

Um estudo sobre os modelos ARIMA e sua capacidade preditiva para dados de competição de previsão de séries temporais

Aluno Responsável:

Marcos Augusto Daza Barbosa

Orientador

José Augusto Fiorucci

Palavras-chave: séries temporais; `auto.arima`; ARIMA; SARIMA; M3; forecasting

Introdução

A utilização de modelos preditivos em séries temporais assume fundamental importância, para o planejamento operacional de diversas empresas que buscam antecipar e tomar proveito de eventos futuros. Além disso, vale ressaltar que esse tipo de modelagem também é indispensável para estudar fenômenos da natureza, como previsão do tempo, ocorrência de chuvas, mudanças no nível de água de rios e de mares e etc. Sendo assim, é ponto pacífico dizer que é de grande interesse econômico e científico, estudar e aprimorar tais métodos.

Nesse cenário, a família de modelos estatísticos ARIMA/SARIMA assume destaque e, à vista disso, este projeto visa estudá-la e analisar sua capacidade preditiva, com a utilização da função `auto.arima()` do pacote *forecast* [5] do *software* R [10]. Baseado em diversos critérios técnicos, esta função seleciona automaticamente o melhor modelo entre aqueles pertencentes a família do modelo.

Para tal fim, utilizou-se o conjunto de dados da competição de previsão de séries temporais M3 [6], que foi um evento em que diversos profissionais e pesquisadores de diferentes áreas competiram para tentar obter as melhores previsões, para mais de 3000 séries temporais.

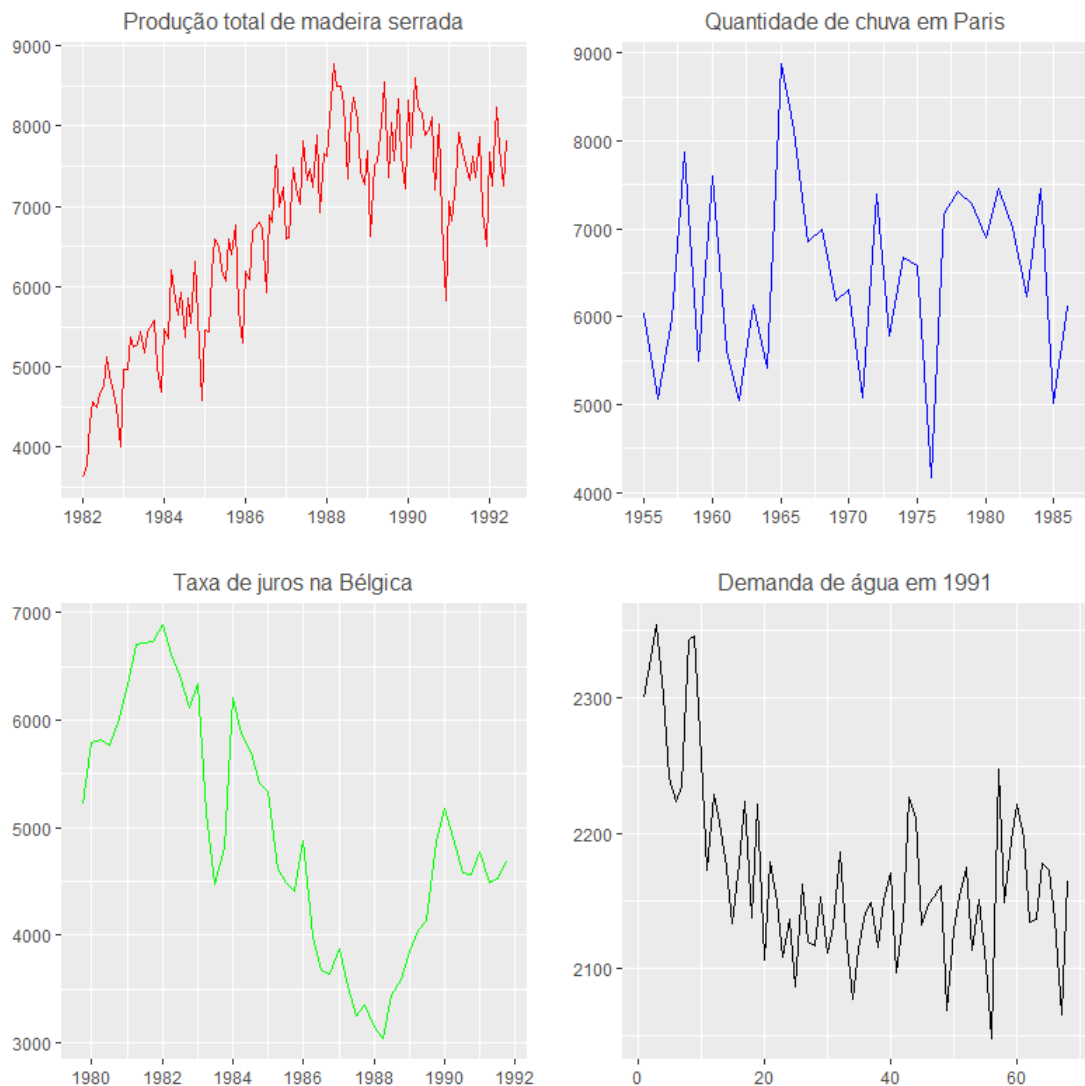
O presente documento está organizado da seguinte maneira, a seção 1 apresenta uma revisão dos modelos envolvidos com as abordagens consideradas e detalha o banco de dados utilizado neste estudo; a seção 2 apresenta os resultados obtidos e as análises envolvidas; por fim, a seção 3 apresenta as conclusões e detalha algumas possíveis abordagens para trabalhos futuros.

