principalmente do comportamento da série temporal.

A seguir está descrito o modelo para a média  $\mu_t$  variante no tempo. Note que é exatamente o mesmo que seria definido para média  $\lambda_t$ .

Para a série de carga, dado seu perfil não estacionário, o parâmetro de média será modelado com um passeio aleatório amortecido com *drift* como componentes de tendência e com sazonalidade trigonométrica. As equações abaixo descrevem este modelo:

$$\begin{aligned} \mu_{t+1|t} &= m_{t+1|t} + \gamma_{t+1|t} & (4-28) \\ m_{t+1|t} &= m_{t|t-1} + b_{t|t-1} + \kappa_m s_t \\ b_{t+1|t} &= \phi b_{t|t-1} + \kappa_b s_t \\ \gamma_{t+1|t} &= \sum_{i=1}^{S/2} \gamma_{i,t} \\ \gamma_{i,t} &= \gamma_{i,t-1} \cos(\omega_{i,t}) + \gamma_{i,t-1}^* \sin(\omega_{i,t}) + \kappa_S s_t \\ \gamma_{i,t}^* &= -\gamma_{i,t-1} \sin(\omega_{i,t}) + \gamma_{i,t-1}^* \cos(\omega_{i,t}) + \kappa_S s_t \end{aligned}$$

onde  $S$  é o período sazonal da série (12 para séries mensais),  $s_t$  é o vetor *score*,  $m_{t+1|t}$  é a componente de tendência,  $\gamma_{t+1|t}$  a componente de sazonalidade,  $\omega_{i,t} = 2\pi i t / S$  e  $s_t = I_{t|t-1}^{-d} \nabla_t$  é o *score* padronizado com função de ligação identidade.

A série de ENA, por sua vez, não apresenta tendência, o que torna o modelo acima inadequado. Sendo assim, será mantida a componente sazonal via funções trigonométricas, mas a componente de tendência será descrita por meio de um processo  $AR(p)$ , com  $p \leq 2$ . As equações abaixo descrevem este modelo:

$$\begin{aligned}\mu_{t+1|t} &= m_{t+1|t} + \gamma_{t+1|t} && (4-29) \\ m_{t+1|t} &= \phi_0 + \sum_{i=1}^p \phi_i m_{t|t-i} + \kappa_m s_t \\ \gamma_{t+1|t} &= \sum_{i=1}^{S/2} \gamma_{i,t} \\ \gamma_{i,t} &= \gamma_{i,t-1} \cos(\omega_{i,t}) + \gamma_{i,t-1}^* \sin(\omega_{i,t}) + \kappa_S s_t \\ \gamma_{i,t}^* &= -\gamma_{i,t-1} \sin(\omega_{i,t}) + \gamma_{i,t-1}^* \cos(\omega_{i,t}) + \kappa_S s_t\end{aligned}$$

Todos os elementos dessas equações são os mesmos do modelo para série de carga.

#### 4.2.6

#### Diagnósticos dos Resíduos

Parte essencial da modelagem de séries temporais é realizar diagnósticos nos resíduos do modelo estimado, com o intuito e verificar a qualidade da estimação. De acordo com [Hyndman e Athanasopoulos 2021], os resíduos de um bom modelo de série temporal devem apresentar como principal característica ausência de autocorrelação. Além disso, também é desejável que eles sejam homocedásticos e que sejam oriundos de uma distribuição normal.

Como explicado na sessão 4.1, os resíduos mais adequados para avaliar modelos GAS são os resíduos quantílicos.

Para compor o diagnóstico dos resíduos e verificar se as características desejadas foram obtidas, é comum realizar testes de hipóteses. Em especial, os testes de Jarque-Bera, que avalia normalidade dos resíduos, o teste de Ljung-Box, que avalia presença de autocorrelação nos resíduos e o teste ARCH, que investigar presença de heterocedasticidade. As hipóteses nulas destes 3 testes são:

- Teste JarqueBera:  $H_0$  supõe resíduos normalmente distribuídos
- Teste ARHLM:  $H_0$  supõe resíduos homocedásticos
- Teste LjungBox:  $H_0$  supõe resíduos descorrelatados

Para fins de metodologia, será utilizado nível de significância  $\alpha = 5\%$  para esses testes. Além disso, vale comentar que o tamanho da amostra dos

testes pode influenciar fortemente a rejeição da hipótese nula do teste, de tal forma que um resultado, mesmo que seja estatisticamente significante, pode não ser relevante.

Para complementar o teste de hipótese para verificar a normalidade dos resíduos, é comum utilizar o QQ-Plot (*quantile-quantile plot*). Ele permite comparar os quantis dos nossos dados com os quantis de uma distribuição de probabilidade qualquer. Para fins de diagnóstico de resíduos, comparamos os quantis dos resíduos com os quantis de uma distribuição normal padrão (média 0 e desvio padrão 1). Por fim, é essencial avaliar a função de autocorrelação (FAC) dos resíduos, para verificar se toda a dependência linear da série temporal foi capturada pelo modelo estimado.

#### 4.2.7

##### Métrica de avaliação

Por fim, a última etapa da avaliação do modelo é o cálculo de uma métrica de aderência dos valores estimados ou previstos pelo modelo com os valores reais da série. Muitas são essas métricas de erro: erro quadrático médio (*MSE*), raiz quadrada do erro quadrático médio (*RMSE*), erro absoluto médio (*MAE*), dentre outros.

A métrica escolhida para avaliar a estimação e previsão do modelo neste trabalho é o erro médio percentual absoluto, ou *MAPE*, em inglês. Ele foi escolhido por sua interpretabilidade, uma vez que trata de um erro percentual. O MAPE é definido como:

$$MAPE = 100 \times \frac{1}{T} \sum_{t=1}^T \left| \frac{y_t - \hat{y}_{t|t-1}}{y_t} \right|, \quad y_t \neq 0$$

onde  $T$  é o tamanho da série,  $y_t$  é a série temporal e  $\hat{y}_t$  é o valor estimado ou previsto pelo modelo.

Esta métrica será utilizada em dois contextos: para avaliar a qualidade da estimação *in sample* e a qualidade da previsão *out of sample*.

#### 4.2.8

##### Combinação não linear de tendência e sazonalidade

Aqui estão apresentados os modelos GAS-CNO com combinação não linear de componentes de tendência e sazonalidade implementados para cada uma das séries. Primeiramente, o que será chamado de "modelo multiplicativo 1", foi definido como a simples multiplicação das duas componentes. Sendo assim, para a série de carga, o modelo a ser implementado é:

$$\begin{aligned}
\mu_{t+1|t} &= m_{t+1|t} \times \gamma_{t+1|t} & (4-30) \\
m_{t+1|t} &= m_{t|t-1} + b_{t|t-1} + \kappa_m s_t \\
b_{t+1|t} &= \phi b_{t|t-1} + \kappa_b s_t \\
\gamma_{t+1|t} &= \sum_{i=1}^{S/2} \gamma_{i,t} \\
\gamma_{i,t} &= \gamma_{i,t-1} \cos(\omega_{i,t}) + \gamma_{i,t-1}^* \sin(\omega_{i,t}) + \kappa_S s_t \\
\gamma_{i,t}^* &= -\gamma_{i,t-1} \sin(\omega_{i,t}) + \gamma_{i,t-1}^* \cos(\omega_{i,t}) + \kappa_S s_t
\end{aligned}$$

onde  $S$  é o período sazonal da série (12 para séries mensais),  $s_t$  é o vetor score,  $m_{t+1|t}$  é a componente de tendência,  $\gamma_{t+1|t}$  a componente de sazonalidade e  $\omega_{i,t} = 2\pi i t / S$  e  $s_t = I_{t|t-1}^{-d} \nabla_t$ .

Já para a série de ENA:

$$\begin{aligned}
\mu_{t+1|t} &= m_{t+1|t} \times \gamma_{t+1|t} & (4-31) \\
m_{t+1|t} &= \phi_0 + \sum_{i=1}^p \phi_i m_{t|t-i} + \kappa_m s_t \\
\gamma_{t+1|t} &= \sum_{i=1}^{S/2} \gamma_{i,t} \\
\gamma_{i,t} &= \gamma_{i,t-1} \cos(\omega_{i,t}) + \gamma_{i,t-1}^* \sin(\omega_{i,t}) + \kappa_S s_t \\
\gamma_{i,t}^* &= -\gamma_{i,t-1} \sin(\omega_{i,t}) + \gamma_{i,t-1}^* \cos(\omega_{i,t}) + \kappa_S s_t
\end{aligned}$$

Somado a ele, o modelo denominado "multiplicativo 2" é uma pequena alteração, sendo definido, para a série de carga, como:

$$\begin{aligned}
\mu_{t+1|t} &= m_{t+1|t} \times (1 + \gamma_{t+1|t}) & (4-32) \\
m_{t+1|t} &= m_{t|t-1} + b_{t|t-1} + \kappa_m s_t \\
b_{t+1|t} &= \phi b_{t|t-1} + \kappa_b s_t \\
\gamma_{t+1|t} &= \sum_{i=1}^{S/2} \gamma_{i,t} \\
\gamma_{i,t} &= \gamma_{i,t-1} \cos(\omega_{i,t}) + \gamma_{i,t-1}^* \sin(\omega_{i,t}) + \kappa_S s_t \\
\gamma_{i,t}^* &= -\gamma_{i,t-1} \sin(\omega_{i,t}) + \gamma_{i,t-1}^* \cos(\omega_{i,t}) + \kappa_S s_t
\end{aligned}$$

e, para a série de ENA, como:

$$\begin{aligned}
 \mu_{t+1|t} &= m_{t+1|t} \times (1 + \gamma_{t+1|t}) & (4-33) \\
 m_{t+1|t} &= \phi_0 + \sum_{i=1}^p \phi_i m_{t|t-i} + \kappa_m s_t \\
 \gamma_{t+1|t} &= \sum_{i=1}^{S/2} \gamma_{i,t} \\
 \gamma_{i,t} &= \gamma_{i,t-1} \cos(\omega_{i,t}) + \gamma_{i,t-1}^* \sin(\omega_{i,t}) + \kappa_S s_t \\
 \gamma_{i,t}^* &= -\gamma_{i,t-1} \sin(\omega_{i,t}) + \gamma_{i,t-1}^* \cos(\omega_{i,t}) + \kappa_S s_t
 \end{aligned}$$

Vale relembrar que, para todos esses modelos, o *score padronizado* é definido como

$$s_t = I_{t|t-1}^{-d} \nabla_t$$

Para a distribuição gama, temos:

$$s_t = \begin{pmatrix} \psi_2(\alpha_t) - \frac{1}{\alpha_t} & 0 \\ 0 & \frac{\alpha_t}{\lambda_t^2} \end{pmatrix}^{-d} \begin{pmatrix} \ln(y_t) - \frac{y_t}{\lambda_t} + \ln(\alpha_t) - \psi_1(\alpha_t) - \ln(\lambda_t) + 1 \\ \frac{\alpha_t}{\lambda_t} \left( \frac{y_t}{\lambda_t} - 1 \right) \end{pmatrix}$$

Já para a distribuição normal no log da série, temos

$$s_t = \begin{pmatrix} \frac{1}{\sigma_t^2} & 0 \\ 0 & \frac{1}{2\sigma_t^4} \end{pmatrix}^{-d} \begin{pmatrix} \frac{\ln(y_t) - \mu_t}{\sigma_t^2} \\ \frac{-1}{2\sigma_t^2} + \frac{1}{2\sigma_t^4} (\ln(y_t) - \mu_t) \end{pmatrix}$$

## **5**

## **Resultados**

Neste capítulo estão apresentados todos os resultados das alterações descritas nas seções do capítulo 4.

A apresentação dos resultados segue o seguinte padrão: para cada série temporal, estão apresentados os gráficos comparativos do modelo GAS-CNO aditivo convencional para as distribuições log-normal e gama, seguido dos gráficos de cada modelo multiplicativo, para cada distribuição e para cada valor de  $d$ . Os gráficos apresentados são: estimativa no período de treino (*fit in sample*), FAC dos resíduos quantílicos e previsão 12 passos à frente fora da amostra. Além disso, também há uma tabela com o resultado dos testes de hipóteses e com os MAPEs dentro e fora da amostra.

Todos os comentários e conclusões encontram-se sempre ao final das sessões.

### **5.1**

#### **Série de carga**

### 5.1.1

#### Carga - GAS-CNO Multiplicativo 1

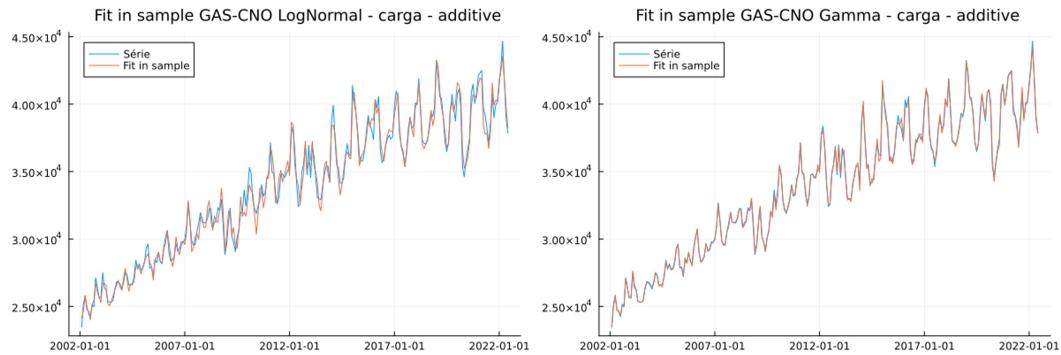


Figura 5.1: Período de treino série de carga GAS-CNO aditivo.

