Uz_Inflation – Time Series Analysis of Uzbekistan's Inflation

By Arslonbek Ishanov

This report provides a detailed description of each step taken to analyse the inflation levels in Uzbekistan from January 2010 to October 2024. After the initial analysis, the author fit the time series on a ARMA model, uses it to predict the next step in the time series (inflation in November 2024), and evaluates the model.

Task 1. Obtain the time series.

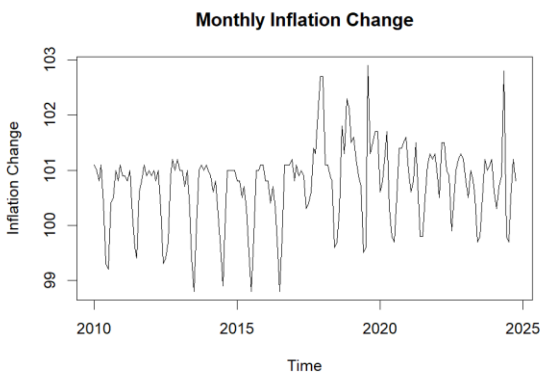

Figure 1. Monthly Inflation Change in Uzbekistan (Compared to Previous Month).

Figure 1 shows monthly inflation change in Uzbekistan compared to previous month in percentage. The figure was obtained using the following code:

```
plot(ts_data, main = "Monthly Inflation Change in Uzb", ylab = "Inflation Change", xlab = "Time")
```

The time series span from January 2010 to October 2024. There is a noticeable change which occurred around 2017, where the month on month inflation rose to approximately 102.5% and kept varying withing 2% over the remaining period.

Task 2. Detrend the series.

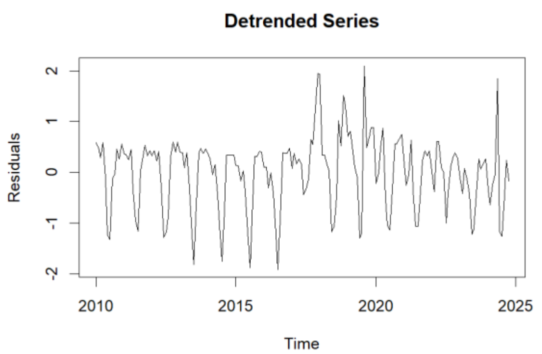

Figure 2. Detrended Time Series.

Figure 2 demonstrates the detrended time series (DTS) from Figure 1 which was obtained in two steps:

1. Identified the trend using a linear regression model (which has a formula of $y_t = \beta_0 + \beta_1 * t + \epsilon_t$).
2. Subtracted the trend from the series.

The line graph was plotted with the following code:

```
plot(detrended_series, main = "Detrended Series", ylab = "Residuals", xlab = "Time")
```

It is worth noting that there is a noticeable seasonality throughout the period.

Task 3. Create ACF & PACF. Is it a unit root?

The Figure 3 demonstrates the AutoCorrelation Function (ACF) and Partial AutoCorrelation Function (PACF). Because the DTS and its lagged version plumet around lag 0.1, the time series seem to be stationary.

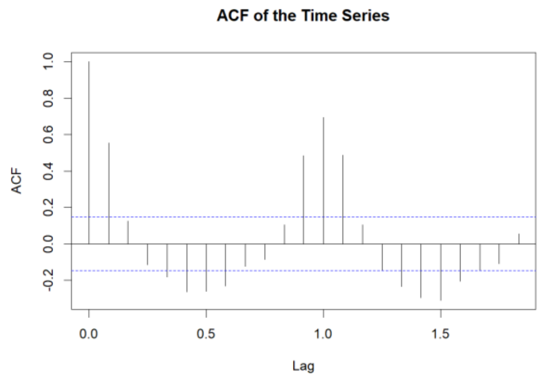

Figure 3.1. The ACF Plot of the DTS.