1 Metodologia

1.1 Série temporal

Uma série temporal consiste em uma coleção de observações, feitas sequencialmente, ao longo do tempo. Esse fenômeno se caracteriza por possuir dependência entre as observações que estão próximas ou que estão em um mesmo ciclo, ou seja, em um mesmo período de recorrência. Abaixo, encontra-se uma figura com quatro gráficos de séries temporais utilizadas na elaboração deste artigo.

Figura 1: Exemplos de séries temporais



Um aspecto importante a ser entendido é o de estacionaridade. Uma série temporal é estacionária se, ao longo do tempo, desenvolve-se de modo aleatório em torno de uma média, refletindo alguma forma de equilíbrio estável. Contudo, devido à complexidade e ao caos inerentes dos fenômenos estudados pelas séries temporais, estas muitas vezes são não-estacionárias.

As séries temporais utilizadas na confecção deste artigo são ditas *discretas*, pois as observações são feitas em tempos específicos, geralmente equiespaçados. Com a definição

do conjunto $T = \{t_1, \dots, t_n\}$, a série temporal será denotada por $\{X_t : t \in T\}$. De um modo mais simples, pode-se fazer $T = \{1, 2, \dots, n\}$

1.2 Família de modelos ARIMA/SARIMA

Segundo o livro *Análise de Séries Temporais* [9], a classe dos modelos ARIMA será capaz de descrever de maneira satisfatória séries estacionárias e não estacionárias, contanto que não apresentem comportamento explosivo. Aqui será definido de forma sucinta e objetiva esse modelo.

Sabe-se que uma série temporal X_t , substituída pela sua d-ésima diferença $\nabla^d X_t$, é apropriada para descrever certos tipos de séries não estacionárias. Denota-se

$$W_t = \nabla^d X_t = (1 - B)^d X_t$$

Em que B é operador de retardo tal que $B^j X_t = X_{t-j}$, para todo j . Dada essa definição, tem-se que um processo autoregressivo médias móveis - $ARIMA(p, d, q)$ - é dado por

$$W_t = \alpha_1 W_{t-1} + \dots + \alpha_p W_{t-p} + \epsilon_t + \beta_1 \epsilon_{t-1} + \dots + \beta_q \epsilon_{t-q}$$

ou, equivalentemente

$$\phi(B)(1 - B)^d X_t = \theta(B)\epsilon_t$$

Em que $\phi(B)$ é o operador de médias móveis de ordem q , $\theta(B)$ é o operador autoregressivo estacionário de ordem p e ϵ_t é um processo aleatório com média zero e variância σ_ϵ^2 .

De acordo com o livro *Análise de séries temporais* [2], para incluir uma componente periódica sazonal, que está contida em muitas séries temporais, há o modelo ARIMA sazonal multiplicativo $SARIMA(p, d, q) \times (P, D, Q)_s$, definido por:

$$\phi(B)\Phi(B^s)W_t = \theta(B)\Theta(B^s)\epsilon_t$$

em que

$$\begin{aligned}\phi(B) &= (1 - \alpha_1 B - \dots - \alpha_p B^p) \\ \Phi(B^s) &= (1 - \phi_s B^s - \dots - \phi_P B^{Ps}) \\ W_t &= \nabla^d \nabla_s^D X_t = (1 - B^s)^D (1 - B)^d X_t \\ \theta(B) &= (1 + \beta_1 B + \dots + \beta_q B^q) \\ \Theta(B^s) &= (1 + \theta_s B^s + \dots + \theta_Q B^{Qs})\end{aligned}$$

1.3 Banco de dados da competição M3

A M3 foi a terceira competição de séries temporais da *Makridakis Competitions*, que foi um evento em que diversos profissionais e pesquisadores da área testaram diversos modelos preditivos, com o intuito de estudar empiricamente quais foram os melhores procedimentos.

