*J. Health Inform.* 2024 Janeiro-Dezembro; 16

**Artigo Original**

DOI: https://doi.org/10.59681/2175-4411.v16.2024.1003

JHI JOURNAL OF HEALTH INFORMATICS

# Implementation of a time series forecasting model to estimate excess deaths in Brazil in 2020

Implementação de um modelo de previsão usando séries temporais para estimar excesso de óbitos no Brasil em 2020

Implementación de un modelo de pronóstico de series de tiempo para estimar exceso de muertes en Brasil en 2020

**Lucas Felippe Mateus[1], Fabrício Ourique[1], Analucia Schiaffino Morales[1], Millena Nayara da Silva[2]**

**Keywords:** Time series; Predictive model; Excess deaths; Underreporting of deaths by COVID-19.

## ABSTRACT

Goals: The aim of this paper is to understand the behavior of the Covid-19 pandemic on the national Brazilian scenario and describe how it affected the mortality rate. Methods: Implement a predictor model using ARIMA modeling concepts and data extracted from the Unified Health System database, in order to estimate the number of deaths caused by COVID-19 in Brazil during 2020. Results: COVID-19 is estimated to have contributed, on average, to a surplus of 713 daily deaths. Conclusion: Even considering the records of deaths by COVID-19 on the result of the prediction, it is observed that the combination is below the real curve, which indicates that there is underreporting of deaths caused by this disease during the year 2020 in Brazil.

**Descritores:** Séries temporais; Modelo preditivo; Excesso de mortes; Sub-notificação de mortes por COVID-19.

## RESUMO

Objetivos: Compreender o comportamento da pandemia da Covid-19 no cenário nacional e descrever como ela afetou o índice de mortalidade. Métodos: Implementar um modelo preditor utilizando conceitos de modelagem ARIMA e dados extraídos do banco de dados do Sistema Único de Saúde, a fim de estimar o número de óbitos causados pela COVID-19 no Brasil durante 2020. Resultados: Estima-se que a COVID-19 tenha contribuído, em média, para um excedente de 713 mortes diárias. Conclusão: Mesmo considerando os registros de óbitos por COVID-19 sobre o resultado da predição, observa-se que a combinação fica abaixo da curva real, indicando que há subnotificação de óbitos causados por essa doença durante o ano de 2020 no Brasil.

**Descriptores:** Serie de tiempo. modelo predictivo; Demasiadas muertes; Subnotificación de muertes por COVID-19.

## RESUMEN

Objetivos: Comprender el comportamiento de la pandemia de Covid-19 en el escenario nacional y describir cómo afectó la tasa de mortalidad. Métodos: Implementar un modelo predictor, utilizando conceptos de modelado ARIMA y datos extraídos de la base de datos del Sistema Único de Salud, para estimar el número de muertes causadas por COVID-19 en Brasil durante 2020. Resultados: se estima que el COVID-19 ha contribuido, en promedio, a un superávit de 713 muertes diarias. Conclusión: incluso considerando los registros de muertes por COVID-19 sobre el resultado de la predicción, se observa que la combinación está por debajo de la curva real, lo que indica que hay subregistro de muertes causadas por esta enfermedad durante el año 2020 en Brasil.

[1] *Department of Computer – Federal University of Santa Catarina (UFSC) – Santa Catarina – Brazil*
[2] *Center of Health Sciences – Federal University of Santa Maria (UFSM) – Rio Grande do Sul – Brazil*

Autor Correspondente: **Fabrício Ourique**
e-mail: **fabricio.ourique@ufsc.br**

Artigo recebido: 11/10/2022
Aprovado: 02/10/2023
https://jhi.sbis.org.br/

## INTRODUCTION

At the end of 2019, China reported cases of pneumonia, which occurred in Hubei province, more specifically in Wuhan, to the World Health Organization (WHO). These cases were due to a new coronavirus, which became known as 2019-nCoV, and health officials found evidence that human-to-human transmission was able to sustain itself. Moreover, other countries confirmed travel-associated cases. In this way, the increasing number of cases and deaths began challenging public health and government policies.[1] After cases outside China increased and the number of countries with confirmed cases tripled, the World Health Organization (WHO), concerned about alarming levels of spread, declared the novel coronavirus outbreak a pandemic in March 2020.[2]

The COVID-19 pandemic has rapidly affected daily life and businesses in general, beyond disrupting world trade overall. In health care, significant challenges such as diagnosis, treatment options, professional overload and insufficient hospital supply have arisen. As for the economy sphere, the world has faced a conspicuous slowdown in revenue growth. Furthermore, in the interpersonal relationship sphere, there was social distancing, excessive emotional stress among the population and the shutdown of entertainment establishments and restaurants.[3]

