International Conference on Actuarial Sciences (ICActS)
Available on-line at: **www.sicash.sliit.lk**

Development of Time Series Model to Predict the Weekly Percentage of Python Programming Language usage

D. M. N. M. Gunawardane[1*], H. M. P. T. Herath[1], W. S. Pitiyekumbura[1], P. L. D. Samodhika[1], A. M. B. T. Athauda[1], E. J. C. U. Amarasinghe[1], T. S. G. Peiris[1]

[1]*Department of Mathematics and Statistics, Faculty of Humanities and Sciences, Sri Lanka Institute of Information Technology (SLIIT), Malabe, Sri Lanka*

Corresponding author*: HS23788244@my.sliit.lk

Abstract

Python's super popular and getting bigger fast. Figuring out how it will be used is super important for planning what to teach, training tech workers, and making good rules, especially in places like Sri Lanka that are just now getting into digital stuff. Therefore, this study aims to predict the weekly global usage of Python. We looked at data from April 21, 2019, to April 21, 2024. We got 262 weeks. This data is entered into Kaggle from Google search interest scores (Nextmillionaire, 2023). This dataset shows the highest interest score for Python in the general world. After trying out a bunch of models, the ARIMA (1,1,1) model with seasonal stuff seemed like the best fit. We taught the model with data from April 21, 2019, to January 28, 2024 (250 weeks) and checked it with data from February 4, 2024, to April 21, 2024 (12 weeks). We tested the model to make sure it was doing things right, and the leftovers looked random, which is a good thing. The MAPE (Mean Absolute Percentage Error) for the validation data is 6.04%. This shows the ARIMA model is pretty good at guessing Python usage over time. Because the guesses are pretty accurate and consistent, it looks like Python usage of global is going up steadily. This means Python is a big deal with both Data Science & Analytics, Machine Learning & AI, Cloud Computing & DevOps, Automation & Scripting. This info should help schools, training places, and the government make smart choices about teaching digital skills.

Keywords: Python usage, Time series forecasting, ARIMA, MAPE

Introduction

Pythons become super popular in the last ten years or so. It's easy to read and use, which is why it's all over the place – data stuff, making things automatic, websites, and even AI. Knowing Python is a big deal now for students, people at work, and all kinds of businesses everywhere. Since everyone's going digital, figuring out what's up with programming languages like Python can really help with teaching, job planning, and making progress with tech (Python Software Foundation, 2024). Even though Python's taking over, there aren't a ton of studies that guess how much it'll be used later, especially if we're talking about the whole world. Most of what's out there looks at how it's used in certain places, but not with solid number-crunching. So, this study tries to fill that hole. We're using some time-predicting tricks to check out how Python's been used worldwide for five years. That might give some clues to help people make choices in schools and companies (Iqbal & Qureshi, 2020; Van Rossum, 2019).

https://doi.org/10.54389/IJDU3949

Other studies on what's hot in tech show that models like ARIMA can grab short-term ups and downs and when things usually happen at certain times. But not a lot of folks have used these models to look at which programming languages are getting big. We looked at weekly info between April 21, 2019, and April 21, 2024 (that's 262 weeks total). This shows how Python's been growing. We trained the model with 250 weeks of data (from April 2019 to January 2024) and then tested it with the last 12 weeks (February 2024 to April 2024). We used programs like EViews (EViews, 2023) and Minitab to build and test the predicting models. The aim of the study is to develop time series ARIMA model to the Weekly Percentage of Python Programming Language usage**.**

Materials and Methodology

Secondary Data

This study aims to forecast weekly Python programming language usage globally using time series analysis. Secondary data was collected from online databases that track programming trends, covering the period from 21st April 2019 to 28th January 2024 with the ARIMA model. Data analysis was performed using Excel, Minitab and E-Views8 software. It was selected due to its reliability and relevance to the research objective.

Statistical Analysis

In this study, the time series Auto Regressive Moving Average (ARIMA) model was used. It includes both autoregressive (AR) and moving average (MA) components. This model is especially useful for analysing and forecasting future values in time series data. It is also known as the Box–Jenkins ARMA (p, q) model and can be written as:

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + e_t - \theta_1 e_t - \theta_2 e_2 \cdots - \theta_q e_{t-q}$$

First, the series was tested for stationarity using the Augmented Dickey Fuller (ADF). If it is not stationary the first difference $Z_t = Y_t - Y_{t-1}$ was obtained to make the series stationary. (Box et al., 2015; Hamilton, 1994).

