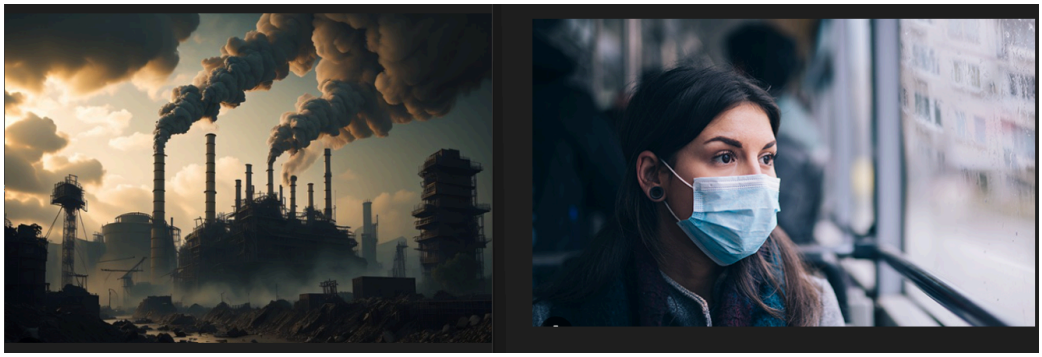
Air Quality Forecasting

Introduction

- Essential for evaluating ecosystem health, as poor air quality affects wildlife, forests, and water bodies, impacting biodiversity.
- Identifying pollutants causing respiratory and cardiovascular diseases helps in improving public health.



Project Objective:

- To develop a forecasting model that accurately forecasts the Air Quality Index (AQI) in Buffalo City, utilizing historical air quality data with a focus on key pollutants, including particulate matter and carbon monoxide concentrations.

Loading the necessary libraries:

```
library(readr)        # For reading the CSV file
library(tsibble)      # For handling time series data
```

```
Attaching package: 'tsibble'

The following objects are masked from 'package:base':

    intersect, setdiff, union
```

```
library(dplyr)        # For data manipulation
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':
```

```
    intersect, setdiff, setequal, union
```

```
library(ggplot2)      # For data visualization
library(forecast)     # For ARIMA and forecasting functions
```

```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

```
library(tseries)      # For the ADF test
library(lubridate)
```

```
Attaching package: 'lubridate'

The following object is masked from 'package:tsibble':

    interval

The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```
library(vars)
```

```
Loading required package: MASS


Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select

Loading required package: strucchange

Loading required package: zoo


Attaching package: 'zoo'

The following object is masked from 'package:tsibble':

    index

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: sandwich

Loading required package: urca
```

```
Loading required package: lmtest
```

```
        library(corrplot)
```

```
corrplot 0.92 loaded
```

Dataset Overview:

- The dataset is downloaded from the United States Environmental Protection Agency's official website - https://www.epa.gov/outdoor-air-quality-data/download-daily-data.
- Data is available for years from 1980 to 2023, and a wide range of pollutants to choose from.

```
        air_data <- read.csv("aq_dataset.csv")
        summary(air_data)
```

```
    Date              Source             Site.ID               POC        
Length:364         Length:364         Min.   :360610135   Min.   :1  
Class :character   Class :character   1st Qu.:360610135   1st Qu.:1  
Mode  :character   Mode  :character   Median :360610135   Median :1  
                                      Mean   :360610135   Mean   :1  
                                      3rd Qu.:360610135   3rd Qu.:1  
                                      Max.   :360610135   Max.   :1  
 Coconcentration    UNITS           DAILY_AQI_VALUE    Site.Name        
 Min.   :0.100   Length:364         Min.   : 1.000   Length:364        
 1st Qu.:0.300   Class :character   1st Qu.: 3.000   Class :character  
 Median :0.300   Mode  :character   Median : 3.000   Mode  :character  
 Mean   :0.381                      Mean   : 4.288                     
 3rd Qu.:0.400                      3rd Qu.: 5.000                     
 Max.   :1.200                      Max.   :14.000                     
 DAILY_OBS_COUNT PERCENT_COMPLETE AQS_PARAMETER_CODE AQS_PARAMETER_DESC
 Min.   :13.00   Min.   : 54.00   Min.   :42101      Length:364        
 1st Qu.:24.00   1st Qu.:100.00   1st Qu.:42101      Class :character  
 Median :24.00   Median :100.00   Median :42101      Mode  :character  
 Mean   :23.23   Mean   : 96.82   Mean   :42101                        
 3rd Qu.:24.00   3rd Qu.:100.00   3rd Qu.:42101                        
 Max.   :24.00   Max.   :100.00   Max.   :42101                        
   CBSA_CODE      CBSA_NAME           STATE_CODE    STATE          
 Min.   :35620   Length:364         Min.   :36   Length:364        
 1st Qu.:35620   Class :character   1st Qu.:36   Class :character  
 Median :35620   Mode  :character   Median :36   Mode  :character  
 Mean   :35620                      Mean   :36                     
 3rd Qu.:35620                      3rd Qu.:36                     
 Max.   :35620                      Max.   :36                     
  COUNTY_CODE    COUNTY           SITE_LATITUDE   SITE_LONGITUDE   
 Min.   :61   Length:364         Min.   :40.82   Min.   :-73.95  
 1st Qu.:61   Class :character   1st Qu.:40.82   1st Qu.:-73.95  
 Median :61   Mode  :character   Median :40.82   Median :-73.95  
 Mean   :61                      Mean   :40.82   Mean   :-73.95  
 3rd Qu.:61                      3rd Qu.:40.82   3rd Qu.:-73.95  
 Max.   :61                      Max.   :40.82   Max.   :-73.95  
```

Data Pre-processing:

- Eliminated rows with missing values across all variables, ensuring a complete dataset for analysis.
- Checked for remaining missing values in key variables.
- Ensured all critical fields were fully populated for accurate modeling.

```
air_data$Date <- as.Date(air_data$Date, format = "%Y-%m-%d")

air_data <- na.omit(air_data)

# Check for NA values in the key variables
sum(is.na(air_data$`Coconcentration`))
```

```
[1] 0
```

```
sum(is.na(air_data$Date))
```

```
[1] 0
```

```
sum(is.na(air_data$DAILY_AQI_VALUE))
```

```
[1] 0
```

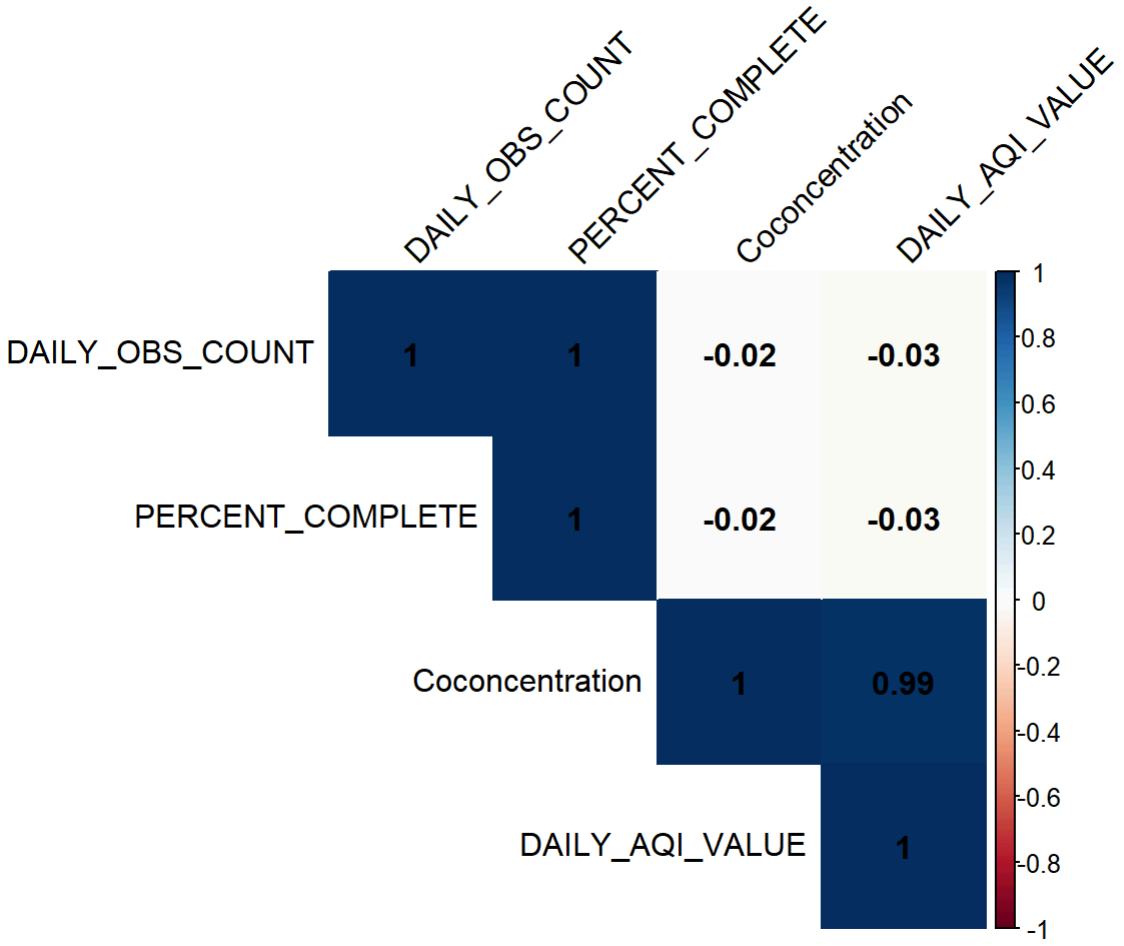
Empirical Data Analysis:

All the below data analysis were performed:

- Correlation HeatMap.
- Time Plots of Target variable and the covariate.
- Outliers Detection (BoxPlots).
- Scatter Plot (CO concentration Vs Air Quality Index).
- ACF and PACF plots.
- Seasonal and Trend Decomposition.