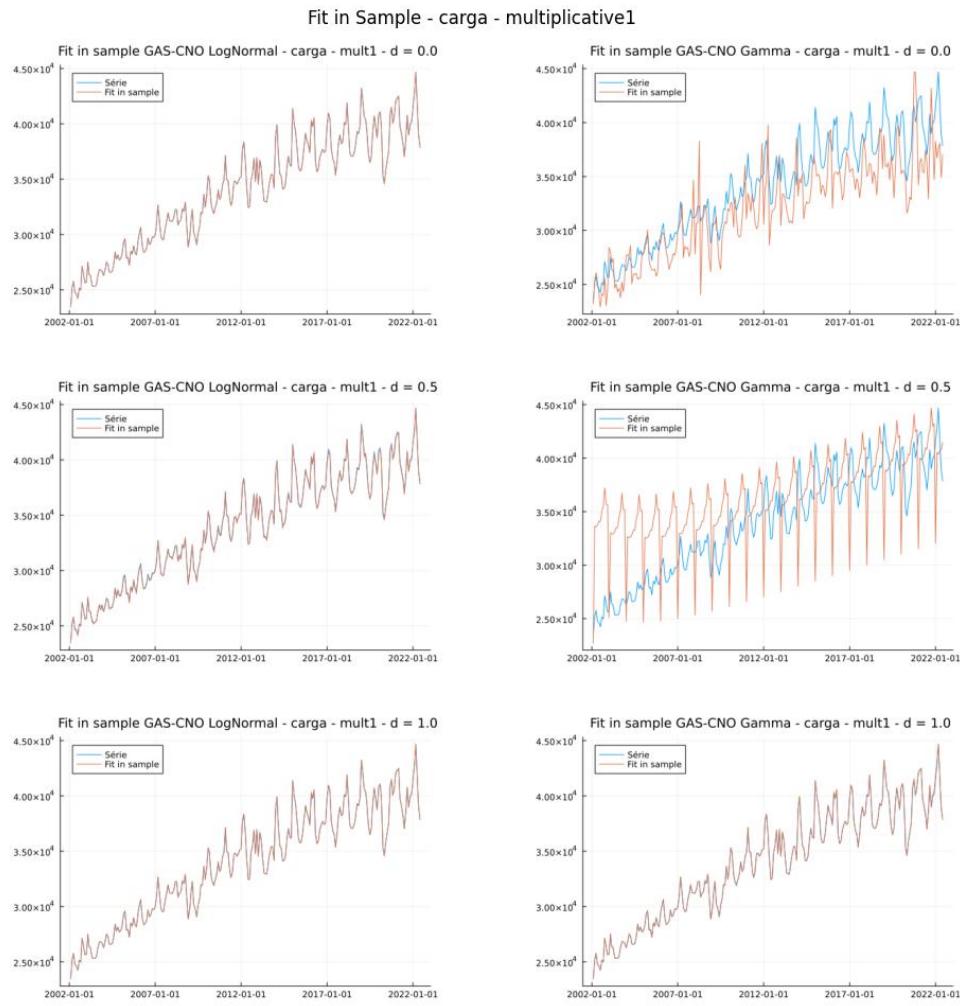


Figura 5.2: Período de treino série de carga GAS-CNO multiplicativo 1.

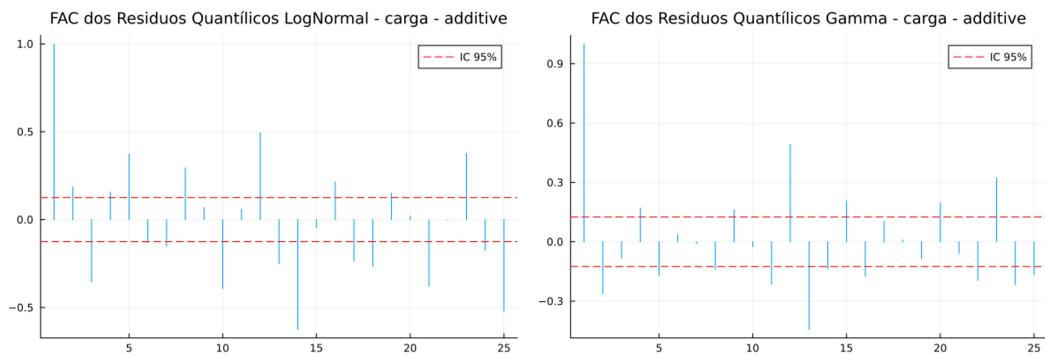


Figura 5.3: FAC dos resíduos quantílicos série de carga GAS-CNO aditivo.

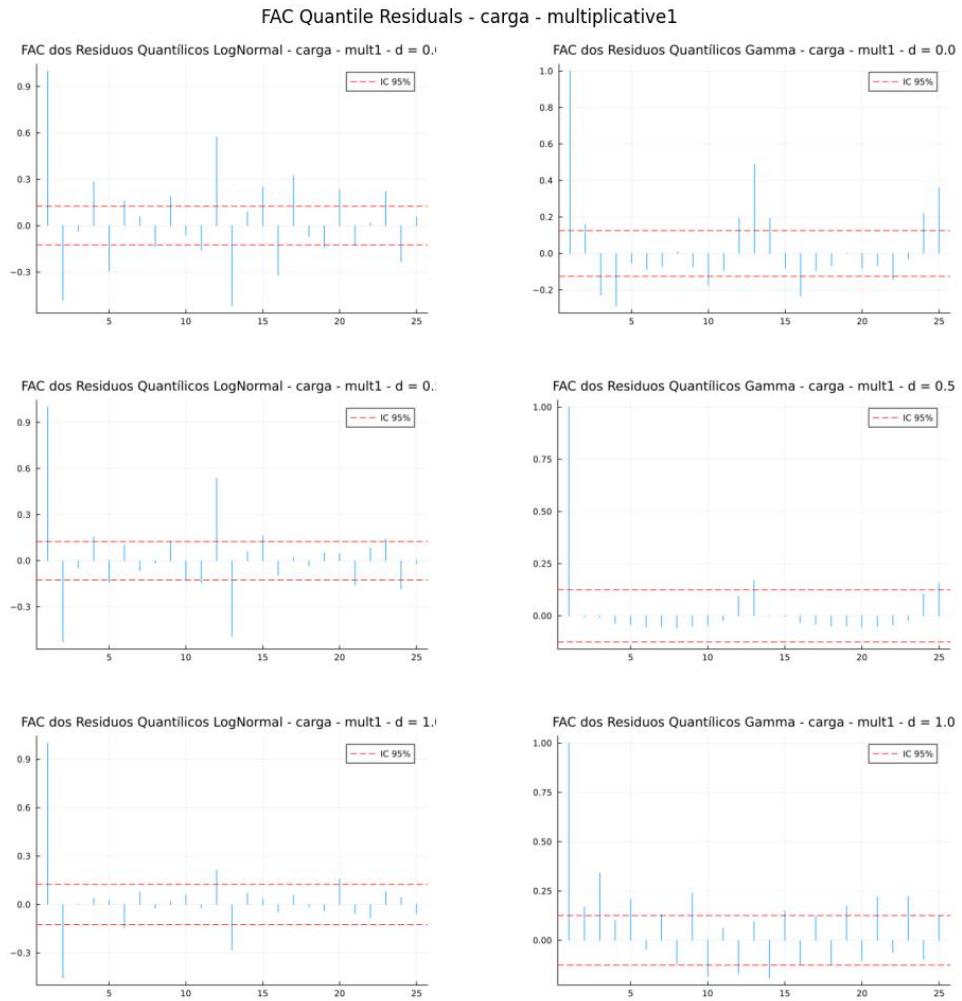


Figura 5.4: FAC dos resíduos quantílicos série de carga GAS-CNO multiplicativo 1.

<b>d</b>	<b>Distribuição</b>	<b>pvalor</b>	<b>Rejeição</b>
0.0	LogNormal	0.1229	Não rejeita $H_0$
0.0	Gamma	0.0	Rejeita $H_0$
0.5	LogNormal	0.0076	Rejeita $H_0$
0.5	Gamma	0.0	Rejeita $H_0$
1.0	LogNormal	0.0	Rejeita $H_0$
1.0	Gamma	0.0	Rejeita $H_0$

Tabela 5.1: Resultados teste Jarque-Bera para série de carga GAS-CNO multiplicativo 1.

<b>d</b>	<b>Distribuição</b>	<b>pvalor</b>	<b>Rejeição</b>
0.0	LogNormal	0.0	Rejeita $H_0$
0.0	Gamma	0.0001	Rejeita $H_0$
0.5	LogNormal	0.0	Rejeita $H_0$
0.5	Gamma	0.0	Rejeita $H_0$
1.0	LogNormal	0.0	Rejeita $H_0$
1.0	Gamma	0.0	Rejeita $H_0$

Tabela 5.2: Resultados teste ARCH para série de carga GAS-CNO multiplicativo 1.

<b>d</b>	<b>Distribuição</b>	<b>pvalor</b>	<b>Rejeição</b>
0.0	LogNormal	0.0	Rejeita $H_0$
0.0	Gamma	0.0	Rejeita $H_0$
0.5	LogNormal	0.0	Rejeita $H_0$
0.5	Gamma	0.0002	Rejeita $H_0$
1.0	LogNormal	0.0	Rejeita $H_0$
1.0	Gamma	0.0	Rejeita $H_0$

Tabela 5.3: Resultados teste Ljung-Box para série de carga GAS-CNO multiplicativo 1.

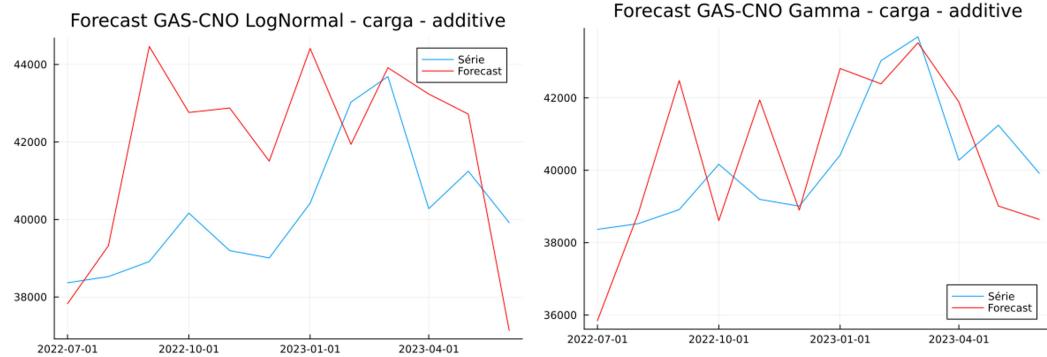


Figura 5.5: Previsão 12 passos à frente para série de carga GAS-CNO aditivo.

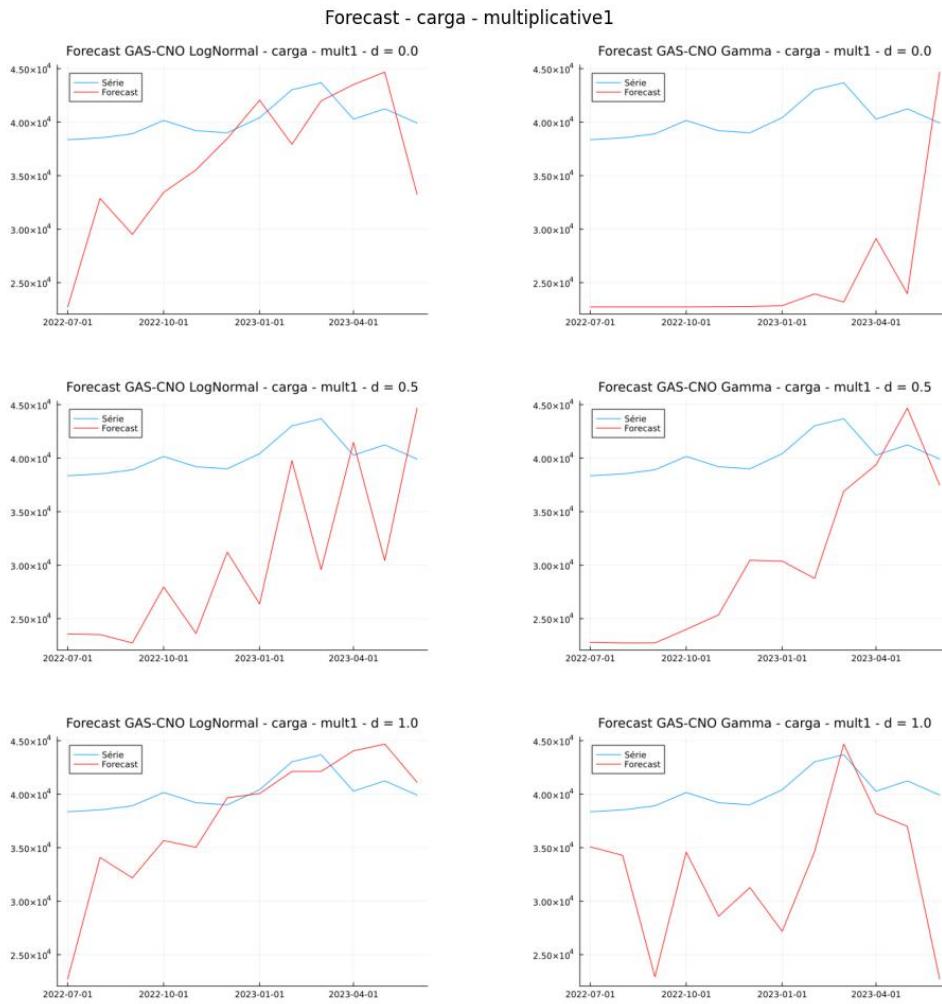


Figura 5.6: Previsão 12 passos à frente para série de carga GAS-CNO multiplicativo 1.

O primeiro comentário a ser feito é que o valor de  $d$  impacta fortemente a estimativa da previsão do modelo quando a distribuição escolhida é a gama, de tal forma que, para  $d = 0.0$  e  $d = 0.5$  o resultado obtido não foi razoável. Ao analisar a Figura 5.4, nota-se que mesmo o GAS-CNO aditivo não gerou resíduos quantílicos com a FAC desprezível e, além disso, o único modelo que

foi capaz de diminuir a magnitude dos *lags* na FAC foi o modelo log-normal com  $d = 1$ . Quando analisamos as tabelas 5.1 a 5.3, notamos que todos os testes são fortemente rejeitados, o que pode indicar má especificação do modelo a partir da combinação multiplicativa 1. Por fim, as previsões fora da amostra também se mostraram muito ruins para todos os modelos multiplicativos 1.

