Time Series Analysis and Forecasting of Air Quality Index in Delhi

Name: Devendra Sanjay Patil - 2234

Abstract:

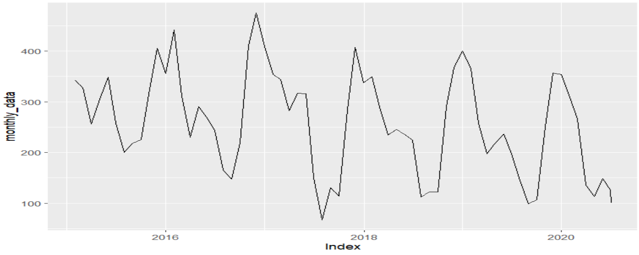
Air pollution is a significant concern in urban areas, particularly in Delhi, the capital city of India. The Air Quality Index (AQI) is a crucial measure that indicates the air quality level and its potential health impacts. The AQI is a numerical scale that ranges from 0 to 500, with higher values indicating more severe air pollution levels.

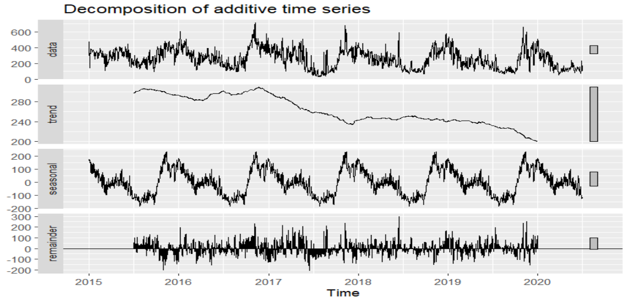
This project aims to analyze and forecast the AQI in Delhi using time series analysis on daily AQI data from 2015 to 2020. The objective is to understand the trends and patterns in the AQI data and predict future AQI values. This analysis can provide valuable insights into the air pollution situation in Delhi and aid in the development of effective pollution control strategies.

Data Description:

The dataset used in this project initially consisted of daily Air Quality Index (AQI) values for Delhi from 2015 to 2020. However, **analyzing daily data** can be **challenging** due to its high variability and volume. To simplify the analysis and capture broader trends, the daily data was **converted into monthly data**. This was achieved by calculating the average of all daily AQI values within each month.

This transformation reduces the granularity of the data from daily to monthly, making it more manageable and easier to analyze. It also helps to smooth out short-term fluctuations and highlight longer-term trends or cycles. The resulting monthly AQI dataset provides a clearer picture of the air quality situation in Delhi over the specified period. This data is then used for time series analysis and forecasting of AQI values.





Plot 1 shows the monthly data and plot 2 is decomposition of data into trend, seasonal and random error the series is additive

```
head(monthly_data)
```

Year	AQI
2015-01-31	342.2903
2015-02-28	327.9286
2015-03-31	256.0645
2015-04-30	305.2667
2015-05-31	348.5806
2015-06-30	258.3333

Stationarity Test:

```
> adf.test(monthly_data)
        Augmented Dickey-Fuller Test
data:  monthly_data
Dickey-Fuller = -3.8237, Lag order = 4,
p-value = 0.2295
alternative hypothesis: stationary
```

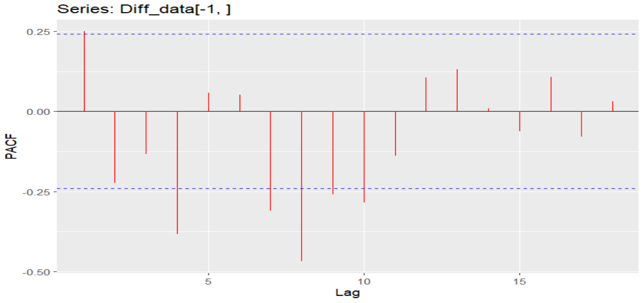
From the above output the data is non-stationary because p value greater than 0.05 which we accept Ho:non-stationary.

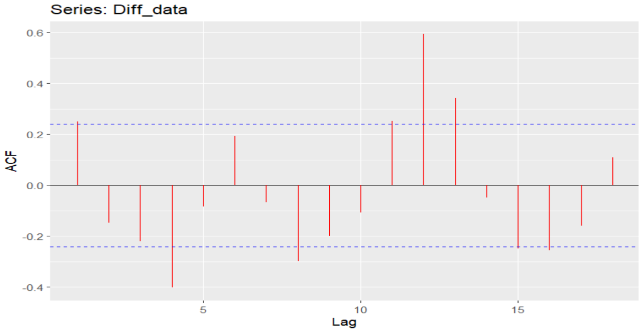
By using the Differencing method we gone make the data stationary.

```
> Diff_data=diff(monthly_data)
> adf.test(Diff_data[-1,])
        Augmented Dickey-Fuller Test

data:  Diff_data[-1, ]
Dickey-Fuller = -4.4724, Lag order = 4,
p-value = 0.01
alternative hypothesis: stationary
Warning message:
In adf.test(Diff_data[-1, ]) : p-value s
maller than printed p-value
```

Now the data is stationary p-value is less than 0.05.