```
air_data <- air_data[, sapply(air_data, function(x) length(unique(na.omit(x))) > 1)]

# Convert factors to numeric if necessary
numeric_columns <- sapply(air_data, is.numeric)
air_data_numeric <- air_data[, numeric_columns]

# Compute the correlation matrix
correlation_matrix <- cor(air_data_numeric, use = "pairwise.complete.obs") # Handle m

# Plot the heatmap
```

```
corrplot(correlation_matrix, method = "color", type = "upper", order = "hclust",
         addCoef.col = "black", # Add correlation coefficients on the heatmap
         tl.col = "black", tl.srt = 45) # Text label color and rotation
```

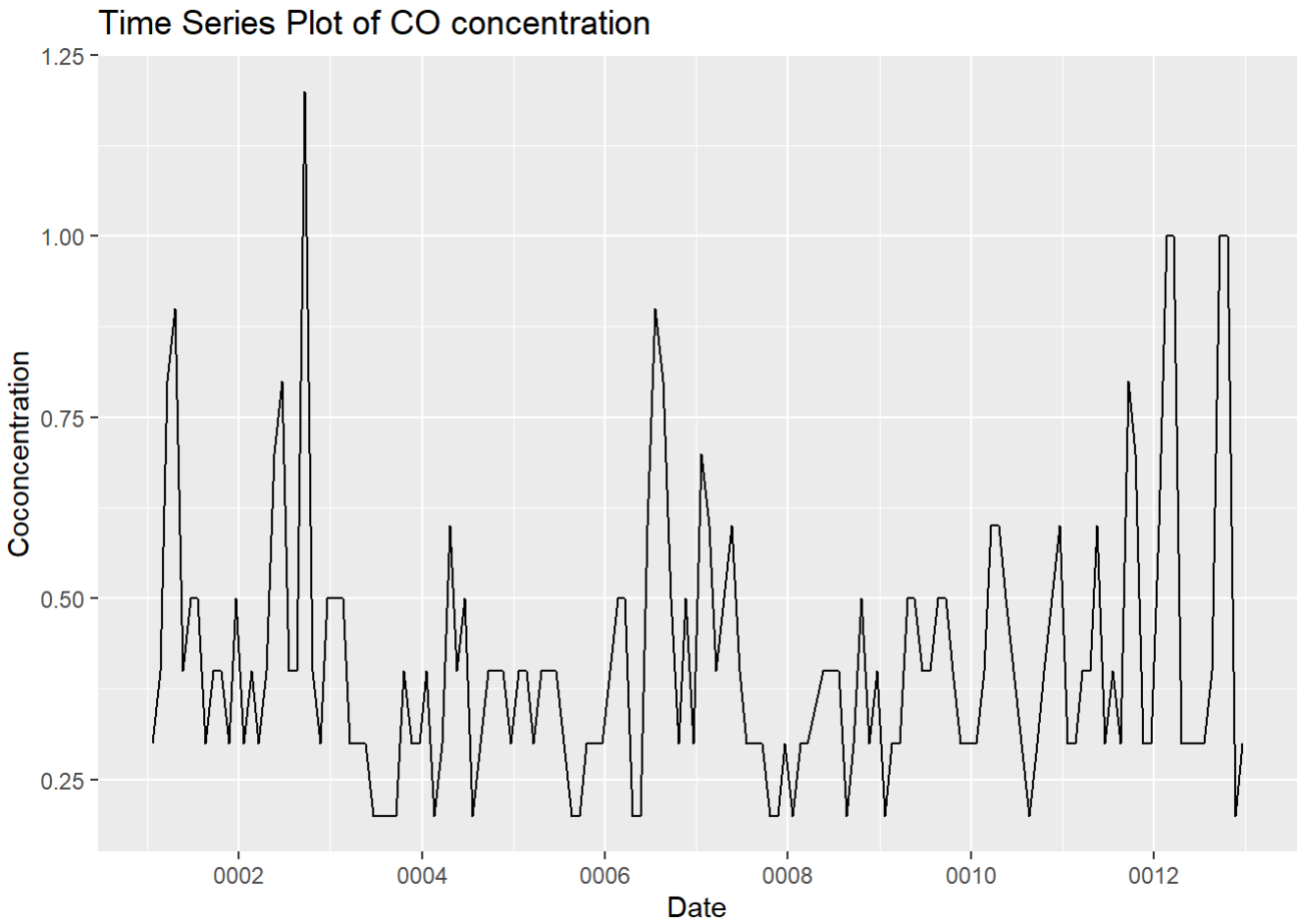


Correlation HeatMap.

HeatMap Observations:

- Correlation Values: Darker blue circles represent stronger positive correlations, where one variable increases as the other does. No circles or smaller circles indicate a weaker or no correlation.
- Key Relationships: The variable “CO concentration” seems to have a strong positive correlation with “DAILY_AQI_VALUE,” which is consistent with our analysis of the scatter plot.
- Data Integrity: Variables like “PERCENT_COMPLETE” might relate to the completeness of data records, although the specific nature of the correlation with other variables isn’t clear from this graph.

```
ggplot(air_data, aes(x = Date, y = Coconcentration)) +
    geom_line() +
    labs(title = "Time Series Plot of CO concentration")
```

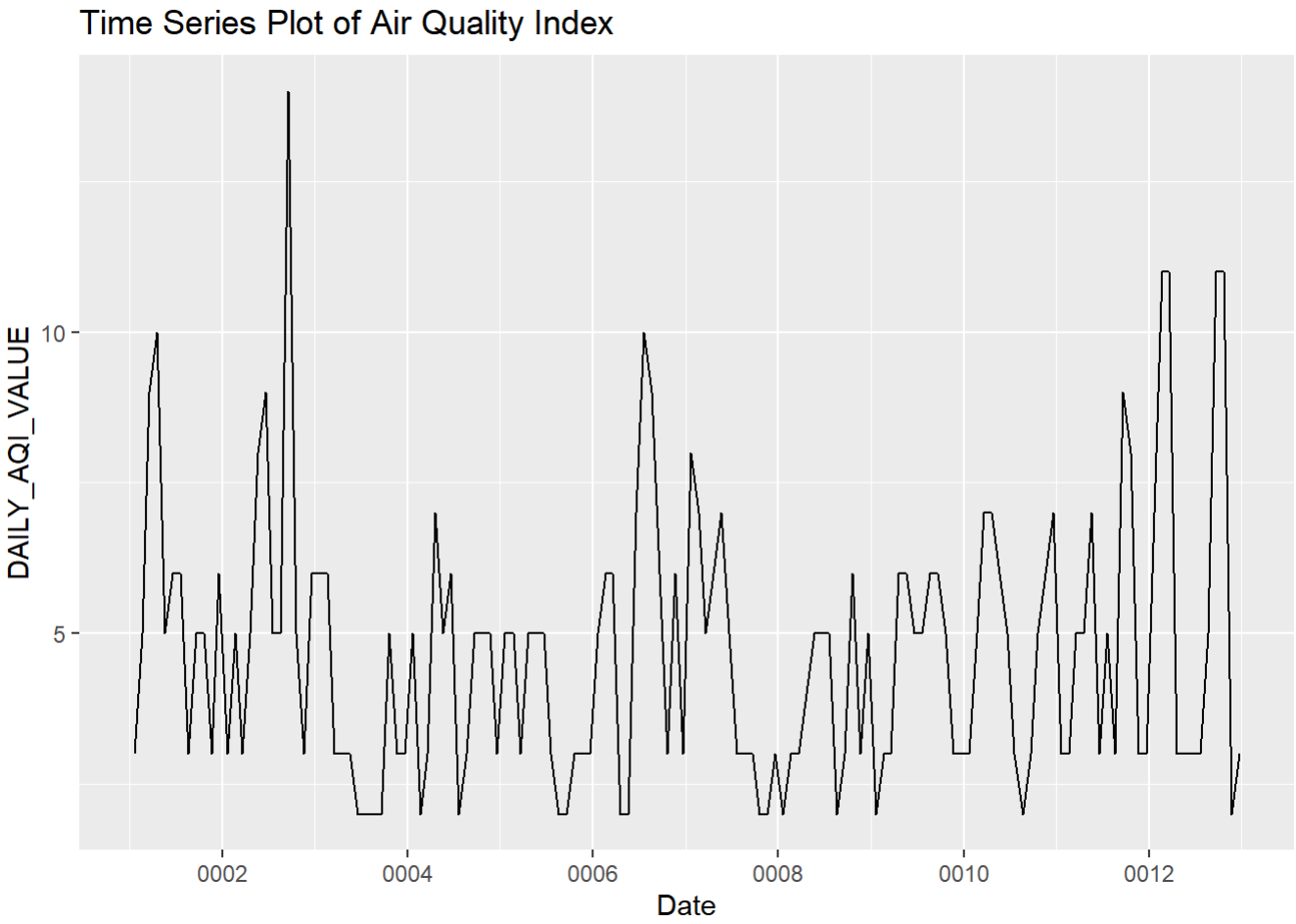


Time Plot of Carbon Monoxide Concentration.

Time Plots Observations:

- Seasonal Variation: The plot suggests possible seasonal variation in CO levels, with peaks that might correspond to particular times of the year when CO emissions increase due to factors such as heating during colder months.
- Daily Fluctuations: There is considerable day-to-day volatility in CO concentration levels, with several extreme spikes indicating episodic high pollution events, possibly linked to specific environmental or human activities.
- Baseline Levels: Aside from the prominent spikes, there is a somewhat consistent baseline CO concentration level, with values often returning to a range between approximately 0.25 to 0.5.
- This suggests a regular emission source that is constant over time, punctuated by irregular high emission events.

