ISSN 2318-2377

Cedeplar
UFMG

TEXTO PARA DISCUSSÃO N° 661

TIME SERIES FORECASTING:
A TEST OF AUTOMATED ECONOMETRIC METHODS

Erick Inácio Ferreira
Igor Viveiros Melo Souza

Outubro de 2023

http://www.cedeplar.ufmg.br

Textos para Discussão

A série de Textos para Discussão divulga resultados preliminares de estudos desenvolvidos no âmbito do Cedeplar, com o objetivo de compartilhar ideias e obter comentários e críticas da comunidade científica antes de seu envio para publicação final. Os Textos para Discussão do Cedeplar começaram a ser publicados em 1974 e têm se destacado pela diversidade de temas e áreas de pesquisa.

Ficha catalográfica

F383t
2023

Time series forecasting : a test of automated econometric methods / Erick Inácio Ferreira, Igor Viveiros Melo Souza. - Belo Horizonte: UFMG / CEDEPLAR, 2023.

33 p. - (Texto para discussão, 661)
Inclui bibliografia.
ISSN 2318-2377
1. Econometria. 2. Análise de séries temporais. I. Souza, Igor Viveiros Melo. II.. Universidade Federal de Minas Gerais. Centro de Desenvolvimento e Planejamento Regional. III. Título. IV. Série.

CDD:330.028

Elaborado por Adriana Kelly Rodrigues CRB-6/2572
Biblioteca da FACE/UFMG. – AKR/132/2023

UNIVERSIDADE FEDERAL DE MINAS GERAIS
FACULDADE DE CIÊNCIAS ECONÔMICAS
CENTRO DE DESENVOLVIMENTO E PLANEJAMENTO REGIONAL

TIME SERIES FORECASTING: A TEST OF AUTOMATED ECONOMETRIC METHODS

Erick Inácio Ferreira
UFMG

Igor Viveiros Melo Souza
UFMG

CEDEPLAR/FACE/UFMG
BELO HORIZONTE
2023

SUMMARY

ABSTRACT

The aim of this study is to assess the performance of two well-known algorithms which automate the process of modeling and forecasting time series, each applying a different econometric technic: ARIMA or exponential smoothing. We provide a brief discussion of how these algorithms work and results of a Monte Carlo experiment, which was conducted to evaluate the capabilities of auto.arima and ets, available in Rob Hyndman's forecast package for the statistical software R, commonly used by economists to study and forecast time series. Over 200.000 synthetic series were simulated, with several different characteristics, used to test both methods and report metrics of correct modeling and out-of-sample forecast errors of the algorithms, on top of which we provide a brief discussion of the successes and shortcomings that happened while applying each algorithm.

Keywords: *Time series econometrics, ARIMA, exponential smoothing, auto.arima.*

RESUMO

O objetivo deste estudo é avaliar a performance de dois algoritmos populares que automatizam o processo de modelar e prever valores futuros de séries temporais, cada um aplicando uma técnica econométrica diferente: ARIMA ou suavização exponencial. É fornecida uma breve discussão de como esses algoritmos funcionam e os resultados de um experimento de Monte Carlo, que foi conduzido para avaliar as capacidades do auto.arima e do ets, disponíveis no pacote forecast, de Rob Hyndman, para o programa estatístico R, frequentemente usado por economistas para estudar e fazer previsões de séries temporais. Mais de 200.000 séries sintéticas foram simuladas, com características diversas, que foram usadas para testar os métodos e reportar suas taxas de acerto na modelagem e métricas de erros de previsão fora da amostra, com uma breve discussão dos acertos e falhas que ocorreram quando aplicamos cada algoritmo.

Palavras-chave: Econometria de séries temporais, ARIMA, suavização exponencial, *auto.arima.*

1. INTRODUCTION

In the time series literature, to compare the predictive capacity of different methods, it is usual to apply the methods in a large number of series and compute their forecast errors for one or more steps ahead (HYDMAN, 2020). An applied example of this technique is provided by the M Competitions, which in its fourth edition, realized in 2019, had 61 forecasting algorithms, tested in 100.000 real-world time series, with various characteristics, such as trends, seasonality and variable variance (MAKRIDAKIS; SPILIOTIS; ASSIMAKOPOULOS, 2020).

The need to apply a forecasting method in thousands of series requires the development of a computational algorithm that autonomously sets its parameters, aiming to get the best forecasts possible. In the case of the ARIMA method, it is required to choose a model, such as an AR(1), to make forecasts for the next values of a series. Regarding the exponential smoothing method, Hyndman et al. (2008) showed that there are 18 different models for use, each own being optimal for modeling the presence and association of various characteristics of a time series, such as trend and/or seasonality. It is required to choose the appropriate model for the process before forecasting, otherwise, the forecasts will not be optimal. Willing to make this process of choosing a model and producing forecasts automatically, Hyndman and Khandakar (2008) developed the algorithms auto.arima and ets, each trying to fit an appropriate ARIMA or exponential smoothing model to a time series, respectively, and enabling, if needed, to produce forecasts for any steps ahead. Both algorithms are available for free in the package forecast, for use in the popular statistical software R (R CORE TEAM, 2019).

This work is about those two algorithms, whose performance was tested in simulated time series, given that both have been already extensively assessed in real-world time series (MAKRIDAKIS; SPILIOTIS; ASSIMAKOPOULOS, 2020). In Chapter 2 we detail the method used for this study and some hypotheses for tests. Chapter 3 reports our results and a brief discussion about them. Lastly, chapter 4 presents our conclusions.

2. METHODOLOGY

Due to the advance in the processing power and availability of computers during the second half of the twentieth century, many automated algorithms for time series forecasting started being assessed in real-world applications, being the exponential smoothing ones the most used in industry and business (HYNDMAN, 2020). Since 1979, a handful of competitions have been held, aiming to test those algorithms in real-world time series. Most of them received the name of M Competitions, in honor of Spyros Makridakis, who organized the events. The last one so far, M4, occurred in 2020, with over 60 algorithms for forecasting being applied to more than 100,000 real-world time series from several fields, such as demography, industrial supply, macroeconomics, labor, finance and others. Among the other methods being tested, auto.arima and ets ranked among the top 23, with an advantage for the former

(MAKRIDAKIS S.; SPILIOTIS E.; ASSIMAKOPOULOS. V., 2020). The conclusions reported by the M Competitions analysts are similar to the first one, M1, which was made with only 3.000 time series. The first one is that more complex methods, in terms of operation and non-linearity, do not perform better than simpler techniques. The other one is that there is not a preferred algorithm for use, for being more accurate than any other, in terms of out-of-sample forecasting error (HYNDMAN, 2020).

Another applied use of auto.arima and ets is shown by the United Kingdom's Office for National Statistics (ONS) (2008). In their report, the British agency states that they use automated forecasting methods to inform point predictions for multiple series, which are used by other agencies for planning an adequate supply of public services and goods. Forecasts are made for time series of several fields, such as education, social assistance, financial costs of fire departments, and many others. Until 2008, ONS used an automated exponential smoothing method, which relied on the Holt-Winters variation. However, in 2008 they were forced to replace this software, for technical reasons. This new software also forecasts automatically, but using the ARIMA method. Before ending the transition, ONS made forecasts for their series using both algorithms and comparing their errors, using the Mean Absolute Percentage Error (MAPE), which is an average between the absolute value of the errors and the actual value of the series. The results led ONS to conclude that there was virtually no difference regarding the MAPE of each method, at least for the ones made one step ahead.

The methodology we chose for this work is a Monte Carlo simulation. According to Enders (2014), this method is the workhorse of modern time series research, being particularly useful for assessing the properties of small – or finite – samples. The technique consists of using a computer to generate data from a specified process of interest. The simulation runs by creating thousands of random samples of random numbers for the process. Each sample will yield its own statistics, given that they are random variables, allowing us to collect and build the probability distribution of the process, its critical values and confidence intervals. Enders (2014) states that the reason for running a Monte Carlo simulation is the Law of Large Numbers, which ensures that the sample mean converges to the populational mean as the sample size increases, resulting in unbiased estimates for the population parameters.

