UNIVERSITY OF LEEDS

School of Mathematics

Declaration of Academic Integrity for Individual Pieces of Work

I declare that I am aware that as a member of the University community at the University of Leeds I have committed to working with Academic Integrity and that this means that my work must be a true expression of my own understanding and ideas, giving credit to others where their work contributes to mine.

I declare that the attached submission is my own work.

Where the work of others has contributed to my work, I have given full acknowledgement using the appropriate referencing conventions for my programme of study.

I confirm that the attached submission has not been submitted for marks or credits in a different module or for a different qualification or completed prior to entry to the University.

I have read and understood the University's rules on Academic Misconduct. I know that if I commit an academic misconduct offence there can be serious disciplinary consequences.

I re-confirm my consent to the University copying and distributing any or all of my work in any form and using third parties to verify that this is my own work, and for quality assurance purposes.

I confirm that I have declared all mitigating circumstances that may be relevant to the assessment of this piece of work and I wish to have taken into account.

Student Signature:	Manvendra Kumar Mishra	**Student Number:**	201911048
Student Name:	Manvendra Kumar Mishra	**Date:**	22-Nov-2024

Please note:

When you become a registered student of the University at first and any subsequent registration you sign the following authorisation and declaration:

"I confirm that the information I have given on this form is correct. I agree to observe the provisions of the University's Charter, Statutes, Ordinances, Regulations and Codes of Practice for the time being in force. I know that it is my responsibility to be aware of their contents and that I can read them on the University web site. I acknowledge my obligation under the Payment of Fees Section in the Handbook to pay all charges to the University on demand.

I agree to the University processing my personal data (including sensitive data) in accordance with its Code of

Pond Level Data Analysis (MATH3802)

Step 1: Load and Plot the Data

R Code: -

```
# Load necessary libraries
library(ggplot2)
library(forecast)

# Load the data
read.csv("pond_level.csv")

# Plot the data
plot(X, main="Pond Level Data", ylab="Level", xlab="Time")
```

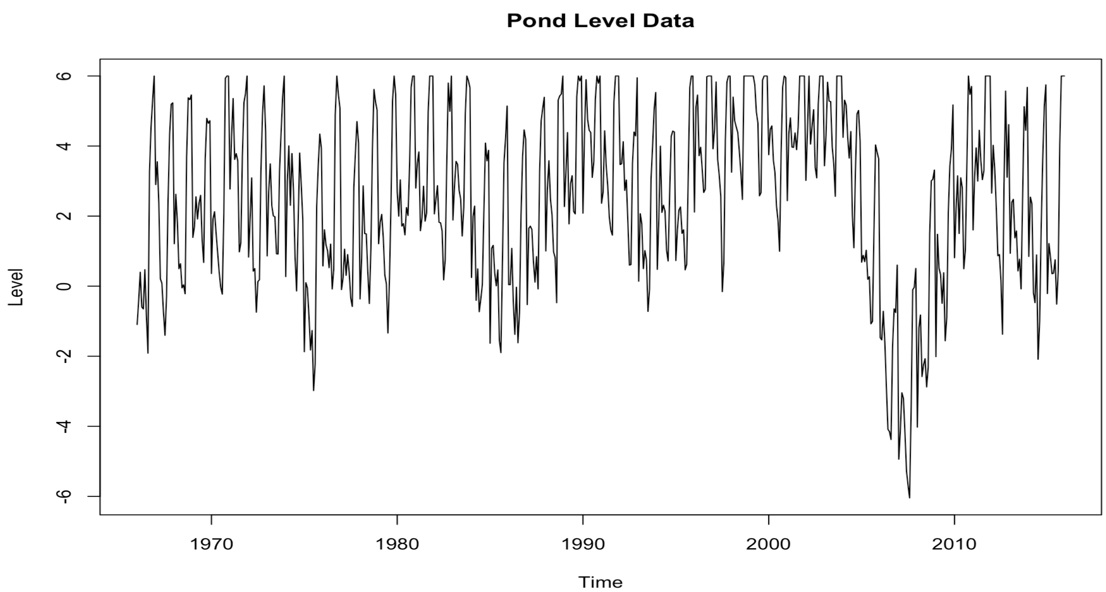

Fig1. Graph Showing Pond Level Data Across Decades

Key Observations: -

Overall Trend: -

There is no clear long-term upward or downward trend in data. The levels appear to fluctuate over the time, but the data oscillates around a mean without showing a distinct increase or decrease over the decades.