In Brazil, only patients with severe symptoms or who died are considered COVID-19 cases due to the lack of tests, an approach that ends up promoting a large percentage of the disease underreporting. Thus, the greater the number of infected, the more underreporting occurs, creating uncertainty in the face of confirmed cases and deaths. If these uncertainties are not considered, precipitous decisions can be made, which makes understanding the underreporting cases one of the critical points in this pandemic analysis.[4] Moreover, another important indicator during the pandemic analysis is excess mortality. The high excess of deaths, due to those that COVID-19 does not directly explain and those that occur outside the hospital, suggest a high underreporting of deaths from COVID-19, reinforcing the spread of SARS-CoV-2.[5]

An epidemiologist interested in understanding a flu virus over time could use time series analysis to better understand the number of observed cases. Time series analysis is a tool that significantly impacts several scientific applications that seek to analyze experimental data observed over different points in time to achieve modeling and statistical inference. Furthermore, this analysis is efficient when it comes to the necessary data to build its model, helps to identify possible trends and seasonality, contains techniques such as smoothing and seasonality adjustments that facilitates data cleansing and uses historical data to predict future values.[6] A time series is a collection of observations made sequentially in time, describing the behavior of a given variable.[7]

This research intends to analyze a time series that represents the number of daily deaths in Brazil from 2014 to 2019 and implement a forecast model to estimate the number of daily deaths in Brazil during 2020. From this estimate, it is possible to compare it with the actual data and then extract, from this difference, a notion of the excess of death for this year.

## MATERIALS AND METHODS

The Pandas library offers high-performance tools for loading and manipulating data in Python, returning convenient and easy structures to represent the data as Data Frame and Series. While the Stats models library provides tools for statistical modeling and testing, it also has tools dedicated to time series analysis and forecasting. Among the features of Stats models, it is worth highlighting those that are relevant for time series prediction, which are: statistical tests, such as the augmented Dickey-Fuller test; time series analysis graphs, like the autocorrelation function (ACF) and the partial autocorrelation (PACF); and time series models, including the autoregressive (AR).[8]

For this research, the data were selected from the Unified Health System (SUS) open database, known as openDataSus. More specifically, the Mortality Information System (SIM) was used among the available databases. In addition to the database, TABWIN, an exploratory analysis software used to decompress and tabulate the files obtained, was necessary since the data available in openDataSus are not in the desired format for analysis.

The subset of interest was selected from the dataset obtained through open DataSus. The death dates range from 2014 to 2020. This subset is a series containing 9,361,816 records of the dates mentioned above. In the series obtained, the dates were not standardized. That is, there is a difference in the number of digits allocated for the day and this can be an obstacle during processing and later in the data visualization. Therefore, a routine was applied to the data subset that checks each date and standardizes them. In this case, standardizing the dates consists of assigning four digits for the year, two for the month, two for the day, adding a separator between them and formatting in that respective order.

The invalid data was also verified, seeking to identify discrepancies between the records provided by the database and those obtained after pre-processing. As this check did not return any outliers, the data subset was considered ready for a transformation. Considering that one of the objectives of this research is to apply the time series predictor model, it is necessary to obtain the time series. To do so, the occurrence of each date was counted in the pre-processed data, and then the number of deaths was grouped by dates. This procedure returns the number of deaths per day. Time series are collections of observations made over time of one or more variables.

According to the observations, they are classified as continuous or discrete. When working with time series, two essential characteristics must be kept in mind: observations are usually not independent and the order in which they were made must be maintained. Such a dependence feature allows using past values to predict future ones.[7]

Thereby, grouping by date - where each date carries a value - represents a discrete time series, since equally spaced observations of a variable were made sequentially in time. It is possible to maintain the order of the observations by transforming the date column data to the 'data time' type, setting a daily frequency and placing the date column as the Data Frame's index. Time series analysis evaluates the properties of the probabilistic model that generated the observed time series.[7] One of the first steps that can be done in time series analysis is to build a graph, Figure 1.