In our simulation, we collected the forecasting errors made by auto.arima and ets when applied in synthetic time series generated by ARIMA models. It is worth noting that both algorithms were used as benchmarks in the last M Competition, whose real-world time series comes from unknown data-generating processes. Spiliotis et al. (2020) argue that those series have characteristics that can potentially bias the algorithm's performance. Therefore, the reasoning for using simulated series is knowing exactly which is the stochastic process generating the data, allowing us to assess if the algorithms are detecting correctly the features of the processes, such as trends, cycles, seasonality, and others that may happen in time series. Both auto.arima and ets work by fitting many models to the series to which they are applied and choosing the one with the smallest information criteria, that is Akaike's one by default. This work reports the results of how often these algorithms choose the correct model for the process being analyzed and also four common forecast error metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) and Root Mean Absolute Percentage Error (RMSPE). The stochastic processes are five:

A) A stationary AR(1);

B) A stochastic trend ARIMA(1,1,1);

C) A trend stationary with ARMA(1,1)

D) A seasonal SARIMA(0,0,0)(1,0,1)[4]

E) A fractionally integrated ARFIMA with d = 0.4.

For processes A, B and C, 10,000 series were generated, ranging from 100 up to 1,000 observations. The random shocks followed three different random variables: a Gaussian one, with mean 0 and variance 1, a T Student with 3 degrees of freedom and, lastly, a chi-squared with two degrees of freedom and focused in zero. The reason for using such random variables is to assess how algorithms are affected by them since in applied studies we often find out that the shocks cannot be considered a random normal variable. In all series, we added a mean u = 100, avoiding having series with too small observations, which would lead to huge percentage errors. For computational reasons, the ARFIMA process made 5,000 time series and the seasonal one only a thousand. For processes A, B and E, auto.arima and ets received only the first n-1 observations of each time series, being n the length of the series, and forecast for one step ahead. In the case of process C, both algorithms got the first n-5 observations and produced forecasts for the next five steps ahead. Finally, for process D, the algorithms received the first n-4 values of each series and forecasted for the next four steps ahead.

Before carrying out the simulations, we conceived five hypotheses, listed below:

Both algorithms choose the model correctly 80% of the time for the series to which they are applied, and this rate grows as the series gets larger;

The random variable of the shocks does not affect the rate at which the algorithms choose the correct model for the series;

Forecasting errors are similar for both algorithms, not exceeding 10% of each other;

Forecasting errors get smaller as the time series grows larger;

All the last hypotheses are valid independently of which information criteria is used for running the algorithms.

3. RESULTS

We start by showing the results obtained for the stationary AR(1) series. The models used followed this equation:

$$y_t = u + \phi y_{t-1} + \varepsilon_t \quad (3.1)$$

With $u = 100$, $\phi = 0.5$ or 0.8 and ε a random independent and identically distributed variable (the random shock, or disturbance).

Table 1 presents the rate of correct modeling of the algorithms when the shocks followed a normal distribution $N\sim(0,1)$, with n being the length of the series, ranging from 100 up to 1,000.

Table 1 — Rate of correct modeling for AR(1) process with normal random shocks

Length (n)	Algorithm	Criteria	AR(1) - ϕ = 0,5		AR(1) - ϕ = 0,8	
			Correct (%)	Incorrect (%)	Correct (%)	Incorrect (%)
100	*auto. arima*	AIC	57,06	42,94	42,41	57,59
100	*ets*	AIC	50,17	49,83	54,73	45,27
100	*auto. arima*	BIC	70,20	29,80	55,30	44,70
100	*Ets*	BIC	51,52	48,48	54,88	45,12
250	*auto.arima*	AIC	56,30	43,70	43,83	56,17
250	*Ets*	AIC	53,29	46,71	57,45	42,55
250	*auto.arima*	BIC	81,52	18,18	63,17	36,83
250	*Ets*	BIC	53,62	46,38	57,46	42,54
500	*auto.arima*	AIC	59,00	41,00	49,77	50,23
500	*Ets*	AIC	55,92	44,08	60,66	39,34
500	*auto.arima*	BIC	87,48	12,52	73,62	26,38
500	*Ets*	BIC	56,07	43,93	60,69	39,31
1000	*auto.arima*	AIC	60,76	39,24	53,03	46,97
1000	*Ets*	AIC	58,57	41,43	65,19	34,81
1000	*auto.arima*	BIC	90,09	9,91	79,14	20,86
1000	*Ets*	BIC	58,63	41,37	65,25	34,75

Source: self elaboration.

Notice that auto.arima's performance was consistently better in detecting correctly the process generating the series when ϕ=0.5, independently of what information criteria was used. However, this did not happen when ϕ=0.8, a case when ets performed better, at least when used with AIC. That reinforces the discussion of Enders (2014), who points out that one should use BIC for modeling because AIC would be inconsistent. The author's argument is verified in table 1, given that the use of AIC did not yield significantly better results as series grew, whereas BIC did. The difference in the rate of correct modeling only improved up to 10.62 percentual points for AIC, and, in the most extreme scenario, the use of BIC returned the correct model 90.09% of times, which is 29.33 percentual points more than AIC in the same scenario.

The discussion in the last paragraph does not hold for the algorithm ets. Usually, both information criteria chose the same model, independently of the series' length. Unlike auto.arima, both AIC and BIC led ets to choose de correct model as the series grew larger. However, this performance increase was smaller, being about only 10 percentual points, making ets model choosing performance worse for AR(1) time series, if compared to auto.arima, even though the latter had much more models to choose from than the former.

If we compare the rate of correct modeling of both algorithms when we increase the autoregressive parameter (ϕ), it is noted that ets' performance increases, whereas auto.arima's decreases, for both information criteria. Anyway, this small increase in ets' performance should be ignored, given that this algorithm has a much smaller pool of models to choose from, especially if compared to auto.arima. Since we did not define any seasonal frequency when generating the series, both algorithms had no extra work checking if the process had seasonality, so ets needed only to decide if the series contained a trend and if the random part variance was stable or increasing through time, while auto.arima could choose many ARMA models to fit the process. Nevertheless, the automated exponential smoothing algorithm ended up choosing the wrong model more frequently, noticing an increasing variance in the processes, that do not exist in AR(1) time series with the parameters reported. The decay of auto.arima's performance is because the increase in the parameter ϕ made the algorithm identify (wrongly) a unit root in the series.

Table 2 reports the one-step-ahead forecasting errors of the algorithms in the AR(1) series with the parameter ϕ=0.8.

Table 2 – Forecasting errors for AR(1) - ϕ = 0,8 with normal shocks.

Length (n)	Algorithm	Criteria	AR(1) - ϕ = 0,8			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	1,083	1,041	0,830	1,041
100	*Ets*	AIC	1,117	1,057	0,844	1,057
100	*auto.arima*	BIC	1,073	1,036	0,826	1,036
100	*Ets*	BIC	1,116	1,057	0,844	1,057
250	*auto.arima*	AIC	1,026	1,013	0,812	1,013
250	*Ets*	AIC	1,097	1,047	0,840	1,048
250	*auto.arima*	BIC	1,035	1,017	0,815	1,018
250	*Ets*	BIC	1,097	1,047	0,840	1,048
500	*auto.arima*	AIC	1,024	1,012	0,811	1,013
500	*Ets*	AIC	1,100	1,049	0,840	1,050
500	*auto.arima*	BIC	1,022	1,011	0,811	1,012

500	*Ets*	BIC	1,100	1,049	0,840	1,050
1000	*auto.arima*	AIC	1,000	1,000	0,795	1,000
1000	ets	AIC	1,089	1,044	0,835	1,044
1000	*auto.arima*	BIC	1,003	1,002	0,797	1,002
1000	*ets*	BIC	1,089	1,044	0,835	1,044

Source: self elaboration.