Esse conjunto de dados está reunido no pacote *Mcomp* e possui 3003 séries temporais com diferentes tipos de sazonalidade, com os valores históricos para serem utilizados no

ajuste dos modelos e os valores futuros para serem utilizados no cálculo das métricas de acurácia. A Tabela 1 apresenta a distribuição das 3003 séries do conjunto de dados de acordo com o tipo de frequência e a área de origem [6].

Tabela 1: Classificação das 3003 séries temporais da competição M3

Frequência	Tipo de série temporal						Total
	Micro	Indústria	Macro	Finança	Demográfico	Outro	
Anual	146	102	83	58	245	11	645
Trimestral	204	83	336	76	57		756
Mensal	474	334	312	145	111	52	1428
Outro	4			29		141	174
Total	828	519	731	308	413	204	3003

Nota-se que esse conjunto de dados reúne uma coleção diversa de séries temporais e, por conta disso, foi selecionado para ser utilizado neste estudo. Para avaliar a capacidade preditiva das abordagens consideradas, os seguintes números de observações futuras para previsão foram utilizados, conforme o tipo de frequência da série temporal, veja Tabela 2.

Tabela 2: Número de previsões requeridas de acordo com o tipo de cada série

Frequência	Número de previsões
Anual	6
Trimestral	8
Mensal	18
Outro	8

1.4 A função *auto.arima()*

A função *auto.arima()* do pacote *forecast* [5] encontra de forma totalmente automática uma configuração do modelo ARIMA que melhor se ajusta a série temporal observada, conforme uma árvore de decisão baseada em vários critérios técnicos. Além disso, disponibiliza uma série de funcionalidades úteis para aprimorar o modelo.

A equação do modelo utilizada para um processo ARIMA não sazonal $ARIMA(p, d, q)$ é definida, neste caso, com a adição do termo c :

$$\phi(B)(1 - B)^d y_t = c + \theta(B)\epsilon_t$$

tal que se $c \neq 0$, há um polinômio de ordem d na função de forecast.

No caso de um processo ARIMA sazonal $SARIMA(p, d, q) \times (P, D, Q)_s$, a equação também leva em conta a adição do termo c :

$$\phi(B)\Phi(B^s)W_t = c + \theta(B)\Theta(B^s)\epsilon_t$$

tal que se $c \neq 0$, há um polinômio de ordem $d + D$ na função de forecast.

Para encontrar d e D , utiliza-se testes de raízes unitárias, em que a hipótese nula é de que não há raiz unitária. Vale ressaltar que caso fosse empregado um teste de raiz unitária em que a hipótese nula é de que há raiz unitária, isto poderia criar um viés para tomar mais diferenças e, para propósitos de predição, é melhor que esse número de diferenças seja o menor possível.

No caso de uma série temporal não sazonal, considera-se modelos $ARIMA(p, d, q)$ em que d é selecionado, por meio de sucessivos testes de raízes unitárias KPSS. Primeiro, testa-se os dados para existência de raiz unitária. Caso seja significativo, testa-se para existência de raiz unitária dos dados com diferenciação. O procedimento continua até a obtenção do primeiro resultado não significativo.

No caso de uma série temporal sazonal, é considerado um $ARIMA(p, d, q)(P, D, Q)_s$, como a estrutura do modelo, em que s é a frequência sazonal e $D = 0$ ou $D = 1$, dependendo do teste estendido de Canova-Hansen (*Canova e Hansen 1995*). Na implementação desse teste, é permitido qualquer valor de $s > 1$, ao contrário do teste original de Canova e Hansen que apenas disponibiliza valores críticos para $2 < s < 13$. Defina que C_s é o valor crítico para o período sazonal s . Foi feito o gráfico de C_s para os 2 valores para s até 365 e notou-se que eles se ajustam à linha $C_s = 0.269s^{0.928}$ quase perfeitamente. Sendo assim, para $s > 12$, é utilizada a essa simples expressão para obter o valor crítico.

Após a seleção de D , é escolhido d por meio da aplicação sucessiva de testes de raiz-unitária KPSS para os dados sazonalmente diferenciados (se $D = 1$) ou para os dados originais (se $D = 0$).

