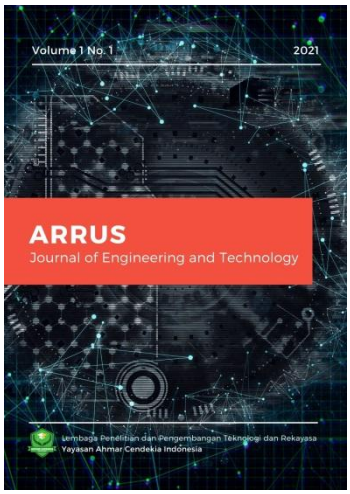
https://doi.org/10.35877/jetech4398

OPEN ACCESS

ISSN : 2776-7914 (Print) / 2807-3045 (Online)



RESEARCH ARTICLE

TSA App by R Shiny: Time Series Analysis Application for Univariate Series Data

Agung Tri Utomo[1]*, Ansari Saleh Ahmar[1], Muhammad Kasim Aidid[1], Zulkifli Rais[1], Muh. Qodri Alfairus[2]

[1]Statistics Department, Faculty of Mathematics and Natural Science, State University of Makassar, Makassar, Indonesia.
[2]Accounting Department, Faculty of Econimics and Bussiness, State University of Makassar, Makassar, Indonesia.

*Corresponding author: Agung Tri Utomo, Statistics Department, Faculty of Mathematics and Natural Science, State University of Makassar, Makassar, Indonesia

E-mail: agungtriutomo@unm.ac.id

Abstract: Time series analysis is a statistical method used to model and forecast sequential data over time. This modeling is typically performed using software, but most analytical tools require paid licenses. To address this issue, the TSA App by R Shiny is developed as an open-source application that is easily accessible. The application features a dashboard-based interface designed to help users perform univariate time series analysis without requiring programming skills. This study compares the analysis results of the TSA App with other software such as R Studio, Minitab, and Python. The results show that the TSA App produces comparable outputs in terms of visualization, ARIMA modeling, and forecasting accuracy. Therefore, the TSA App provides a practical and legal solution for time series analysis, especially for users who are unfamiliar with coding.

Keywords: Programming Skills, Time Series Analysis, TSA App, R Shiny, Univariate Data.

1. Introduction

In current statistics learning, the ability to use applications for analysis is essential. One commonly used analysis for forecasting is time series analysis. According to Wei (2006), time series analysis is a statistical methodology used to analyze, model, and forecast data that is collected sequentially over time. Forecasting time series data can be performed using various applications to predict market trends, economic performance, and more. This analysis has several methods or models that can be used for time series data, such as the Autoregressive Integrated Moving Average (ARIMA) model. The ARIMA model is the most widely used model for analyzing and predicting time series data (Cryer & Chan, 2008). However, in the field of computing, users still require analysis applications that simplify the modeling process. Many users rely on pirated applications to facilitate such analyses.

Nowadays, technology develops rapidly and becomes increasingly sophisticated by the day. This advancement brings both positive and negative impacts. One negative impact is software piracy. According to Majid & Ahmadian, (2020), software piracy is a crime related to Copyright Law. Applications that are protected by copyright typically come with paid licenses. Due to the licensing system, software becomes the exclusive right of the owner, which means it can only be used, distributed, reproduced, or modified by others under the conditions stated in the license agreement (Danendra et al., 2022). This issue becomes the reason for developing a free and open-source application for analysis.

https://doi.org/10.35877/jetech4398

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

One of the most widely known and legal open-source applications is R Studio/R Console. R Studio or R Console is an open-source application that is very popular in the field of data science and statistics. This application allows users to conduct analyses using coding. However, coding presents its own challenges for users. Based on previous studies regarding the development of R packages for GUI-based time series analysis by Warella (2010) and Utomo et al. (2018), this study aims to update the previous research by adding new features using R Shiny and addressing issues related to information system ethics violations caused by the use of pirated software (Jerry et al., 2023).

2. Literature Review

2.1. *Time Series Data*

Time series data is a sequence of data measured over time at equal intervals (Masha & Rinaldi, 2023). This data is attained through a series of nonstop compliances measured grounded on time, such as financial data, which is distributed as a series of compliances of arbitrary variables.

2.2. *Stationary of Data*

Stationarity of data in terms of friction can be examined using the Box-Cox Transformation, where the data is considered stationary if the rounded value is equal to 1 (Muzakki et al., 2022). According to Gujarati (2003), stationarity is a critical property of a time series in which its main statistical characteristics mean, variance, and covariance remain constant over time.