Table 2 shows that *auto.arima* yielded better forecasts than *ets*, in all of the error metrics. Another point to be noticed is the irrelevance of the information criteria for forecasting, since the difference between the forecast errors, regarding the use of AIC or BIC is too small. When we had 1,000 observations in each series, a convergence between the error metrics for both criteria, reinforcing our point. Also, there was a small increase in forecast accuracy as the series grew larger.

When we changed the random variable of (3.1) for a Student's T with 3 degrees of freedom, the rate of correcting modeling of both algorithms was similar than the ones reported for a normal variable. The discussion about information criteria and *auto.arima's* rate convergence when using BIC is also held. The results are available in table 10 in the appendix. Regarding forecasting errors for one step ahead using Student's T random variable for the shocks, the greater variability of this probability distribution made errors be larger, but the same points we noted about this topic, regarding normal distribution, also were maintained. This can be seen in table 11 of the appendix. On the other hand, when the shocks followed a chi-squared distribution with two degrees of freedom, *ets* started choosing the wrong model for the series more frequently, returning more often the multiplicative errors model, as if the variable of the series were increasing (which were not), as seen in table 12 of the appendix.

We show next the results of series with trend, starting by the series with stochastic trend, which were produced by an ARIMA(0,1,1) process, which as chosen for being the ARIMA representation of the simple exponential smoothing model (HYNDMAN *et al.*, 2008). The moving average parameter (Θ) was set to 0.5. The results of modeling are shown below, in table 3, with random shocks following a normal distribution.

Table 3 — Rate of correct modeling for ARIMA(0,1,1) with normal shocks.

(continues)

Length (n)	Algorithm	Criteria	ARIMA(0,1,1) - Θ = 0,5	
			Correct (%)	Incorrect (%)
100	*auto.arima*	AIC	58,20	41,80
100	*ets*	AIC	24,71	75,29
100	*auto.arima*	BIC	72,58	27,42
100	*ets*	BIC	47,33	52,67

250	*auto.arima*	AIC	54,18	45,82
250	*ets*	AIC	8,41	91,59
250	*auto.arima*	BIC	83,43	16,57
250	*ets*	BIC	35,86	64,14
500	auto.arima	AIC	55,06	44,94
500	ets	AIC	0,90	99,10
500	auto.arima	BIC	88,19	11,81
500	ets	BIC	11,26	88,74

Table 3 — Rate of correct modeling for ARIMA(0,1,1) with normal shocks.

(conclusion)

Length (n)	Algorithm	Criteria	ARIMA(0,1,1) - θ = 0,5	
			Correct (%)	Incorrect (%)
1000	*auto.arima*	AIC	55,66	44,34
1000	*ets*	AIC	0	100
1000	*auto.arima*	BIC	90,60	9,40
1000	*ets*	BIC	0,37	99,63

Source: Self elaboration

It is noted that ets' performed poorly, inclusive when the series grew larger, up to reaching a zero rate of correct modeling. Usually, ets modeled our stochastic trend series as if they had a damped trend, which does not exist in an ARIMA(0,1,1) process, since it has an unforecastable trend, due to the accumulation of the random shocks (unit root) (BUENO, 2012). Hyndman (2020) states that the damped trend is a variation of Holt's exponential smoothing, used for series with clear and continuous trends. For forecasting purposes, the use of the damped trend performed better than Holt's simple linear trend method (MAKRIDAKIS; SPILIOTIS; ASSIMAKOPOULOS, 2018). However, this result was obtained when forecasting real-world series, which have unknown data-generating process and the continuity of the trend is uncertain. In our simulation, we know the real model that describes the time series, and thus we know there is not any deterministic trend in the process. This way, one should model the series using simple exponential smoothing, proposed originally by Brown (1959).

The algorithm auto.arima had a similar performance, if compared to the stationary series we analyzed. Once more, the use of BIC resulted in a greater rate of correct modeling, which grew for larger series, which did not happen when using AIC. Even in smaller time series, with only 100 observations, the performance of the algorithm was better with BIC, a scenario in which AIC was supposed to perform greater (small samples) (ENDERS, 2014).

Table 4 reports the one step ahead forecasting errors of the two algorithms, for ARIMA(0,1,1) with normal shocks. The incorrect modeling of a deterministic trend impaired ets' performance, in which MSE was about 20% larger than auto.arima's. As in stationary series, the use of larger series did not lead to more accurate forecasts, for both algorithms. However, Hyndman and Koehler (2006) argue that the use of simple error metrics, such as MSE and RMSE, should not be considered for series with trend, because they rely on the series' level. Anyway, our analysis is not changed if we use free-of-scale error metrics – the percentual errors – such as MAPE and RMSPE. The use of another probability distribution for the shocks, as reported in tables 14 to 17 of the appendix, leads to the same conclusions we made here.

Table 4 — Forecasting errors for ARIMA(0,1,1) - θ = 0,5 with normal shocks

Length (n)	Algorithm	Criteria	ARIMA(0,1,1) - θ = 0,5			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	1,069	1,034	0,842	1,073
100	*ets*	AIC	1,246	1,116	0,911	1,160
100	*auto.arima*	BIC	1,059	1,029	0,839	1,067
100	*ets*	BIC	1,262	1,123	0,917	1,166
250	*auto.arima*	AIC	1,019	1,009	0,863	1,125
250	*ets*	AIC	1,193	1,092	0,933	1,214
250	*auto.arima*	BIC	1,017	1,008	0,862	1,125
250	*ets*	BIC	1,217	1,103	0,941	1,225
500	*auto.arima*	AIC	1,019	1,010	1,030	3,637
500	*ets*	AIC	1,165	1,079	1,101	4,702
500	*auto.arima*	BIC	1,015	1,007	1,026	3,619
500	*ets*	BIC	1,174	1,083	1,105	4,704
1000	*auto.arima*	AIC	1,037	1,018	1,997	27,38
1000	*ets*	AIC	1,180	1,086	2,151	29,55
1000	*auto.arima*	BIC	1,035	1,017	1,972	25,87
1000	*ets*	BIC	1,180	1,086	2,151	29,55

Source: self elaboration.

For testing algorithms in series with deterministic trends, we simulated series that followed the process:

$$y_t = 0{,}3t + 0{,}5y_{t-1} + \varepsilon_t + 0{,}5\varepsilon_{t-1} \quad (3.2)$$

With ε_t being a white noise. Therefore, (3.2) is stationary if detrended, what is called trend stationary in time series textbooks (GUJARATI; PORTER, 2011). As explained by Enders (2014), the correct procedure for modeling a trend stationary process is first to filter the deterministic trend, using linear regression techniques, and then apply the Box-Jenkins method in the regression residuals. Differentiating a series generated by (3.2) is incorrect since there is no unit root in the process, resulting in an artificial moving average, that does not exist in the series. Checking if a time series has a unit root can be done by an appropriate test, such as ADF (DICKEY; FULLER, 1981). Note that the regression of a unit root process against a deterministic time variable is not stationary, showing the inadequacy of this method for detrending unit root series. Table 5 shows the rate of correct modeling of the algorithms when applied to the time series described by the process of (3.2).

Table 5 — Rate of correct modeling for *trend stationary* time series with normal shocks.