Após d e possivelmente D serem selecionados, o objetivo é encontrar os valores de p , q , P e Q que minimizam o AIC.

O algoritmo opera na seguinte estrutura:

1. Ajusta-se os seguintes 4 modelos:

- $ARIMA(2, d, 2)$, se $s = 1$; $ARIMA(2, d, 2)(1, D, 1)$, se $s > 1$
- $ARIMA(0, d, 0)$, se $s = 1$; $ARIMA(0, d, 0)(0, D, 0)$, se $s > 1$
- $ARIMA(1, d, 0)$, se $s = 1$; $ARIMA(1, d, 0)(1, D, 0)$, se $s > 1$
- $ARIMA(0, d, 1)$, se $s = 1$; $ARIMA(0, d, 1)(0, D, 1)$, se $s > 1$

Se $d + D \leq 1$, esse modelos são ajustados com $c \neq 0$. Caso contrário, com $c = 0$. Dentre esses quatro modelos, é escolhido o de menor AIC. O modelo selecionado é denominado de modelo atual.

2. Ajusta-se, se possível, os seguintes 13 modelos:

- Um entre os valores p , q , P e Q pode variar em uma unidade para cima ou para baixo do modelo atual.
- p e q variam em uma unidade para cima ou para baixo do modelo atual
- P e Q variam em uma unidade para cima ou para baixo do modelo atual
- a constante c é incluída se o modelo atual tem $c = 0$ ou excluída se o modelo atual tem $c \neq 0$.

Sempre quando um modelo com menor AIC é encontrado, ele passa a ser o modelo atual e o processo é repetido. O processo termina quando não é possível encontrar um modelo perto do modelo atual com menor AIC.

Há diversas restrições no modelos ajustados para evitar problemas com convergência ou raízes-unitárias próximas.

1.5 Argumentos da função `auto.arima()`

Ainda que uma versão default esteja disponível, a função disponibiliza uma série de argumentos que podem ser escolhidos de modo a melhorar o algoritmo que escolhe o melhor modelo. De um modo geral, esses argumentos estão relacionados a escolha do critério de informação, do teste de raiz unitária e da utilização ou não da transformação de Box-Cox. Abaixo, esses argumentos serão detalhados.

1.5.1 Critério de informação

É possível variar o critério de informação com as seguintes opções: Critério de Akaike AIC - argumento `ic = "aic"` - em que $L(\hat{\theta})$ é o valor maximizado da função de verossimilhança do modelo e m é o número de parâmetros estimados, dado por

$$AIC = -2 \ln(L(\hat{\theta})) + 2m$$

Outra opção, por default, é o Critério de Akaike Corrigido AIC_c (`ic = "aicc"`), em que n é o número de pontos de dados, definido por

$$AIC_c = AIC + \frac{2m(m+1)}{n-m-1}$$

E, também, o Critério de Informação Bayesiano BIC (`ic = "bic"`) determinado por

$$BIC = -2 \ln(L(\hat{\theta})) + m \ln(n)$$

1.5.2 Teste de raiz unitária

Para testes de raiz unitária, as alternativas são as seguintes: por, default, Teste Kwiatkowski-Phillips-Schmidt-Shin (`test = "kpss"`), em que a hipótese nula é de que a série temporal tem uma raiz estacionária contra a alternativa de raiz unitária. Outra opção é o Teste de Dickey-Fuller Aumentado (`test = "adf"`) e o Teste Phillips-Perron (`test = "pp"`), em que ambos possuem a hipótese nula de que a série temporal tem raiz unitária contra a alternativa de raiz estacionária.

1.5.3 Testes sazonais de raiz unitária

Para testes sazonais de raiz unitária, há diversas escolhas possíveis:

- Se `seasonal.test = "seas"` (default), uma medida de força sazonal é utilizada, em que a diferenciação é selecionada se a força sazonal (Wang, Smith & Hyndman, 2006) excede 0,64 (baseado na minimização de MASE quando é feita predição `auto.arima` com os dados da M3 e M4)
- Se `seasonal.test = "ch"`, então o Teste de Canova-Hansen (1995) é usado, com hipótese nula de sazonalidade determinística
- Se `seasonal.test = "hegy"`, o Teste de Hylleberg, Engle, Granger & Yoo (1990) é usado
- Se `seasonal.test = "ocsb"`, o Teste de Osborn-Chui-Smith-Birchenhall (1988) é utilizado, com hipótese nula de que a raiz unitária sazonal existe

1.5.4 Transformação Box-Cox

É possível também optar para selecionar automaticamente um parâmetro de transformação Box-Cox (`lambda = "auto"`), ou ignorar essa possibilidade, por default, `lambda = "NULL"`.