```
ggplot(air_data, aes(x = Date, y = DAILY_AQI_VALUE)) +
    geom_line() +
    labs(title = "Time Series Plot of Air Quality Index")
```

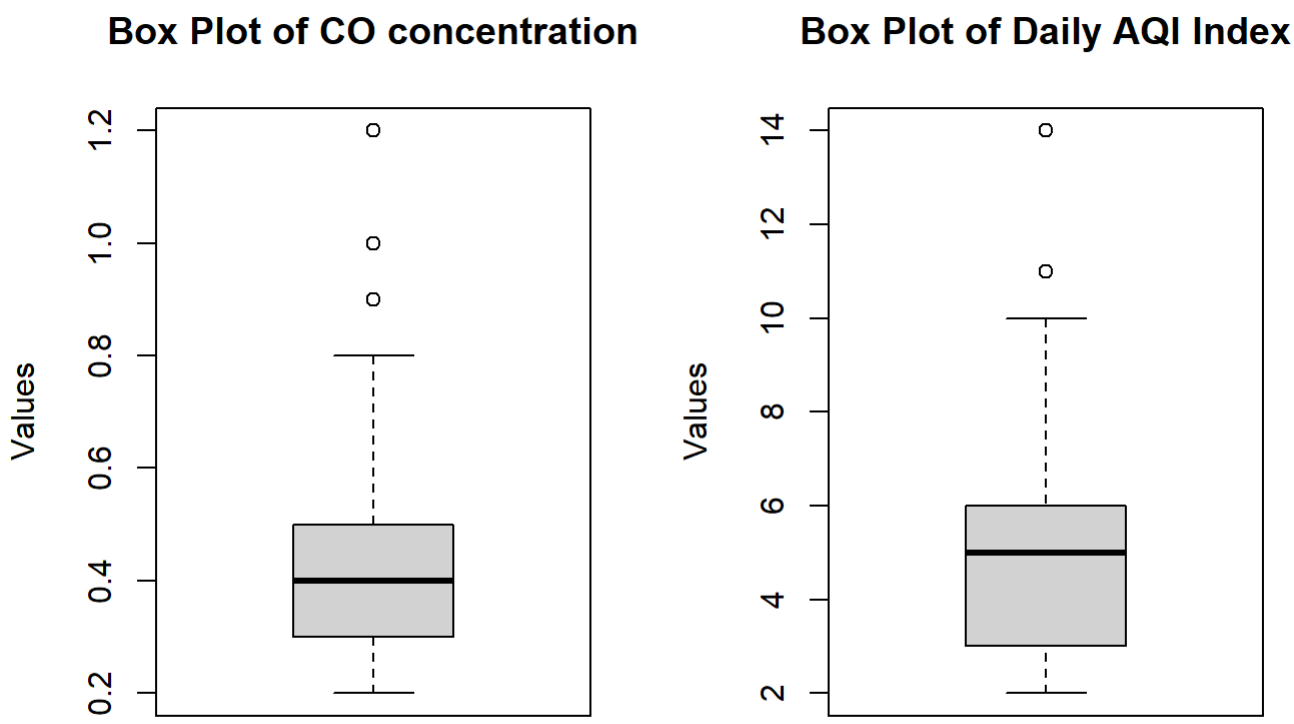


Time Plot of Air Quality Index.

Time Plots Observations:

- Volatile Indicator: The Air Quality Indicator is highly volatile, with many sharp peaks and troughs.
- It suggests frequent changes in air quality, with certain days experiencing significantly poorer air quality.
- Extreme Values: There are several notable extreme values, particularly high peaks, which could indicate days with very poor air quality.
- These may correspond to specific events or conditions that dramatically worsen air quality.
- Baseline Fluctuation: The baseline level of the Air Quality Indicator seems to vary between 5 to around 15 for the most part.
- However, this is interrupted by the spikes, suggesting that while the air quality does have a general range it fluctuates within, there are numerous instances where it deviates significantly from this range.

```
par(mfrow = c(1, 2))
boxplot(air_data$Coconcentration, main="Box Plot of CO concentration", ylab="Values")
boxplot(air_data$DAILY_AQI_VALUE, main="Box Plot of Daily AQI Index", ylab="Values")
```



BoxPlots of CO Concentration and AQI Index.

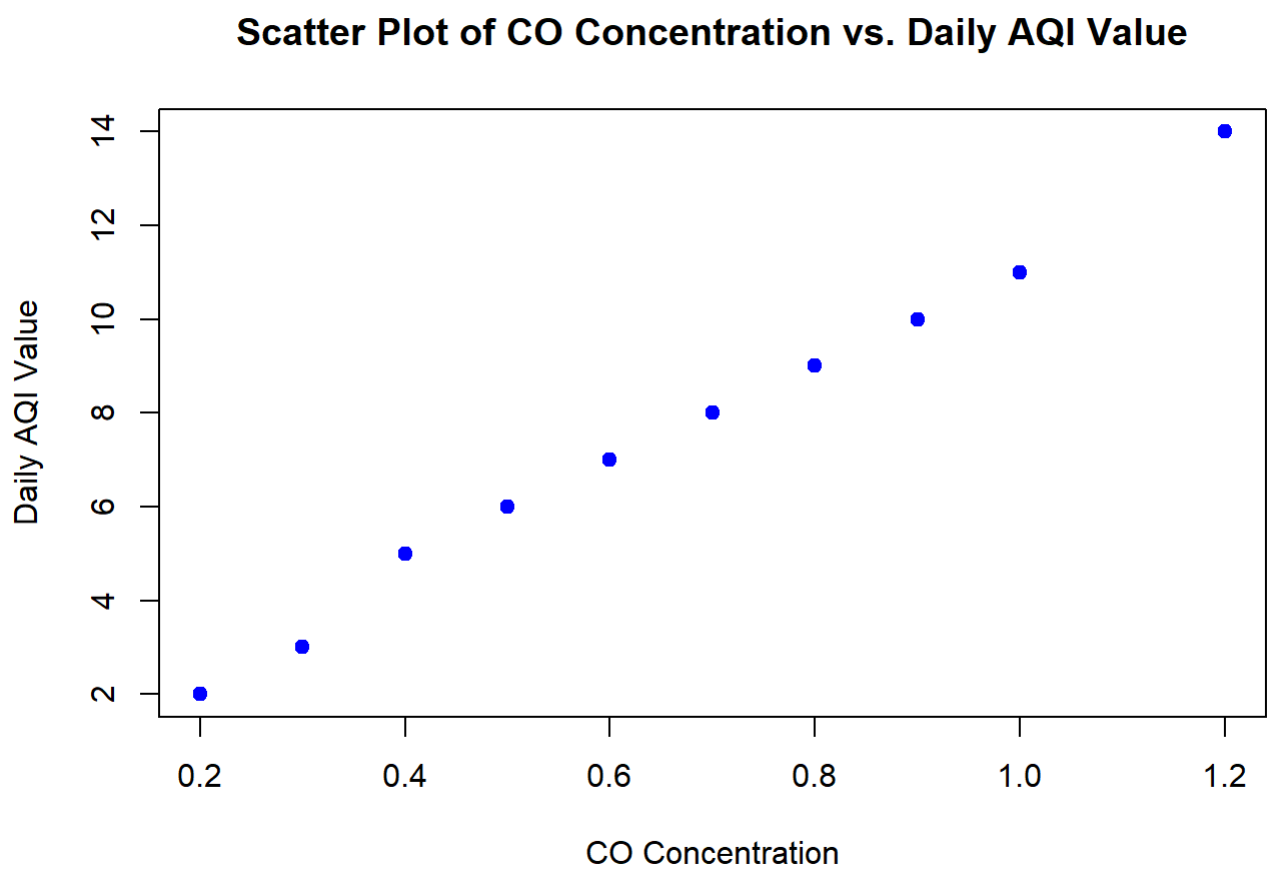
```
par(mfrow = c(1, 1))
```

BoxPlots Observations:

- Median: The line within the box indicates the median CO concentration, which appears to be just above 0.3 and Daily AQI is around 5.
- Interquartile Range (IQR): The box represents the middle 50% of the data, known as the interquartile range. It's relatively narrow, suggesting that the middle 50% of readings are not spread out over a wide range of values.
- Outliers: Outliers in both CO concentration and AQI values suggest sporadic days with unusually high pollution levels, indicating episodes of significantly poor air quality.
- The box plots indicate that while daily AQI values and CO concentrations generally hover around a stable median, there are exceptional days with high AQI and CO readings, raising concerns for air quality and public health.