Length (n)	Algorithm	Criteria	*Trend Stationary*	
			Correct (%)	Incorrect (%)
100	*auto.arima*	AIC	0	100
100	*ets*	AIC	54,64	45,36
100	*auto.arima*	BIC	0	100
100	*ets*	BIC	2,58	97,42
250	*auto.arima*	AIC	0	100
250	*ets*	AIC	92,22	7,78
250	*auto.arima*	BIC	0	100
250	*ets*	BIC	88,80	11,20
500	*auto.arima*	AIC	0	100
500	*ets*	AIC	99,86	0,14
500	*auto.arima*	BIC	0	100
500	*ets*	BIC	99,87	0,13
1000	*auto.arima*	AIC	0	100
1000	*ets*	AIC	99,97	0,03
1000	*auto.arima*	BIC	0	100
1000	*ets*	BIC	99,99	0,01

Source: Self elaboration.

The results obtained reveal a critical failure of auto.arima, because it always took the first difference of the series in which it was applied, independently of the length of the series or the distribution of probabilities of the shocks. The algorithm modeled the synthetic series as having stochastic trend and drift, which is a deterministic trend. Due to incorrectly differentiating all the time series, auto.arima's rate of correct modeling was considered zero, even though it also noted the presence of a deterministic trend. It is worth stating that auto.arima automatically runs unit root tests, which are capable of detecting both types of trends. In larger series and given the ARMA parameters of (3.2), the power of those tests would be enough to conclude that the series had a deterministic trend, but not stochastic (DICKEY; FULLER, 1981).

Regarding ets, there was a clear advantage in correct modeling when using AIC, mainly in smaller series, such as the ones with 100 and 250 observations. In this scenario, the use of BIC made ets choose simple exponential smoothing more often, ignoring the deterministic trend of the series. As the series grew larger, and thus clearer the linear trend, the use of BIC led to the choice of the correct exponential smoothing model more often, but no as much as AIC. This point also happens for other shock distributions, available in tables 18 and 19 of the appendix.

Even though the was an advantage for ets in modeling the trend stationary process, the same did not happen for forecasting. Since the time series of this process had a deterministic trend, the forecast errors were collected five steps ahead. As shown in Table 6, auto.arima had a better performance and smaller error metrics, although it incorrectly modeled the series. The same happened for other types of shocks, as seen in Tables 20 and 21 of the appendix. We also note that, unlike stationary processes, both algorithms started having smaller forecasting errors as the series grew larger.

Table 6 — Forecasting errors for *trend stationary* series with normal shocks.

Length (n)	Algorithm	Criteria	*Trend stationary*			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	52,14	7,221	4,41	5,57
100	*ets*	AIC	65,79	8,111	4,95	6,26
100	*auto.arima*	BIC	62,59	7,911	4,81	6,10
100	*ets*	BIC	80,85	8,992	5,53	6,90
250	*auto.arima*	AIC	46,87	6,846	3,10	3,92
250	*ets*	AIC	58,84	7,671	3,49	4,39
250	*auto.arima*	BIC	46,62	6,828	3,08	3,91
250	*ets*	BIC	60,15	7,756	3,53	4,44
500	*auto.arima*	AIC	45,69	6,760	2,15	2,71
500	*ets*	AIC	59,70	7,727	2,47	3,09

500	*auto.arima*	BIC	43,81	6,619	2,11	2,65
500	*ets*	BIC	59,71	7,727	2,47	3,09
1000	*auto.arima*	AIC	48,06	6,931	1,36	1,73
1000	*ets*	AIC	59,90	7,740	1,54	1,93
1000	*auto.arima*	BIC	46,60	6,827	1,34	1,70
1000	*ets*	BIC	59,91	7,740	1,54	1,93

Source: self elaboration.

Regarding seasonality, we simulated series following a SARIMA(0,0,0)(1,0,1)[4] model, which can be represented by the following equation:

$$y_t = \varepsilon_t + 0{,}5y_{t-4} + 0{,}5\varepsilon_{t-4} \qquad (3.3)$$

In which ε_t is a white noise. This way, (3.3) has no trend and its seasonality is additive and stochastic. The results about correct modeling, when the shocks were normal, are in table 7.

Table 7 — Rate of correcting modeling for SARIMA(0,0,0)(1,0,1)[4] with normal shocks

Length (n)	Algorithm	Criteria	SARIMA(0,0,0)(1,0,1)[4]	
			Correct (%)	Incorrect (%)
100	*auto.arima*	AIC	11,60	88,40
100	*ets*	AIC	46,50	53,50
100	*auto.arima*	BIC	15,60	84,40
100	*ets*	BIC	45,50	54,50
250	*auto.arima*	AIC	5,70	94,30
250	*ets*	AIC	47,20	52,80
250	*auto.arima*	BIC	17,20	82,80
250	*ets*	BIC	47,60	52,40
500	*auto.arima*	AIC	4,80	95,20
500	*ets*	AIC	47,80	52,20
500	*auto.arima*	BIC	22,40	77,60
500	*ets*	BIC	48,80	51,20
1000	*auto.arima*	AIC	4,30	95,70
1000	*ets*	AIC	49,30	50,70
1000	*auto.arima*	BIC	24,80	75,20

1000	*ets*	BIC	49,90	50,10

Source: self elaboration.

Both algorithms had a rate of correct modeling below 50%, with auto.arima not even reaching 30%. On the other hand, this method noticed the presence of seasonality and its correct frequency in 95% of the series, even in the smaller ones, with only 100 observations. The mistakes made are due to fitting parameters that do not exist in the real data-generating process, and incorrectly differentiating the seasonal part of the series, which did not have a unit root, inducing an artificial moving average. On the other hand, ets' mistakes are related to not identifying the presence of seasonality, mainly when applied using BIC and in smaller series, or marking the series variance as increasing, which was not the case. It should be noticed that both algorithms did not increase their rate of correct modeling as the series grew larger, except for auto.arima, when using BIC, which had an increase in performance, but was still limited to 30%. These results also happened for other distributions of shocks, as reported by tables 22 and 23 in the appendix. Once more, ets modeled multiplicative variance more often for series made with shocks that followed a chi-squared random variable.

Regarding the forecasting errors, table 8 presents the statistics for four steps ahead errors of series with normal shocks. For other types of shocks, check tables 24 and 25, available at the appendix. As happened in other processes, *auto.arima* performed better for all error metrics, being 10% smaller than *ets'* ones, on average.

Table 8 — Forecasting errors for SARIMA(0,0,0)(1,0,1)[4] with normal shocks.

Length (n)	Algorithm	Criteria	SARIMA(0,0,0)(1,0,1)[4]			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	5,054	2,248	1,77	2,24
100	*ets*	AIC	5,647	2,376	1,89	2,37
100	*auto.arima*	BIC	5,032	2,243	1,77	2,24
100	*ets*	BIC	5,661	2,379	1,88	2,38
250	*auto.arima*	AIC	4,439	2,107	1,66	2,10
250	*ets*	AIC	5,297	2,301	1,83	2,30
250	*auto.arima*	BIC	4,429	2,105	1,68	2,10
250	*ets*	BIC	5,289	2,300	1,83	2,30
500	*auto.arima*	AIC	4,753	2,180	1,73	2,18
500	*ets*	AIC	5,602	2,367	1,91	2,36
500	*auto.arima*	BIC	4,667	2,160	1,72	2,16
500	*ets*	BIC	5,580	2,362	1,91	2,36

1000	*auto.arima*	AIC	4,632	2,152	1,70	2,15
1000	*ets*	AIC	5,467	2,338	1,83	2,33
1000	*auto.arima*	BIC	4,552	2,133	1,70	2,13
1000	*ets*	BIC	5,466	2,338	1,83	2,37

Source: self elaboration.

The last process simulated is a white noise with fractional integration of order $d = 0.4$. The equation for a time series that follows this process is given by:

(3.4)

Where $0<d<1$ is the order of fractional integration and ε_t is a white noise.

Since none of the algorithms is made to deal with fractionally integrated processes, they cannot model correctly time series with this characteristic. In fact, in about half of times, *auto.arima* modeled a unit root in the series, showing its indecision about modeling the series as integrated of order 1 or 0. Therefore, we opted for reporting only the statistics about forecasting errors of the two algorithms being analyzed. Table 9 presents those metrics when (3.4) had normally distributed shocks. Tables 26 and 27, in the appendix, show metrics for shocks following other probability distributions.