Seasonality: -

- While there are repetitive fluctuations in the data, its not immediately obvious if these are seasonal in nature without additional decomposition or statistical tests.
- To confirm seasonality, we would need to perform a seasonal decomposition or look at the autocorrelation functions to identify periodic patterns.

Variability: -

- The amplitude of fluctuations seems to vary across different periods. For instance:
    - The period between 1966 and 1975 appears to have high variability
    - Later periods, such as the 2000s, appear to have reduced variability

Anomalies: -

- There are noticeable peaks & troughs in some years, which could be outliers or significant events affecting the pond levels.

Interpretation: -

The data fluctuates around a mean without a clear long-term trend, though variability changes over time, with higher fluctuations between 1966-1975 and reduced variability in later periods like the 2000s. While repetitive patterns suggest possible seasonality, and the data may not be strictly stationary due to changing fluctuation levels.

Step 2: Remove Linear Trend and Seasonality

R Code: -

```
# Fit a linear model to remove trend
time <- 1:length(X)
linear_model <- lm(X ~ time)
res <- residuals(linear_model)
# Plot detrended data
plot(res, main="Detrended Pond Level Data", ylab="Detrended
Level", xlab="Tim
```

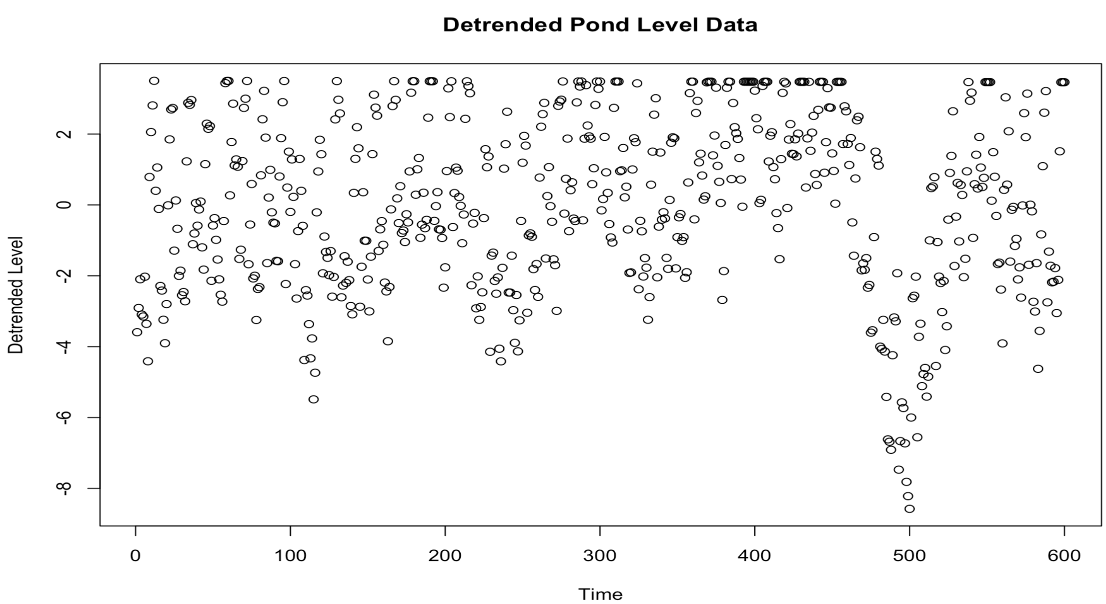

Fig2. Graph Showing Residuals After Removing Linear Trend/Seasonality

Step 3: Check for AR or MA Structure in Residuals (Y)

The next step is to inspect the cleaned (potentially de-seasonalized) data, Y to determine if an Autoregressive (AR) or Moving Average (MA) model might be appropriate for this data. For that, we can check the autocorrelation function (ACF).