ACF/PACF plot





Plot 1 is PACF and Plot2 is ACF from ACF we see than spike at 12 and from this plots we can there is significant relationship between the values

SARIMA MODEL:

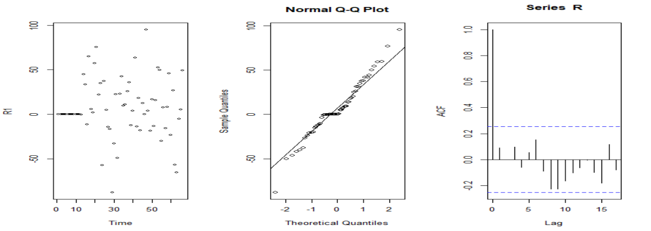
The Seasonal Autoregressive Integrated Moving Average (SARIMA) model is a popular model for time series forecasting. It extends the non-seasonal ARIMA model to effectively handle data with seasonal patterns. SARIMA combines the concepts of autoregressive (AR), integrated (I), and moving average (MA) models, along with seasonal components. It captures both short-term and long-term dependencies in the data, making it a powerful tool for forecasting. The SARIMA model is specifically designed to support univariate time series data with a seasonal component.

I have tried so many combinations of lags and finally I get one best model which mentioned below,

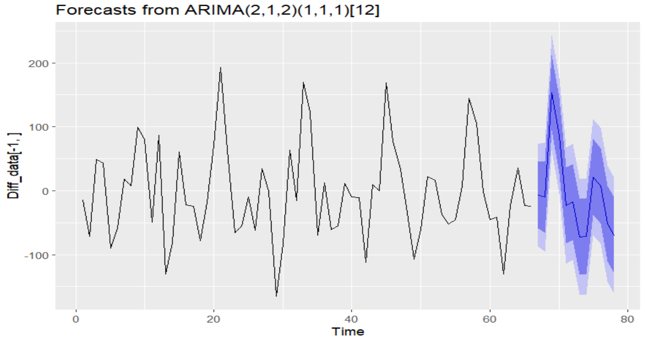
```
> summary(SM1)
Series: Diff_data[-1, ]
ARIMA(2,1,2)(1,1,1)[12]

Coefficients:
ar1       ar2       ma1      ma2
0.5760  -0.1450  -1.9927  0.9997

sar1      sma1
-0.2767  -0.6093

s.e.  0.1665   0.1541   0.1735  0.1738
0.2435   0.3724

sigma^2 = 1590:  log likelihood = -277.9
2
AIC=569.83   AICc=572.32   BIC=583.62

Training set error measures:
ME          RMSE        MAPE
7.312001  33.64611  328.2265
```

Residual Analysis:

```
> shapiro.test(R1)

        Shapiro-Wilk normality test

data:  R1
W = 0.96926, p-value = 0.1005
```



From the above out puts we can say the residuals follow all the assumption normality, independence and constant variance .



Conclusion:

From the forecasted plot we can see that the model fitted well and from the RMSE it is quit good so over all conclusion that SARIMA(2,1,2)(1,1,1)[12] fitted well on Air quality data of Delhi

Code:

```
rm(list=ls(all=T))
library(readxl)
library(tseries)
# Library Simple Moving Average
library(TTR)
# Library to forecast
library(forecast)
# Data visualisation
library(plotly)
library(xts)
library(TSstudio)
library(zoo)
data=read_xlsx("C:\\Users\\Shiv\\Documents\\Time Series\\Time_projects\\AIR_POL_DELHI.xlsx")
View(data)
g(data)
dim(data)
data1=ts(data$AQI,start = c(2015,1),frequency = 365)
autoplot(data1)
View(data)
d=decompose(data1)
autoplot(d)
# Assume 'data' is your daily data and 'dates' are the corresponding dates
daily_data =xts(data$AQI, order.by=as.Date(data$Date))

# Convert daily data to monthly data
monthly_data =apply.monthly(daily_data, FUN = mean)
colnames(monthly_data)=c("Year","API")
head(monthly_data)
autoplot(monthly_data)
plot(decompose(monthly_data))
##### check the stationarity
adf.test(monthly_data)  ### non-stationary

### making series stationary
Diff_data=diff(monthly_data)
adf.test(Diff_data[-1,])
plot(Diff_data)


####### ACF and PACF plot
ggAcf(Diff_data,col="red")
ggPacf(Diff_data[-1,],col="red")

####### Model building
SM1=Arima(Diff_data[-1,], order = c(2, 1, 2), seasonal = list(order = c(1, 1, 1), period = 12))
summary(SM1)
F2=forecast(SM1,12)
F2
autoplot(F2)

R1=resid(SM1)
shapiro.test(R1)
par(mfrow=c(1,3))
plot(R1,type="p")
qqnorm(R)
qqline(R)
acf(R)
```