Table 9 — Forecasting errors for fractionally integrated (d=0.4) process with normal shocks

Length (n)	Algorithm	Criteria	Fractional integration (d=0,4)			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	1,059	1,029	0,81	1,03
100	*ets*	AIC	1,094	1,046	0,83	1,04
100	*auto.arima*	BIC	1,070	1,035	0,82	1,03
100	*ets*	BIC	1,090	1,044	0,82	1,04
250	*auto.arima*	AIC	1,022	1,011	0,81	1,01
250	*ets*	AIC	1,061	1,030	0,82	1,03
250	*auto.arima*	BIC	1,028	1,014	0,81	1,01
250	*ets*	BIC	1,061	1,030	0,82	1,03
500	*auto.arima*	AIC	1,042	1,021	0,81	1,02
500	*ets*	AIC	1,081	1,040	0,83	1,04
500	*auto.arima*	BIC	1,049	1,024	0,817	1,02
500	*ets*	BIC	1,081	1,040	0,83	1,04
1000	*auto.arima*	AIC	1,045	1,022	0,81	1,02

1000	*ets*	AIC	1,085	1,042	0,83	1,04
1000	*auto.arima*	BIC	1,044	1,022	0,81	1,02
1000	*ets*	BIC	1,085	1,042	0,83	1,04

Source: self elaboration.

In the series with fractional integration, we notice that the difference between the performance of both algorithms, mainly in percentual errors, was very small, being able to be inconsiderable, even though *auto.arima* did better. As in the case of other processes, there was no clear advantage when using BIC or AIC, at least for forecasting purposes. Also, larger series did not make the algorithms produce smaller forecast errors.

4. CONCLUSIONS

The results from our Monte Carlo experiment show that researchers should act carefully when using an automated time series algorithm, mainly if the intent is to model a series. Commonly used algorithms for that task, such as auto.arima and ets, have as a strategy applying many different preset models for the series to which they are applied and then choosing the one with the smallest information criteria, which by default is AIC. Our simulations showed that the use of BIC yields better results, even in smaller samples, at least for auto.arima. Regarding ets, changing the information criteria did not increase the performance – both led to choosing an exponential smoothing model that is appropriate for time series with increasing variance, which did not exist in our processes. Therefore, we noted a bad performance of ets, which often failed to use the correct exponential smoothing method for the series, even though the possibilities are few. This problem happened even more often when our series had random shocks that followed a chi-squared distribution, which is asymmetric. Another point to notice about ets is the absence of converge towards choosing the right exponential smoothing model as the series grow larger, even if using BIC.

On the other hand, auto.arima's most critical point is its incapacity of modeling correctly some deterministic patterns that may be present in a data-generating process. As we reported, in the case of trend stationary series, the algorithm always modeled a unit root and a drift in the series, which did not exist. Regarding seasonality, auto.arima could not correctly model more than 30% of times the series, even in the ones with 1.000 observations, which are rarely available for real-world economic processes. Also, the algorithm will always model series seasonality as being stochastic, reinforcing our point regarding its failure in dealing with deterministic aspects of a series.

Regarding forecasting, auto.arima had a clear advantage in all kinds of series we simulated, except for the ones with fractional integration, given that it cannot model this type of process. Opposed to the forecasting literature, this advantage was noted for all error metrics we calculated, for one or more steps ahead. Surprisingly, the use of larger series did not lead to more accurate forecasts sometimes, and the choice of information criteria was irrelevant.

About our initial hypothesis, posted in chapter 2, we conclude that we should reject all of them. The rate of correct modeling of both algorithms is far below 80%, even in large series, and convergence towards choosing the correct model as the series grew larger was not always noticed, mainly for more complex processes, such as the ones with trend or seasonality. We also saw that the random variable that controls the shocks of the series influenced ets. Also, auto.arima had a clear advantage in forecasting, being more than 10% more accurate, on average, than its competitor. Regarding the information criteria, it is beneficial to use BIC over AIC, at least for modeling time series using auto.arima.

REFERENCES

BOX, George E. P. et al. *Time series analysis: forecasting and control*. 1.ed. New Jersey: Wiley, 1970.

BROWN, Robert G. *Statistical Forecasting for Inventory control*. New York: McGraw-Hill, 1959.

BUENO, Rodrigo De Losso da Silveira. *Econometria de séries temporais.* 2.ed. São Paulo: Cengage Learning, 2012.

CRONE S.; HIBON M.; NIKOLOPOULOS K. Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting, [s.l],* v. 27, n. 3, p. 635-660, Jul. 2011.

DICKEY, David A; FULLER, Wayne A. Distribution of the Estimators for Autoregressive Time Series With a Unit Root. *Journal of the American Statistical Association, [s.l.]*, v.74, n. 366, p. 427-431, jun. 1979.

DICKEY, David A; FULLER, Wayne A. Likelihood Ratio Statistics for Autoregressive Time Series with a Unit Root. *Econometrica, [s.l.]*, v. 49, n. 4, p. 1057-1072, jul. 1981.

ENDERS, Walter. *Applied Econometric Time Series.* 4.ed. [s.l.]: Wiley, 2014.

GILLIAND, Michael. The value added by machine learning approaches in forecasting. *International Journal of Forecasting, [s.l.]*, v. 36, n. 1, p. 161-166, jan. 2020.

GUJARATI, Damodar; PORTER, Dawn. *Econometria Básica.* 5.ed. Porto Alegre: AMGH, 2011.

HOLT, Charles C. *Forecasting seasonals and trends by exponentially weighted averages.* Pittsburgh: Carnegie Institute of Technology, 1957.

HYNDMAN, Rob J. A brief history of forecasting competitions. *International Journal of Forecasting, [s.l.]*, v. 36, n. 1, p. 7-14, jan. 2020.

HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: Principles and practice.* 3.ed. Melbourne: Otexts, 2021. Disponível em: otexts.com/fpp3. Acesso em: 16 out. 2022.

HYNDMAN, Rob J. *et al*. *Forecasting with Exponential Smoothing: The State Space Approach.* Berlim: Springer, 2008.

HYNDMAN, R. J.; KHANDAKAR, Yeasmin. Automatic Time Series Forecasting: The forecast package for R. *Journal of Statistical Software*, *[s.l.]*, v. 27, n. 3, p. 1-22, jul. 2008.

HYNDMAN, R. J.; KOEHLER, A. B. Another look at measures of forecast accuracy. *International Journal of Forecasting, [s.l.]*, v. 22, n. 4, p. 679-688, out. 2006.

MAKRIDAKIS, S.; SPILIOTIS, E.; ASSIMAKOPOULOS, V. The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting, [s.l.]*, v. 36, n. 1, p. 54-74, jan. 2020.

MAKRIDAKIS, S.; SPILIOTIS, E.; ASSIMAKOPOULOS, V. The M4 Competition: Results, findings, conclusions and way forward. *International Journal of Forecasting, [s.l.]*, v. 34, n. 4, p. 802-808, out. 2018.

MONTGOMERY, Douglas C.; RUNGER, George C. *Estatística aplicada e probabilidade para engenheiros.* 6. ed. Rio de Janeiro: LTC, 2020.

MORETTIN, Pedro A.; TOLOI, Clélia M. C. *Análise de séries temporais.* 2. ed. São Paulo: Egard Blucher, 2006.

OFFICE FOR NATIONAL STATISTICS. *From Holt-Winters to ARIMA Modeling: measuring the Impact on Forecasting Errors for components of quarterly estimates of public service output*. Reino Unido, 2008.

R CORE TEAM. *R: a language and environment for statistical computing*. R Foundation for Statistical Computing. Áustria, 2019.