Let Y_t represent an observation at time t, and $Y_t$is a random variable; the sequence of random variables $\{Y_1, Y_2, \dots, Y_t\}$ is appertained to as a stochastic process (Riyanto & Mulyono, 2019). Thus, the stationarity of a time series is not told by specific time points at which the data is observed. Statistically, the conditions for stationarity are as follows:

$$E(Y_t) = \mu$$

$$Var\ (Y_t) = E(Y_t - \mu)^2 = \sigma^2$$

$$\gamma_k = E[(Y_t - \mu)(Y_{t-k} - \mu)]$$

Where:

$E(Y_t)$: The expected value of the observation at time t
μ : The mean of the time series
$Var\ (Y_t)$: The variance of the observation at time t
σ^2 : The constant variance for all observations over time
γ_k : The autocovariance at lag k between observations

2.3. *ARIMA Modelling*

The ARIMA modeling approach, which combines the components of autoregressive, integrated, and moving average, is one of the classical methods for forecasting time series data (Hamzah et al., 2024). According to Cryer & Chan (2008), the ARIMA model consists of three main components, which are represented as ARIMA(p, d, q).

$$\left(1 - \phi_1 B - \phi_2 B - \cdots - \phi_p B^p\right)(1 - B)^d Y_t = \left(1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^p\right)\backslash \epsilon_t$$

Where $B^k Y_t = Y_{t-k}$

2.4. *Forecasting*

Forecasting is the science used to predict or estimate future events, serving as a foundation for planning, monitoring, and decision-making regarding future occurrences, based on known values from past issues (Fauzani & Rahmi, 2023).

2.5. *R Shiny*

https://doi.org/10.35877/jetech4398

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

R Shiny is one of the application packages available in R Studio/R Console that allows users to create dashboard applications which can be run online. This package can be accessed via a dedicated website to run the application.

The components of the R Shiny program are divided into two main parts: the server and the user interface (UI) (Andreanto et al., 2021). The server part is responsible for performing the data analysis simulations, which are then sent to the output section, while the user interface part is responsible for designing the layout based on the user's needs.

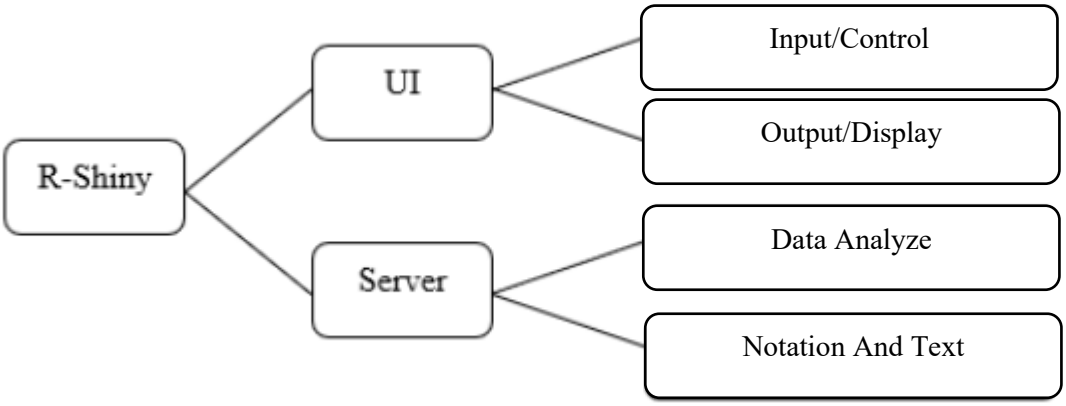


Figure 1. Structure of R Shuny Components

3. Research Method and Materials

3.1. *Application Development*

Based on the analysis stages from previous research by Utomo et al. (2018), the following stages can be carried out:

3.1.1. System Needs Analysis

Based on the analysis stages from previous research by Utomo et al. (2018), the following stages can be carried out:

3.1.2. System Design

The design stage involves adding to the system the features that are still needed in the current system.

3.1.3. System Implementation

The design stage involves adding to the system the features that are still needed in the current system.

3.1.4. System Integration and Testing

This stage allows the system to be tested using simulation data based on the system's requirements to evaluate the system's accuracy in analyzing data. Comparisons will then be made between the results of this system and other applications.

3.1.5. Operation and Maintenance

In this stage, the application will be tested with real data to determine how the system performs and to assess how well it can carry out the analysis tasks.