R Code: -

```
# Plot ACF and PACF of residuals
acf(res, main="ACF of Residuals (Y)")
```

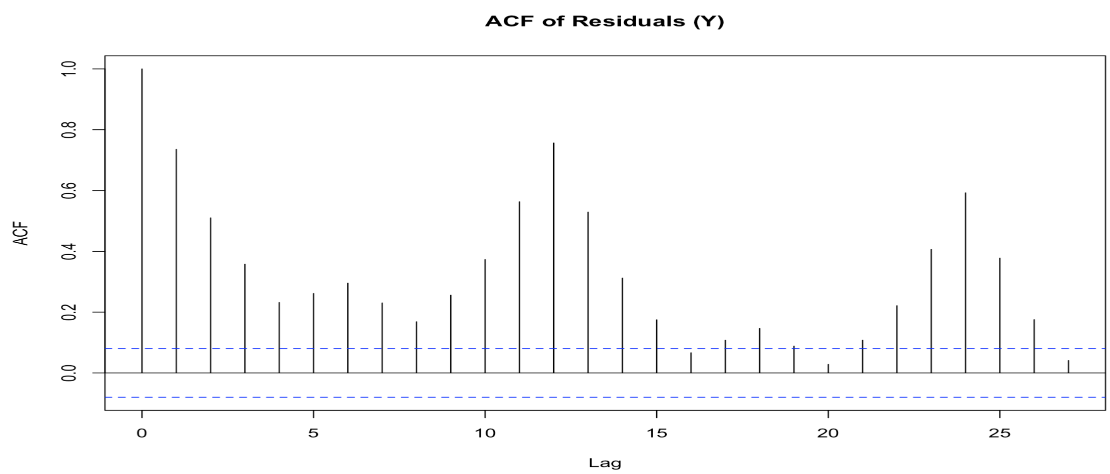

Fig3. Graph Showing ACF Of Residuals

```
pacf(res, main="PACF of Residuals (Y)")
```

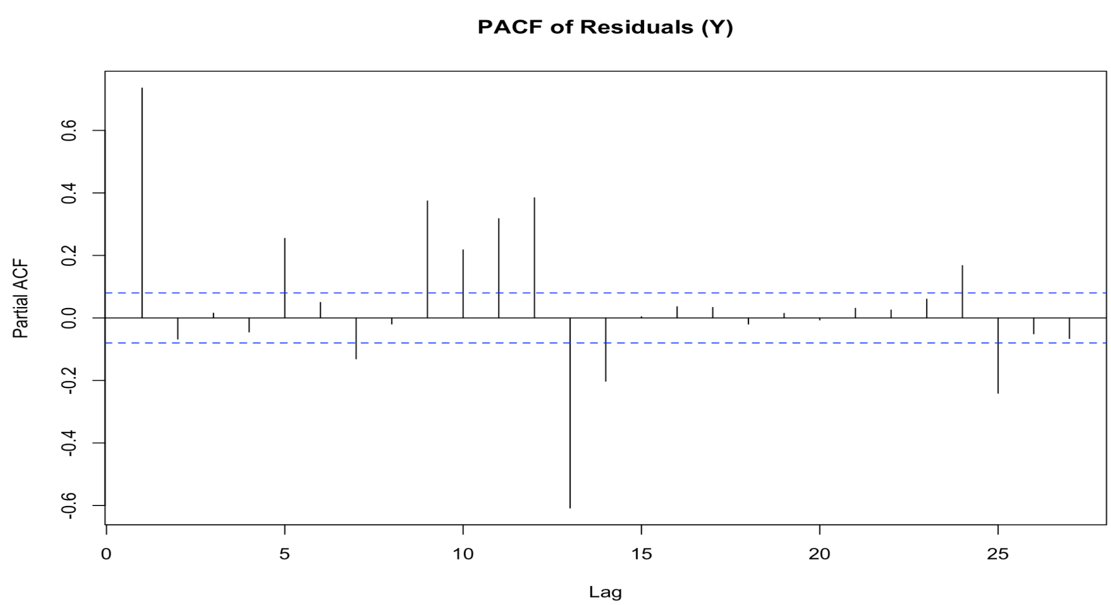

Fig3. Graph Showing PACF Of Residuals

- **Autoregressive model:** If the ACF tapers off gradually while the PACF cuts off after a few lags, it means we should go ahead with an AR model.
- **Moving Average model:** If the ACF cuts off after a few lags while the PACF tapers off, the MA model might fit better than AR model.