### 5.1.2

#### Carga - GAS-CNO Multiplicativo 2

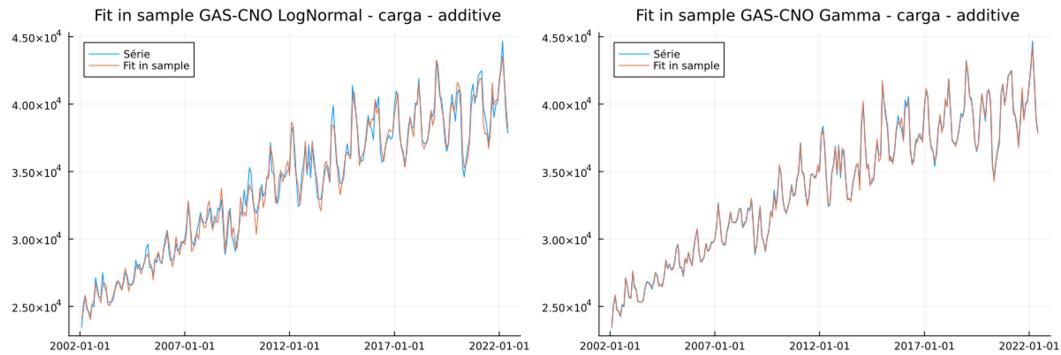


Figura 5.7: Período de treino série de carga GAS-CNO aditivo.

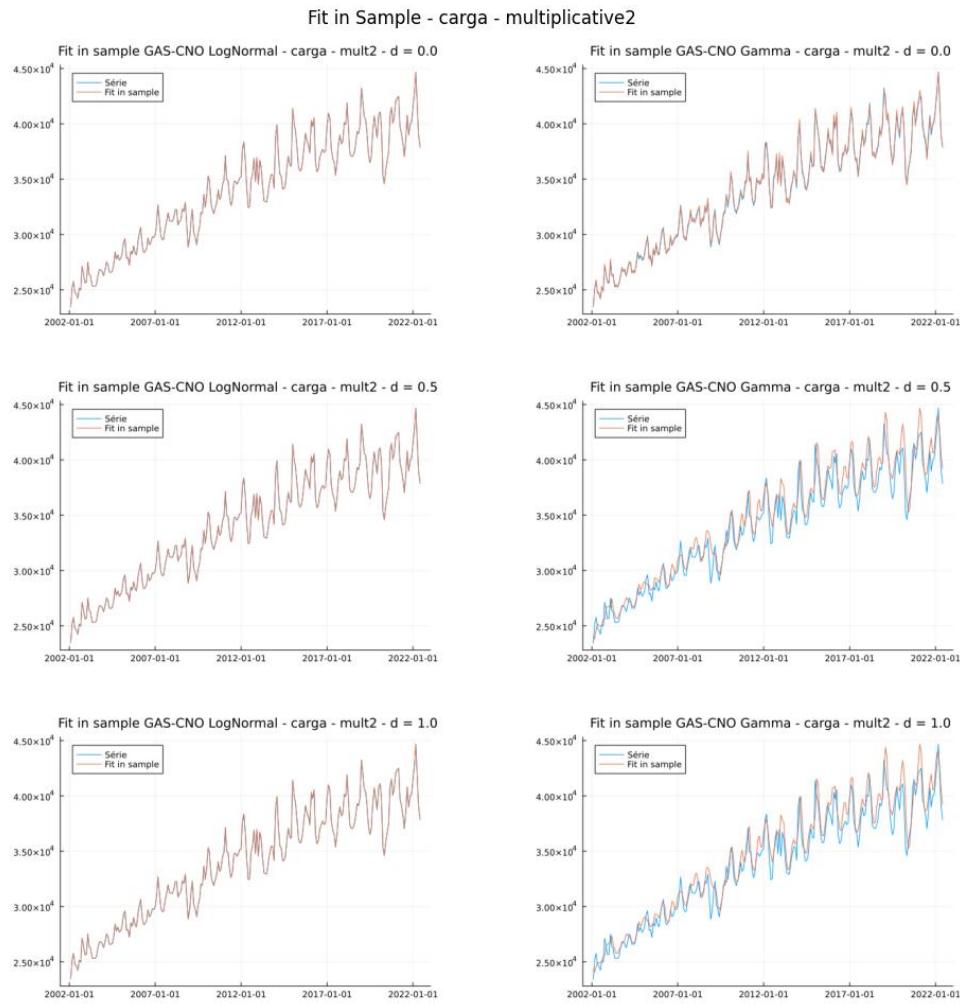


Figura 5.8: Período de treino série de carga GAS-CNO multiplicativo 2.

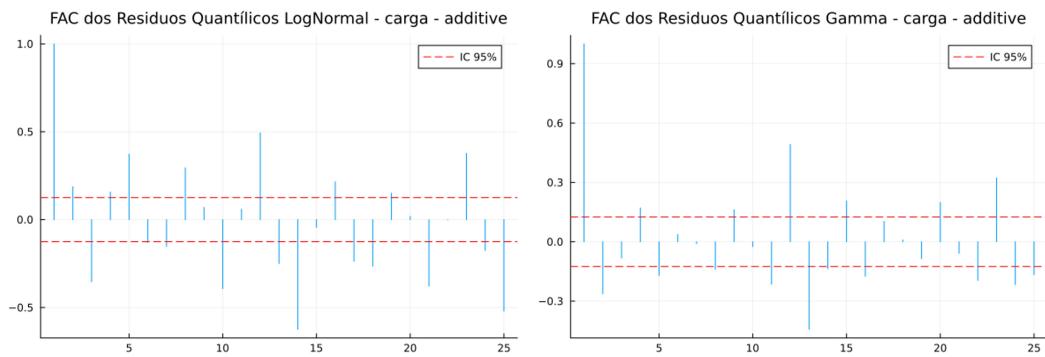


Figura 5.9: FAC dos resíduos quantílicos série de carga GAS-CNO aditivo.

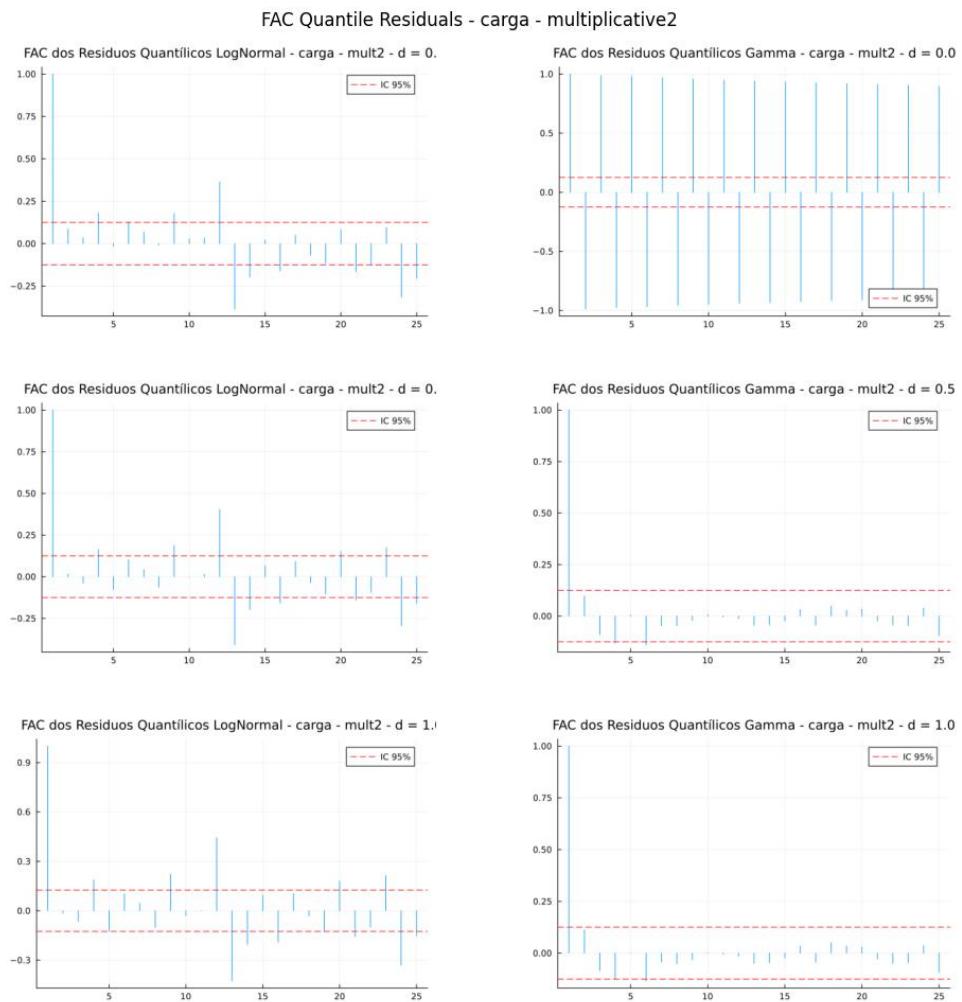


Figura 5.10: FAC dos resíduos quantílicos série de carga GAS-CNO multiplicativo 2.

<b>d</b>	<b>Distribuição</b>	<b>pvalor</b>	<b>Rejeição</b>
0.0	LogNormal	0.5951	Não rejeita $H_0$
0.0	Gamma	0.0	Rejeita $H_0$
0.5	LogNormal	0.6039	Não rejeita $H_0$
0.5	Gamma	0.0	Rejeita $H_0$
1.0	LogNormal	0.5943	Não rejeita $H_0$
1.0	Gamma	0.0	Rejeita $H_0$

Tabela 5.4: Resultados teste Jarque-Bera para série de carga GAS-CNO multiplicativo 2.

<b>d</b>	<b>Distribuição</b>	<b>pvalor</b>	<b>Rejeição</b>
0.0	LogNormal	0.0938	Não rejeita $H_0$
0.0	Gamma	0.0	Rejeita $H_0$
0.5	LogNormal	0.0537	Não rejeita $H_0$
0.5	Gamma	0.0146	Rejeita $H_0$
1.0	LogNormal	0.0021	Rejeita $H_0$
1.0	Gamma	0.0169	Rejeita $H_0$

Tabela 5.5: Resultados teste ARCH para série de carga GAS-CNO multiplicativo 2.

<b>d</b>	<b>Distribuição</b>	<b>pvalor</b>	<b>Rejeição</b>
0.0	LogNormal	0.0	Rejeita $H_0$
0.0	Gamma	0.0	Rejeita $H_0$
0.5	LogNormal	0.0	Rejeita $H_0$
0.5	Gamma	0.0045	Rejeita $H_0$
1.0	LogNormal	0.0	Rejeita $H_0$
1.0	Gamma	0.004	Rejeita $H_0$

Tabela 5.6: Resultados teste Ljung-Box para série de carga GAS-CNO multiplicativo 2.

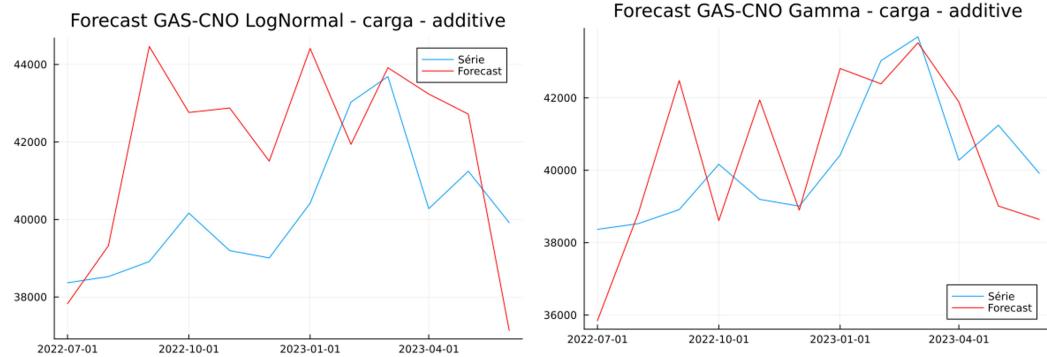


Figura 5.11: Previsão 12 passos à frente para série de carga GAS-CNO aditivo.

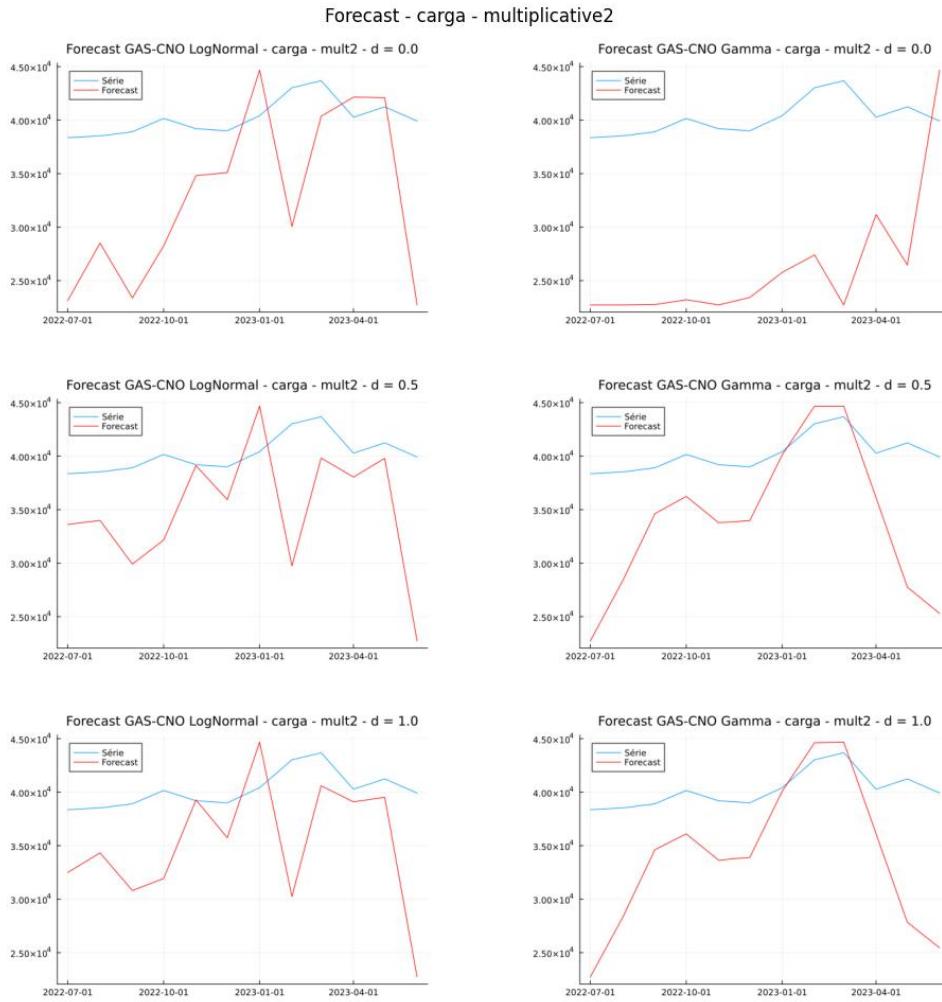


Figura 5.12: Previsão 12 passos à frente para série de carga GAS-CNO multiplicativo 2.