3.2. *Analysis Stages*

(1). Generate simulation data in the form of non-stationary time series data in variance.
(2). Identify the pattern of the time series data, along with the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF)
(3). Perform the Augmented Dickey-Fuller (ADF) test to determine the stationarity of the data.

https://doi.org/10.35877/jetech4398

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

(4). Apply transformations if:

(a). Non-stationary in mean

The transformation used is differencing the data. Differencing can be done up to a maximum of three times.

(b). Non-stationary in variance

The transformation used when the data is non-stationary in variance is the Box-Cox transformation.

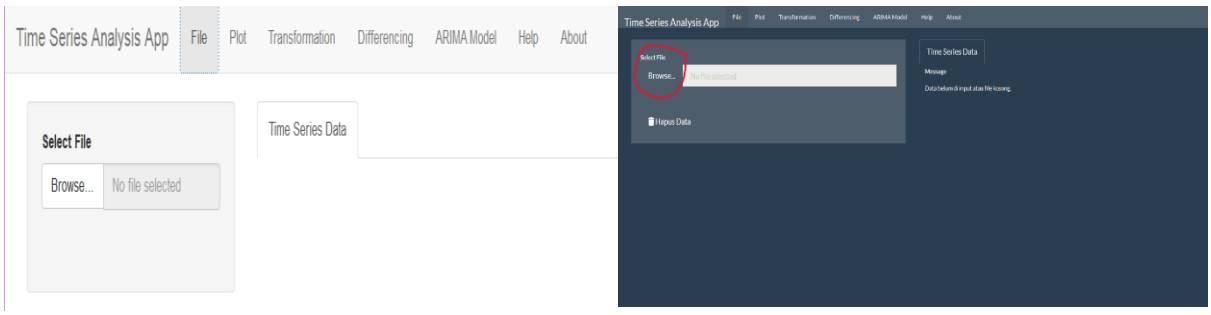
(5). Build the model based on the required data. If no transformation is applied, the model will use AR, MA, or ARMA. If the data undergoes transformation, the model used will be ARIMA.
(6). Determine the best model and compare the results with other applications.
(7). Based on the best model, forecast the next 10 data points.
(8). Compare the analysis results across different applications to assess the consistency of the analysis from the developed application.

4. Results and Discussion

4.1. Application Design

The design of the TSA Apps has undergone changes compared to the previous version of TSA Apps (Utomo et al., 2018). The differences in the application design can be observed in the following sections:

(1). Files Tab



(a) (b)

Figure 2. Files Tab (a) Old, (b) Latest

Figure 2 illustrates the differences between the previous version of the TSA Application and the new one. The new TSA Application includes a new feature: Delete Data. This feature provides users with the ability to remove previously used data and enter new data for re-analysis. Other differences include the information shown in Figure 2(a), which was not available when no data had been entered, but is now present in Figure 2(b).

(2). Plot Tab

https://doi.org/10.35877/jetech4398

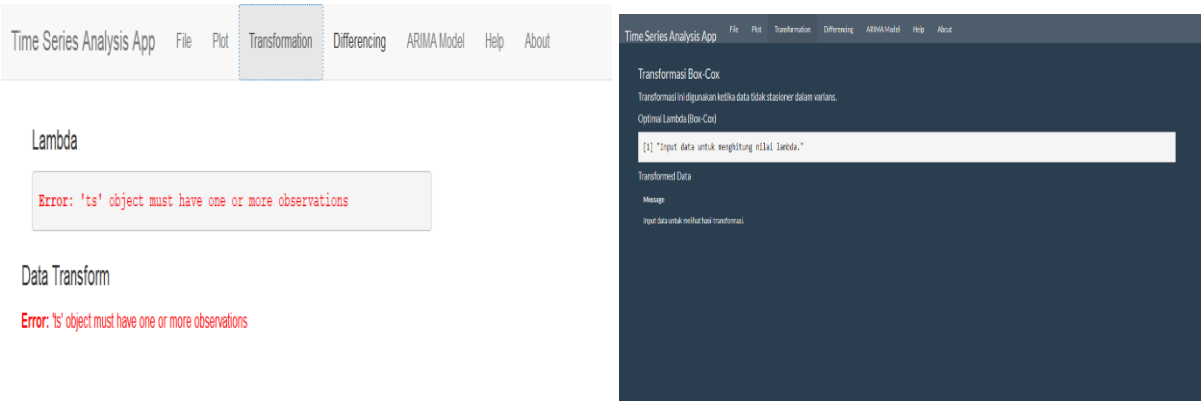
ISSN : 2776-7914 (Print) / 2807-3045 (Online)

(a) (b)

Figure 3. Plot Tab (a) Old, (b) Latest

Figure 3 displays a comparison of the output plots, where Figure 3(a) still shows an error, making it unsuitable for viewing. In Figure 3(b), the error has been replaced with an empty plot, which, once data is input, will display the corresponding output.