**Figure 1. Time Series that represents deaths per day in Brazil since 2014 until 2020**

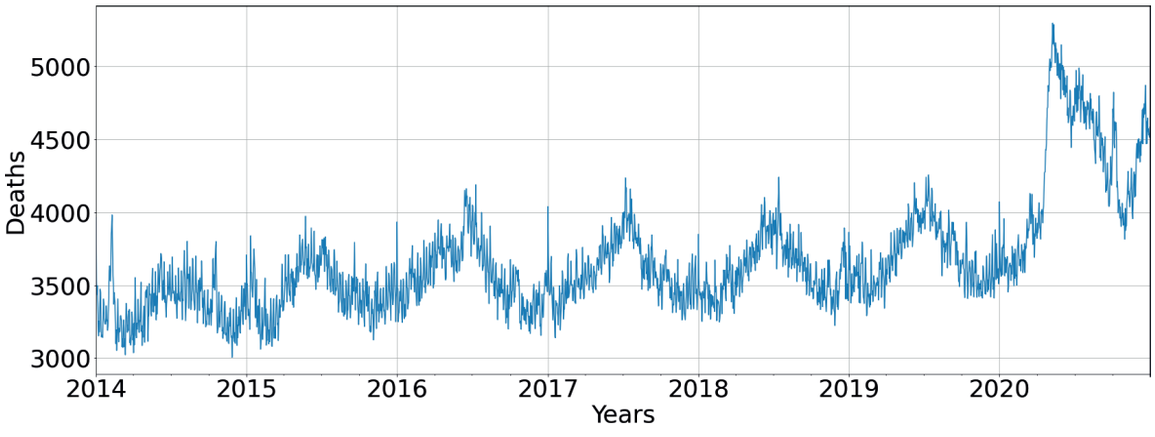


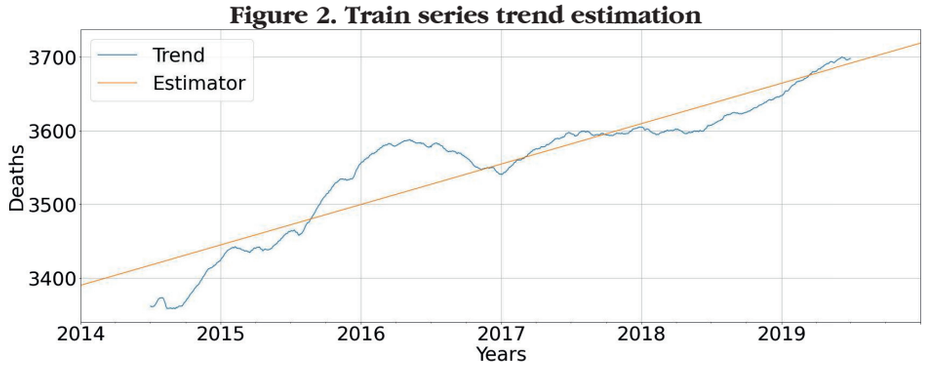

Figure 1 shows the time series that describes the number of daily deaths in Brazil from the beginning of 2014 to the end of 2020. It is possible to notice that there has been an increase in deaths over the years. Even though it is mild, this represents a trend. In addition, there is also a seasonal behavior between the years, that is, a pattern that repeats itself. Finally, 2020 stands out for its anomalous behavior, which occurred due to the COVID-19 pandemic.

It is necessary to define the part of the series that will be used to train the model and the part that will serve as a comparison to verify its performance. Such parts are known respectively as training set and testing set. Remembering that the separation of training and testing sets for time series does not occur in the same way as in sets intended for other machine learning algorithms, since the chronological sequence of observations is relevant. For many machine learning methods, we shuffle the data before splitting it to represent the two sets equally. However, time series data needs to maintain the chronological order of the values within the set, which means that shuffling is not applicable in this context. Thus, the training and testing sets would have to be unbroken sequences of values. Therefore, the training set must include all values from the beginning of the data to a specific point in time, while the testing set must include the rest. Keeping in mind that this research aims to use the time series forecast model to understand excess deaths in 2020, the point of separation to obtain both sets is clear. As six years will be used to predict the seventh, the training set represents 86% of the original set while the testing represents the remaining 14%.

It is possible to decompose the time series into four parts: level, trend, seasonality and noise. The level is the average value of the series. In contrast, the trend is the increasing or decreasing behavior of the series over time. Seasonality, on the other hand, represents the patterns of repetition or cycles of behavior over time. Lastly, the noise is the variability in the observations.[8] In order to obtain a good description of the training set behavior, it is possible to decompose it in level, trend, seasonality, and noise with the additive decomposition, which is a straightforward decomposition that expects a linear relationship between the four parts and the original series. This is the simplest type of decomposition, in which it is expected to observe a linear additive relationship between the three parts and the observed series. Two behaviors were observed over time in the training set, and it is expected that such behaviors will persist in future observations. Therefore, it is necessary to have functions describing these patterns to reproduce them.

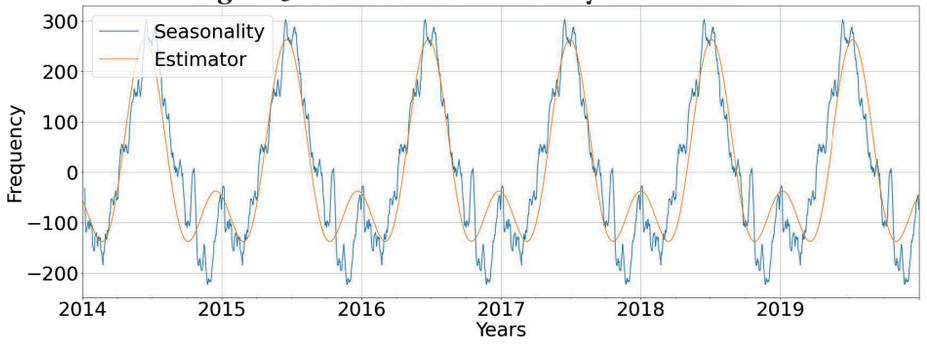
Starting with the trend, it is possible to do linear regression and approximate it with a first-degree function. Despite being a very simple approximation, this function manages to meet the purposes of this research. At Figure 2, two curves are displayed, the blue one represents the trend obtained by Python's stats models and the orangeone represents the approximation function. In more than two thousand observations, the number of deaths increased by approximately three hundred in a setting where the data describe values in the unit of thousands, meaning that this trend is smooth over time.