```
plot(air_data$Coconcentration, air_data$DAILY_AQI_VALUE,
     main = "Scatter Plot of CO Concentration vs. Daily AQI Value",
```

```
        xlab = "CO Concentration",
        ylab = "Daily AQI Value",
        pch = 19, col = "blue")
```

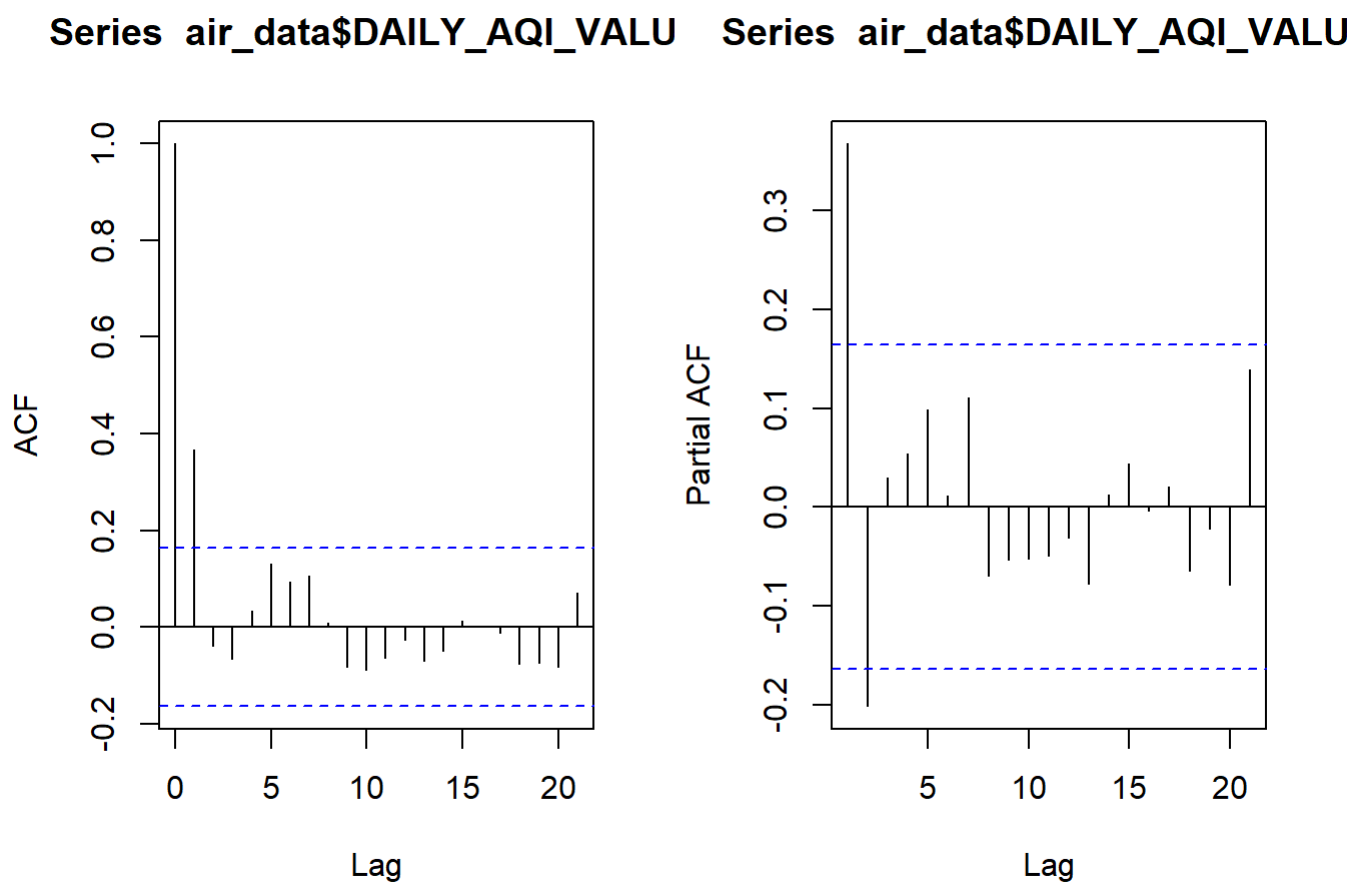


Scatter Plot of CO Concentration and AQI value.

ScatterPlot Observations:

- Positive Correlation: CO concentration levels and Daily AQI values show a direct, positive relationship.
- Increased Rate: The rate of increase in AQI values is greater at higher CO concentrations.
- Data Clustering: Most data points are concentrated at lower CO levels, with significant AQI variations at higher CO levels.

```
par(mfrow = c(1, 2))
acf(air_data$DAILY_AQI_VALUE)
pacf(air_data$DAILY_AQI_VALUE)
```



ACF and PACF Plots of Air Quality Index.

```
par(mfrow = c(1, 1))
```

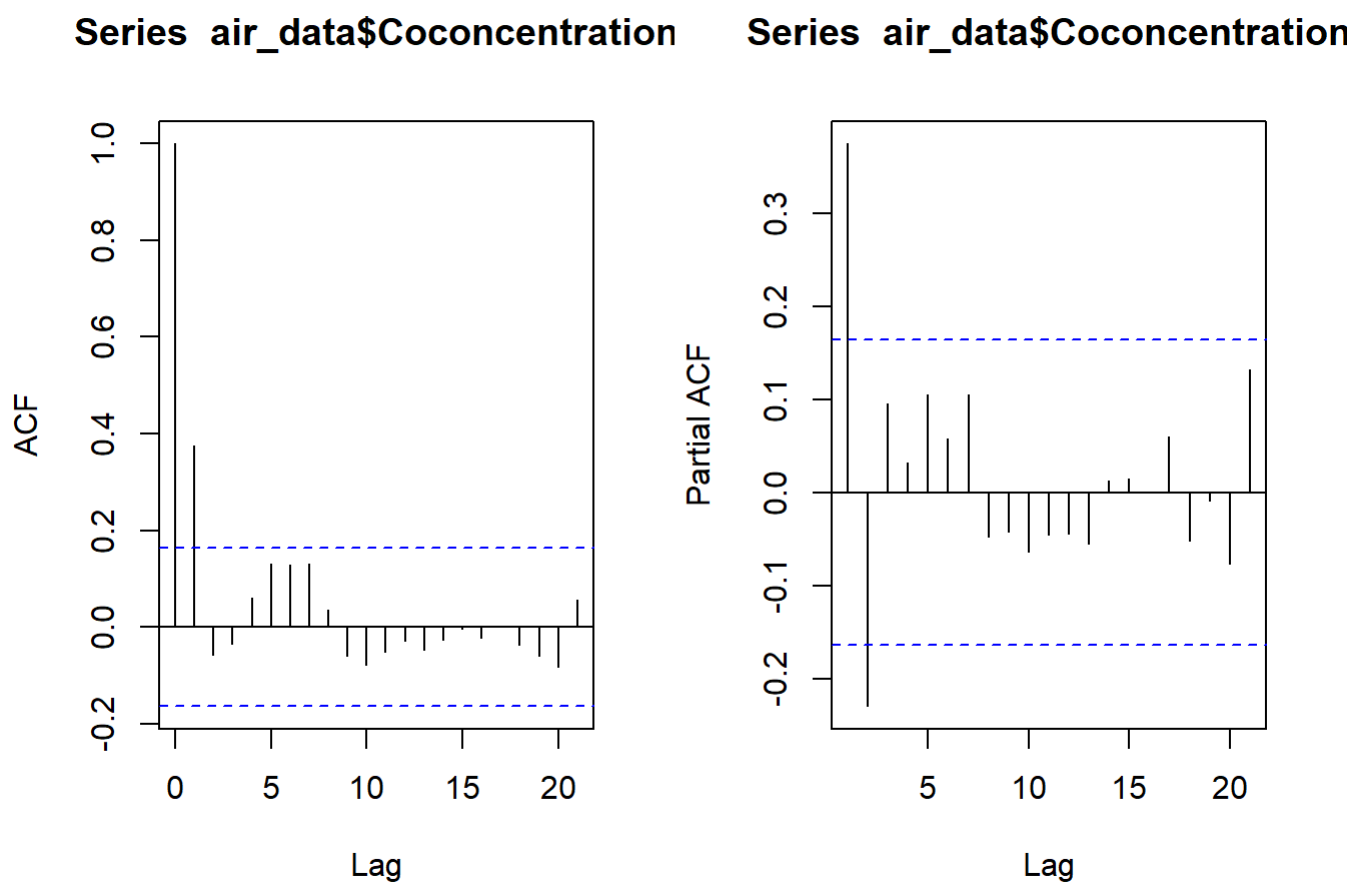
ACF Analysis:

- High autocorrelation at lag 0 with a sharp initial decline.
- Subsequent lags within confidence bounds, indicating no significant autocorrelation.
- Suggests AQI values do not depend on past values; forecasting models might be less effective.

PACF Analysis:

- Strong initial partial auto correlation at lag 0.
- No significant partial auto correlations beyond lag 0.
- Implies limited autoregressive behavior; AR models may not be useful beyond the first lag.

```
par(mfrow = c(1, 2))
acf(air_data$Coconcentration)
pacf(air_data$Coconcentration)
```



ACF and PACF Plots of CO Concentration.