1.6 Abordagens

Este estudo contempla a avaliação do impacto na acurácia gerado pela alteração dos argumentos da função `auto.arima()`, por meio de 10 abordagens próprias da função, sendo uma com todos os argumentos default (`Default`), duas alterando o critério de informação (`ic`), duas alterando o teste de raiz unitária (`test`), três alterando o tipo de teste sazonal (`seasonal.test`) e uma para a transformação de Box-Cox (`lambda`).

Além das 10 abordagens baseadas na mudança dos argumentos próprios da função `auto.arima()`, outras 4 abordagens foram testadas com o intuito de aprimorar as previsões. Essas últimas quatro abordagens utilizam a forma default da função, no entanto, elas variam no tratamento em que é dado às séries temporais externamente à função `auto.arima()`. Esses tratamentos estão relacionados à exclusão de valores extremos e alguns tipos de técnicas de decomposição, que serão detalhados a seguir.

1.6.1 Exclusão de valores extremos

Esta abordagem consiste na execução do seguinte procedimento: primeiro, se ajusta o modelo ARIMA, com os argumentos padrões da função `auto.arima()`. Em seguida, é identificado, por meio dos resíduos, quais pontos podem ser considerados extremos, com o a utilização de três desvios-padrão. Depois, substitui-se os valores extremos da série pelos valores ajustados. Finalmente, ajusta-se novamente o modelo ARIMA na série resultante e as métricas de predição são computadas.

1.6.2 Decomposição Clássica Aditiva

Uma série temporal pode ser decomposta em três componentes, que seriam mais simples de serem modeladas. Na Decomposição Clássica Aditiva, a série temporal assume a seguinte estrutura:

$$X_t = T_t + S_t + E_t$$

Em que T_t é a componente de tendência, S_t é a componente de sazonalidade e E_t representa o ruído, ou seja, a parte não explicada, que se espera ser puramente aleatória.

Para calcular as predições, primeiro foi subtraída a componente S_t da série temporal X_t .

$$X_t - S_t = T_t + E_t$$

Em seguida, é ajustado o modelo ARIMA com a função `auto.arima()` para a série temporal sem sazonalidade e, em seguida, são estimados os *forecasts*. Após isso, é computada a predição para a componente sazonal S_t por meio de um modelo $ARIMA(0, 0, 0)(0, 1, 0)_s$, com a função `snaive()`. Finalmente, essas duas estimativas são somadas e essa soma constitui a predição pontual final.

É importante dizer que essa decomposição foi aplicada apenas para as séries temporais de período mensal e trimestral, pois, para as séries anuais, não há frequência sazonal e,

para as séries do tipo outros, é desconhecido o parâmetro s . Sendo assim, para esses dois casos em que não é possível modelar a sazonalidade, as previsões foram calculadas por meio do ajuste de um modelo ARIMA, com as opções padrão da função `auto.arima()`.

1.6.3 Decomposição Clássica Multiplicativa

Na Decomposição Clássica Multiplicativa, a série temporal assume a seguinte estrutura:

$$X_t = T_t \times S_t \times E_t$$

As três componentes são iguais às da decomposição clássica aditiva, porém agora a série temporal é o produto delas.

Para calcular as previsões, primeiro foi feita a divisão entre a série temporal X_t e a componente S_t .

$$\frac{X_t}{S_t} = T_t \times E_t$$

Em seguida, é ajustado o modelo ARIMA com a função `auto.arima()` para a série temporal sem sazonalidade e, em seguida, são estimados os *forecasts*. Após isso, é computada a previsão para a componente sazonal S_t por meio de um modelo $ARIMA(0, 0, 0)(0, 1, 0)_s$, com a função `snaive()`. Finalmente, essas duas estimativas são multiplicadas e esse produto constitui a previsão pontual final.

Novamente, essa decomposição foi aplicada apenas para as séries temporais de período mensal e trimestral, pelo fato já explicado anteriormente e, para as outras séries temporais, as previsões foram calculadas por meio do ajuste de um modelo ARIMA, com as opções padrão da função `auto.arima()`.

1.6.4 Decomposição por Seasonal e Trend Loess

O método STL, que é uma abreviação para Seasonal and Trend decomposition using Loess, consiste em uma abordagem versátil e robusta, para decompor séries temporais, capaz de capturar relações não lineares com a utilização do Loess. A decomposição por Seasonal e Trend Loess foi desenvolvida por Robert B. Cleveland, William S. Cleveland, Jean E. McRae e Irma Terpenning [1], em 1990.