**Figure 2. Train series trend estimation**




The following behavior to be estimated is seasonality. In this case, the best- approximated function was composed of a sum of two cosines. To adjust them for seasonality, it was necessary to advance their signals and put the frequency of one cosine as half the frequency of the other. The amplitude was regulated according to the seasonality obtained through Python stats models. The seasonality estimator can be observed in Figure 3. Again, the blue curve is the seasonality signal of the training set, and the orange curve is the approximation function.

**Figure 3. Train series seasonality estimation**

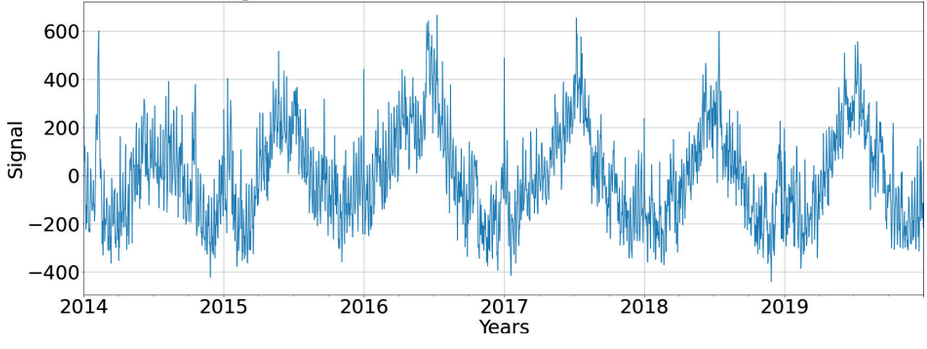



A series that does not have systematic changes in either its mean or its variance and has undergone a process of removing periodic variations is known as a stationary series. Most of the probability theory around time series is directed towards stationary series. Therefore, before applying specific probabilistic models, it is necessary to treat the time series to remove these sources of variation, making it stationary.[7]

As highlighted, the training set has patterns over time that cause variations in statistical metrics. A time series that has a time-varying mean is a non-stationary series. In addition, the models that are being used as theoretical references are intended for stationary time series. Therefore, it is essential to make the training set series stationary.

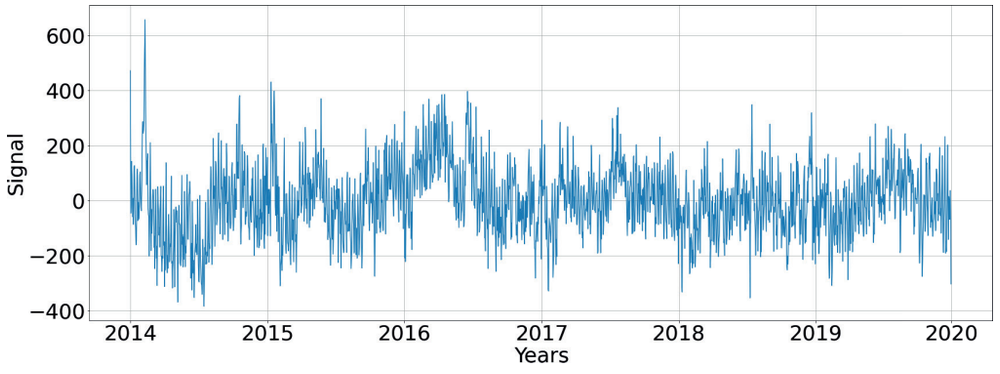
It is necessary to remove the components that cause the unwanted variations, so the first one to be removed is the trend. The technique to detrend the training set chosen here was the detrending method of the SciPy signal, because it was the one that managed to get the average of the signal closer to zero. After removing the trend, the signal shown in Figure 4 is obtained.

**Figure 4. Train time series without trend**




The next component that needs to be removed is seasonality. For this case, it was decided to do it through seasonal decomposition. This process is nothing more than taking the series without trend, previously obtained, and subtracting from it the seasonality returned by the decomposition of the training series made with stats models. Thus, we arrive at the signal shown in Figure 5.

**Figure 5. Train time series without trend and seasonality**




To ensure that the series to be modeled is stationary, a statistical test like the augmented Dick-Fuller is applied, aiming to identify unit roots in time series. The Dick-Fuller augmented test considers two hypotheses, the null hypothesis and the alternative hypothesis. The null hypothesis assumes that the series is not stationary, while the alternative hypothesis assumes that the series is stationary. To know which of the hypotheses the series fits into, it is necessary to interpret the p-value returned by the test. The null hypothesis can be rejected if the p-value is less than 0.05. That is, the series is stationary.[8] This test can be computed in Python using the ad fuller function from the stat stools library. Look at Table 1 to view the information returned by this test.