Here, after looking at the graphs for PACF and ACF-

- The PACF shows few significant spikes at early lags suggesting that AR model can be a better fit.
- The ACF graph does not have any clear cutoff, so MA model might not be a good fit.

Step 4: Fit AR Models of Orders p = 1, 2, 3 & Evaluate Model Fit Using Residuals

R Code: -

```
# Fit AR models of different orders
ar1 <- ar(res, order.max=1, method="yule-walker")
ar2 <- ar(res, order.max=2, method="yule-walker")
ar3 <- ar(res, order.max=3, method="yule-walker")
```

After fitting all three AR models, we can assess the residuals to check if the residuals resemble white noise (or are uncorrelated). In an ideal scenario, the AR model should have no significant lags if the model must fit well.

R Code: -

```
# Residual analysis
par(mfrow=c(1,3))
acf(residuals(ar1), main="AR(1) Residuals")
acf(residuals(ar2), main="AR(2) Residuals")
acf(residuals(ar3), main="AR(3) Residuals")
par(mfrow=c(1,1))
```

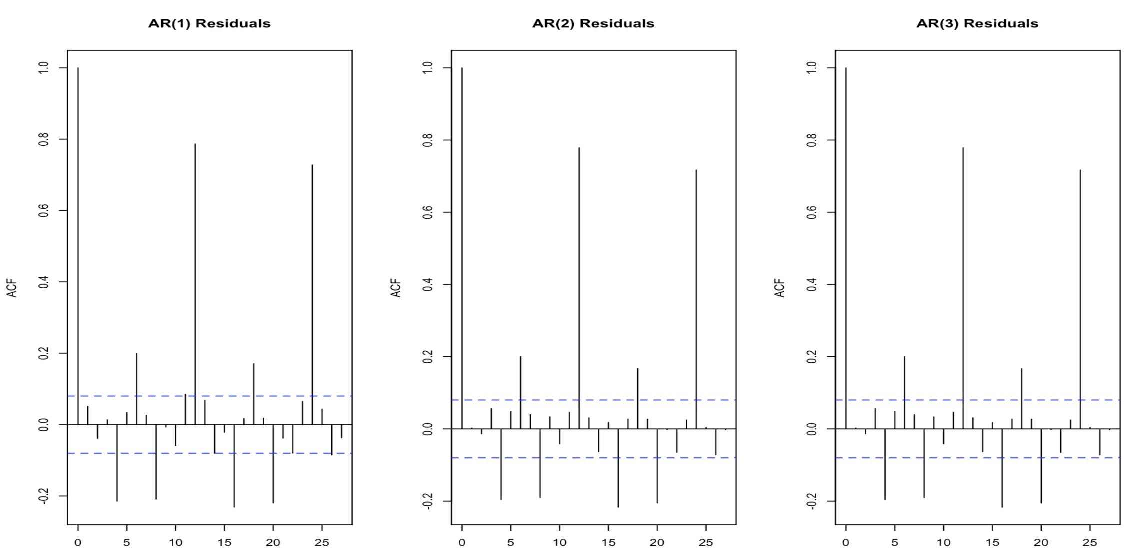

Fig4. Graphs Showing AR models of order 1, 2 & 3

Based on above ACF plots for AR(1), AR(2) and AR(3) models-

- **AR(1): -**There is significant autocorrelation at some lags, suggesting that this model does not cover complete time series dynamics.
- **AR(2): -** There is a reduction in autocorrelation in this model as compared to AR(1) at most lags. Though, there are some spikes suggesting improvement over AR(1) but still with some remaining structure.
- **AR(3): -** The residual for this model suggests the least autocorrelation, though the graph looks somewhat similar to AR(2) model, with most bands falling under the confidence band, suggesting the best fit amongst all the three AR models as this captures the structure more effectively

To summarize, the AR(3) model is likely the best fit amongst all the other orders of AR models as it minimizes the autocorrelation in the residuals which indicates a better capture of the structure.