Novamente, analisando a Figura 5.8 podemos notar que os modelos gama se mostraram sensíveis à escolha de  $d$ , enquanto que os modelos log-normais ficaram mais consistentes. Interessante perceber o caso do modelo gama com  $d = 0.0$ , o qual, via análise gráfica da Figura 5.8 parece ser o de melhor estimação mas, ao analisar a FAC de seus resíduos na Figura 5.10, nota-se

que ela é a pior das 6. Mais uma vez, todas as FAC dos modelos log-normais indicam forte dependência no *lag* sazonal e todas as previsões fora da amostra ficaram muito ruins.

### 5.1.3 MAPES

Combinacao	d	Distribuicao	MAPE Treino (%)	MAPE Teste (%)
aditivo	-	LogNormal	1.41	5.9
aditivo	-	Gamma	0.47	4.0
multiplicativo1	0.0	LogNormal	0.03	13.33
multiplicativo1	0.0	Gamma	6.83	38.91
multiplicativo1	0.5	LogNormal	0.3	27.09
multiplicativo1	0.5	Gamma	12.66	25.92
multiplicativo1	1.0	LogNormal	0.01	10.04
multiplicativo1	1.0	Gamma	0.07	19.56
multiplicativo2	0.0	LogNormal	0.0	21.23
multiplicativo2	0.0	Gamma	0.75	36.54
multiplicativo2	0.5	LogNormal	0.01	14.81
multiplicativo2	0.5	Gamma	2.63	16.77
multiplicativo2	1.0	LogNormal	0.05	14.47
multiplicativo2	1.0	Gamma	2.6	16.8

Tabela 5.7: MAPEs dentro e fora da amostra para série de carga.

Analisar a Tabela 5.7 corrobora com alguns comentários feitos anteriormente. Primeiramente, nenhum modelo multiplicativo conseguiu superar o modelo aditivo convencional em termos de MAPE dentro e fora da amostra. Por outro lado, vários foram aqueles que superaram o modelo convencional ao olhar apenas para o MAPE da estimação. Ao olhar apenas para os erros fora da amostra, a conclusão da análise visual torna-se ainda mais forte: nenhum modelo multiplicativo foi capaz de gerar boas previsões, ainda mais de gerar previsões melhores que o modelo aditivo.

## 5.2 Série de ENA

### 5.2.1 ENA - GAS-CNO Multiplicativo 1

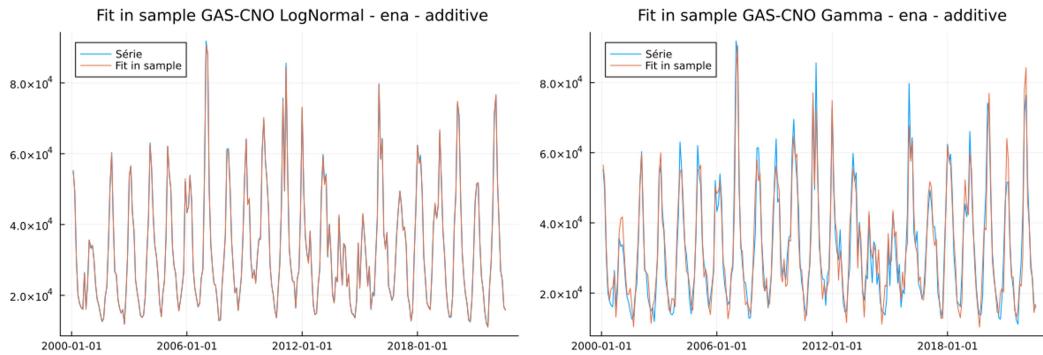


Figura 5.13: Período de treino série de ENA GAS-CNO aditivo.

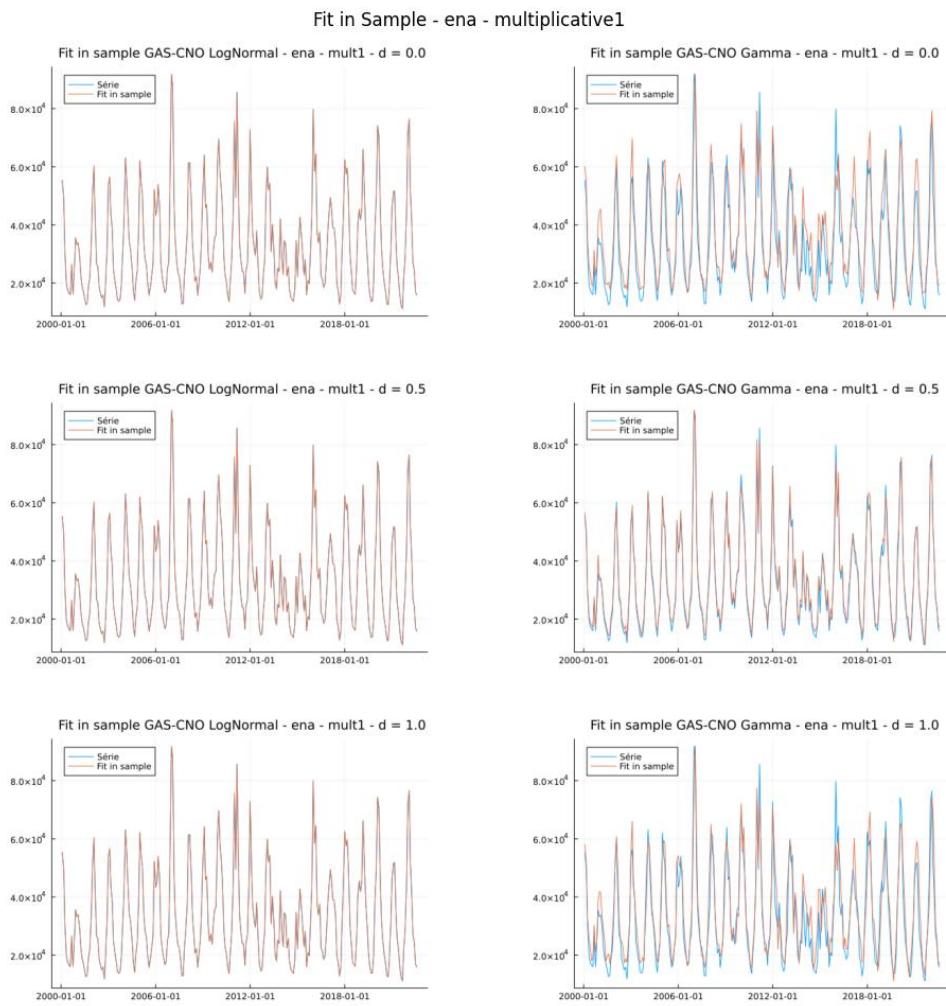


Figura 5.14: Período de treino série de ENA GAS-CNO multiplicativo 1.

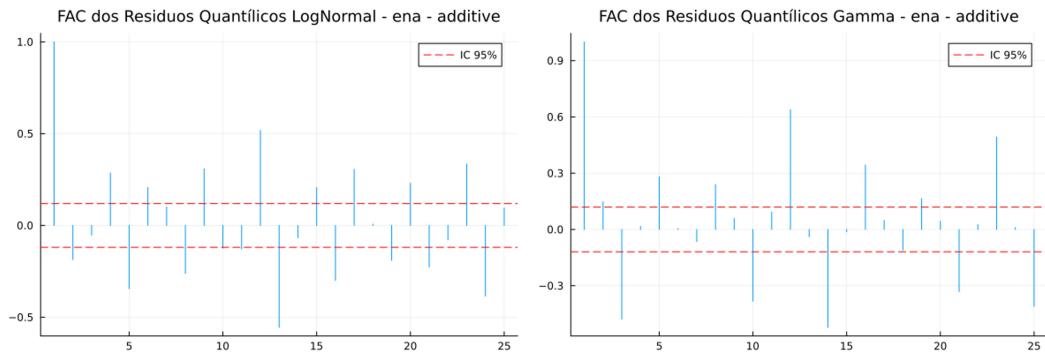


Figura 5.15: FAC dos resíduos quantílicos série de ENA GAS-CNO aditivo.

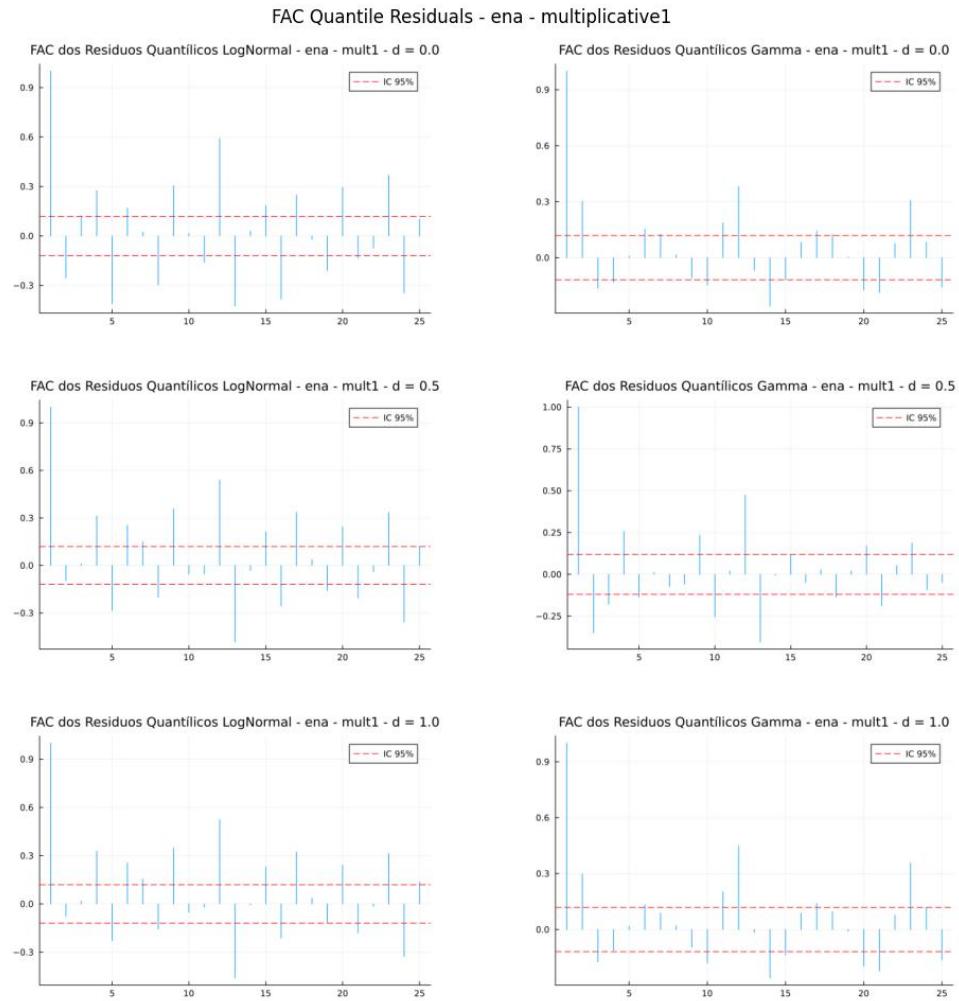


Figura 5.16: FAC dos resíduos quantílicos série de ENA GAS-CNO multiplicativo 1.

d	Distribuição	pvalor	Rejeição
0.0	LogNormal	0.0	Rejeita $H_0$
0.0	Gamma	0.0	Rejeita $H_0$
0.5	LogNormal	0.623	Não rejeita $H_0$
0.5	Gamma	0.0	Rejeita $H_0$
1.0	LogNormal	0.8038	Não rejeita $H_0$
1.0	Gamma	0.0013	Rejeita $H_0$

Tabela 5.8: Resultados teste de normalidade Jarque-Bera para série de ENA GAS-CNO multiplicativo 1.

d	Distribuição	pvalor	Rejeição
0.0	LogNormal	0.0	Rejeita $H_0$
0.0	Gamma	0.0414	Rejeita $H_0$
0.5	LogNormal	0.0	Rejeita $H_0$
0.5	Gamma	0.0	Rejeita $H_0$
1.0	LogNormal	0.0014	Rejeita $H_0$
1.0	Gamma	0.0003	Rejeita $H_0$

Tabela 5.9: Resultados teste de heterocedasticidade ARCH para série de ENA GAS-CNO multiplicativo 1.

d	Distribuição	pvalor	Rejeição
0.0	LogNormal	0.0	Rejeita $H_0$
0.0	Gamma	0.0	Rejeita $H_0$
0.5	LogNormal	0.0	Rejeita $H_0$
0.5	Gamma	0.0	Rejeita $H_0$
1.0	LogNormal	0.0	Rejeita $H_0$
1.0	Gamma	0.0	Rejeita $H_0$

Tabela 5.10: Resultados teste Ljung-Box para série de ENA GAS-CNO multiplicativo 1.