**Table 1: Augmented Dick-Fuller test**

| Parameters | Values |
|---|---|
| Statistic test value | -5.30 |
| P value | 5.41e-06 |
| Lags | 26 |
| Observations | 2164 |
| Critical values | 1%: -3.43<br>5%: -2.86<br>10%: -2.56 |

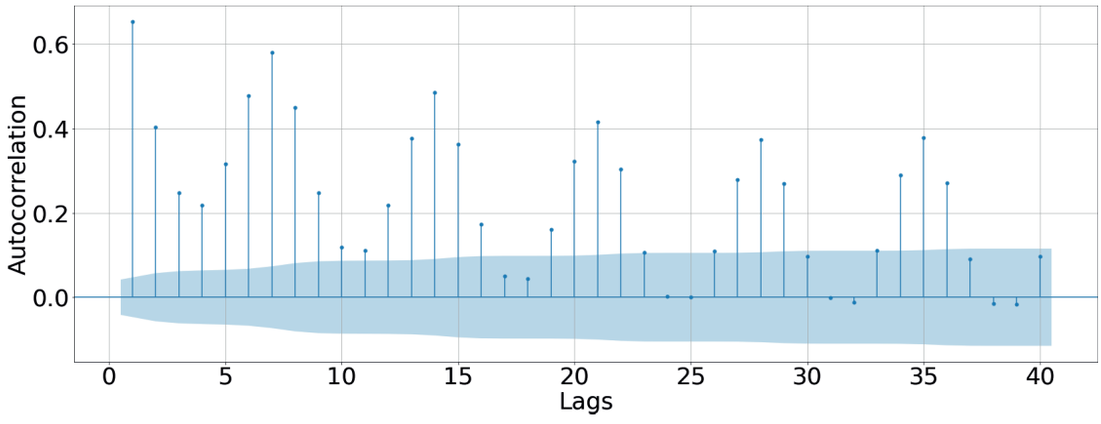
The table above shows: the value of the statistical test, which represents the critical value of the series submitted; the p-value, which indicates the probability of rejection or not of the null hypothesis; the lag, which indicates the number of delays used in the regression to determine confidence levels, number of observations used in the analysis and critical values of confidence levels for 1%, 5%, and 10%.

For this case, it is observed that the augmented Dick--Fuller test returned a p-value less than 0.05. Also, the critical value of the series is less than the critical value of 1%, which gives a confidence level of 99%. Under these circumstances, the null hypothesis is rejected, and it is concluded that after removing the components the series became stationary.

An important property in time series analysis is known as autocorrelation, which measures the correlation between observations at different times within the series. The autocorrelation coefficients clarify the probabilistic model that generated the observed series. These coefficients can be better observed through the correlogram.[7]

It was previously mentioned that the data could not be shuffled because the set's chronological order must be preserved - however, there is one more reason why the data should not be rearranged. It is necessary to find links between past and present observations to understand the behavior assumed by the series and even make pre- dictions about it. To accomplish that, autocorrelation is used, representing the correlation between a series and itself lagged. This autocorrelation function can be observed through the correlogram in Figure 6.

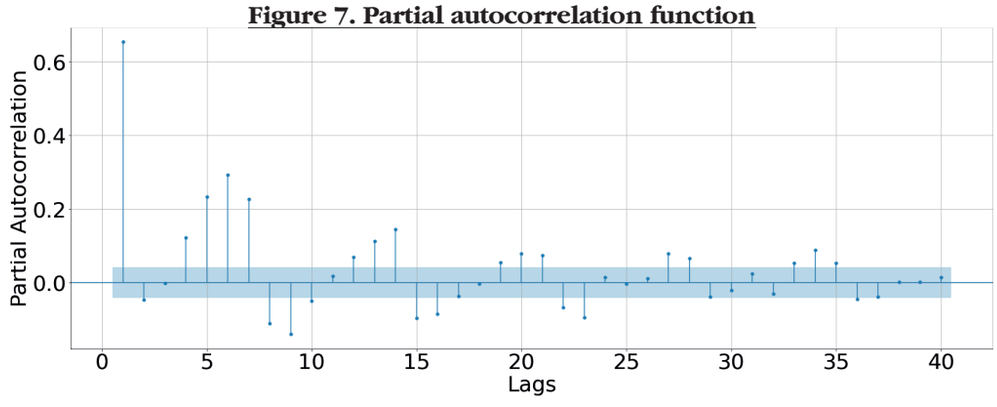
**Figure 6. Autocorrelation function**



Values on the x-axis represent lags, while the y-axis indicates possible values for the autocorrelation coefficient. The correlation can only take values between one and minus one, which is why the maximum amplitude of the graph is one. The thin line along the graph represents the autocorrelation between the time series and a lagged copy of itself. The first line indicates the autocorrelation one period ago, the second line represents the coefficient value for two periods ago and so on. The blue area around the x-axis represents the significance, that is, the values situated outside are significantly different from zero, which suggests the existence of autocorrelation for that specific lag. This area expands as lag values increase, so this autocorrelation is more unlikely to persist at high lags. It needs to be ensured that the autocorrelation coefficient at higher lags is more pronounced to be significantly different from zero. The lines higher than the blue region suggest that the coefficients are significant, indicating dependence between the data over time.