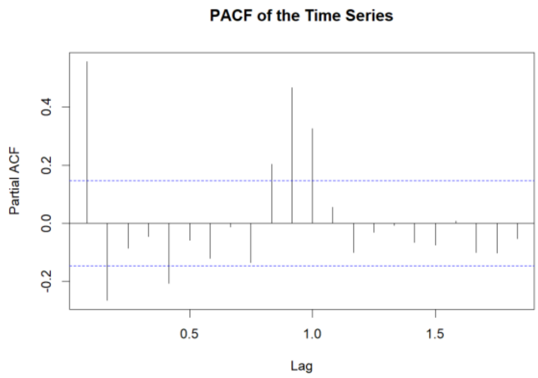

Figure 3.2. The PACF Plot of the DTS.

The ACF is defined by the function at lag k:

$$\rho_k = \frac{E[(x_t - \mu)(x_{t-k} - \mu)]}{E[x_t - \mu]^2}$$

Where:

E – the expected value,

x_t – DTS

μ – the mean of the time series (Adhikari and Agrawal, 2013).

While PACF is defined as follows:

$$\phi_k = \mathrm{Corr}(x_t, x_{t+k} \mid x_{t+1}, x_{t+2}, \ldots, x_{t+k-1})$$

Where:

ϕ_k is the PACF,

x_t – DTS,

k – the lag order

Corr is the correlation. (Frain, 1992).

Task 4. Check for autocorrelation. Compare the results with the task 3.

After performing the Ljung-Box test with the code below, where the null hypothesis (H_0) suggests that the residuals are distributed independently, while the alternative (H_1) implies that there is significant autocorrelation:

Box.test(detrended_series, type = 'Ljung-Box')

The resulted p-value = 7.772e-14 < 0.05 provided enough evidence to reject the null hypothesis and concluded that there was significant autocorrelation on at least one lag. This was consistent with both ACF and PACF plots which had noticeable spikes at lag 1.0 suggesting that there as significant correlation at that lag.

Task 5. Check for Stationarity and compare the results with the task 3.

To accurately assess the non-/stationarity of the DTS, two tests were performed.

1. Augmented Dickey-Fuller Test, where H_0 suggests that the series had a unit root, and therefore, was not stationary, while the alternative was the stationarity.
   After running the test:
   adf.test(detrended_series)
   The resulting p-value of $0.01 < 0.05$ provided enough evidence to reject H_0. Thus, the DTS was stationary.
2. To ensure the accuracy of such statement, the second test, Kwiatkowski-Phillios-Schmidt-Shin (KPSS), was performed:
   kpss.test(detrended_series)
   The H_0 meant stationary DTS, while H_1 – non-stationarity. The p-value = 0.1 did not provide enough evidence to reject H_0, implying the stationarity of the DTS.

Both results were consistent with the observations from ACF and PACF in task 3. Therefore, there was significant evidence that the series are stationary.

Task 6. Normality test.

To ensure the accuracy of the assessment, three tests for normality were performed. In all three tests, H_0 suggested a normally distributed time series, while H_1 stated that the DTS was not normally distributed.

1. Lilliefors (Kolmogoros-Smirnov):
   lillie.test(coredata(detrended_series))   # p-value = 2.975e-05
2. Anderson-Darling:
   ad.test(as.numeric(coredata(detrended_series)))   # p-value = 4.002e-10
3. Shapiro-Francia:
   sf.test(detrended_series)   # p-value = 1.032e-05

Small p-values in all the three tests provided enough evidence to reject H_0. Thus, the series was not normally distributed.

Task 7. Fit an ARMA model & determine the best lag.

As noted in task 2, the DTS had clear seasonality. To fit the data into the suitable model, the seasonality had to be accounted for. Thus, Seasonal AutoRegressive Integrated Moving Average (SARIMA) model was the best candidate. The model is defined in r by the formula:

$$model = ARMA(p, d, q)[P, D, Q](s)$$

Where:

p – is the number of AR terms,

d – differencing,

q – the number of MA terms,

P – is the number of seasonal AR terms,

D – seasonal differencing,

Q – number of seasonal MA terms with specified period *s* (Frain, 1992).