```
par(mfrow = c(1, 1))
```

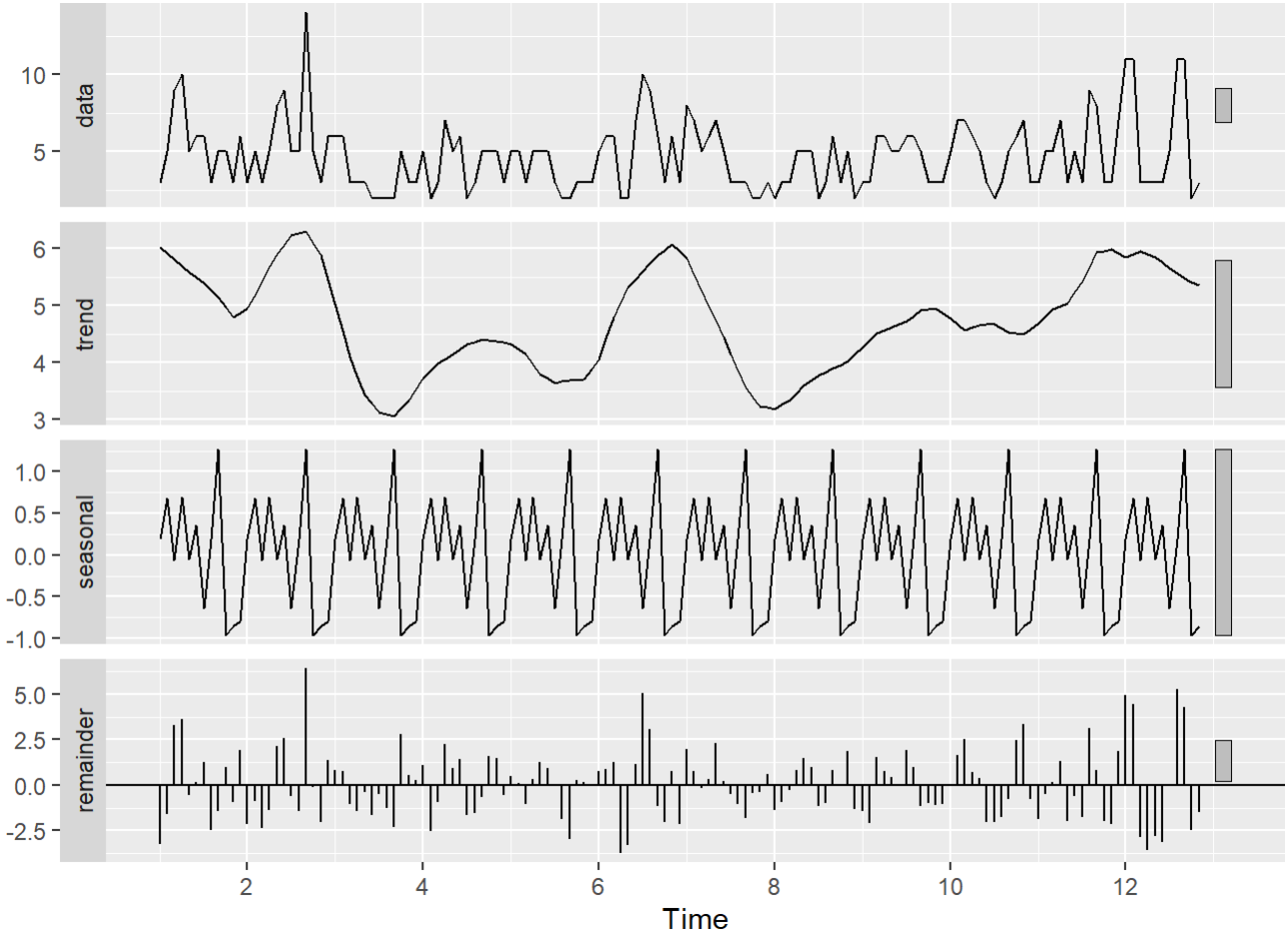
ACF Analysis:

- High initial autocorrelation at lag 0.
- Sharp decline after lag 0, indicating minimal autocorrelation.
- Values within confidence bounds suggest no significant autocorrelation.

PACF Analysis:

- Dominant initial partial autocorrelation at lag 0.
- Rapid decline in PACF values, with no significant lags beyond 0.
- Implies little autoregressive behavior for predictive modeling.

```
air_data_ts_var1 <- ts(air_data$DAILY_AQI_VALUE, frequency = 12)
stl_decomp <- stl(air_data_ts_var1, s.window = "periodic")
autoplot(stl_decomp)
```

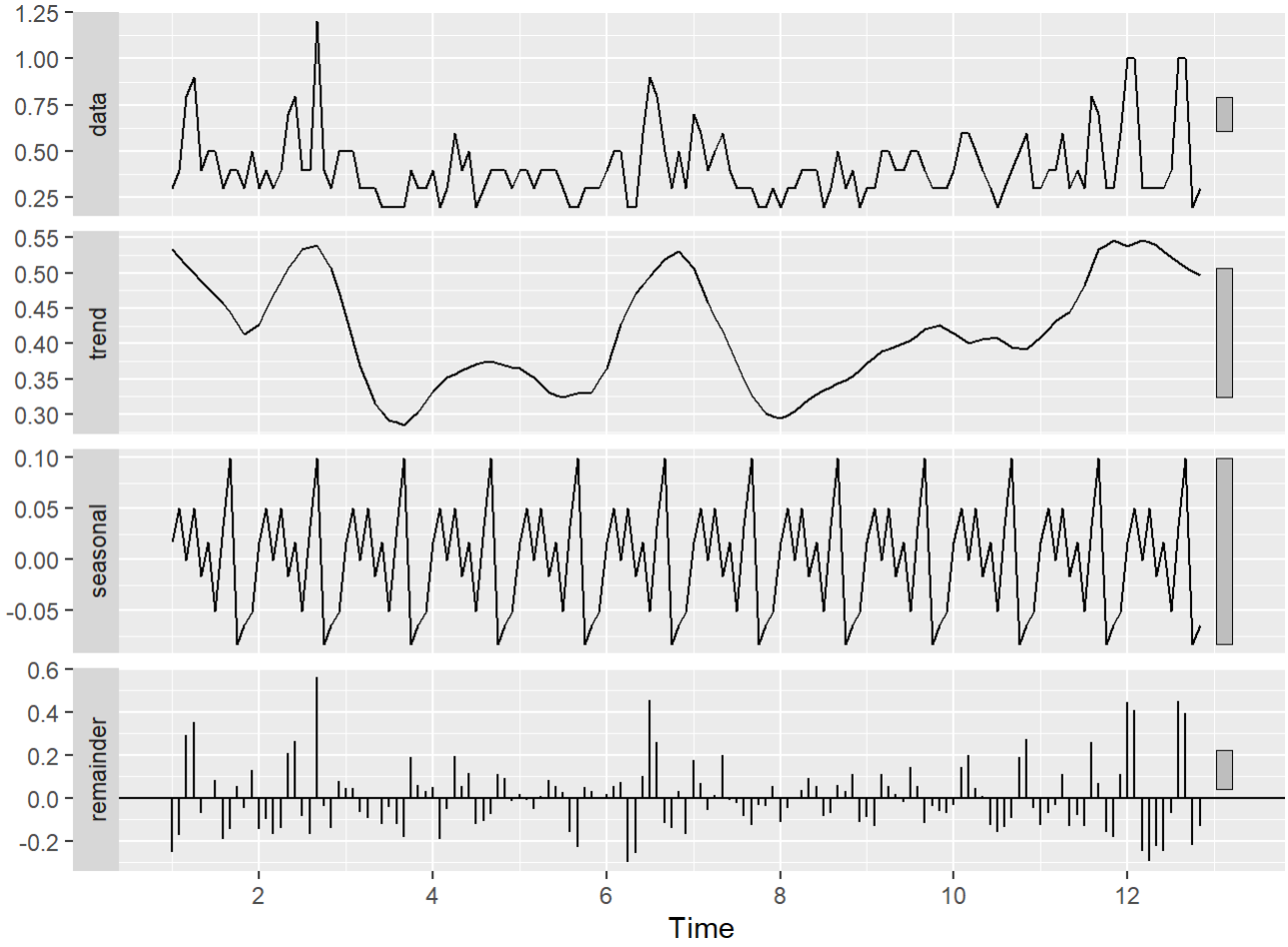


Seasonal and Trend Decomposition of AQI Value.

STL Decomposition of AQI Value:

- Data: Fluctuating Daily AQI Value time series.
- Trend: Shows a cyclical pattern with no clear directional trend.
- Seasonal: Regular, repeating patterns in the series.
- Remainder: Residual random variation after detrending and deseasonalizing
- The analysis suggests that the Daily AQI Value is influenced by a repeating seasonal pattern, while the trend component does not show a clear increase or decrease over the period represented on the x-axis. The remainder component indicates the presence of random noise and possibly outliers or unusual events affecting air quality.

```
air_data_ts_var2 <- ts(air_data$Coconcentration, frequency = 12)
stl_decomp <- stl(air_data_ts_var2, s.window = "periodic")
autoplot(stl_decomp)
```



Seasonal and Trend Decomposition of CO Concentration.

STL Decomposition of CO Concentration:

- Data: Shows observed CO concentration variability and spikes.
- Trend: Undulating pattern without clear long-term change, suggesting a stable trend.
- Seasonal: Subtle seasonal effects with small amplitude, indicating minor seasonal impact.
- Remainder: Volatile residuals highlight irregular fluctuations or external impacts on CO levels.
- The decomposition suggests that while there might be a slight seasonal pattern, the CO concentration data's volatility is captured more in the irregular remainder component than in the seasonal component. The trend doesn't show a strong long-term directional movement over the period analyzed.

Augmented Dickey-Fuller Test

```
adf_result_var1 <- adf.test(air_data_ts_var1)
print(adf_result_var1)
```

```
    Augmented Dickey-Fuller Test

data:  air_data_ts_var1
Dickey-Fuller = -4.0677, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary
```