In addition to the autocorrelation function, there is a partial autocorrelation function that determines only the direct relationship between the time series and its lagged version. To better understand this idea, it is necessary to remember that autocorrelation measures the similarity between a time series and a lagged version of itself. However, the coefficients also capture secondary effects. For example, when examining the correlation of the current period, $X_t$, with the value of two periods ago, $X_{t-2}$, it is observed that there are two ways for $X_{t-2}$ to influence $X_t$: the first would be direct, that is, $X_{t-2}$ infers a tendency for $X_t$, and the second would be indirect through $X_{t-1}$, that is, $X_{t-2}$ influences $X_{t-1}$ which in turn influences $X_t$. The partial autocorrelation function is in the correlogram of Figure 7.

**Figure 7. Partial autocorrelation function**




The correlogram of the partial autocorrelation function is very different from that obtained by the autocorrelation function; this is precisely because the partial autocorrelation function considers only the direct correlation between the series and its lagged version, while the autocorrelation function considers all the influences, whether direct or indirect. In this case, it is observed that the partial autocorrelation function indicates a very significant correlation for the one-period lag. In addition, it can be seen that it assumes the same value presented by the autocorrelation function seen in Figure 6, which is expected. Both functions must return the same correlation value for the one-period lag, as there is no other channel that $X^{t-1}$ would influence $X^t$. The correlogram of the partial autocorrelation function presents negative correlations for certain lags, which means that higher values in these periods result in lower values in the current period and vice versa. Another essential point to be highlighted is that while the autocorrelation function indicates that the correlation up to forty lags is still significantly different from zero, the partial autocorrelation function shows that from the sixth lag, the correlation starts to fall. The more lags are considered, more insignificant they become, i.e., they are not significantly different from zero, so the numerical values associated with them are not necessary as they can be all assumed as essentially zero, whether positive or negative and with no lasting effects.

The analysis carried out so far has brought the necessary basic knowledge about the series, so it is possible to start working with the predictor model. Forecasts are usually necessary when projections are needed, which vary with each problem. The objective of predictions is to obtain a function that uses the observations available up to a certain point in time, aiming to estimate the values the series will assume in subsequent moments with the smallest possible mean squared deviation between the estimated value and the value that it assumed.[9]

Many forecasting procedures are based on a time series model. Therefore, it is helpful to be familiar with a range of models and understand the modeling process applied to series data before starting to look at forecasting methods. The ARIMA class of models is the basis of many fundamental ideas in time series analysis and is a crucial forecasting tool. The original reference for this study is Box and Jenkins (1970), which is why ARIMA models are sometimes called Box-Jenkins models.[7]

In order to adjust the predictor model, the idea is to make predictions in known years, that is, in years covered by the training set. Once the model is adjusted, it can be extrapolated to the unknown year, the year in the test

set and, finally, compared with the real data. To start the modeling, the concepts of auto-regression were applied in 2014 and 2015 to predict 2016. Then, the same methodology was applied using the years from 2014 to 2016 to predict 2017. And so was done until the 2019 forecast was obtained using only auto-regression. In addition to the auto-regression predictions, the following metrics were calculated to evaluate the model's performance: root mean square error, mean absolute error and R2 score.

The forecast error (or residual error) is calculated as the expected value minus the predicted value - the closer to zero this error is, the better the prediction model.[8] Performance measures for time series forecasting provide a summary of the model's ability and the ability to make such predictions.

**Table 2: Metrics from 2016 until 2019**

| 2016 | | | 2017 | | | 2018 | | | 2019 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **RMSE** | **MAE** | **$R_2$** | **RMSE** | **MAE** | **$R_2$** | **RMSE** | **MAE** | **$R_2$** | **RMSE** | **MAE** | **$R_2$** |
| 162.92 | 128.69 | 0.41 | 116.49 | 91.82 | 0.64 | 116.45 | 91.70 | 0.63 | 122.89 | 98.11 | 0.64 |

Except for the year 2019, it is possible to see that as the forecasts progressed, the errors reduced. With each prediction made, the group of past values increased, which made it possible to work with more lags. The error rose again in the 2019 forecast because the seasonality estimator could not well follow the behavior of the first three months of that year. The latest observations can be seen in Figure 3, showing how this mismatch occurs.

Continuing with the modeling, one can recap the concepts of the moving average (MA) model. This one, as well as the autoregressive model, seeks to make a combi- nation through regression. The difference is that the autoregressive model seeks to perform an autoregression on the time series values. In contrast, the MA model aims at an autoregression on the errors made in past forecasts. If it is possible to predict the expected error, then the model can be improved, where its output will be the sum of the predicted values and errors.