SPILIOTIS, Evangelos *et al.* Are forecasting competitions data representative of reality? *International Journal of Forecasting, [s.l.]*, v. 36, n. 1, p. 37-53, jan. 2020.

WICKHAM, Hadley; GROLEMUND, Garrett. *R para Data Science: Importe, Arrume, Transforme, Visualize e Modele Dados.* Rio de Janeiro: Alta Books, 2019.

WINTERS, Peter R. Forecasting Sales by Exponentially Weighted Moving Averages. *Management Science. [s.l.]* v.6, n.3, p. 324-342. abr, 1960**.**

APPENDIX

Table 10 — Rate of correct modeling for AR(1) with Student's T shocks (3 d.f)

Length (n)	Algorithm	Criteria	AR(1) - ϕ = 0,5		AR(1) - ϕ = 0,8	
			Acerto (%)	Erro (%)	Correct (%)	Incorrect (%)
100	*auto.arima*	AIC	59,19%	40,81%	44,22%	55,78%
100	*ets*	AIC	49,17%	50,83%	53,14%	46,86%
100	*auto.arima*	BIC	71,10%	28,9%	56,05%	43,95%
100	*ets*	BIC	51,03%	48,97%	53,40%	46,60%
250	*auto.arima*	AIC	62,27%	37,73%	46,33%	53,67%

250	*ets*	AIC	53,01%	46,99%	55,60%	44,40%
250	*auto.arima*	BIC	81,36%	18,64%	61,81%	38,19%
250	*ets*	BIC	53,56%	46,44%	55,84%	44,16%
500	*auto.arima*	AIC	64,92%	35,08%	53,34%	46,66%
500	*ets*	AIC	55,40%	44,60%	58,21%	41,79%
500	*auto.arima*	BIC	87,40%	12,60%	72,31%	27,69%
500	*ets*	BIC	55,71%	44,29%	58,49%	41,51%
1000	*auto.arima*	AIC	66,13%	33,87%	57,17%	42,83%
1000	*ets*	AIC	56%	44%	61,58%	38,42%
1000	*auto.arima*	BIC	89,64%	10,36%	77,91%	22,09%
1000	*ets*	BIC	56,39%	43,61%	61,71%	38,29%

Source: self elaboration.

Table 11 — Forecasting errors for AR(1) - ϕ = 0,8 with Student's T shocks (3 d.f)

Length (n)	Algorithm	Criteria	AR(1) - ϕ = 0,8			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	4,326	2,080	1,218	3,807
100	*ets*	AIC	3,869	1,967	1,222	1,936
100	*auto.arima*	BIC	4,292	2,072	1,212	1,936
100	*ets*	BIC	3,869	1,967	1,222	1,936
250	*auto.arima*	AIC	2,953	1,718	1,142	1,778
250	*ets*	AIC	3,148	1,774	1,211	1,828
250	*auto.arima*	BIC	2,972	1,724	1,148	1,780
250	*ets*	BIC	3,148	1,774	1,211	1,828
500	*auto.arima*	AIC	2,859	1,691	1,118	1,696
500	*ets*	AIC	3,099	1,760	1,189	1,764
500	*auto.arima*	BIC	2,873	1,695	1,119	1,700
500	*ets*	BIC	3,098	1,760	1,189	1,764
1000	*auto.arima*	AIC	2,682	1,638	1,112	1,671
1000	*ets*	AIC	2,900	1,703	1,183	1,732
1000	*auto.arima*	BIC	2,685	1,639	1,114	1,732
1000	*ets*	BIC	2,900	1,703	1,183	1,732

Fonte: Elaboração própria.

Table 12 — Rate of correct modeling for AR(1) with chi-squared shocks (2 d.f)

Length (n)	Algorithm	Criteria	AR(1) - ϕ = 0,5		AR(1) - ϕ = 0,8	
			Acerto (%)	Erro (%)	Correct (%)	Incorrect (%)
100	*auto.arima*	AIC	58,74	41,26	45,33	54,67
100	*ets*	AIC	35,89	64,11	57,29	42,71
100	*auto.arima*	BIC	70,92	29,08	57,01	42,99
100	*ets*	BIC	37,15	62,85	57,32	42,68
250	*auto.arima*	AIC	62,16	37,84	46,19	53,81
250	*ets*	AIC	28,13	71,87	51,47	48,53
250	*auto.arima*	BIC	82,18	17,88	61,17	38,83
250	*ets*	BIC	28,38	71,62	51,52	48,48
500	*auto.arima*	AIC	63,55	36,45	53,24	46,76
500	*ets*	AIC	20,22	79,78	46,69	53,31
500	*auto.arima*	BIC	86,29	13,71	72,10	27,90
500	*ets*	BIC	20,29	79,71	46,73	53,27
1000	*auto.arima*	AIC	64,59	35,41	56,77	43,23
1000	*ets*	AIC	11,64	88,36	40,98	59,02
1000	*auto.arima*	BIC	88,44	11,56	78,05	21,95
1000	*ets*	BIC	11,68	88,32	41,01	58,99

Source: self elaboration.

Table 13 — Forecasting errors for AR(1) - ϕ = 0,8 with chi-squared shocks (2 d.f)

Length (n)	Algorithm	Criteria	AR(1) - ϕ = 0,8			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	4,345	2,084	1,519	2,012
100	*ets*	AIC	4,489	2,119	1,539	2,046
100	*auto.arima*	BIC	4,310	2,076	1,512	2,004
100	*ets*	BIC	4,490	2,119	1,539	2,046
250	*auto.arima*	AIC	4,073	2,018	1,465	1,948
250	*ets*	AIC	4,359	2,088	1,512	2,016
250	*auto.arima*	BIC	4,108	2,027	1,473	1,956
250	*ets*	BIC	4,359	2,088	1,512	2,016

500	*auto.arima*	AIC	4,193	2,048	1,482	1,974
500	*ets*	AIC	4,512	2,124	1,537	2,050
500	*auto.arima*	BIC	4,183	2,045	1,479	1,971
500	*ets*	BIC	4,512	2,124	1,537	2,050
1000	*auto.arima*	AIC	4,169	2,042	1,467	1,964
1000	*ets*	AIC	4,458	2,111	1,511	2,033
1000	*auto.arima*	BIC	4,174	2,043	1,467	1,965
1000	*ets*	BIC	4,458	2,111	1,511	2,033

Source: self elaboration

Table 14 — Rate of correct modeling for ARIMA(0,1,1with Student's T shocks (3 d.f)

Length (n)	Algorithm	Criteria	ARIMA(0,1,1) - θ = 0,5	
			Correct (%)	Incorrect (%)
100	*auto.arima*	AIC	60,06	39,94
100	*ets*	AIC	24,33	75,67
100	*auto.arima*	BIC	73,69	26,31
100	*ets*	BIC	47,50	52,50
250	*auto.arima*	AIC	60,07	39,93
250	*ets*	AIC	8,05	91,95
250	*auto.arima*	BIC	83,93	16,07
250	*ets*	BIC	33,98	66,02
500	*auto.arima*	AIC	62,08	37,92
500	*ets*	AIC	0,90	99,10
500	*auto.arima*	BIC	88,65	11,35
500	*ets*	BIC	10,10	89,90
1000	*auto.arima*	AIC	62,64	37,36
1000	*ets*	AIC	0,08	99,92
1000	*auto.arima*	BIC	89,64	10,36
1000	*ets*	BIC	0,44	99,56

Source: self elaboration.