Based on the pattern of ACF and PACF of the stationary series, several models are postulated. To select the best model from the selected models, several biased models are assumed by comparing the sample ACF and sample PACF (Partial Autocorrelation Function) patterns with the corresponding theoretical ACF and PACF. The best fitted model from those identified models is selected based on AIC, SC, HQ and Log-likelihood (Box et al., 2015).

Results and Discussion

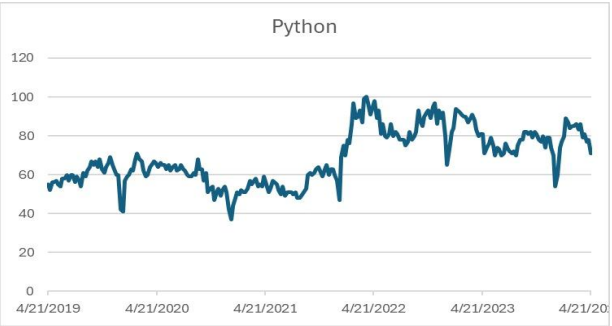


Figure 1: *The popularity trend of Python Variability of the original series*

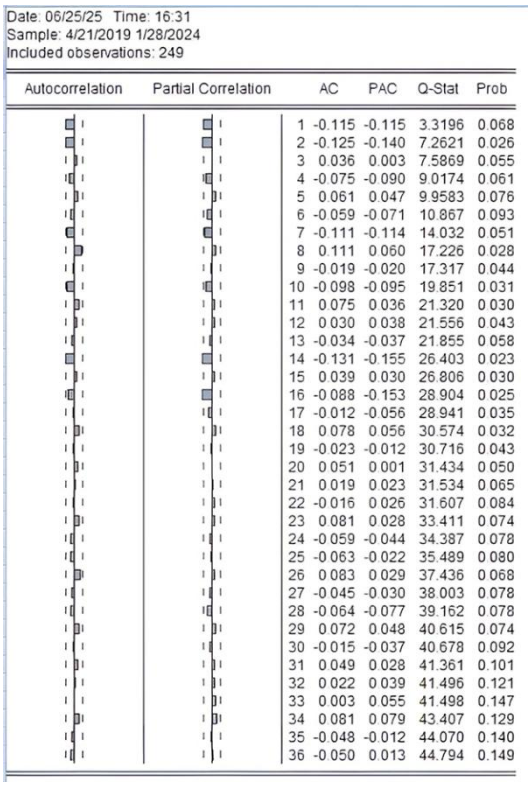
The time series data from 2019 to 2024 shows the popularity trend of Python, with values ranging between 37 and 100. The average popularity is 68.34, and the standard deviation is 14.07. From 2019 to early 2021, interest remained steady between 50–70. P value is significant. It can colude by 95% confidence data significantly deviated from normalityHowever, in mid-2021, there was a sharp rise, reaching peak values near 100. From 2022 onwards, Python's popularity remained high with moderate fluctuations, showing overall growth in its global usage.

The ADF test is used to check whether a time series is stationary or has a unit root (non-stationary) (Hamilton, 1994). Accordingly, two hypotheses were: Null Hypothesis (H_0): The series has a unit root (non-stationary), vs. Alternative Hypothesis (H_1): The series is stationary.

The original series' Augmented Dickey-Fuller (ADF) test result yields a p-value of 0.0905, above the significance level of 0.05. As a result, the null hypothesis cannot be rejected, proving that the original series has a unit root and is not stationary. But after applying the first difference, the associated p-value is 0.0000, which is significantly below 0.05, and the ADF test statistic is -17.52158, which is significantly less than the critical values at the 1%, 5%, and 10% levels. As a result, the differenced series' null hypothesis is rejected, proving its stationarity. Therefore, following initial differencing, the original series becomes stable, indicating that it is integrated of order one.

Correlogram of 1st difference (stationary) series

The figure 2 shows significant lags at 1 for both ACF and PACF, but everything else is within the confidence bands. This suggests an AR(1), MA(1), or a mix of both might work. By comparing expected ACF and PACF along with the observed sample ACF and PACF the following five models: ARIMA(1,1,0), ARIMA (0,1,1), ARIMA(1,1,1), ARIMA(0,1,2), and ARIMA(2,1,1) were considered as parsimonious models. We chose them because they're simple and make sense theoretically, plus they seem like the best, most basic models for our stationary series after the first difference.