When applying the same auto-regression done for the series values, on errors made in past forecasts, an error predictor that would improve the model's performancewas not obtained. However, in an exploratory analysis of past forecast errors, it was observed the possibility to improve the model's performance if the expected error was considered to be the average of the last two errors for each date. For example, the expected error for 01/ 01/2019 is equal to the average between the error obtained on 01/01/2018 and the one obtained on 01/01/2017. After performing this adjustment in the model, the error in the prediction was reduced. Therefore, its performance increased. These new metrics for 2019 are shown in table 3.

**Table 3: Metrics for 2019 prediction considering the expected error**

| Year | Metrics | Values |
|---|---|---|
| 2019 | RMSE | 112.32 |
| | MAE | 91.67 |
| | $R_2$ | 0.70 |

## RESULTS AND DISCUSSION

Once the model has been trained and properly adjusted, it is possible to make the forecast for 2020. The forecast can be seen in figure 3 and its metrics, in table 4.

**Figure 8. 2020 prediction**

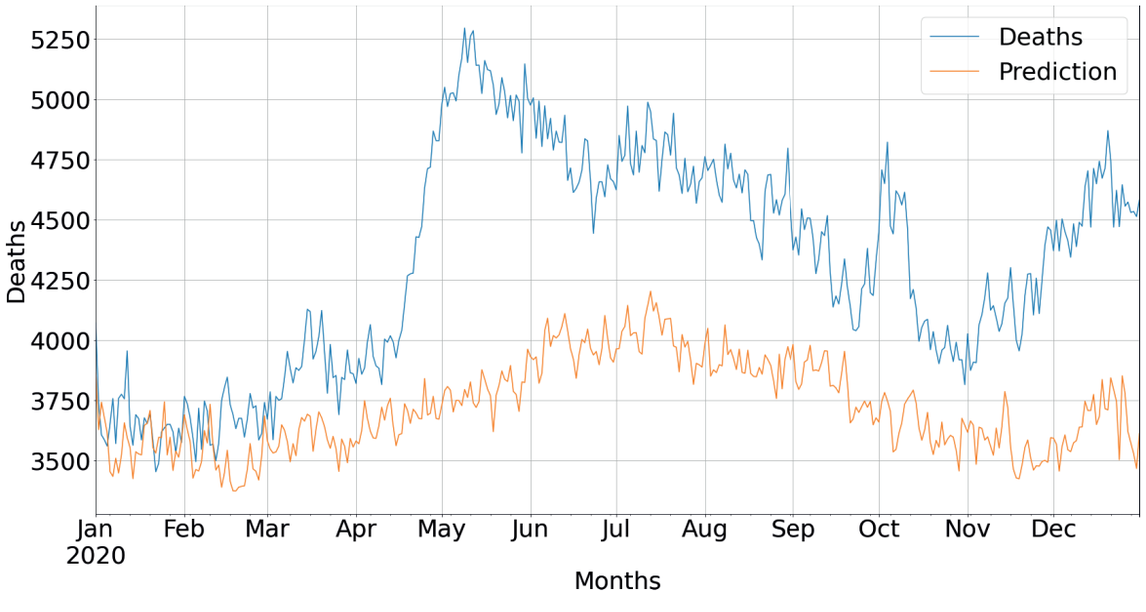


**Table 4: Metrics for 2020 prediction considering the expected error**

| Year | Metrics | Values |
|---|---|---|
| 2020 | RMSE | 694.78 |
| | MAE | 601.12 |
| | $R_2$ | -1.26 |

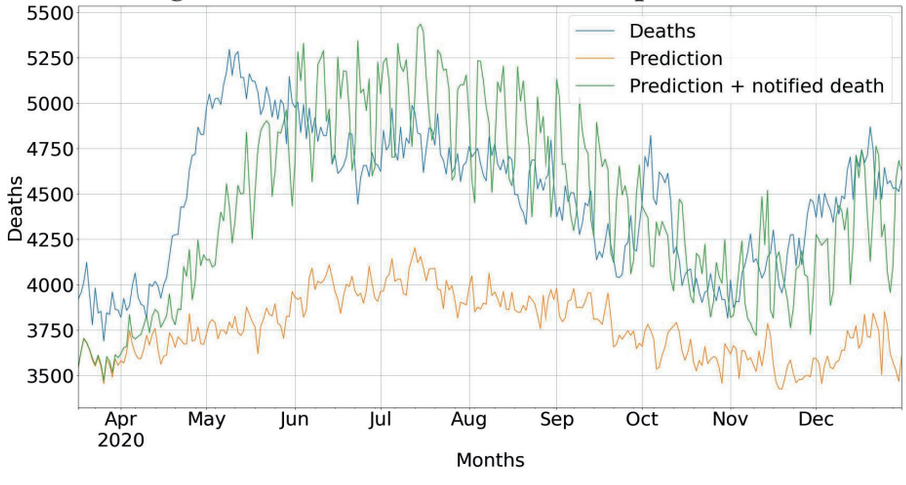
According to the metrics, it is concluded that this model is not a good predictor for 2020, which was already expected. To be considered a model with predictive power, it would be necessary to predict the pandemic effect using only data from past years, which is impossible. What matters in this research is the error contained in this prediction. In other words, a prediction was made of what would have been expected if the COVID-19 pandemic had not happened. In this way, it is possible to estimate the impact caused by the pandemic.