Step 6: Peridograms for X, Y and Z

<u>R Code:</u> -

```
# Periodograms
par(mfrow=c(1,3))
spectrum(X, main="Periodogram of X")
spectrum(res, main="Periodogram of Y")
spectrum(na.omit(residuals(ar3)), main="Periodogram of Z")

# assuming AR(3) is best
par(mfrow=c(1,1))
```

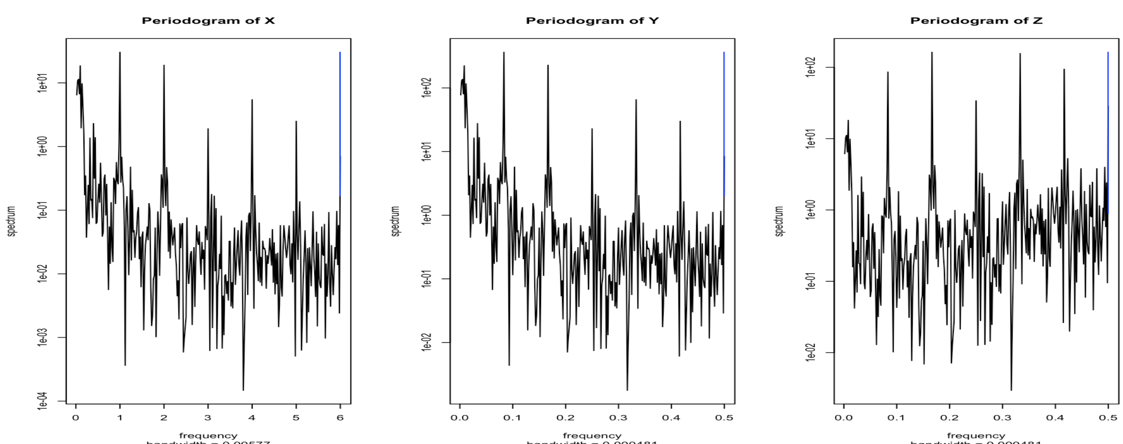

Fig5. Graphs Showing Periodogram for X, Y & Z variables

The periodogram for X (original time series), Y(residual from the AR model), and Z(residual from the AR(3) model) includes the frequency for each graph.

- The periodogram for X shows raw frequency distribution, while Y shows the unexplained variance after fitting the model.

- The periodogram for Z reveals significantly reduced frequency distribution, suggesting a better-fit model, with less periodic factor remaining in the residual.

Step 7: Re-fit the Model Using ARIMA

Using ARIMA function to re-fit the AR model to the original data, allows for a refined model with optimal parameters.
Below is the R code with the output for the same-

R Code: -

```
# Assuming AR(3) was selected
final_model <- arima(time_series, order=c(3,0,0))
print(final_model)
```

Result:-

```
Coefficients:
         ar1      ar2     ar3  intercept
      0.7896  -0.0799  0.0169     2.5169
s.e.  0.0408   0.0519  0.0409     0.2478

sigma^2 estimated as 2.777:  log likelihood = -1158.23,  aic =
2326.45
```

R Code: -

```
# adjust the order as per the chosen model
summary(final_model)
Training set error measures:
```

	MASE	ACF1	ME	RMSE	MAE	MPE	MAPE
Training set	0.005875307	1.666575	1.303106	-17.31281	187.5872	1.02424	-0.000427908

Mathematical Expression for AR(3) model: -

As the model is an AR(3) model having order 3. The mathematical expression can be written in form of.

```
X_t = φ1 X_{t-1} + φ2 X_{t-2} + φ3 X_{t-3} + ε_t
```

where:
- φ1, φ2, φ3 are the AR coefficients,
- ε_t is the white noise error term.