(3). Transformation Tab

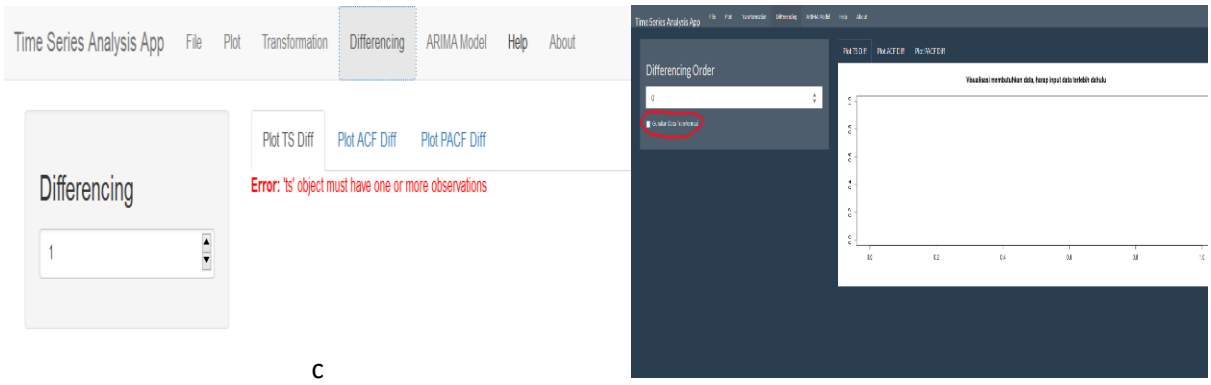


(a) (b)

Figure 4. Transform Box-Cox Tab (a) Old, (b) Latest

Figure 4 illustrates the data transformation section when the data is non-stationary in variance. In Figure 4(a), there is still an error message, but in Figure 4(b), information about the data is displayed. Data that is non-stationary in variance will generate an output from the transformation, whereas if the data is stationary in variance, it will display information indicating that the data is already stationary in variance.

(4). Drift Plot Tab



(a) (b)

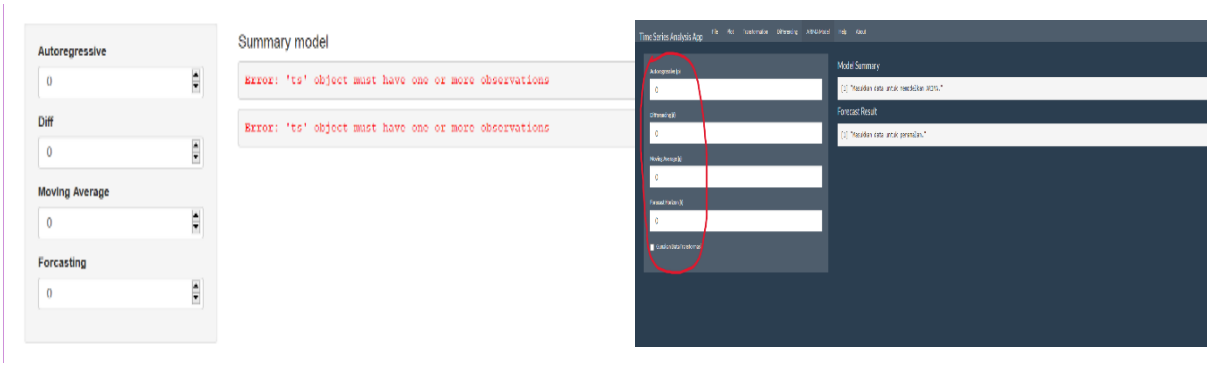
Figure 5. Drift Plot Tab (a) Old, (b) Latest

https://doi.org/10.35877/jetech4398

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

Figure 5 displays the Differencing feature. In Figure 5(a), the feature is still limited, and there is an error message present in the system. This feature is then upgraded by adding the ability for users to utilize the results of the transformation.