No STL, a componente sazonal pode mudar ao longo do tempo e poder ser mais robusta para lidar com valores muito discrepantes. Esse método funciona para o caso da Decomposição Clássica Aditiva. Então, sua decomposição é basicamente a soma das componentes. Neste caso, foi escolhida uma janela com número de defasagens ímpar e maior que 7, como foi orientado no trabalho científico *STL: A Seasonal Trend Decomposition Procedure Based on Loess* [1]. Dito isso, foi escolhida uma janela de tamanho 11. Além disso, esse procedimento também foi executado com ajuste robusto.

Para calcular as previsões, primeiro foi feita a subtração da componente sazonal S_t da série temporal X_t .

$$X_t - S_t = T_t + E_t$$

Em seguida, é ajustado o modelo ARIMA com a função `auto.arima()` para a série temporal sem sazonalidade e, em seguida, são estimados os *forecasts*. Após isso, é computada a previsão para a componente sazonal S_t por meio de um modelo $ARIMA(0, 0, 0)(0, 1, 0)_s$,

com a função `snaive()`. Finalmente, essas duas estimativas são somadas e essa soma constitui a predição pontual final.

Novamente, essa decomposição foi aplicada apenas para as séries temporais de período mensal e trimestral, pelo fato já explicado anteriormente e, para as outras séries temporais, as predições foram calculadas por meio do ajuste de um modelo ARIMA, com as opções padrão da função `auto.arima()`.

1.7 Métricas sMAPE e MASE

Com o intuito de avaliar o desempenho das predições derivadas pelos diversos métodos e abordagens elencados, duas métricas foram escolhidas. Uma delas é a sMAPE (Symmetric Mean Absolute Percentage Error), que corresponde a média do seguinte componente:

$$sAPE = 200 \frac{|Y_{n+i} - \hat{Y}_{n+i|n}|}{|Y_{n+i}| + |\hat{Y}_{n+i|n}|}$$

A outra métrica utilizada, ao longo deste artigo, é a MASE (Mean Absolute Standard Error), que consiste também em uma média, porém agora do seguinte componente:

$$ASE = \frac{|Y_{n+i} - \hat{Y}_{n+i|n}|}{(n-m)^{-1} \sum_{t=m+1}^n |Y_t - Y_{t-m}|}$$

Nessas duas equações apresentadas, tem-se as seguintes definições: n representa o número de dados históricos da série temporal; Y_{n+i} , a observação i passos à frente da origem n ; $\hat{Y}_{n+i|n}$, predição i passos à frente da origem n e m corresponde ao número de períodos dentro de um ano (1 para anual, 4 para Trimestral, 12 para mensal e 1 para outros)

Finalmente, é importante salientar que, para ambas métricas, um menor valor quando comparado com outros derivados de distintos métodos de modelagem representa uma melhor predição.

2 Resultados

Tabela 3: Resultados para as métricas sMAPE e MASE

Métrica sMAPE					
Argumentos	Anual	Trimestral	Mensal	Outro	Tudo
Default	17,10	10,01	14,92	4,51	13,95
ic = "aic"	17,56	9,97	14,99	4,54	14,05
ic = "bic"	17,14	10,10	14,98	4,33	14,01
test = "adf"	21,66	10,83	15,25	5,00	14,81
test = "pp"	21,45	10,27	15,61	4,93	14,94
seasonal.test = "ocsb"	17,10	9,90	14,76	4,51	13,83
seasonal.test = "hegy"	17,10	10,78	17,55	4,51	15,91
seasonal.test = "ch"	17,10	10,21	15,16	4,51	14,16
lambda = "auto"	20,25	10,28	14,32	4,49	13,91

Métrica MASE					
Argumentos	Anual	Trimestral	Mensal	Outro	Tudo
Default	2,96	1,19	0,87	1,84	1,18
ic = "aic"	2,99	1,18	0,87	1,83	1,18
ic = "bic"	3,00	1,19	0,88	1,80	1,19
test = "adf"	3,54	1,32	0,91	2,06	1,29
test = "pp"	3,52	1,23	0,92	2,02	1,29
seasonal.test = "ocsb"	2,96	1,18	0,87	1,84	1,18
seasonal.test = "hegy"	2,96	1,18	0,87	1,84	1,22
seasonal.test = "ch"	2,96	1,22	0,89	1,84	1,20
lambda = "auto"	6,65	1,26	1,01	1,88	1,67