```
adf_result_var2 <- adf.test(air_data_ts_var2)
print(adf_result_var2)
```

```
    Augmented Dickey-Fuller Test

data:  air_data_ts_var2
Dickey-Fuller = -3.8155, Lag order = 5, p-value = 0.02028
alternative hypothesis: stationary
```

Modelling

The below are the 2 forecasting models used for the Air Quality Index Forecasting:

- RegARIMA Model:
    - Combines linear regression with ARIMA: Integrates linear regression with ARIMA, incorporating external regressors into time series forecasting.
    - Suitable for forecasting with external predictors: Ideal for forecasting a target time series influenced by external variables, enhancing forecast accuracy by capturing additional information.
- VAR model:
    - Models interdependencies among time series: Captures linear interdependencies among multiple time series variables, enabling analysis of complex relationships.
    - Facilitates impulse response analysis: Allows examination of how one variable affects another over time, providing insights into dynamic interactions between variables for decision-making.

RegARIMA Model

The RegARIMA model combines regression analysis with ARIMA to analyze time series data with external regressors.

Notation:

- y_t: Target time series variable at time t.
- x_t: External regressor (covariate) at time t.
- p, d, q: ARIMA model parameters, where p is the autoregressive order, d is the degree of differencing, and q is the moving average order.

Mathematical Background:

The model is given by:

$$y'_t = \beta x_t + ARIMA(p, d, q)$$

Where y'_t is the differenced series if $d > 0$, β represents the coefficients of the external regressor x_t, and the ARIMA component captures the autocorrelation within the residuals.

Equation:

air_data_ts_var1 = -0.2511 + 0.1756 * L12(air_data_ts_var1) + 11.9372 * air_data_ts_var2

Annotations:

air_data_ts_var1: Dependent variable representing the time series data.

air_data_ts_var2: Exogenous variable.

-0.2511: Intercept term.

0.1756: Coefficient for the lagged value of the time series (seasonal lag of 12).

11.9372: Coefficient for the exogenous variable.

This equation indicates that the predicted value at time t is a linear combination of an intercept term (-0.2511), the value of the time series lagged by 12 months (seasonal AR term with coefficient 0.1756), and the value of the exogenous variable at time t (with coefficient 11.9372).

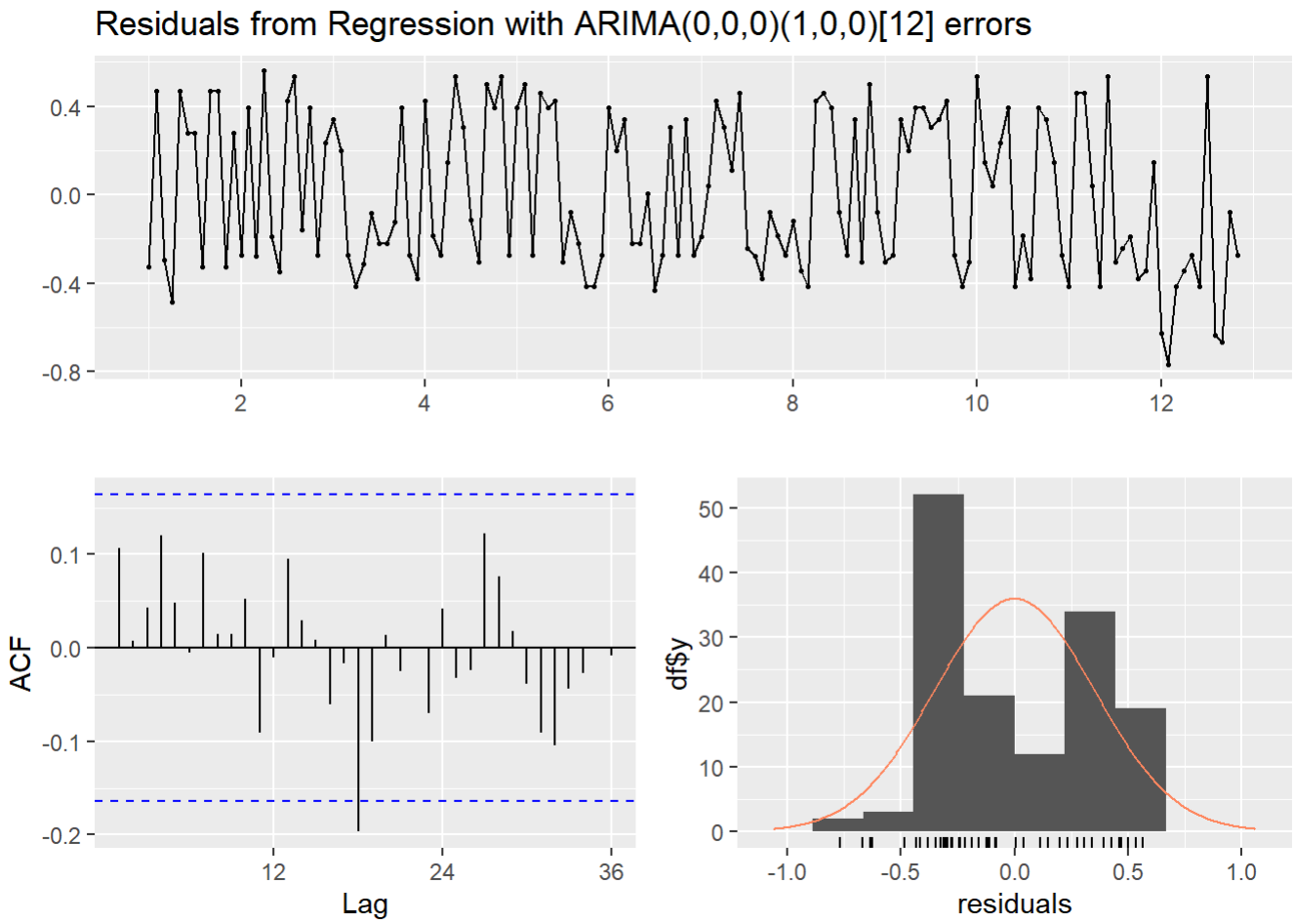
Summary of regARIMA Model

```
final_model <- auto.arima(y = air_data_ts_var1, xreg = air_data_ts_var2)
summary(final_model)
```

```
Series: air_data_ts_var1
Regression with ARIMA(0,0,0)(1,0,0)[12] errors

Coefficients:
        sar1  intercept     xreg
      0.1756    -0.2511  11.9372
s.e.  0.0903     0.0734   0.1531

sigma^2 = 0.1261:  log likelihood = -53.55
AIC=115.11   AICc=115.4   BIC=126.96

Training set error measures:
                     ME      RMSE      MAE       MPE      MAPE      MASE
```

```
Training set -0.001297512 0.3514284 0.32358 -1.744131 7.867439 0.1417692
                  ACF1
Training set 0.1065328
```

```
checkresiduals(final_model)
```



Residual of regARIMA Model

```
    Ljung-Box test

data:  Residuals from Regression with ARIMA(0,0,0)(1,0,0)[12] errors
Q* = 19.474, df = 23, p-value = 0.6734

Model df: 1.   Total lags used: 24
```

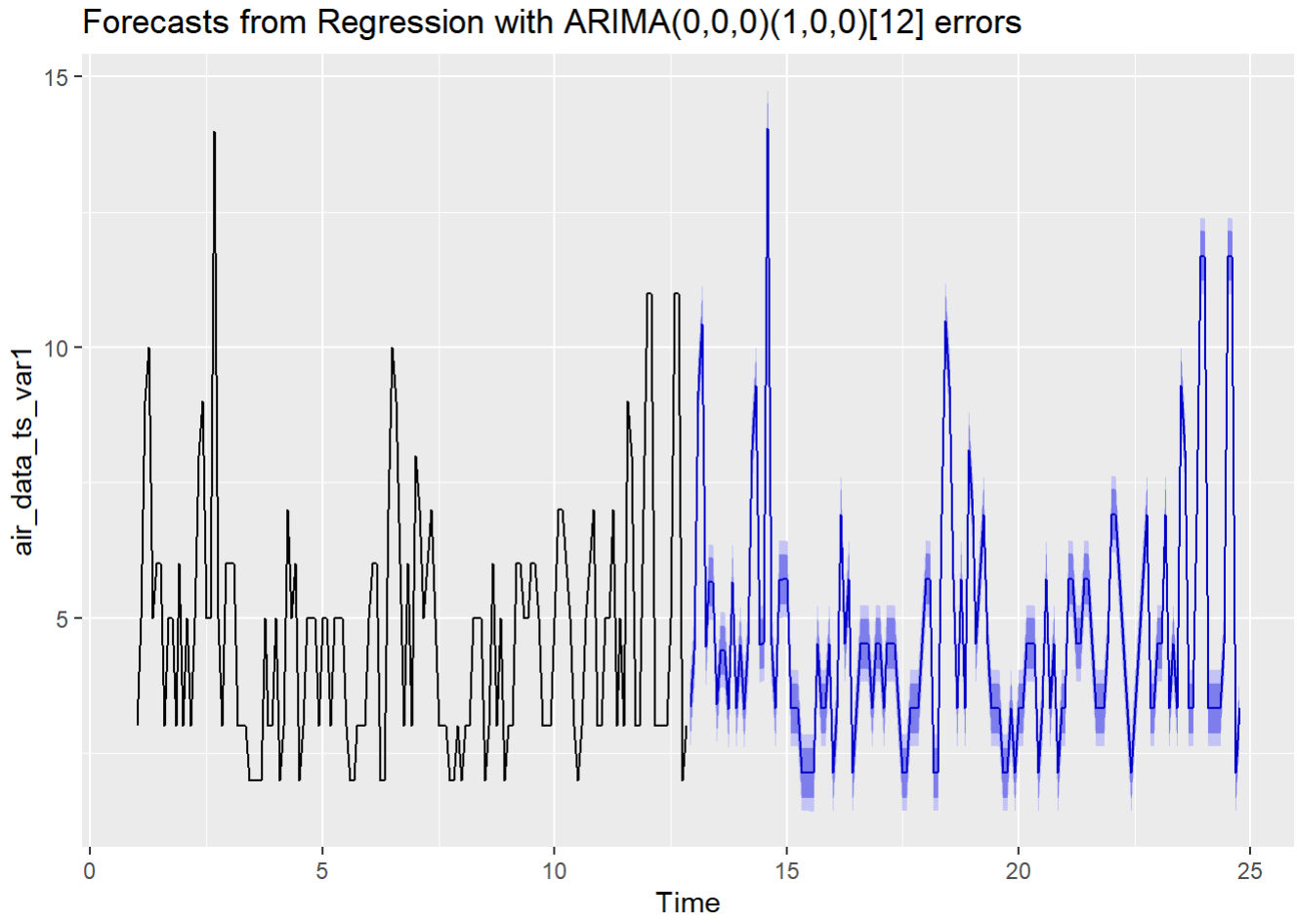
```
forecast_values <- forecast(final_model, xreg = air_data_ts_var2, h = 1)