The time series analysis gathered the information necessary for the model adjustment, which, in turn, predicted the number of deaths per day of the desired year. The result to be analyzed is the difference between the actual and the predicted data. This error represents the error associated with the model plus the deaths caused by COVID-19. Considering that in the auto-regression equation there is a component of randomness, it is difficult to estimate with certainty the error of the model, so to simplify the analysis, it is considered that the difference between the actual values and the prediction was caused only by the COVID-19 deaths. Thus, considering the date of the first death record by the disease in Brazil (03/17/2020) and extracting the average error, it is concluded that COVID-19 contributed, in average, to 713 more daily deaths than it was expected.

In addition to observing the COVID-19 impact on national deaths through the difference between the real value and the forecast, it is also possible to understand the underreporting of deaths by this disease. Considering that the difference between the two curves represents the deaths caused by COVID-19, it is enough to add these deaths to the forecast curve to arrive at the real one. The record of daily COVID-19 deaths was obtained through the Johns Hopkins University Center for Systems Science and Engineering (CSSE) Data Repository.

The green curve represents the deaths forecast for 2020 plus the COVID-19 deaths; adding these two, a new curve is obtained in Figure 9. With this new curve, it is possible to see that the notification of deaths by COVID-19 does not directly explain the number of real deaths. At other times, the number of expected deaths plus the deaths by COVID-19 exceed the real curve of deaths. That wise, to understand whether, in the general picture, deaths by COVID-19 in Brazil are being underreported or overreported, it is necessary to calculate the error average in the new curve obtained. Thus, if the average is positive, it means there are more deaths from COVID-19 than what is being reported, that is, there is underreporting. On the other hand, if the average is negative, it means there is an overreporting of these deaths.

**Figure 9. Notified deaths added to 2020 prediction**




The errors average is positive, which means that even considering the reported deaths, the prediction stayed below the real curve. So, it is estimated that in 2020 there will be more deaths from COVID-19 than reported.

## CONCLUSION

The analysis of time series and the study of their predictive models proved to be significant when modeling the scenario of the pandemic in Brazil. This research showed that the number of deaths per day could be described in the form of a time series. There- fore, it can be better understood through two tasks: analysis and prediction of time series. The analysis brings behavior characteristics, while the data's extrapolation by the forecast delimits a scenario from which decisions can be made.

When the prediction of the number of deaths per day is made, it is possible to interpret the associated error to get a sense of the excess deaths, which can be

used to estimate the underreporting of deaths caused by COVID-19. The model presented here proved to be a good predictor for years wherein the mortality behavior was well known. However, compared with 2020, its performance was greatly affected, which makes it not an efficient predictor for that year. This performance loss was expected, and it is in line with the study context, which is to use the error of this prediction to estimate the impact of COVID-19 on daily deaths. With this, it is possible to get an idea of how much COVID-19 influenced the number of daily deaths, in addition to estimating the underreporting of deaths caused by this disease. It is recommended to improve the trend and seasonality estimators for future work. It is also possible to approach models not included in this research, such as SARIMA, which extends the ARIMA model to encompass the time series seasonality. It is also worth mentioning the possibility of research in the field of Artificial Intelligence (AI), that is, using artificial neural networks (ANN) to see how these networks can describe this data and how their predictions would differ from the predictions made by time series models.

## REFERENCES


1. Phelan AL, Katz R, Gostin LO. The novel coronavirus originating in Wuhan, China: challenges for global health governance. *JAMA.* 2020;323(8):709-10.
2. Cucinotta D, Vanelli M. Who declares COVID-19 a pandemic. *Acta Bio Medica: Atenei Parmensis.* 2020;91(1):157.
3. Haleem A, Javaid M, Vaishya R. Effects of COVID-19 pandemic in daily life. *Current Medicine Research and Practice.* 2020;10(2):78.
4. Morato MM, Bastos SB, Cajueiro DO, Normey-Rico JE. An optimal predictive control strategy for COVID-19 (Sars-Cov-2) social distancing policies in Brazil. *Annual Reviews in Control.* 2020;50:417-31.
5. Orellana JDY, da Cunha GM, Marrero L, Moreira RI, da Costa Leite I, Horta BL. Excesso de mortes durante a pandemia de COVID-19: subnotificação e desigualdades regionais no Brasil. *Cadernos de Saúde Pública.* 2021;37(1):E00259120.
6. Shumway RH, Stoffer DS. *Time series analysis and its applications.* New York: Springer; 2000.
7. Chatfield C. *The analysis of time series: an introduction.* New York: Chapman And Hall/CRC, Routledge, 7th edition; 2019.
8. Brownlee J. *Introduction to time series forecasting with Python: how to prepare data and develop models to predict the future.* Machine Learning Mastery; 2017.
9. Box GEP, Jenkins GM, Reinsel GC, Ljung GM. *Time series analysis: forecasting and control.* Wiley; 5th edition; 2015.