Table 15 — Rate of correct modeling for ARIMA(0,1,1) with chi-squared shocks (2 d.f)

Length (n)	Algorithm	Criteria	ARIMA(0,1,1) - $\theta = 0,5$	
			Correct (%)	Incorrect (%)
100	*auto.arima*	AIC	57,97	42,03
100	*ets*	AIC	25,43	74,57
100	*auto.arima*	BIC	71,84	28,16
100	*ets*	BIC	47,43	52,57
250	*auto.arima*	AIC	53,19	46,81
250	*ets*	AIC	8,23	91,77
250	*auto.arima*	BIC	83,31	16,69
250	*ets*	BIC	35,11	64,89
500	*auto.arima*	AIC	54,99	45,01
500	*ets*	AIC	0,92	99,08
500	*auto.arima*	BIC	87,98	12,02
500	*ets*	BIC	11,28	88,71
1000	*auto.arima*	AIC	56,03	43,97
1000	*ets*	AIC	0	100
1000	*auto.arima*	BIC	90,29	9,71
1000	*ets*	BIC	0,33	99,67

Source: self elaboration

Table 16 — Forecasting errors for ARIMA(0,1,1) - θ = 0,5 with Student's T shocks (3 d.f)

Length (n)	Algorithm	Criteria	ARIMA(0,1,1) - $\theta = 0,5$			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	3,200	1,789	1,273	2,246
100	*ets*	AIC	3,782	1,945	1,422	2,407
100	*auto.arima*	BIC	3,171	1,781	1,267	2,283
100	*ets*	BIC	3,864	1,966	1,442	2,494
250	*auto.arima*	AIC	3,701	1,924	2,923	53,60
250	*ets*	AIC	4,201	2,050	3,435	66,99
250	*auto.arima*	BIC	3,681	1,919	2,907	55,08
250	*ets*	BIC	4,296	2,073	3,447	67,05

500	*auto.arima*	AIC	2,924	1,710	2,739	15,99
500	*ets*	AIC	3,423	1,850	3,323	24,55
500	*auto.arima*	BIC	2,914	1,707	2,736	15,33
500	*ets*	BIC	3,437	1,854	3,33	24,63
1000	*auto.arima*	AIC	2,880	1,697	4,976	48,16
1000	*ets*	AIC	3,264	1,807	5,16	48,25
1000	*auto.arima*	BIC	2,876	1,696	4,961	47,47
1000	*ets*	BIC	3,265	1,807	5,16	48,25

Source: self elaboration.

Table 17 — Forecasting errors for ARIMA(0,1,1) - θ = 0,5 with chi-squared shocks (2 d.f)

Length (n)	Algorithm	Criteria	ARIMA(0,1,1) - θ = 0,5			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	1,052	1,026	0,839	1,067
100	*ets*	AIC	1,223	1,106	0,901	1,149
100	*auto.arima*	BIC	1,043	1,021	0,835	1,063
100	*ets*	BIC	1,247	1,117	0,910	1,159
250	*auto.arima*	AIC	1,018	1,009	0,858	1,125
250	*ets*	AIC	1,184	1,088	0,926	1,212
250	*auto.arima*	BIC	1,009	1,004	0,855	1,121
250	*ets*	BIC	1,208	1,099	0,936	1,224
500	*auto.arima*	AIC	0,983	0,992	1,421	42,02
500	*ets*	AIC	1,136	1,066	1,595	52,50
500	*auto.arima*	BIC	0,980	0,990	1,419	42,02
500	*ets*	BIC	1,143	1,069	1,599	52,50
1000	*auto.arima*	AIC	1,007	1,003	3,151	124,12
1000	*ets*	AIC	1,143	1,069	2,90	88,05
1000	*auto.arima*	BIC	1,004	1,002	3,214	128,83
1000	*ets*	BIC	1,143	1,069	2,90	88,05

Source: self elaboration.

Table 18 — Rate of correct modeling for trend stationary series with Student's T shocks (3 d.f)

Length (n)	Algorithm	Criteria	*Trend Stationary*	
			Correct (%)	Incorrect (%)
100	*auto.arima*	AIC	0	100
100	*ets*	AIC	4,66	95,34
100	*auto.arima*	BIC	0	100
100	*ets*	BIC	0,03	99,97
250	*auto.arima*	AIC	0	100
250	*ets*	AIC	59,77	40,23
250	*auto.arima*	BIC	0	100
250	*ets*	BIC	0,18	99,82
500	*auto.arima*	AIC	0	100
500	*ets*	AIC	88,63	11,37
500	*auto.arima*	BIC	0	100
500	*ets*	BIC	33,81	66,19
1000	*auto.arima*	AIC	0	100
1000	*ets*	AIC	97,71	2,29
1000	*auto.arima*	BIC	0	100
1000	*ets*	BIC	93,70	6,30

Source: self elaboration.

Table 19 — Rate of correct modeling for trend stationary series with chi-squared shocks (2 d.f)

Length (n)	Algorithm	Criteria	*Trend Stationary*	
			Correct (%)	Incorrect (%)
100	*auto.arima*	AIC	0	100
100	*ets*	AIC	0,24	99,76
100	*auto.arima*	BIC	0	100
100	*ets*	BIC	0,01	99,99
250	*auto.arima*	AIC	0	100
250	*ets*	AIC	27,15	72,85
250	*auto.arima*	BIC	0	100
250	*ets*	BIC	0	100

500	*auto.arima*	AIC	0	100
500	*ets*	AIC	91,77	8,23
500	*auto.arima*	BIC	0	100
500	*ets*	BIC	0,07	99,93
1000	*auto.arima*	AIC	0	100
1000	*ets*	AIC	98,93	1,07
1000	*auto.arima*	BIC	0	100
1000	*ets*	BIC	85,83	14,17

Source: self elaboration.

Table 20 — Forecasting errors for trend stationary series with Student's T shocks (3 d.f)

Length (n)	Algorithm	Criteria	*Trend stationary*			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	170,42	13,05	7,20	10,75
100	*ets*	AIC	200,47	14,15	8,06	11,51
100	*auto.arima*	BIC	192,97	13,89	7,83	11,34
100	*ets*	BIC	195,88	13,99	7,96	11,37
250	*auto.arima*	AIC	135,51	11,64	4,94	6,77
250	*ets*	AIC	174,61	13,21	5,64	7,67
250	*auto.arima*	BIC	140,65	11,86	5,02	6,89
250	*ets*	BIC	184,45	13,58	5,85	7,84
500	*auto.arima*	AIC	127,31	11,28	3,38	4,51
500	*ets*	AIC	168,62	12,98	3,86	5,20
500	*auto.arima*	BIC	125,76	11,21	3,36	4,48
500	*ets*	BIC	183,92	13,56	4,07	5,42
1000	*auto.arima*	AIC	129,00	11,35	2,09	2,84
1000	*ets*	AIC	175,08	13,23	2,40	3,31
1000	*auto.arima*	BIC	127,08	11,27	2,06	2,82
1000	*ets*	BIC	176,23	13,27	2,41	3,32

Source: self elaboration.

Table 21 — Forecasting errors for trend stationary series with chi-squared shocks (2 g.l)

Length (n)	Algorithm	Criteria	*Trend stationary*			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	212,99	14,59	8,57	11,09
100	*ets*	AIC	254,96	15,96	9,47	12,10
100	*auto.arima*	BIC	244,56	15,63	9,24	11,85
100	*ets*	BIC	252,42	15,88	9,44	12,04
250	*auto.arima*	AIC	183,34	13,54	5,91	7,64
250	*ets*	AIC	256,07	16,00	7,01	9,05
250	*auto.arima*	BIC	187,86	13,70	5,99	7,73
250	*ets*	BIC	252,08	15,87	6,96	8,96
500	*auto.arima*	AIC	171,15	13,08	4,02	5,18
500	*ets*	AIC	233,89	15,29	4,68	6,08
500	*auto.arima*	BIC	169,19	13,00	4,00	5,15
500	*ets*	BIC	250,82	15,83	4,90	6,28
1000	*auto.arima*	AIC	178,27	13,35	2,57	3,31
1000	*ets*	AIC	236,34	15,37	2,96	3,83
1000	*auto.arima*	BIC	174,51	13,21	2,55	3,28
1000	*ets*	BIC	240,67	15,51	2,99	3,86

Source: self elaboration.