After implementing the model in r:

sarima_model = arima(detrended_series, order = c(1, 0, 1), seasonal = list(order = c(0, 0, 1), period = 12))

Where:

p=1,

d=0,

q=1,

P=D=0,

Q=1,

s=12.

The parameters were selected from ACF (q, Q, because lag 1 had a significant spike, outside the blue line) and PACF (p, P, because lag 1 had a significant spike, outside the blue line). D and d were set to 0 as the time series was detrended in task 2.

After experimenting with the parameters above, the final model was constructed:

sarima_model = arima(detrended_series, order = c(1, 0, 2), seasonal = list(order = c(0, 1, 2), period = 12))

The initial model's Akaike Information Criterion (AIC) was 283.31, while the final model's AIC = 204.92, with the lower value suggesting that the final model fit the series better.

The AIC of the final model was even slightly better than the auto.arma suggested model which produced AIC = 205.02.

Task 8. The coefficients, the equation for the model. Make 1-step ahead prediction.

The final model's coefficients were:

ar1	ma1	ma2	sma1	sma2
0.9383	-0.6924	-0.1264	-0.7106	-0.1064

This implies that the final model's mathematical equation is:

$$(1 - 0.9383B)x_t = (1 - 0.6924B - 0.1264B^2)(1 - 0.7106B^{12} - 0.1064B^{24})\epsilon_t$$

Where:

B is the backshift operator,

x_t – the DTS,

ϵ_t – is the white noise. (Adhikari and Agrawal, 2013).

To perform a 1-step forecasting with the model, the following code was used:

```
forecast(sarima_model, h=1)
```

Which produced the following result:

```
         Point Forecast      Lo 80     Hi 80      Lo 95    Hi 95
Nov 2024      0.3328849 -0.2026685 0.8684383 -0.4861734 1.151943
```

Indicating that the November's inflation rate would be around 99.5138266% (slight decrease), or 101.151943% (slight increase) compared to previous month (i.e., October) with 95% confidence, and the most likely value being 100.3328849%.

Task 9. Check for residual mean, finite & constant variance, zero autocovariance.

To access whether the fitted model is a good fit for the data, the residuals should behave like white noise. Therefore, three tests were performed to examine these criteria:

1. **Mean**: mean(residuals(sarima_model)) # -0.006575348, which is close enough to 0.
2. Studentized Breusch-Pagan test for **constant variance**: bptest(residuals(sarima_model) ~ fitted(sarima_model)) # p-value = 0.3843, i.e., not enough evidence to reject H_0, thus suggesting that the variance is constant.
   H_0 – Homoscedasticity (variance is constant)
   H_1 – heteroscedasticity (variance is not constant)
3. Ljung-Box for **autocorrelation**: Box.test(residuals(sarima_model), lag = 12, type = "Ljung-Box") # see Task 4 for the hypothesises; p-value = 0.7837 => not enough evidence to reject H_0, thus, suggesting no significant autocorrelation.

To conclude, the tests above showcase that the residuals behave as white noise implying that the constructed SARIMA model is the good fit for the data.

References:

Adhikari, R. & Agrawal, R.K. (2013), "Deep learning for time series modelling". Available at: https://arxiv.org/pdf/1302.6613 (Accessed: 1 December 2024).

Frain, J. (1992), *Lecture Notes on Univariate Time Series Analysis and Box Jenkins Forecasting*. Available at: https://www.tcd.ie/Economics/staff/frainj/main/MSc%20Material/Session_notes/UNIVAR4.pdf (Accessed: 5 December 2024).

OpenAI (2023) *ChatGPT*[1] *response to user queries*, 27 November - 8 December. Available at: https://chat.openai.com/ (Accessed: 27 November 2024).

[1] The code was partially written using ChatGPT.