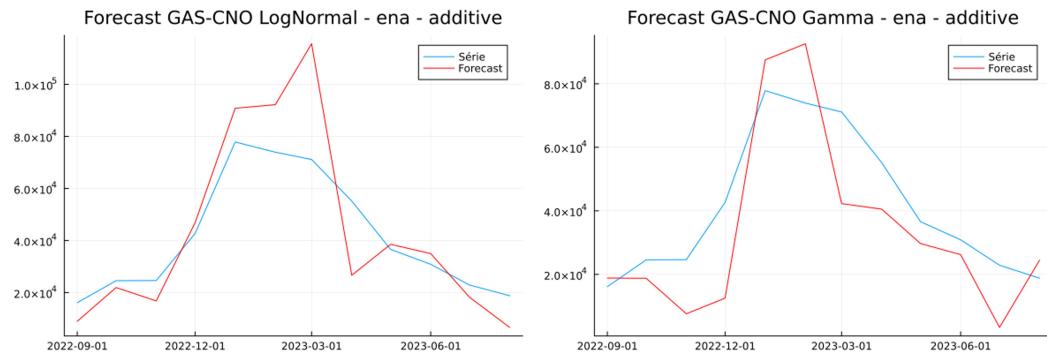


Figura 5.17: Previsão 12 passos à frente para série de ENA GAS-CNO aditivo.

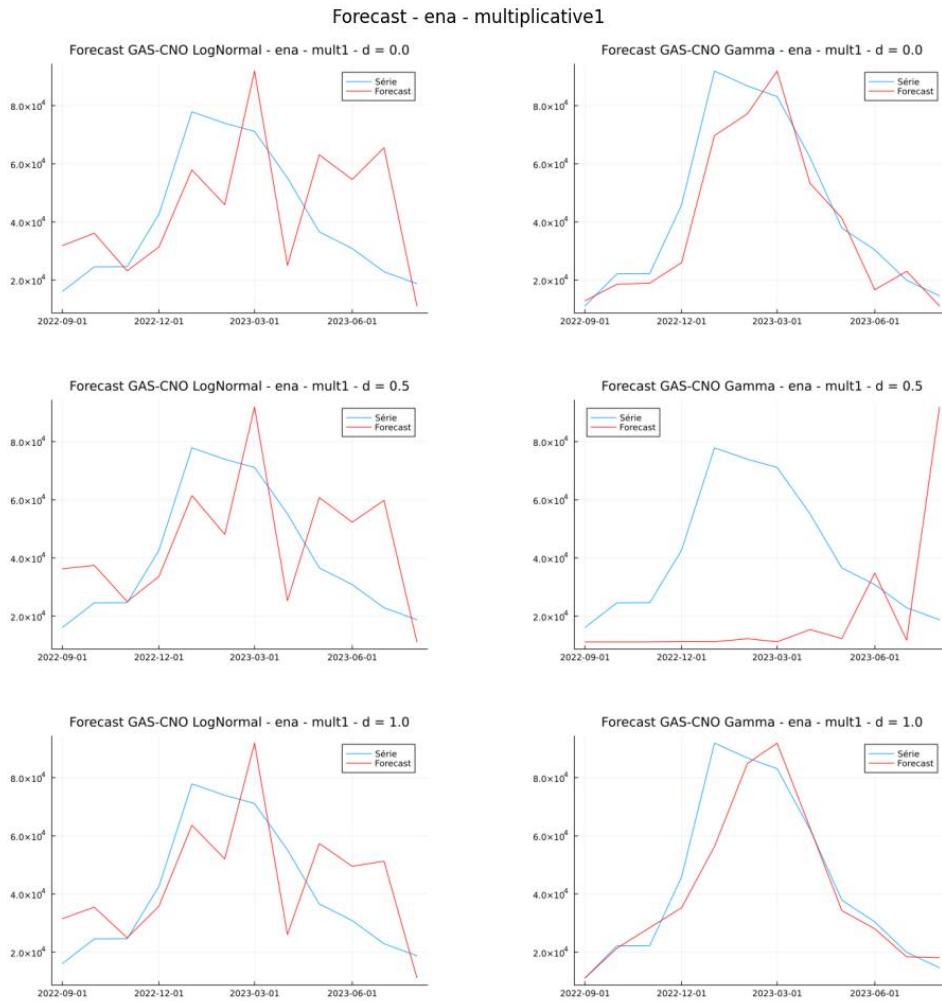


Figura 5.18: Previsão 12 passos à frente para série de ENA GAS-CNO multiplicativo 1.

O primeiro comentário a ser feito para a série de ENA é que, analisando a Figura 5.14, notamos, mais uma vez, que os modelos gama foram mais sensíveis aos valores de  $d$ . Por outro lado, as FACs dos resíduos, indicadas pela Figura 5.16 ainda revelam estrutura não capturada pelo modelo multiplicativo 1, principalmente no *lag* sazonal. Por fim, a Figura 5.18 mostra que os modelos

gama de  $d = 0.0$  e  $d = 1.0$  conseguiram gerar uma previsão que segue a dinâmica da série fora da amostra, resultado que não foi visto para nenhum modelo de série de carga.

### 5.2.2 ENA - GAS-CNO Multiplicativo 2

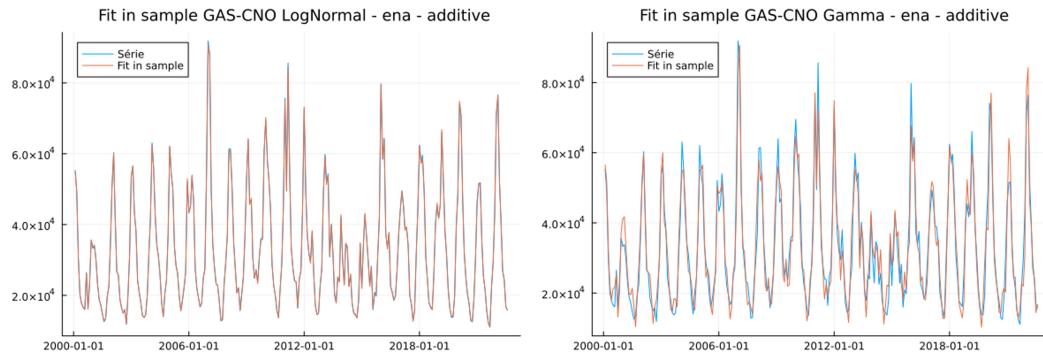


Figura 5.19: Período de treino série de ENA GAS-CNO aditivo.

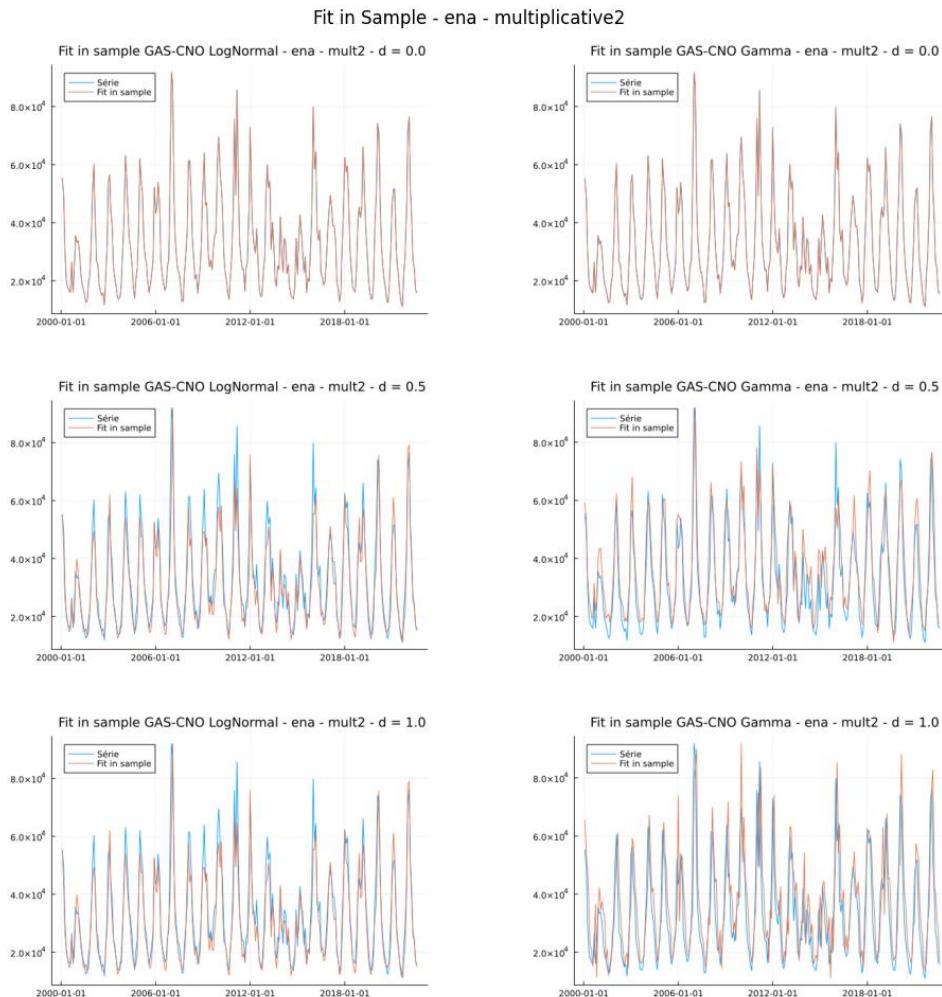


Figura 5.20: Período de treino série de ENA GAS-CNO multiplicativo 2.

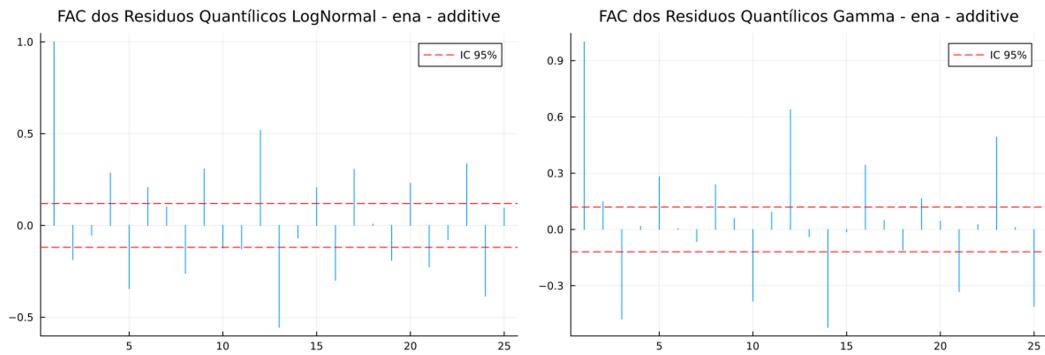


Figura 5.21: FAC dos resíduos quantílicos série de ENA GAS-CNO aditivo.

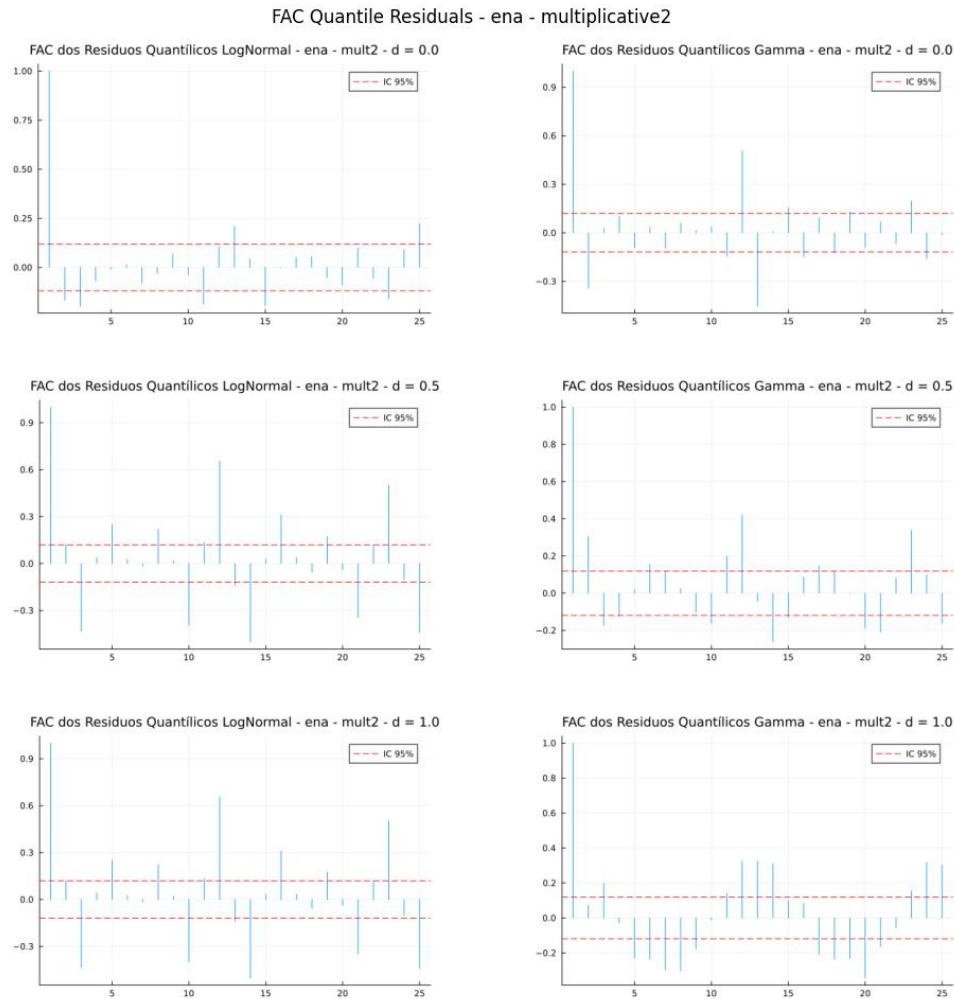


Figura 5.22: FAC dos resíduos quantílicos série de ENA GAS-CNO multiplicativo 2.