Tabela 4: Resultados para as métricas sMAPE e MASE

Métrica sMAPE					
Abordagem	Anual	Trimestral	Mensal	Outros	Tudo
Exclusão de Valores Extremos	17,80	9,90	14,76	4,71	13,91
Decomposição Aditiva	17,10	9,91	17,30	4,51	15,59
Decomposição Multiplicativa	17,10	9,96	16,32	4,51	14,92
Seasonal and Trend Loess	17,10	9,65	15,94	4,51	14,61

Métrica MASE					
Abordagem	Anual	Trimestral	Mensal	Outros	Tudo
Exclusão de Valores Extremos	3,09	1,19	0,88	2,02	1,21
Decomposição Aditiva	2,96	1,2	0,93	1,84	1,22
Decomposição Multiplicativa	2,96	1,18	0,92	1,84	1,21
Seasonal and Trend Loess	2,96	1,17	0,87	1,84	1,18

Para as séries anuais, não houve tanto para métrica sMAPE quanto para a MASE um destaque positivo discrepante. O melhores valores encontrados foram de 17,10 e 2,96, respectivamente. Contudo, houve claramente um desempenho pior com a utilização da transformação Box-Cox (`lambda = "auto"`).

Para o caso do período trimestral, nota-se que por meio da abordagem de predição da série temporal que foi recomposta por meio de Seasonal and Trend Loess, ambas as

métricas assumirem os valores mais baixos. 9,65 para sMAPE e 1,17 para MASE. Como foi enunciado na seção 1, a modelagem estatística ARIMA para séries com comportamento explosivo pode ser prejudicada. Sendo assim, a seleção automática do melhor modelo pela função `auto.arima()` de início já recebe dados mais robustos para outliers e, além disso, a componente sazonal foi alterada ao longo do tempo.

As séries mensais, por sua vez, apresentaram resultados mais complexos de se analisar. Em primeiro lugar, vale ressaltar que para a métrica MASE a abordagem de Seasonal and Trend Loess foi a melhor, porém se for analisada a métrica sMAPE, esta assumiu o maior valor, ou seja, um resultado pior. Em segundo lugar, a transformação Box-Cox foi a responsável pelo melhor desempenho preditivo para a métrica sMAPE. Contudo, para a métrica MASE, essa abordagem resultou no pior valor.

Sendo assim, é proposto que para as séries mensais, o destaque seria para a abordagem de Exclusão de Valores Extremos, uma vez que para ambas as métricas, obteve-se resultados satisfatórios, quando comparados com os das outras abordagens.

É importante ressaltar que a abordagem de Exclusão de Valores Extremos atuou por meio da substituição de outliers com três desvios-padrão de distância da média dos resíduos por valores ajustados, o que nesse caso possibilitou uma maior suavização dos dados antes da seleção automática do melhor modelo ARIMA.

Para as séries do tipo Outro, a utilização do Critério de Informação Bayesiano (`ic = "bic"`) gerou os melhores resultados para ambas as métricas. Esse argumento tem um impacto profundo ao longo de toda a árvore de decisão executada pela função `auto.arima()`, uma vez que é constantemente usado para selecionar o melhor modelo, em cada etapa. Neste caso, é importante ressaltar que o Critério de Informação Bayesiano penaliza mais modelos com maior complexidade, em comparação com o Critério de Akaike.

Finalmente, se for considerado todo o conjunto das 3003 séries temporais, para Erro Percentual Absoluto Médio Simétrico, a utilização do teste sazonal de raiz unitária de Osborn-Chui-Smith-Birchenhall gerou um melhor resultado, apesar de não ter sido por uma grande diferença. O Erro Médio Absoluto de Escala, por sua vez, obteve seu melhor valor também com utilização desse teste, mas a função `auto.arima()` padrão e a utilização do Critério de Akaike também levaram ao mesmo resultado.

É relevante analisar o melhor desempenho dessas variações discutidas acima do modelo ARIMA, em comparação com outros modelos, conforme o período. Sendo assim, se for examinada a tabela 4: *Empirical results for all methods using the sMAPE and the MASE* do artigo *Models for optimising the theta method and their relationship to state space models* [3], pode-se avaliar essa comparação.

Para as séries anuais e a função default de `auto.arima()`, a métrica sMAPE apresentou claramente um pior resultado, quando comparada com o método Simple Exponential Smoothing with drift model - SES-d ($b = B_n/2$) - e os métodos Theta: Dynamic Standard Theta Model (DSTM), Dynamic Optimised Theta Model (DOTM), Optimised Theta Model (OTM) e Standard Theta Model (STM). Contudo, teve claramente um melhor desempenho, em relação a Naive, Simple Exponential Smoothing (SES) e SES-d (b optimised). A métrica MASE, por sua vez, apresentou um menor valor apenas para os métodos Naive e SES.