(5). ARIMA Model Tab



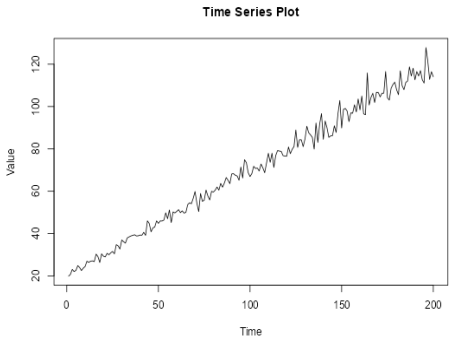
(a) (b)

Figure 6. ARIMA Model tab (a) Old, (b) Latest

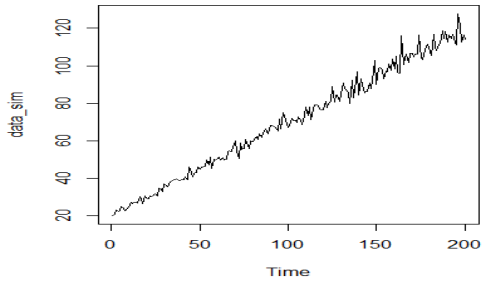
Figure 6 displays the ARIMA modeling feature. The previous version still contains an error message in the system. This feature has since been updated to include an option to use the transformed data. This update makes it easier for users to perform ARIMA modeling, as the application now only requires users to input the necessary values to construct the model.

4.2. Application Implementation Using Simulation Data

In this section, a simulation will be conducted using data that is non-stationary in mean. The data will be analyzed using three different applications to see if the results from the analyses are similar. The initial step will use the results from Auto ARIMA to identify the appropriate model. The time series analysis results will include both stationary data in mean and variance. Below are some sections generated by the applications used.