Date: 06/25/25 Time: 16:31
Sample: 4/21/2019 1/28/2024
Included observations: 249

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.115	-0.115	3.3196	0.068
		2	-0.125	-0.140	7.2621	0.026
		3	0.036	0.003	7.5869	0.055
		4	-0.075	-0.090	9.0174	0.061
		5	0.061	0.047	9.9583	0.076
		6	-0.059	-0.071	10.867	0.093
		7	-0.111	-0.114	14.032	0.051
		8	0.111	0.060	17.226	0.028
		9	-0.019	-0.020	17.317	0.044
		10	-0.098	-0.095	19.851	0.031
		11	0.075	0.036	21.320	0.030
		12	0.030	0.038	21.556	0.043
		13	-0.034	-0.037	21.855	0.058
		14	-0.131	-0.155	26.403	0.023
		15	0.039	0.030	26.806	0.030
		16	-0.088	-0.153	28.904	0.025
		17	-0.012	-0.056	28.941	0.035
		18	0.078	0.056	30.574	0.032
		19	-0.023	-0.012	30.716	0.043
		20	0.051	0.001	31.434	0.050
		21	0.019	0.023	31.534	0.065
		22	-0.016	0.026	31.607	0.084
		23	0.081	0.028	33.411	0.074
		24	-0.059	-0.044	34.387	0.078
		25	-0.063	-0.022	35.489	0.080
		26	0.083	0.029	37.436	0.068
		27	-0.045	-0.030	38.003	0.078
		28	-0.064	-0.077	39.162	0.078
		29	0.072	0.048	40.615	0.074
		30	-0.015	-0.037	40.678	0.092
		31	0.049	0.028	41.361	0.101
		32	0.022	0.039	41.496	0.121
		33	0.003	0.055	41.498	0.147
		34	0.081	0.079	43.407	0.129
		35	-0.048	-0.012	44.070	0.140
		36	-0.050	0.013	44.794	0.149

Figure 2: *ACF and PACF of the First Difference Series*

Identification of the Best-Fitted Model

Based on the results in Table 1, the ARIMA(1,1,1) model was picked as the best fitted based on the above four indices and the status of significance of the parameters. It had the lowest AIC and SC values, suggesting a solid balance of fit and complexity. Its log likelihood was highest, meaning it explained the data well. The AR and MA parameters in the model are statistically sound, backing up its choice. Furthermore, the correlogram of the residual of the fitted model showed that all the ACF and PACF at different lags are not significant (Fig. 3) confirming that the residuals are random. It was also found that the residuals are not significantly deviated from the normality and not significantly deviated from constant variance. Hence, it can be concluded with 95% confidence that the residuals are random, identically distributed, and not correlated. The randomness in the residual correlogram shows the model is good enough. It means the model has captured the time series structure. This checks out with the idea that a good ARIMA model should have residuals that are statistically white noise. So, we don't need to tweak the model anymore (Hyndman & Athanasopulos, 2021; Markridakis et al., 1998).

Table1: *Comparison of Estimated Models*

Model Criteria	ARIMA(0,1,2)	ARIMA(2,1,0)	ARIMA(0,1,1)	ARIMA(1,1,0)	ARIMA(1,1,1)
AR(1)	N/A	Significant	N/A	Not Significant	Significant
AR(2)	N/A	Significant	N/A	N/A	N/A
MA(1)	Significant	N/A	Significant	N/A	Significant
MA(2)	Significant	N/A	N/A	N/A	N/A
AIC	5.968688	5.976496	6.007694	5.986758	5.966418
SC	6.011067	6.01912	6.007694	6.015092	6.008919
HQ	5.985746	5.993657	5.990814	5.998164	5.983528
Log Likelihood	-740.1017	-735.0973	-742.4405	-740.358	-736.8359

Comparison of Observed and Predicted Values of Training Data Set and Independent Data

The Figure 3 shows the relationship between the actual values and the predicted values in python usage. According to the scatterplot, the red line is close to the actual data, so the predicted values are close to the actual data. Considering the required data point description, it shows a strong positive correlation. It was found that the percentage error between the actual values and the fitted values for the training data set (April 21, 2019, to January 28, 2024) varies from -0.0131% to 1.0000%.