No caso das séries trimestrais e a abordagem de Seasonal and Trend Loess, para ambas as métricas, os resultados foram razoavelmente competitivos, quando comparados com o desempenho dos melhores métodos Theta, STheta e os métodos de suavização exponencial com modelo de drift, apesar de terem apresentado valores maiores.

Para as séries mensais e a abordagem de Exclusão de Valores Extremos, houve signi-

ficativamente uma melhor performance de predição da métrica sMAPE, somente quando é comparada com os métodos NAIVE, SES, Damped no caso de dados originais e Damped para dados sazonalmente ajustados. No caso da métrica MASE, houve um resultado bem competitivo de 0,88, uma vez que é bastante próximo do menor valor de 0,85 que foi encontrado pelo método DOTM.

A utilização do Critério de Informação Bayesiano nas séries de período Outro, por sua vez, assume grande competitividade, visto que, para ambas as métricas, valores baixíssimos que quase se igualam ao desempenho dos melhores modelos foram encontrados.

No caso mais geral que engloba as 3003 séries temporais e a utilização do teste sazonal de raiz unitária de Osborn-Chui-Smith-Birchenhall (`seasonal.test = 'ocsb'`), para ambas as métricas, de um modo geral, houve um pior desempenho de predição, mas não por uma expressiva diferença.

3 Conclusão

- Para as séries anuais, não há um destaque para uma abordagem ou mudança de argumento. A implementação default da função, quando comparada com outros métodos, obteve um desempenho intermediário.
- No caso das séries trimestrais, houve um destaque para abordagem de Seasonal and Trend Loess. Em comparação com outros modelos, há um desempenho razoavelmente competitivo, para ambas as métricas, com essa combinação.
- Em relação às séries mensais, há a proeminência da abordagem de Exclusão de Valores Extremos. Em relação a outros modelos, essa combinação resultou em uma predição bastante competitiva para a métrica MASE, com essa configuração.
- No que se refere às séries de período Outro, é notório um melhor desempenho do Critério de Informação Bayesiano. Em comparação com as outras modelagens, apresentou resultados bastante satisfatórios para ambas as métricas, com a utilização desse critério.
- Para todas as 3003 séries temporais, há uma maior, mas não expressiva, importância para o teste sazonal de raiz unitária de Osborn-Chui-Smith-Birchenhall. Em comparação com outros modelos, essa combinação apresenta em geral um pior desempenho, apesar da diferença não ser tão expressiva.
- Para todas as 3003 séries temporais, os resultados mostram que a versão default da função `auto.arima()` está bem calibrada, uma vez que as variações aqui estudadas obtiveram resultados sempre competitivos.

O desenvolvimento de modelos de previsão de séries temporais atrai atenção de diversas áreas dada a sua importância e aplicabilidade e várias abordagens estão constantemente sendo elaboradas e testadas em competições [7]. Outras abordagens envolvendo o modelo ARIMA poderiam ser propostas utilizando validação cruzada [4], bem como a combinação desses modelos com técnicas de aprendizado de máquina [8].

Referências

- [1] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition. *Journal of official statistics*, 6(1):3–73, 1990.
- [2] Ricardo S Ehlers. Análise de séries temporais. *Laboratório de Estatística e Geoinformação. Universidade Federal do Paraná*, 2007.
- [3] Jose A. Fiorucci, Tiago Pellegrini, Francisco Louzada, Fotios Petropoulos, and Anne Koehler. Models for optimising the theta method and their relationship to state space models. *International Journal of Forecasting*, 32, 02 2016.
- [4] Jose Augusto Fiorucci and Francisco Louzada. GROEC: combination method via generalized rolling origin evaluation. *International Journal of Forecasting*, 36(1):105–109, 2020.
- [5] Rob J Hyndman and Yeasmin Khandakar. Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 26(3):1–22, 2008.
- [6] Spyros Makridakis and Michele Hibon. The m3-competition: results, conclusions and implications. *International journal of forecasting*, 16(4):451–476, 2000.
- [7] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020.
- [8] Pablo Montero-Manso, George Athanasopoulos, Rob J Hyndman, and Thiyanga S Talagala. Fforma: Feature-based forecast model averaging. *International Journal of Forecasting*, 36(1):86–92, 2020.
- [9] Pedro A Morettin and Clélia Toloi. Análise de séries temporais. In *Análise de séries temporais*, pages 538–538. 2006.
- [10] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020.