TSA App

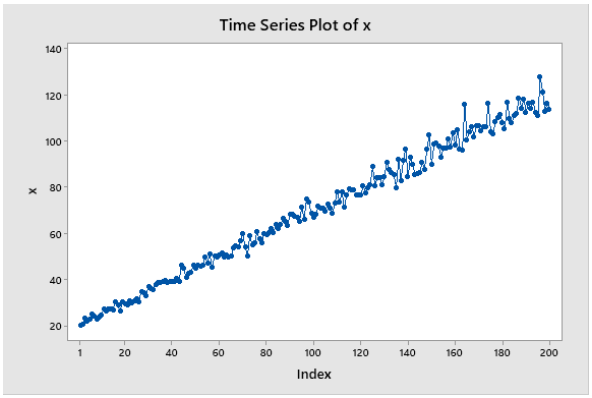


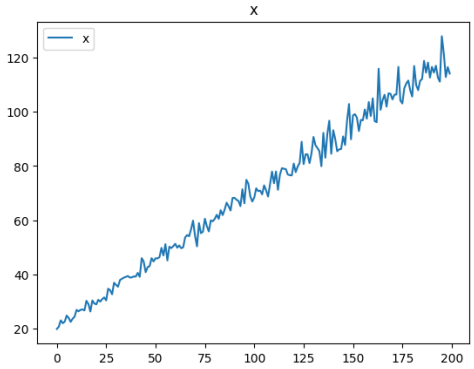
R Studio

https://doi.org/10.35877/jetech4398

OPEN ACCESS

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

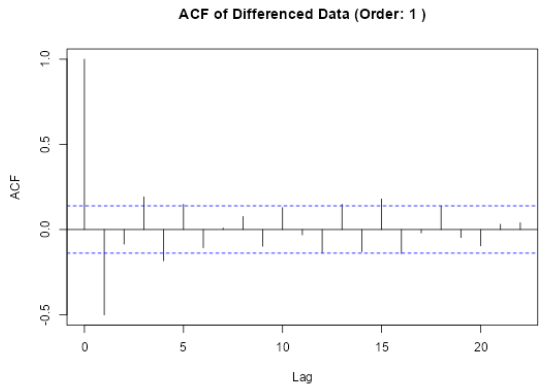


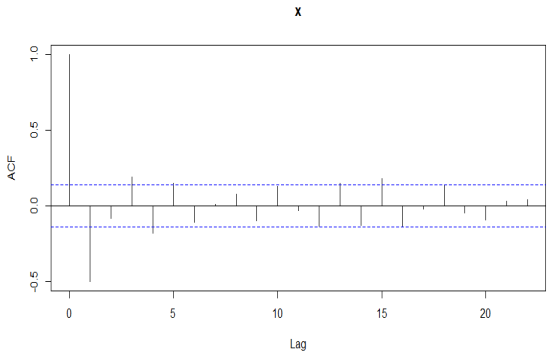


Minitab Python

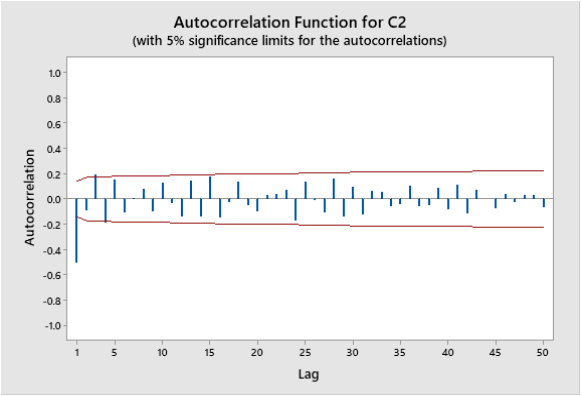
Figure 7. Comparison of Time Series Plots

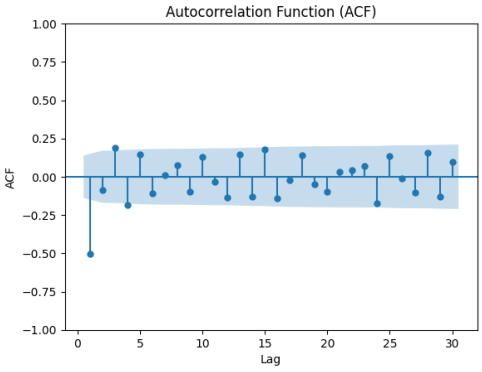
Based on Figure 7, there is no difference in the plots generated from data that is non-stationary in mean. This means that the TSA Apps is able to produce the same output as other applications.