<b>d</b>	<b>Distribuição</b>	<b>pvalor</b>	<b>Rejeição</b>
0.0	LogNormal	0.1215	Não rejeita $H_0$
0.0	Gamma	0.0	Rejeita $H_0$
0.5	LogNormal	0.6893	Não rejeita $H_0$
0.5	Gamma	0.0	Rejeita $H_0$
1.0	LogNormal	0.6796	Não rejeita $H_0$
1.0	Gamma	0.0	Rejeita $H_0$

Tabela 5.11: Resultados teste de normalidade Jarque-Bera para série de ENA GAS-CNO multiplicativo 2.

<b>d</b>	<b>Distribuição</b>	<b>pvalor</b>	<b>Rejeição</b>
0.0	LogNormal	0.5782	Não rejeita $H_0$
0.0	Gamma	0.0	Rejeita $H_0$
0.5	LogNormal	0.1227	Não rejeita $H_0$
0.5	Gamma	0.004	Rejeita $H_0$
1.0	LogNormal	0.1149	Não rejeita $H_0$
1.0	Gamma	0.0089	Rejeita $H_0$

Tabela 5.12: Resultados teste de heterocedasticidade ARCH para série de ENA GAS-CNO multiplicativo 2.

<b>d</b>	<b>Distribuição</b>	<b>pvalor</b>	<b>Rejeição</b>
0.0	LogNormal	0.0	Rejeita $H_0$
0.0	Gamma	0.0	Rejeita $H_0$
0.5	LogNormal	0.0	Rejeita $H_0$
0.5	Gamma	0.0	Rejeita $H_0$
1.0	LogNormal	0.0	Rejeita $H_0$
1.0	Gamma	0.0	Rejeita $H_0$

Tabela 5.13: Resultados teste Ljung-Box para série de ENA GAS-CNO multiplicativo 2.

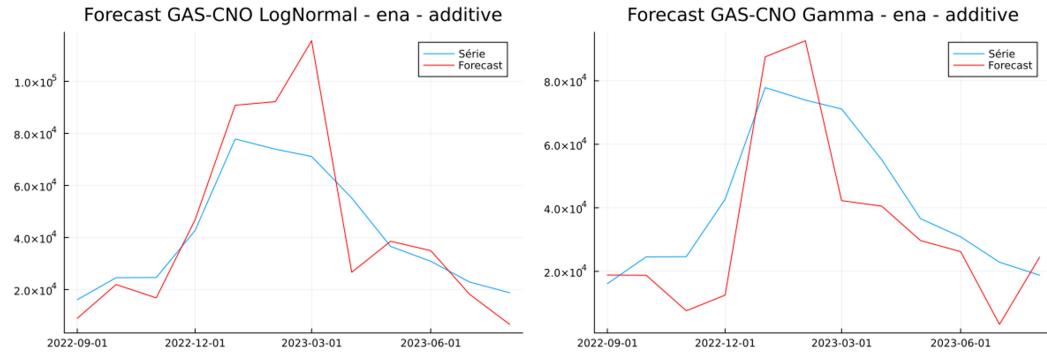


Figura 5.23: Previsão 12 passos à frente para série de ENA GAS-CNO aditivo.

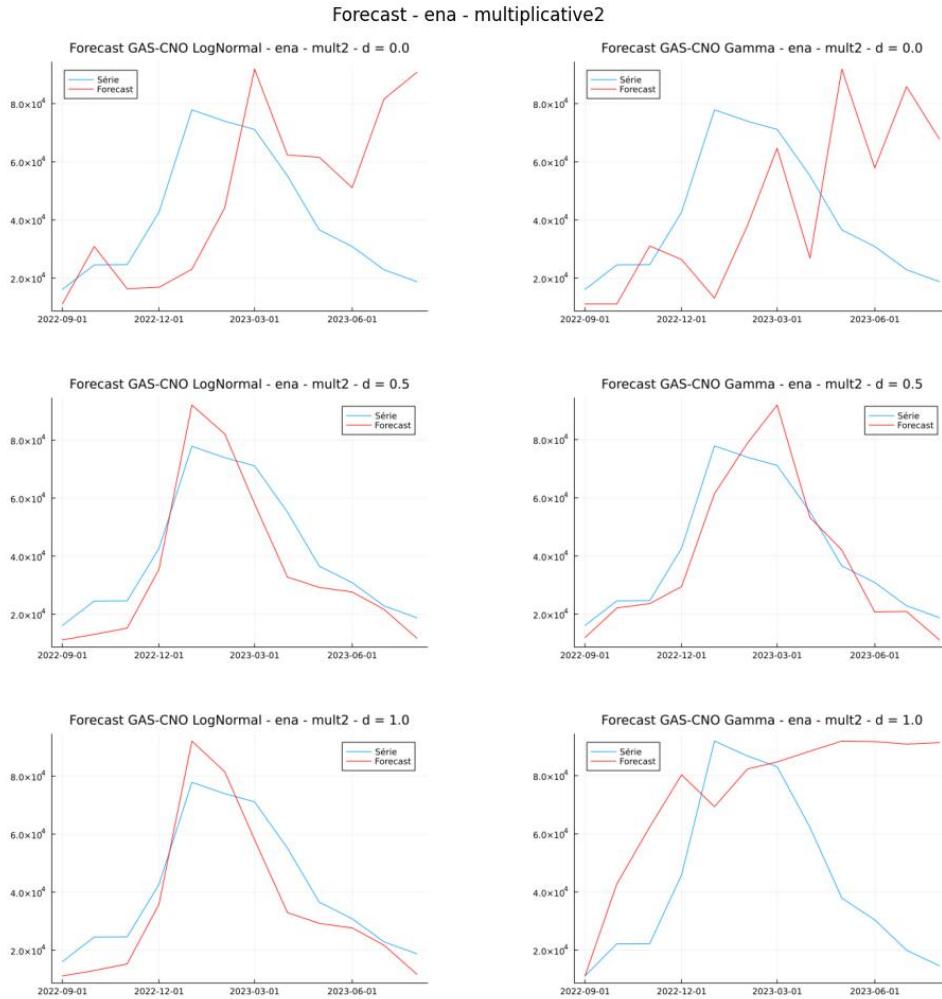


Figura 5.24: Previsão 12 passos à frente para série de ENA GAS-CNO multiplicativo 2.

Pela primeira vez, os modelos log-normais também se mostraram sensíveis, pelo menos visualmente, aos valores de  $d$ , vide Figura 5.20. Para ambas as distribuições, a análise visual indica que os modelos com  $d = 0.0$  obtiveram os melhores valores estimados dentro da amostra. Ao analisar a Figura 5.22, notamos também que a FAC do modelo log-normal com  $d = 0.0$  está

consideravelmente menor que a FAC do modelo aditivo. Todas as demais permanecem indicando *lag* sazonal fortemente significativo. Somado a isso, esse modelo também não é o que, visualmente, gerou a melhor previsão fora da amostra, uma vez que a Figura 5.24 nos mostra que apenas os modelos log-normal com  $d = 0.5$  e  $d = 1.0$  e o modelo gama com  $d = 0.5$  geraram previsões que acompanham a dinâmica da série temporal.

### 5.2.3 ENA - MAPEs

Combinacao	d	Distribuicao	MAPE Treino (%)	MAPE Teste (%)
aditivo	-	LogNormal	0.75	29.69
aditivo	-	Gamma	11.98	36.25
multiplicativo1	0.0	LogNormal	0.01	58.4
multiplicativo1	0.0	Gamma	18.82	20.27
multiplicativo1	0.5	LogNormal	0.05	56.47
multiplicativo1	0.5	Gamma	5.97	88.07
multiplicativo1	1.0	LogNormal	0.19	47.44
multiplicativo1	1.0	Gamma	16.01	13.02
multiplicativo2	0.0	LogNormal	0.0	89.77
multiplicativo2	0.0	Gamma	1.14	93.04
multiplicativo2	0.5	LogNormal	10.8	24.49
multiplicativo2	0.5	Gamma	17.93	19.05
multiplicativo2	1.0	LogNormal	10.81	24.27
multiplicativo2	1.0	Gamma	27.76	137.19

Tabela 5.14: MAPEs dentro e fora da amostra para série de ENA.

Diferente dos resultados obtidos para a série de carga, para a série de ENA foi possível gerar previsões fora da amostra cujo MAPE é menor que o do modelo aditivo convencional. Os modelos que conseguiram superar esse *benchmark* foram os multiplicativos 1 gama com  $d = 0.0$  e  $d = 1.0$ , os modelos multiplicativos 2 log-normais com  $d = 0.5$  e  $d = 1.0$  e o modelo multiplicativo 2 gama com  $d = 0.5$ . Analisando os MAPEs dentro da amostra, muitos foram os que superaram o modelo aditivo, mas isso, por si só, não é suficiente para concluir que os modelos multiplicativos desempenham melhor que o aditivo para a série de ENA.

## 6

### Conclusão

O objetivo deste projeto final da graduação foi estudar se modelos *score-driven* com combinação não linear das componentes de tendência e sazonalidade seriam capazes de resultar em melhor acurácia preditiva para séries temporais não Gaussianas. Para isso, foram escolhidas duas séries temporais com características bem distintas com o intuito de realizar testes empíricos com os modelos multiplicativos sugeridos no projeto e duas distribuições não Gaussianas, log-normal e gama.

Em termos de estimação, posso enumerar algumas conclusões interessantes. Primeiramente, a estimação dos modelos multiplicativos, no geral, foi muito sensível tanto à escolha do parâmetro de escala do *score* padronizado  $d$  quanto à escala da série temporal utilizada. Não é possível realizar uma escolha global que funcione para todas as séries e distribuições. Em segundo lugar, e, possivelmente, a conclusão mais importante, apenas 1 dos diversos modelos implementados foi capaz de reduzir significativamente, a magnitude do *lag* sazonal da FAC dos resíduos do modelo, o que era o principal objetivo deste trabalho.

É importante comentar, também, que os modelos multiplicativos foram mais facilmente estimados para a série de ENA, cuja componente de tendência era estacionária na média. Conseguir gerar modelos GAS-CNO com combinação multiplicativa para séries não estacionárias foi um desafio maior do que para séries estacionárias, mesmo utilizando componentes de passeio aleatório com *drift*.

Por fim, também foi notável que os modelos não geraram previsões fora da amostra coerentes com o período de treino. É possível que, com a função de ligação identidade, não se pode garantir, no processo de simulação, que os parâmetros das distribuições respeitem suas restrições naturais, enquanto que, ao usar uma função logarítmica, por exemplo, esse problema não existiria.

Não é possível, portanto, concluir que os modelos *score-driven* para séries temporais não gaussianas e com combinação não linear das componentes de tendência e sazonalidade são capazes de solucionar o problema da captura da dependência sazonal evidenciada pelo *lag* 12 na FAC dos resíduos. Contudo, é possível, sim, afirmar que esses modelos conseguem competir com o modelo aditivo, em especial quando estimado para uma série estacionária.

Alguns trabalhos futuros interessantes para dar prosseguimento a esta pesquisa seriam:

- Entender a razão da função identidade gerar previsões no período de teste que não são coerentes com o período de treino.
- Substituir a função de ligação identidade pela função logarítmica para as combinações não lineares de componentes.
- Gerar, por simulações de Monte Carlo, séries temporais com combinações não lineares de componentes e função de ligação identidade e logarítmica.
- Considerar normalização da componente sazonal multiplicativa nos modelos.

## Referências bibliográficas

[Alves] ALVES, M. e. a. Unobservedcomponentsgas.jl:. Citado 2 vezes nas páginas 26 e 31.

[Andre e Koopman] ANDRE, L.; KOOPMAN, S. J. *Generalized autoregressive score models*. Disponível em: <<http://www.gasmodel.com/index.htm>>. Citado na página 14.

[Ardia, Boudt e Catania 2016] ARDIA, D.; BOUDT, K.; CATANIA, L. Generalized autoregressive score models in r: The gas package. *arXiv preprint arXiv:1609.02354*, 2016. Disponível em: <<https://cran.r-project.org/web/packages/GAS/GAS.pdf>>. Citado 2 vezes nas páginas 13 e 24.

[Blazsek e Escribano 2023] BLAZSEK, S.; ESCRIBANO, Á. Score-driven threshold ice-age models: benchmark models for long-run climate forecasts. *Energy Economics*, Elsevier, p. 106522, 2023. Citado na página 15.

[Bodin et al. 2020] BODIN, G. et al. Scoredrivenmodels.jl: a julia package for generalized autoregressive score models. *arXiv preprint arXiv:2008.05506*, 2020. Citado 2 vezes nas páginas 13 e 22.

[Bollerslev 1986] BOLLERSLEV, T. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, Elsevier, v. 31, n. 3, p. 307–327, 1986. Citado na página 13.

[Caivano, Harvey e Luati 2016] CAIVANO, M.; HARVEY, A.; LUATI, A. Robust time series models with trend and seasonal components. *SERIEs*, Springer, v. 7, p. 99–120, 2016. Citado 2 vezes nas páginas 30 e 37.

[Cox et al. 1981] COX, D. R. et al. Statistical analysis of time series: Some recent developments [with discussion and reply]. *Scandinavian Journal of Statistics*, JSTOR, p. 93–115, 1981. Citado na página 33.

[Creal, Koopman e Lucas 2013] CREAL, D.; KOOPMAN, S. J.; LUCAS, A. Generalized autoregressive score models with applications. *Journal of Applied Econometrics*, Wiley Online Library, v. 28, n. 5, p. 777–795, 2013. Citado 3 vezes nas páginas 13, 34 e 35.

[Engle e Russell 1998] ENGLE, R. F.; RUSSELL, J. R. Autoregressive conditional duration: a new model for irregularly spaced transaction data. *Econometrica*, JSTOR, p. 1127–1162, 1998. Citado na página 13.