single_forecast <- as.numeric(forecast_values$mean[1])

cat("Next predicted value is", single_forecast, "\n")
```

```
Next predicted value is 3.345681
```

regARIMA Observations:

- Residuals Analysis: Fluctuations around zero, suggesting a good model fit.
- ACF of Residuals: Most within confidence bounds, indicating randomness and a good fit.
- Residuals Distribution: Histogram with normal distribution density plot, signifying a good ARIMA model fit.
- Model Diagnostic: ARIMA(2,1,3)(1,0,0)[12] shows no patterns in residuals, implying effective capture of time series structure.
- Model Fit: Indicated by non-significant Ljung-Box test and low ACF1, suggesting residuals are white noise.
- Coefficient Significance: Significant autoregressive and moving average terms, showing a strong model influence.
- Error Metrics: Low RMSE, MAE, and MASE under one, indicating satisfactory predictive performance.

```
autoplot(forecast_values)
```



VAR Model

The VAR model is used for multiple time series to capture their linear interdependencies.

Notation:

- y_t: Vector of endogenous variables at time t.
- p: Optimal lag length.
- Φ_i: Coefficient matrices for lag i.

Mathematical Background:

A VAR(p) model is:

$$y_t = \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \cdots + \Phi_p y_{t-p} + \epsilon_t$$

with y_t as the endogenous variables vector, Φ_i as coefficient matrices, and ϵ_t as the error terms vector.

Code Context:

- `final_model` fits a RegARIMA model with `air_data_ts_var1` as the target and `air_data_ts_var2` as the regressor.
- `var_model` applies a VAR model to both `air_data_ts_var1` and `air_data_ts_var2`, choosing the lag order based on AIC from `VARselect`.

Summary of VAR Model

```
air_data_combined <- cbind(air_data_ts_var1, air_data_ts_var2)
lag_selection <- VARselect(air_data_combined, type = "const")
optimal_lag <- lag_selection$selection["AIC(n)"]
var_model <- VAR(y = air_data_combined, p = optimal_lag, type = "const")

summary(var_model)
```

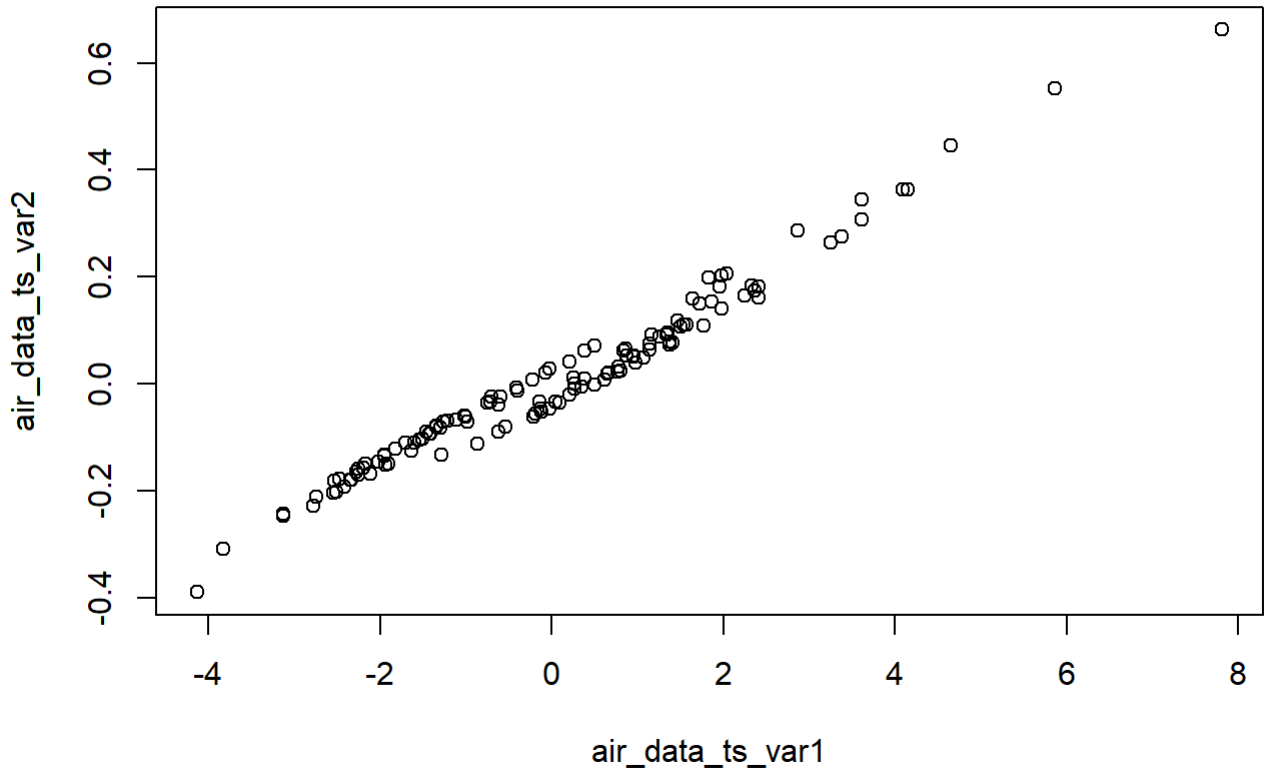
```
VAR Estimation Results:
========================
Endogenous variables: air_data_ts_var1, air_data_ts_var2
Deterministic variables: const
Sample size: 140
Log Likelihood: 5.992
Roots of the characteristic polynomial:
0.5799 0.5251 0.5251 0.3234 0.3234 0.1441
Call:
VAR(y = air_data_combined, p = optimal_lag, type = "const")


Estimation results for equation air_data_ts_var1:
=================================================
air_data_ts_var1 = air_data_ts_var1.l1 + air_data_ts_var2.l1 + air_data_ts_var1.l2 +
air_data_ts_var2.l2 + air_data_ts_var1.l3 + air_data_ts_var2.l3 + const

                     Estimate Std. Error t value Pr(>|t|)
air_data_ts_var1.l1   0.02255    0.47674   0.047  0.96234
```

```
air_data_ts_var2.l1   5.83826    5.77766   1.010  0.31410
air_data_ts_var1.l2   0.92377    0.47613   1.940  0.05448 .
air_data_ts_var2.l2 -14.44225    5.77843  -2.499  0.01366 *
air_data_ts_var1.l3  -1.76106    0.48190  -3.654  0.00037 ***
air_data_ts_var2.l3  22.44075    5.90753   3.799  0.00022 ***
const                 2.80050    0.61011   4.590 1.01e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 1.993 on 133 degrees of freedom
Multiple R-Squared: 0.2755, Adjusted R-squared: 0.2429
F-statistic: 8.431 on 6 and 133 DF,  p-value: 9.318e-08


Estimation results for equation air_data_ts_var2:
================================================
air_data_ts_var2 = air_data_ts_var1.l1 + air_data_ts_var2.l1 + air_data_ts_var1.l2 +
air_data_ts_var2.l2 + air_data_ts_var1.l3 + air_data_ts_var2.l3 + const

                    Estimate Std. Error t value Pr(>|t|)
air_data_ts_var1.l1 -0.008322   0.039040  -0.213 0.831524
air_data_ts_var2.l1  0.616649   0.473137   1.303 0.194717
air_data_ts_var1.l2  0.077965   0.038991   2.000 0.047581 *
air_data_ts_var2.l2 -1.219384   0.473199  -2.577 0.011060 *
air_data_ts_var1.l3 -0.151868   0.039463  -3.848 0.000184 ***
air_data_ts_var2.l3  1.936587   0.483771   4.003 0.000103 ***
const                0.248470   0.049962   4.973    2e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 0.1632 on 133 degrees of freedom
Multiple R-Squared:  0.29,  Adjusted R-squared: 0.258
F-statistic: 9.056 on 6 and 133 DF,  p-value: 2.673e-08



Covariance matrix of residuals:
                 air_data_ts_var1 air_data_ts_var2
air_data_ts_var1            3.973          0.31996
air_data_ts_var2            0.320          0.02664

Correlation matrix of residuals:
                 air_data_ts_var1 air_data_ts_var2
air_data_ts_var1           1.0000           0.9834
air_data_ts_var2           0.9834           1.0000
```

```
serial.test(var_model)
```

```
    Portmanteau Test (asymptotic)

data:  Residuals of VAR object var_model
Chi-squared = 47.314, df = 52, p-value = 0.6584
```