Table 22 — Rate of correct modeling for SARIMA(0,0,0)(1,0,1)[4] with Student's T shocks (3 d.f)

Length (n)	Algorithm	Criteria	SARIMA(0,0,0)(1,0,1)[4]	
			Correct (%)	Incorrect (%)
100	*auto.arima*	AIC	13,80	86,20
100	*ets*	AIC	47,60	52,40
100	*auto.arima*	BIC	18	82
100	*ets*	BIC	47,70	52,60
250	*auto.arima*	AIC	8,20	91,80
250	*ets*	AIC	49,60	50,40
250	*auto.arima*	BIC	17,90	82,10
250	*ets*	BIC	50,20	23,40

500	*auto.arima*	AIC	8,80	91,20
500	*ets*	AIC	51,90	48,10
500	*auto.arima*	BIC	25	75
500	*ets*	BIC	52,70	47,30
1000	*auto.arima*	AIC	9	91
1000	*ets*	AIC	58,70	41,30
1000	*auto.arima*	BIC	25,60	74,40
1000	*ets*	BIC	59,40	40,60

Source: self elaboration.

Table 22 — Rate of correct modeling for SARIMA(0,0,0)(1,0,1)[4] with chi-squared shocks (2 d.f)

Length (n)	Algorithm	Criteria	SARIMA(0,0,0)(1,0,1)[4]	
			Correct (%)	Incorrect (%)
100	*auto.arima*	AIC	12,90	87,10
100	*ets*	AIC	22,60	77,40
100	*auto.arima*	BIC	16,40	83,60
100	*ets*	BIC	21	79
250	*auto.arima*	AIC	7	93
250	*ets*	AIC	9,30	90,70
250	*auto.arima*	BIC	15,40	84,60
250	*ets*	BIC	9,70	90,30
500	*auto.arima*	AIC	6	94
500	*ets*	AIC	2,10	97,90
500	*auto.arima*	BIC	24,70	75,30
500	*ets*	BIC	2,20	97,80
1000	*auto.arima*	AIC	5,50	94,50
1000	*ets*	AIC	0,40	99,60
1000	*auto.arima*	BIC	23,20	76,80
1000	*ets*	BIC	0,40	99,60

Source: self elaboration.

Table 24 – Forecasting errors for SARIMA(0,0,0)(1,0,1)[4] with Student's T shocks (3 d.f)

Length (n)	Algorithm	Criteria	SARIMA(0,0,0)(1,0,1)[4]			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	15,14	3,892	2,85	4,01
100	*ets*	AIC	16,64	4,079	3,00	4,19
100	*auto.arima*	BIC	15,11	3,888	2,86	4,01
100	*ets*	BIC	16,85	4,105	3,02	4,21
250	*auto.arima*	AIC	14,22	3,772	2,69	3,74
250	*ets*	AIC	18,71	4,326	3,09	4,24
250	*auto.arima*	BIC	14,15	3,762	2,69	3,74
250	*ets*	BIC	18,71	4,326	3,09	4,24
500	*auto.arima*	AIC	13,12	3,623	2,71	3,86
500	*ets*	AIC	15,67	3,959	2,93	3,96
500	*auto.arima*	BIC	13,20	3,633	2,70	3,86
500	*ets*	BIC	15,67	3,959	2,93	3,96
1000	*auto.arima*	AIC	15,34	3,918	2,50	3,53
1000	*ets*	AIC	18,47	4,298	2,88	3,97
1000	*auto.arima*	BIC	15,42	3,927	2,51	3,54
1000	*ets*	BIC	18,47	4,298	2,88	3,97

Source: self elaboration.

Table 25 – Forecasting errors for SARIMA(0,0,0)(1,0,1)[4] with chi-squared shocks (2 d.f)

Length (n)	Algorithm	Criteria	SARIMA(0,0,0)(1,0,1)[4]			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	18,67	4,322	3,31	4,20
100	*ets*	AIC	20,71	4,551	3,50	4,42
100	*auto.arima*	BIC	18,85	4,343	3,32	4,22
100	*ets*	BIC	20,95	4,577	3,51	4,44
250	*auto.arima*	AIC	17,65	4,202	3,21	4,06
250	*ets*	AIC	20,93	4,576	3,49	4,44
250	*auto.arima*	BIC	18,28	4,276	3,26	4,13
250	*ets*	BIC	20,89	4,571	3,49	4,43

500	*auto.arima*	AIC	17,85	4,226	3,16	4,10
500	*ets*	AIC	21,41	4,628	3,45	4,49
500	*auto.arima*	BIC	17,42	4,174	3,15	4,05
500	*ets*	BIC	21,38	4,625	3,45	4,49
1000	*auto.arima*	AIC	17,88	4,229	3,15	4,05
1000	*ets*	AIC	20,58	4,537	3,39	4,36
1000	*auto.arima*	BIC	17,84	4,224	3,16	4,05
1000	*ets*	BIC	20,57	4,536	3,39	4,36

Source: self elaboration.

Table 26 — Forecasting errors for fractionally integrated series with Student's T shocks (3 d.f)

Length (n)	Algorithm	Criteria	Fractional integration (d=0,4)			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	3,241	1,800	1,17	1,75
100	*ets*	AIC	3,319	1,822	1,19	1,78
100	*auto.arima*	BIC	3,279	1,811	1,18	1,76
100	*ets*	BIC	3,301	1,817	1,19	1,77
250	*auto.arima*	AIC	3,155	1,776	1,14	1,73
250	*ets*	AIC	3,267	1,807	1,18	1,76
250	*auto.arima*	BIC	3,165	1,779	1,14	1,73
250	*ets*	BIC	3,266	1,807	1,18	1,76
500	*auto.arima*	AIC	3,115	1,765	1,14	1,80
500	*ets*	AIC	3,245	1,801	1,18	1,84
500	*auto.arima*	BIC	3,142	1,773	1,15	1,81
500	*ets*	BIC	3,244	1,801	1,18	1,84
1000	*auto.arima*	AIC	2,623	1,619	1,07	1,61
1000	*ets*	AIC	2,793	1,671	1,13	1,66
1000	*auto.arima*	BIC	2,626	1,621	1,08	1,61
1000	*ets*	BIC	2,793	1,671	1,13	1,66

Source: self elaboration.

Table 27 — Forecasting errors for fractionally integrated series with chi-squared shocks (2 d.f)

Length (n)	Algorithm	Criteria	Fractional integration (d=0,4)			
			MSE	RMSE	MAPE(%)	RMSPE(%)
100	*auto.arima*	AIC	4,219	2,054	1,50	1,98
100	*ets*	AIC	4,419	2,102	1,54	2,03
100	*auto.arima*	BIC	4,252	2,062	1,51	1,99
100	*ets*	BIC	4,389	2,095	1,53	2,02
250	*auto.arima*	AIC	4,287	2,070	1,47	1,99
250	*ets*	AIC	4,417	2,102	1,49	2,02
250	*auto.arima*	BIC	4,307	2,075	1,48	1,99
250	*ets*	BIC	4,418	2,102	1,49	2,02
500	*auto.arima*	AIC	3,950	1,988	1,43	1,91
500	*ets*	AIC	4,144	2,036	1,46	1,96
500	*auto.arima*	BIC	3,970	1,993	1,43	1,92
500	*ets*	BIC	4,144	2,036	1,46	1,96
1000	*auto.arima*	AIC	4,133	2,033	1,46	1,95
1000	*ets*	AIC	4,329	2,081	1,49	2,00
1000	*auto.arima*	BIC	4,133	2,033	1,46	1,95
1000	*ets*	BIC	4,329	2,081	1,49	2,00

Source: self elaboration.