[Fernandes 2020] FERNANDES, C. *Notas de aula da disciplina Score Driven Models para Séries Temporais Não Gaussianas*. [S.I.]: Departamento de Engenharia Elétrica da PUC-Rio, Grupo de Energia Elétrica do LAMPS, 2020. Citado na página 35.

[Fernandes 2023] FERNANDES, C. *Notas de aula – IV Sazonalidade em Modelos Estatísticos*. [S.I.]: Departamento de Engenharia Elétrica da PUC-Rio, Grupo de Energia Elétrica do LAMPS, 2023. Citado na página 79.

[Fuentes, Herrera e Clements 2023] FUENTES, F.; HERRERA, R.; CLEMENTS, A. Forecasting extreme financial risk: A score-driven approach. *International Journal of Forecasting*, Elsevier, v. 39, n. 2, p. 720–735, 2023. Citado na página 15.

[Harvey 2013] HARVEY, A. C. *Dynamic models for volatility and heavy tails: with applications to financial and economic time series*. [S.I.]: Cambridge University Press, 2013. v. 52. Citado 4 vezes nas páginas 13, 30, 37 e 39.

[Harvey 2022] HARVEY, A. C. Score-driven time series models. *Annual Review of Statistics and Its Application*, Annual Reviews, v. 9, p. 321–342, 2022. Citado na página 33.

[Hoeltgebaum et al. 2021] HOELTGEBAUM, H. et al. A score-driven model of short-term demand forecasting for retail distribution centers. *Journal of Retailing*, Elsevier, v. 97, n. 4, p. 715–725, 2021. Citado na página 15.

[Hyndman e Athanasopoulos 2021] HYNDMAN, R.; ATHANASOPOULOS, G. *Forecasting: principles and practice*. [S.I.]: OText, 2021. Citado 3 vezes nas páginas 31, 40 e 48.

[Hyndman et al. 2008] HYNDMAN, R. et al. *Forecasting with exponential smoothing: the state space approach*. [S.I.]: Springer Science & Business Media, 2008. Citado 2 vezes nas páginas 31 e 39.

[Jonathan e Kung-Sik 2008] JONATHAN, D. C.; KUNG-SIK, C. *Time series analysis with applications in R*. [S.I.]: Springer, 2008. Citado na página 35.

[Kalliovirta 2012] KALLIOVIRTA, L. Misspecification tests based on quantile residuals. *The Econometrics Journal*, Oxford University Press Oxford, UK, v. 15, n. 2, p. 358–393, 2012. Citado na página 35.

[Koopman e Lit 2019] KOOPMAN, S. J.; LIT, R. Forecasting football match results in national league competitions using score-driven time series models. *International Journal of Forecasting*, Elsevier, v. 35, n. 2, p. 797–809, 2019. Citado na página 15.

[Kushwah e Wadhvani 2019] KUSHWAH, A. K.; WADHVANI, R. Performance monitoring of wind turbines using advanced statistical methods. *Sādhanā*, Springer, v. 44, p. 1–11, 2019. Citado na página 15.

[Lit] LIT, R. *Time Series Lab Manual*. Disponível em: <<https://timeserieslab.com>>. Citado na página 17.

[Lit, Koopman e Harvey 2021] LIT, R.; KOOPMAN, S.; HARVEY, A. Time series lab - dynamic score edition. 2021. Disponível em: <<https://timeserieslab.com>>. Citado na página 13.

[Sarlo, Fernandes e Borenstein 2023] SARLO, R.; FERNANDES, C.; BORENSTEIN, D. Lumpy and intermittent retail demand forecasts with score-driven models. *European Journal of Operational Research*, Elsevier, v. 307, n. 3, p. 1146–1160, 2023. Citado na página 34.

[Taylor 2016] TAYLOR, R. Pyflux: A time-series analysis library for python. nov. 2016. Disponível em: <<https://pyflux.readthedocs.io/en/latest/index.html>>. Citado na página 13.

[Xu e Lien 2022] XU, Y.; LIEN, D. Forecasting volatilities of oil and gas assets: A comparison of gas, garch, and egarch models. *Journal of Forecasting*, Wiley Online Library, v. 41, n. 2, p. 259–278, 2022. Citado na página 15.

# A

## Desenvolvimento GAS

Neste capítulo são apresentadas todas as contas relacionadas ao modelo GAS para algumas distribuições de probabilidade.

### A.1 GAS gama

Seja  $(y_t | \tilde{y}_{t-1}) \sim Gama(\alpha_t, \lambda_t)$ , ou seja, uma série cuja distribuição é uma gama com ambos os parâmetros variantes no tempo.

Vale que  $E(y_t | \tilde{y}_{t-1}) = \lambda_t$  e  $V(y_t | \tilde{y}_{t-1}) = \lambda_t^2 / \alpha_t$ .

Para essa parametrização escolhida, segue-se que a função densidade é

$$f(y_t | \tilde{y}_{t-1}; \alpha_t, \lambda_t) = \frac{1}{\Gamma(\alpha_t)} \frac{1}{(\alpha_t^{-1} \lambda_t)^{\alpha_t}} y_t^{\alpha_t - 1} e^{-\frac{\alpha_t}{\lambda_t} y_t}$$

Tomando o logaritmo natural da densidade, obtenho:

$$\ln f(y_t | \tilde{y}_{t-1}; \alpha_t, \lambda_t) = -\ln(\Gamma(\alpha_t)) - \alpha_t \ln(1/\alpha_t) - \alpha_t \ln(\lambda_t) + (\alpha_t + 1) \ln(y_t) - (\alpha_t / \lambda_t) y_t$$

O próximo passo é obter o vetor  $\nabla_t = \begin{pmatrix} \nabla_t^\alpha \\ \nabla_t^\lambda \end{pmatrix}$

onde

$$\begin{aligned} \nabla_t^\alpha &= \frac{\partial \ln f(y_t | \tilde{y}_{t-1})}{\partial \alpha_t} = \ln(y_t) - \frac{y_t}{\lambda_t} + \ln(\alpha_t) - \psi_1(\alpha_t) - \ln(\lambda_t) + 1 \\ \nabla_t^\lambda &= \frac{\partial \ln f(y_t | \tilde{y}_{t-1})}{\partial \lambda_t} = \frac{\alpha_t}{\lambda_t} \left( \frac{y_t}{\lambda_t} - 1 \right) \end{aligned}$$

Além disso, foi definido  $\psi_1(\alpha) = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)}$ , também conhecida como função *digamma*.

É importante ressaltar que  $E_{t-1}(\nabla_t^\alpha) = E_{t-1}(\nabla_t^\lambda) = 0$

O próximo passo para a definição de um modelo GAS gama é a obtenção da matriz de informação de Fisher

$$I_{t|t-1} = \begin{pmatrix} I_{t|t-1}^{\alpha\alpha} & I_{t|t-1}^{\alpha\lambda} \\ I_{t|t-1}^{\alpha\lambda} & I_{t|t-1}^{\lambda\lambda} \end{pmatrix}$$

Onde, cada elemento dessa matriz, é definido como

$$\begin{aligned}
I_{t|t-1}^\alpha &= E_{t-1}(\nabla_t^\alpha \nabla_t'^\alpha) = E_{t-1}(\nabla_t^\alpha)^2 = -E_{t-1} \frac{\partial}{\partial \alpha_t} (\nabla_t^\alpha) \\
I_{t|t-1}^\lambda &= E_{t-1}(\nabla_t^\lambda \nabla_t'^\lambda) = E_{t-1}(\nabla_t^\lambda)^2 = -E_{t-1} \frac{\partial}{\partial \lambda} (\nabla_t^\lambda) \\
I_{t|t-1}^{\alpha,\lambda} &= E_{t-1}(\nabla_t^\alpha \nabla_t'^\lambda) = -E_{t-1} \frac{\partial}{\partial \lambda} (\nabla_t^\alpha) = -E_{t-1} \frac{\partial}{\partial \alpha} (\nabla_t^\lambda)
\end{aligned}$$

Agora, desenvolvendo cada um desses termos individualmente, obtenho

$$\begin{aligned}
I_{t|t-1}^\alpha &= E_{t-1} \frac{\partial}{\partial \alpha_t} \left[ \ln(y_t) - \frac{y_t}{\lambda_t} + \ln(\alpha_t) - \psi_1(\alpha_t) - \ln(\lambda_t) + 1 \right] = \dots = \\
&= \psi_2(\alpha_t) - \frac{1}{\alpha_t} \\
I_{t|t-1}^\lambda &= E_{t-1} \frac{\partial}{\partial \lambda_t} \left[ \frac{\alpha_t y_t}{\lambda_t^2} - \frac{\alpha_t}{\lambda_t} \right] = -E_{t-1} \left[ \frac{-2\alpha_t y_t}{\lambda_t^3} + \frac{\alpha_t}{\lambda_t^2} \right] = \dots = \\
&= \frac{\alpha_t}{\lambda_t^2} \\
I_{t|t-1}^{\alpha,\lambda} &= -E_{t-1} \frac{\partial}{\partial \alpha_t} \left[ \frac{\alpha_t y_t}{\lambda_t^2} - \frac{\alpha_t}{\lambda_t} \right] = -E_{t-1} \left[ \frac{\alpha_t y_t}{\lambda_t^2} - \frac{\alpha_t}{\lambda_t} \right] = \dots = \\
&= 0
\end{aligned}$$

Portanto, a matriz de informação de Fisher  $I_{t|t-1}$  para uma distribuição gama de parâmetros  $\alpha_t$  e  $\lambda_t$  é

$$I_{t|t-1} = \begin{pmatrix} \psi_2(\alpha_t) - \frac{1}{\alpha_t} & 0 \\ 0 & \frac{\alpha_t}{\lambda_t^2} \end{pmatrix}$$

onde a função  $\psi_2(\alpha_t)$  é a função *trigamma* definida como  $\psi_2(\alpha_t) = \frac{\psi'_1(\alpha_t)}{\psi_1(\alpha_t)}$

Com isso, estão definidos todos os resultados necessários para estimar um modelo GAS, tanto com dinâmica ARMA quanto CNO, cuja distribuição condicional é  $gama(\alpha_t, \lambda_t)$ .

## B

### Sazonalidade em modelos de espaço de estados

Neste apêndice, está descrito o processo de modelagem de sazonalidade em modelos de espaço de estados, começando pelo tratamento de sazonalidade determinística até sua versão estocástica. O entendimento da modelagem da componente sazonal dentro dessa classe de modelos é fundamental para compreender a modelagem de sazonalidade em modelos *score-driven* cuja dinâmica é modelada via arcabouço CNO. A maior parte do desenvolvimento aqui apresentado encontra-se em [Fernandes 2023].

Primeiramente, é importante definir sazonalidade como a representação de flutuações periódicas associadas a eventos climáticos, como as estações do ano, eventos culturais, como feriados e festas, bem como datas administrativas, como períodos letivos. Vale comentar que a boa modelagem da sazonalidade é essencial para garantir a qualidade da estimativa de um modelo de série temporal (quando esta apresenta sazonalidade), dado o seu caráter repetitivo.

Um conceito importante a ser definido é o período sazonal, que representa o tempo em que a flutuação periódica leva para se repetir e é descrito pela letra  $s$ . Além disso, sazonalidade é definida apenas para flutuações de, no máximo, 1 ano. Para períodos maiores que este existe o conceito de ciclo que não será abordado nesse apêndice. Exemplos de períodos sazonais seriam  $s = 4$  para séries trimestrais,  $s = 12$  para séries mensais e  $s = 52$  para séries semanais. Embora seja possível que uma série possua mais de uma componente sazonal, como é o caso de séries diárias que podem possuir sazonalidade semanal e anual, esse assunto também não será abordado nesse apêndice, dado que o escopo do projeto se limita a séries mensais.

Para começar o desenvolvimento da modelagem de sazonalidade em série temporais, suponha que uma dada série  $y_t$  possa ser decomposta em componentes de tendência  $\mu_t$  e sazonalidade  $\gamma_t$  como se segue:

$$y_t = \mu_t + \gamma_t + \epsilon_t \quad \epsilon_t \sim N(0, \sigma^2) \quad (\text{B-1})$$

A equação B-1 descreve um modelo cuja componente sazonal se combina de forma aditiva e é o padrão para modelos lineares. Também é possível haver sazonalidade multiplicativa em modelos lineares. Para isso, basta que o modelo seja ajustado para log da série. Esse modelo seria descrito como:

$$y_t = \mu_t \cdot \gamma_t \cdot \epsilon_t \quad \epsilon_t \sim N(0, \sigma^2) \quad (\text{B-2})$$

Sabe-se que, dentro do arcabouço de modelos de espaço de estados, a

componente de tendência  $\mu_t$  pode ser definida da seguinte maneira (modelo estrutural básico):

$$\begin{aligned}y_t &= \mu_t + \gamma_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma^2) \\ \mu_{t+1|t} &= \mu_t + \beta_t + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2) \\ \beta_{t+1|t} &= \beta_t + \xi_t, \quad \xi_t \sim N(0, \sigma_\xi^2)\end{aligned}$$

O objetivo desse apêndice é adicionar a formulação para a componente sazonal  $\gamma_t$  nesse modelo. Para isso, primeiramente será apresentada a componente sazonal determinística utilizando variáveis *dummies* e funções trigonométricas para, em seguida, apresentar a formulação da sazonalidade estocástica.

### Sazonalidade por variáveis *dummies*

O tratamento da sazonalidade por variáveis *dummies* é o mais simples de ser implementado, além de trazer uma interpretação direta de cada coeficiente estimado.

Suponha uma série  $y_t$  estacionária trimetral, o que implica que  $s_4$ . Poderíamos utilizar um modelo como:

$$y_t = \beta + \gamma_1 D_{1,t} + \gamma_2 D_{2,t} + \gamma_3 D_{3,t} + \gamma_4 D_{4,t} + \epsilon_t, \quad t = 1, \dots, T$$

$$D_{i,t} = 1, \quad i = t, \quad i = 1, 2, 3, 4$$

$$D_{i,t} = 0 \quad c.c.$$

Note que cada variável *dummie*  $D_{i,t}$  indica se a observação daquele trimestre  $t$  está no trimestre  $i$ . Embora esse modelo capture a periodicidade trimestral, ele apresenta o problema de multicolinearidade perfeita, dado que um dos regressores ( $D_{4,t}$ , por exemplo) pode ser obtido a partir da combinação linear dos demais.