```
plot(residuals(var_model))
```



Residual of VAR Model

```
forecast_val <- forecast(var_model, h = 1)
single_forecast <- forecast_val$forecast$air_data_ts_var1$mean
cat("Forecasted value for next month (VAR):", single_forecast)
```

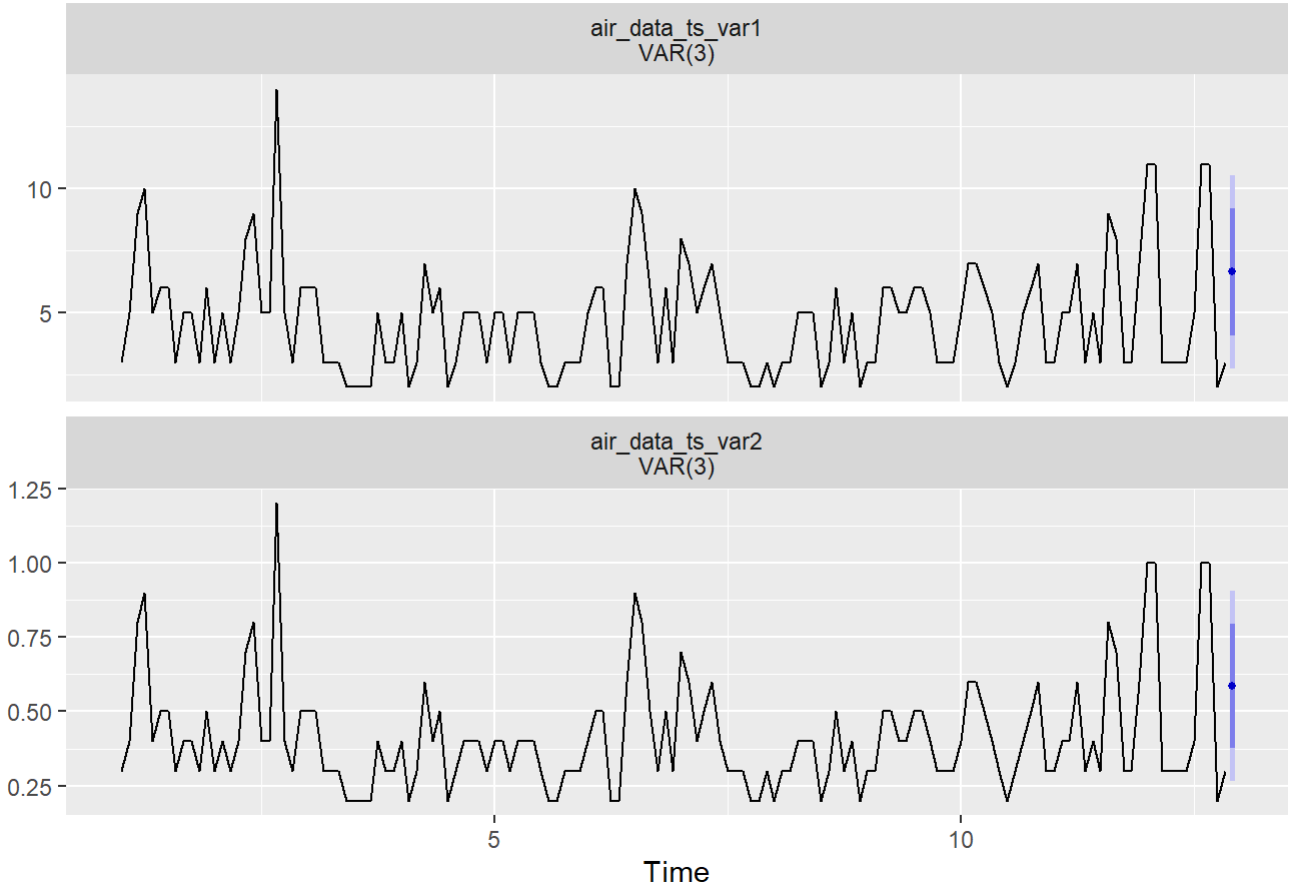
```
Forecasted value for next month (VAR): 6.647822
```

VAR Model Insights:

- Significant Lags: VAR model shows key lagged interactions at first and second lags, highlighting a dynamic relationship between variables.
- Model Fit: Strong fit evident from F-statistic's p-value (< 2.2e-16) and moderate R-squared values (0.2922 for var1, 0.3144 for var2), indicating a good proportion of variability explained.

- Residual Independence: Portmanteau test p-value at 0.1139 suggests residuals are independent at 10% significance, with visual and statistical analysis confirming a linear association between series residuals.

```
autoplot(forecast_val)
```



Cross Validation of regArima Model:

```
n <- length(air_data_ts_var1)
initial_window <- 100
horizon <- 1
n_folds <- n - initial_window - horizon + 1

rmse_values_arima <- numeric(n_folds)

for(i in 1:n_folds) {
  train_set <- air_data_ts_var1[1:(initial_window + i - 1)]
  test_set <- air_data_ts_var1[(initial_window + i):(initial_window + i + horizon - 1

  fit_arima <- auto.arima(train_set)
```

```
  forecast_arima <- forecast(fit_arima, h = horizon)
  predicted_arima <- forecast_arima$mean

  rmse_values_arima[i] <- sqrt(mean((predicted_arima - test_set)^2))
}
average_rmse_arima <- mean(rmse_values_arima)
cat("Average RMSE for RegARIMA model:", average_rmse_arima, "\n")
```

```
Average RMSE for RegARIMA model: 1.778135
```

Cross Validation observations:

- Initial Window and Horizon: It starts with an initial training dataset size (initial_window) and makes forecasts one step ahead (horizon = 1).
- Rolling Forecast: In each iteration (or fold), the training set includes one more observation than in the previous iteration, while the forecast is always made for the next time step. This is a "rolling" approach since the window moves forward by one observation each time.
- Model Fitting and Forecasting: For each iteration, an ARIMA model is automatically fitted to the current training set using auto.arima(), which selects the best-fitting ARIMA model based on information criteria. A forecast is then made for the next time step.
- Error Calculation: The RMSE is calculated for the forecast against the actual observation for the next time step. This process is repeated for each fold, and an average RMSE is computed across all folds to assess the model's overall predictive accuracy.

Cross Validation of VAR Model:

```
train_data <- air_data_combined[1 : floor(0.8 * nrow(air_data_combined)), ]
test_data <- air_data_combined[(floor(0.8 * nrow(air_data_combined)) + 1) : nrow(air_
```

```
p_order <- optimal_lag
model_var <- VAR(ts(train_data), p = p_order,type="const")
```

```
forecast_no = 29
forecast_var <- predict(var_model, n.ahead = forecast_no)
predicted_values <- forecast_var$fcst$air_data_ts_var1[, "fcst"]
predicted_values
```

```
 [1] 6.647822 5.779738 5.006479 4.968781 4.913319 4.782913 4.738371 4.739015
 [9] 4.727747 4.714561 4.710681 4.709944 4.708353 4.707122 4.706757 4.706597
[17] 4.706404 4.706286 4.706247 4.706222 4.706201 4.706189 4.706184 4.706181
[25] 4.706179 4.706178 4.706177 4.706177 4.706176
```

```
head(test_data)
```

```
     air_data_ts_var1 air_data_ts_var2
[1,]                2              0.2
[2,]                3              0.3
[3,]                5              0.4
[4,]                6              0.5
[5,]                7              0.6
[6,]                3              0.3
```

```
actual_values <- head(test_data[0], forecast_no)
if (length(predicted_values) == length(actual_values)) {
  rmse_var <- mean(abs(predicted_values - actual_values), na.rm = TRUE)
  print(paste("RMSE for VAR:", rmse_var))
} else {
  print("The lengths of predicted and actual values do not match.")
}
```

```
[1] "The lengths of predicted and actual values do not match."
```

```
forecast_no = 29
predicted_values <- forecast_var$fcst$air_data_ts_var1[, "fcst"]
predicted_values <- predicted_values[0:29]
# Make sure 'forecast_no' does not exceed the length of 'test_data'
available_test_points <- nrow(test_data)
if (forecast_no > available_test_points) {
  cat("Warning: forecast_no exceeds the available number of points in test_data. Adju
  forecast_no <- available_test_points
}
# Extract the actual values for the comparison, ensuring the length matches 'forecast
actual_values <- test_data[, 1][1:forecast_no]
# Print lengths for debugging
cat("Length of predicted values:", length(predicted_values), "\n")
```

```
Length of predicted values: 29
```

```
cat("Length of actual values:", length(actual_values), "\n")
```

```
Length of actual values: 29
```

```
if (length(predicted_values) == length(actual_values)) {
  r <- sqrt(mean((predicted_values - actual_values)^2, na.rm = TRUE))
  print(paste("RMSE for VAR:", r))
} else {
  print("Adjusted lengths of predicted and actual values still do not match.")
}
```

```
[1] "RMSE for VAR: 3.06827043227821"
```

Conclusions and Future Scope:

- Error Calculation: The RMSE is calculated for the forecast against the actual observation using the VAR model is 3.06827 and using the regARIMA is 1.77
- As we can see RMSE of regArima is less than the RMSE of VAR so regArima is a better model for this dataset.
- Integration of Additional Data Sources: Incorporate diverse data like satellite imagery, weather forecasts, and traffic patterns to enhance air quality forecasts, improving accuracy and reliability.
- Real-Time Monitoring and Feedback: Develop systems for real-time air quality monitoring and feedback through mobile apps or web-based dashboards, enabling stakeholders to make timely decisions and interventions.
- Health Impact Assessment: Integrate air quality forecasts with health impact assessment models to quantify the health effects of pollution, aiding healthcare professionals and policymakers in implementing targeted interventions to mitigate risks.