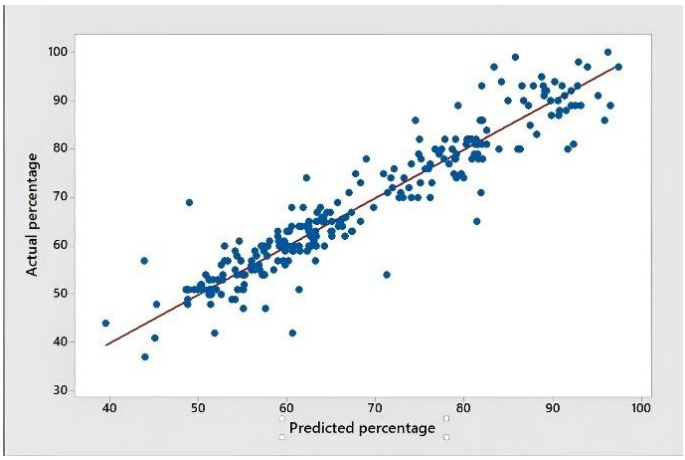


Figure 3: *Comparison of predicted and actual for the training set*

Validation of the Model for the Independent Data Set

It was found that the percentage error between the actual values and the fitted values for the independent data set (February 4, 2024, to April 21, 2024) from -0.1167% to 0.1002%. The mean absolute percentage error (MAPE) is 6.04% and Inequality coefficient (0.039) is closer to 0 – an indication that the model's predicting power was good (James et al., 2013).

Short Term Prediction

Table 2: *Predicted Values of the Subsequent Weeks*

Week	**4/28/2024**	**5/05/2024**	**5/12/2024**	**5/19/2024**
Predicted Values	79.37463	79.46585	79.55706	79.64827

Using the best fitted model, the percentage of python usage for the coming weeks was predicted. These can be used to guide the study to appropriate opportunities.

Conclusions

ARIMA(1,1,1) was recommended as the best fitted model to predict the popularity of Python based on the weekly percentage values of the popularity of using the Python programming language data from April 21, 2019 to April 21, 2024. The errors of the best fitted model were found as white noise. The model was confirmed based on the significance of the parameters and values of statistical indices such as the Akaike Information (AIC), Schwarz Criterion (SC), and log likelihood criteria. The range percentage errors were low for both training data set and validation data set. The model can be written as, $(1-0.7612B)(1-B)Y_t=0.0912+(1-0.8970B)e_t$ and the B represents the backshift operator such that $B^lY_t=Y_{t-l}$.

Acknowledgement

I would like to express my sincere gratitude to the Sri Lanka Institute of Information Technology (SLIIT), Malabe, for providing me the opportunity to conduct this research. I extend my deep appreciation and thanks to the Head of the Department of Mathematics and Statistics and to the esteemed lecturers. Their valuable guidance and support played a significant role throughout this project. Their knowledge, dedication, excellence, and understanding greatly contributed to the success of this work.

References

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). T*ime series analysis: Forecasting and control* (5th ed.). Wiley.

Chatfield, C. (2004). *The analysis of time series: An introduction* (6th ed.). CRC

EViews. (2023). *EViews user guide: Forecasting with ARIMA models*. https://www.eviews.com/help/content/series-Automatic_ARIMA_Forecasting.html

Hamilton, J. D. (1994). *Time series analysis*. Princeton University Press.

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and practice* (3rd ed.). OTexts. https://otexts.com/fpp3/

Iqbal, S., & Qureshi, M. A. (2020). Adoption of Python programming language in academia and industry: A review. *International Journal of Computer Applications, 176*(10), 1–6. https://doi.org/10.5120/ijca2020920373

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in R*. Springer.

Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (1998). *Forecasting: Methods and applications* (3rd ed.). Wiley.

Nextmillionaire. (2023). *Programming languages trend over time* [Dataset]. Kaggle. https://www.kaggle.com/datasets/nextmillionaire/programming-languages-trend-over-time

Python Software Foundation. (2024). *Python success stories*. https://www.python.org/about/success/

Van Rossum, G. (2019). The future of Python: Trends and perspectives. *Communications of the ACM, 62*(3), 28–30.