TSA App R Studio





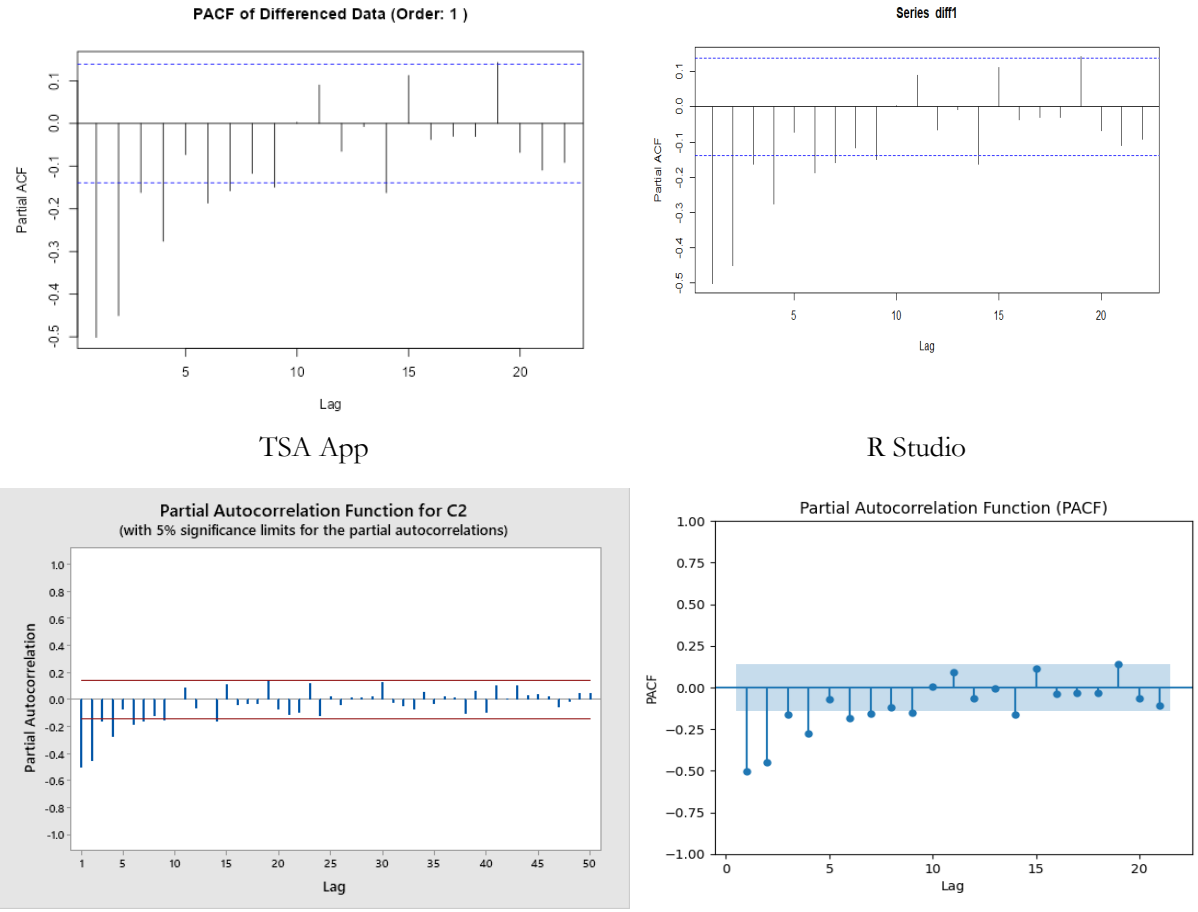
Minitab Python

Figure 8. Comparison of ACF Plots

Based on Figure 8, the differences between the four plots indicate that the plots in TSA App and R Studio start with a lag of 0, while the other two start with a lag of 1. However, when all plots start from lag 1, they will form the same pattern, indicating that TSA App

https://doi.org/10.35877/jetech4398

ISSN : 2776-7914 (Print) / 2807-3045 (Online)



TSA App R Studio

Minitab Python

Figure 9 Comparison of PACF plots

Figure 9 shows a comparison of the PACF plots from each application. It can be observed that each PACF plot displays the same pattern as the TSA App. This indicates that the TSA App is able to analyze the data and produce plots that are consistent with those generated by other applications. Next, for the comparison of the ARIMA model coefficients, refer to Table 1.

Tabel 1. Comparison of Coefficients from Time Series Analysis Results Using TSA Apps, R Studio, Minitab, and Python

Type	TSA Apps	R Studio	Minitab	Python
AR 1	-0.861	-0.861	-0.866	-0.861
AR 2	-0.759	-0.759	-0.767	-0.759
AR 3	-0.433	-0.433	-0.440	-0.433
AR 4	-0.316	-0.316	-0.323	-0.316

Based on Table 1, TSA App produces the same values as R Studio and yields results that are not significantly different from the coefficient values in Minitab and Python. Meanwhile, the coefficient value from Minitab shows a difference of 0.005, which indicates that its value is not far from the one generated by TSA App.

https://doi.org/10.35877/jetech4398

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

Tabel 2. Comparison of SE Coefficients from Time Series Analysis Results Using RStudio, Minitab, and Python

Type	TSA Apps	R Studio	Minitab	Python
AR 1	0.068	0.068	0.069	0.055
AR 2	0.087	0.087	0.088	0.070
AR 3	0.089	0.089	0.091	0.086
AR 4	0.070	0.070	0.072	0.069

The comparison results from Table 2 show the Sum Error Coefficient (SE) values from each application. The comparison between TSA App and the three other applications reveals that they produce nearly identical results in terms of the SE coefficient values.

Tabel 3. Comparison of Forecast Results from Time Series Analysis Using TSA App, R Studio, Minitab, and Python

Forecast	TSA Apps	R Studio	Minitab	Python
1	120.757	120.757	120.862	6.684477
2	119.603	119.603	119.655	-1.153503
3	117.035	117.035	117.007	-2.569036
4	119.619	119.619	119.657	2.584835
5	119.362	119.362	119.378	-0.257322
6	120.738	120.738	120.796	1.375985
7	121.078	121.078	121.122	0.340493
8	120.671	120.671	120.667	-0.407536
9	121.886	121.886	121.928	1.215348
10	122.202	122.202	122.233	0.316084

Table 3 shows the forecast results from each application. The forecast values differ only in the Python application. However, in R Studio and Minitab, the forecast values are slightly different but still not far from the values generated by TSA App. This difference may be caused by variations in the Python function, leading to a different forecast result. The comparison of models can also be observed through the Root Mean Square Error (RMSE) values. The RMSE comparison results from each application are presented in Table 4.

Tabel 4. Comparison of RMSE Values from Time Series Analysis Results Using TSA App, R Studio, Minitab, and Python

TSA App	R Studio	Minitab	Python
3.311	3.311	3.361	3.351

Table 4 shows the model comparison results using the RMSE values for each application. The RMSE values for each application are around 3.3, indicating that the TSA App produces an RMSE value that is very similar to those of the compared applications.