Para resolver esse problema, existem 3 parametrização possíveis.

A primeira baseia-se em abandonar uma das *dummies* do modelo. Suponha, por exemplo, que abandonemos a *dummie* associada ao quarto trimestre. O período que foi abandonado é chamado de período basal. Com isso, obteríamos o modelo:

$$y_t = \beta + \gamma_1 D_{1,t} + \gamma_2 D_{2,t} + \gamma_3 D_{3,t} + \epsilon_t, \quad t = 1, \dots, T$$

$$D_{i,t} = 1, \quad t = i, i + s, i + 2s, \dots, \quad i = 1, 2, 3$$

$$D_{i,t} = 0 \quad t \neq i, i + s, i + 2s, \dots$$

Essa parametrização traz uma interpretação direta para os coeficientes do modelo, que advém do valor esperado de  $y_t$  para cada trimestre, como se segue

abaixo.

$$\begin{aligned} E(y_t | D_{i,t} = 0, i = 1, 2, 3) &= \beta \\ E(y_t | D_{j,t} = 1, D_{i,t} = 0, j \neq i, i = 1, 2, 3) &= \beta + \gamma_j \end{aligned}$$

Portanto,  $\hat{\beta} = \bar{y}_4$ , ou seja, o intercepto representa a média do trimestre basal e  $\gamma_j = \bar{y}_j - \bar{y}_4$ , isto é, os coeficientes sazonais são o desvio da média de cada trimestre em relação ao trimestre basal.

A segunda parametrização opta, por sua vez, em abandonar o intercepto e manter todas as *dummies* sazonais, obtendo o modelo:

$$\begin{aligned} y_t &= \delta_1 D_{1,t} + \delta_2 D_{2,t} + \delta_3 D_{3,t} + \delta_4 D_{4,t} + \epsilon_t, \quad t = 1, \dots, T \\ D_{j,t} &= 1, \quad t = j, i+s, i+2s, \dots, \quad j = 1, 2, 3 \\ D_{j,t} &= 0 \quad t \neq j, j+s, j+2s, \dots \end{aligned}$$

Novamente, a interpretação de cada coeficiente sazonal é direta, mas, dessa vez,  $\hat{\delta}_j = \bar{y}_j$ , que significa que a estimativa de cada coeficiente é a média daquele trimestre. Essa interpretação vem do fato de que  $E(y_t | D_{j,t}) = \delta_j$ .

A terceira e última parametrização opta por manter todas as *dummies* no modelo, mas, para evitar a multicolinearidade, introduz uma restrição nos coeficientes sazonais de tal forma que a soma dos fatores sazonais deve ser zero no período sazonal. Isso significa que:

$$\sum_{j=1}^s \theta_j = 0 \implies \theta_s = -\sum_{j=1}^{s-1} \theta_j$$

A partir dessa restrição, obtemos o modelo:

$$\begin{aligned} y_t &= \alpha + \sum_{j=1}^{s-1} \theta_j D_{j,t} + \epsilon_t, \quad t = 1, \dots, T \\ D_{j,t} &= 1, t = j, j+s, j+2s, \dots \\ D_{j,t} &= 0, t \neq j, j+s, j+2s, \dots \\ D_{j,t} &= -1, t = s, 2s, 3s, \dots \end{aligned}$$

Para conseguir obter uma interpretação útil dos coeficientes sazonais, note

que

$$\begin{aligned} E(y_t|D_{j,t}) &= \alpha + \theta_j \implies \theta_j = E(y_t|D_{j,t}) - \alpha, \quad j = 1, 2, 3 \\ E(y_t|D_{4,t}) &= \alpha - (\theta_1 + \theta_2 + \theta_3) = \alpha + \theta_4 \implies \theta_4 = E(y_t|D_{4,t}) - \alpha \end{aligned}$$

Dada a restrição imposta inicialmente, posso escrever que:

$$0 = \theta_1 + \theta_2 + \theta_3 + \theta_4 = \sum_{j=1}^4 E(y_t|D_{j,t}) - 4\alpha$$

Logo,

$$\alpha = \frac{1}{4} \sum_{j=1}^4 E(y_t|D_{j,t}) \implies \alpha = \frac{1}{4} \sum_{j=1}^4 \bar{y}_j$$

Portanto, se os trimestres forem balanceados,  $\hat{\alpha} = \bar{y}$ , que é a média da série. Com isso,  $\hat{\theta}_j$  serão os desvios das médias dos trimestres em relação à média da série.

A partir da modelagem por *dummies* com a parametrização 3, podemos completar a definição do modelo estrutural básico que foi apresentado no início desse apêndice. Mantendo as componentes de tendência e nível estocástica, mas a nova componente de sazonalidade determinística, obtenho:

$$\begin{aligned} y_t &= \mu_t + \gamma_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma^2) \\ \mu_{t+1|t} &= \mu_t + \beta_t + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2) \\ \beta_{t+1|t} &= \beta_t + \xi_t, \quad \xi_t \sim N(0, \sigma_\xi^2) \\ \gamma_t &= - \sum_{j=1}^{s-1} \gamma_{t-j} \end{aligned}$$

Note que a componente sazonal já está escrita em uma forma recursiva, então, para modificá-la para uma versão estocástica, basta somar um choque aleatório em sua equação de estado. Com isso, finalmente obtemos o modelo estrutural básico com sazonalidade modelada a partir de variáveis *dummies*.

$$\begin{aligned} y_t &= \mu_t + \gamma_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma^2) \\ \mu_{t+1|t} &= \mu_t + \beta_t + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2) \\ \beta_{t+1|t} &= \beta_t + \xi_t, \quad \xi_t \sim N(0, \sigma_\xi^2) \\ \gamma_t &= - \sum_{j=1}^{s-1} \gamma_{t-j} + \omega_t, \quad \omega_t \sim N(0, \sigma_\omega^2) \end{aligned}$$

É importante para o escopo deste projeto ressaltar que, normalmente, todos os trimestres da série temporal possuem uma mesma variância da componente

sazonal. Um dos objetivos deste projeto é, justamente, generalizar essa variância de forma que diferentes trimestres, ou épocas do ano de forma geral, possuam variâncias diferentes, ou seja  $\sigma_\omega^2 \rightarrow \sigma_{\omega,t}^2$

### Sazonalidade por funções trigonométricas

Além de modelar a sazonalidade de uma série temporal a partir de variáveis *dummies* que indicam em qual período uma dada observação se encontra, também é possível modelar o efeito sazonal a partir de uma combinação de funções trigonométricas. Começando, novamente, pela versão determinística, podemos definir a dinâmica da componente sazonal como:

$$\gamma_t = \sum_{j=1}^{[s/2]} (\gamma_j \cos(\lambda_j t) + \gamma_j^* \sin(\lambda_j t)) \quad (\text{B-3})$$

onde  $\lambda_j = 2\pi j/s, j = 1, \dots, [s/2]$  são os harmônicos,  $s$  é o período sazonal da série e  $[s/2] = s/2, s$  par ou  $[s/2] = (s-1)/2, s$  ímpar.

É importante ressaltar que  $\gamma_j^* = 0$  se  $j = [s/2]$ . Além disso, diferente da sazonalidade por *dummies*, o coeficiente sazonal não mais o fator sazonal em si. Agora, os coeficientes sazonais são  $\gamma_j, \gamma_j^*$ , enquanto que o fator sazonal é  $\gamma_t$ .

Uma vez definida a versão determinística da sazonalidade por funções trigonométricas, falta obter sua versão estocástica. Para isso, primeiro vou obter uma forma recursiva da equação e, em seguida, adicionar um choque aleatório.

Antes de começar esse desenvolvimento, atente-se para a notação possivelmente confusa, assumindo série mensal:  $\gamma_t$  é a sazonalidade do mês  $t$ ,  $(\gamma_j, \gamma_j^*)$  são os coeficientes trigonométricos e  $(\gamma_{j,t}, \gamma_{j,t}^*)$  são os coeficientes estocásticos.

O primeiro passo é reescrever a equação B-3 como:

$$\gamma_t = \sum_{j=1}^{[s/2]} (\gamma_j \cos(\lambda_j t) + \gamma_j^* \sin(\lambda_j t)) = \sum_{j=1}^{[s/2]} \gamma_{j,t} \quad (\text{B-4})$$

Note que

$$\gamma_{j,t} = \begin{cases} \gamma_j \cos(\lambda_j t) + \gamma_j^* \sin(\lambda_j t), & \text{se } j = 1, \dots, (s/2) - 1 \\ \gamma_{s/2} \cos \lambda_{s/2} t, & \text{se } j = s/2 \end{cases} \quad (\text{B-5})$$

Para obter a relação recursiva, trocaremos  $t$  por  $t+1$  na expressão acima e desenvolveremos as contas.

$$\begin{aligned}
\gamma_{j,t+1} &= \gamma_j \cos(\lambda_j(t+1)) + \gamma_j^* \sin(\lambda_j(t+1)) \\
\gamma_{j,t+1} &= \gamma_j \cos(\lambda_j t + \lambda_j) + \gamma_j^* \sin(\lambda_j t + \lambda_j) \\
\gamma_{j,t+1} &= \gamma_j [\cos(\lambda_j t) \cos(\lambda_j) - \sin(\lambda_j t) \sin(\lambda_j)] + \\
&\quad + \gamma_j^* [\sin(\lambda_j t) \cos(\lambda_j) + \cos(\lambda_j t) \sin(\lambda_j)] \\
\gamma_{j,t+1} &= \cos \lambda_j [\gamma_j \cos(\lambda_j t) + \gamma_j^* \sin(\lambda_j t)] + \\
&\quad + \sin \lambda_j [-\gamma_j \sin(\lambda_j t) + \gamma_j^* \cos(\lambda_j t)]
\end{aligned}$$

Pode-se perceber que, dentro dos colchetes, obtivemos, justamente, as expressões para  $\gamma_{j,t}$  e  $\gamma_{j,t}^*$ . Reescrevendo a equação acima em forma vetorial obtenho:

$$\gamma_{j,t+1} = \begin{bmatrix} \cos \lambda_j & \sin \lambda_j \end{bmatrix} \begin{bmatrix} \gamma_{j,t} \\ \gamma_{j,t}^* \end{bmatrix}$$

Com essa equação, temos a recursão de  $\gamma_{j,t+1}$  com  $\gamma_{j,t}$ , mas falta ainda obter a recursão para  $\gamma_{j,t}^*$ . Para isso, basta seguir o mesmo procedimento:

$$\begin{aligned}
\gamma_{j,t+1}^* &= \gamma_j \cos^*(\lambda_j(t+1)) - \gamma_j \sin(\lambda_j(t+1)) \\
\gamma_{j,t+1}^* &= -\gamma_j [\sin(\lambda_j t) \cos(\lambda_j) + \sin(\lambda_j) \cos(\lambda_j t)] + \\
&\quad + \gamma_j^* [\cos(\lambda_j t) \cos(\lambda_j) - \sin(\lambda_j t) \sin(\lambda_j)] \\
\gamma_{j,t+1}^* &= \begin{bmatrix} -\sin \lambda_j & \cos \lambda_j \end{bmatrix} \begin{bmatrix} \gamma_j \cos(\lambda_j t) + \gamma_j^* \sin(\lambda_j t) \\ \gamma_j^* \cos(\lambda_j t) + \gamma_j \sin(\lambda_j t) \end{bmatrix} \\
\gamma_{j,t+1}^* &= \begin{bmatrix} -\sin \lambda_j & \cos \lambda_j \end{bmatrix} \begin{bmatrix} \gamma_{j,t} \\ \gamma_{j,t}^* \end{bmatrix}
\end{aligned}$$

Juntando as duas equações, obtenho finalmente a relação recursiva desejada:

$$\begin{bmatrix} \gamma_{j,t+1} \\ \gamma_{j,t+1}^* \end{bmatrix} = \begin{bmatrix} \cos \lambda_j & \sin \lambda_j \\ -\sin \lambda_j & \cos \lambda_j \end{bmatrix} \begin{bmatrix} \gamma_{j,t} \\ \gamma_{j,t}^* \end{bmatrix}$$

Assim como foi o caso para o tratamento da sazonalidade via variáveis *dummies*, para tornar essa componente estocástica basta adicionar choques aleatórios na equação.

$$\begin{bmatrix} \gamma_{j,t+1} \\ \gamma_{j,t+1}^* \end{bmatrix} = \begin{bmatrix} \cos \lambda_j & \sin \lambda_j \\ -\sin \lambda_j & \cos \lambda_j \end{bmatrix} \begin{bmatrix} \gamma_{j,t} \\ \gamma_{j,t}^* \end{bmatrix} + \begin{bmatrix} \omega_{j,t} \\ \omega_{j,t}^* \end{bmatrix}$$

onde  $\omega_{j,t} \sim N(0, \sigma_\omega^2)$  e  $\omega_{j,t}^* \sim N(0, \sigma_\omega^2)$ .

É possível simplificar o modelo utilizando o mesmo choque para todas as

componentes sazonais, de tal forma que :

$$\begin{bmatrix} \gamma_{j,t+1} \\ \gamma_{j,t+1}^* \end{bmatrix} = \begin{bmatrix} \cos \lambda_j & \sin \lambda_j \\ -\sin \lambda_j & \cos \lambda_j \end{bmatrix} \begin{bmatrix} \gamma_{j,t} \\ \gamma_{j,t}^* \end{bmatrix} + \begin{bmatrix} \omega_{j,t} \\ \omega_{j,t} \end{bmatrix}$$

Além disso, note que a variância dos choques da componente sazonal são constantes no tempo e, tal qual foi comentado para o caso do tratamento via variáveis *dummies*, um dos objetivos desse projeto é generalizar  $\sigma_\omega^2 \rightarrow \sigma_{\omega,t}^2$ .