5. Conclusion

For. The TSA App is an application developed to simplify analysis, particularly in time series analysis. The use of R Shiny enables users to perform time series analysis without coding in R Studio, while also helping to reduce reliance on illegal software by providing an open-source alternative. The latest version of the TSA App includes several new features and improvements to existing functions. Simulation and real-data analyses show that the results are nearly equivalent to those produced by other software such as Minitab and Python. Based on these findings, the latest version of the TSA App is capable of performing univariate time series analysis effectively. However, the current limitation is that the app does not yet support multivariate data analysis. Future updates are expected to include features that address this limitation. The TSA App is accessible at: https://triutomo.shinyapps.io/TimeSeriesApps/.

https://doi.org/10.35877/jetech4398

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

References

Andreanto, Y., H., & Rusgiyono, A. (2021). IMPLEMENTASI R-SHINY UNTUK ANALISIS BIPLOT KOMPONEN UTAMA (Studi Kasus: Penggunaan Alat Kontrasepsi pada Peserta Aktif KB di Provinsi Jawa Tengah Tahun 2019). JURNAL GAUSSIAN, 10(4), 499–507. https://doi.org/10.14710/j.gauss.v10i4.33097

Cryer, J. D., & Chan, K.-S. (2008). Time Series Analysis: With Applications in R (Second Edition). Springer. https://doi.org/10.1007/978-0-387-75959-3

Danendra, D. R., Manopo, E. C. K., Wijaya, O. H., & Tanuwijaya, Y. (2022). Kesadaran Etis Dalam Penggunaan Aplikasi Berlisensi Secara Ilegal (Studi Kasus Mahasiswa Pradita University Pada Tahun 2022). Jurnal Pendidikan, Seni, Sains Dan Sosial Humanioral, 1(1), 115–137. https://doi.org/10.11111/nusantara.xxxxxxx

Fauzani, S. P., & Rahmi, D. (2023). Penerapan Metode ARIMA Dalam Peramalan Harga Produksi Karet di Provinsi Riau. Jurnal Teknologi Dan Manajemen Industri Terapan, 2(4), 269–277. https://doi.org/10.55826/tmit.v2i4.283

Gujarati, D. N. (2003). Basic Econometrics (Fourth Edition). McGraw-Hill/Irwin.

Hamzah, M., Fadilah, Novendra, H., Kurnianto, B. A., & Kasih, T. P. (2024). ANALISA DERET WAKTU DENGAN MENGGUNAKAN METODE ARIMA DAN VAR PADA APLIKASI ORANGE. Jurnal Review Pendidikan Dan Pengajaran, 7(1), 330–333.

Jerry, Vendryan, Yanto, A., Nasution, D. F., Tings, V. V., & Sama, H. (2023). PELANGGARAN ETIKA SISTEM INFORMASI DALAM PENGGUNAAN SOFTWARE BAJAKAN. Jurnal Sistem Informasi, Teknologi Informasi, Dan Edukasi Sistem Informasi (JUST-SI), 4(1), 19–30. https://doi.org/10.53682/justsi.v4i1.879

Majid, B. A., & Ahmadian, H. (2020). Overview of Pirated Software on Campus: Educators' Perspective. Cyberspace: Jurnal Pendidikan Teknologi Informasi, 4(2), 118. https://doi.org/10.22373/cj.v4i2.7717

Masha, H. A., & Rinaldi, A. (2023). Metode Peramalan Deret Waktu Menggunakan Model Asymmetric Power Autoregressive Conditional Heteroskedastic (Aparch) Studi Kasus Data Penutupan Harga Saham Mingguan PT Adhi Karya (Persero) Tbk. Jurnal Sains, 2(1).

Muzakki, A. F., Aditama, D., & Anugrah, I. G. (2022). PENERAPAN METODE AUTOREGRESSIVE INTEGRATED MOVING AVERAGE UNTUK MEMPREDIKSI PENGGUNAAN BARANG MEDIS PADA LOGISTIK MEDIS RUMAH SAKIT MUHAMMADIYAH GRESIK. INDEXIA: Informatic and Computational Intelligent Journal, 4(1), 1–16. https://doi.org/10.30587/indexia.v4i1.3595

Utomo, A. T., Ahmar, A. S., & Aidid, M. K. (2018). Pengembangan Paket R untuk Analisis Time Series dengan Grapichal User Interface (GUI). Provided by Repository Universitas Negeri Makassar.

Warella, A. (2010). Pengembangan Paket R untuk Analisis Deret Waktu dengan Antar Muka User Friendly. Institut Pertanian Bogor.

Wei, W. W. S. (2006). Time Series Analysis: Univariate and Multivariate Methods (Second Edition). Pearson Addison Wesley.