505777837_ECON-144_Project-1

Andy Ho

2024-10-16

I. Introduction

Particulate matter present in the air is not a single pollutant, but rather a mixture of many chemical combinations that arises from both solid fragments and liquids. They vary widely in size and chemical composition and are defined by size using their diameter to help measure air quality. PM10 are particulate matter with a diameter of 10 microns or less and can be dangerous as inhaling them can affect the lungs harmfully, leathing to health effects. PM2.5 are particles of 2.5 microns or less, which means they are a portion of PM10. However, PM2.5 tends to be more dangerous as they can more easily transport throughout air and surfaces. As a result, they can go into deeper parts of the lungs, leading to even more adverse health affects compared to PM10. Since these particles can result from pollution, it is important to measure the air quality in many places to ensure at least a moderate level of clean air quality is maintained.

The time series data set that will be utilized for this project originates from from a Korean company called AirKorea, specializing in maintaining air quality across the country of South Korea. It measures the daily levels of PM2.5 and PM10 across different cities and districts in South Korea. The units of PM2.5 and PM10 are expressed in micrograms per cubic meter to express the concentration of a substance in the air or another gaseous medium. The dates of the data ranges from the end of 2013 to the beginning of 2022. Therefore, we will be using data from within the past two decades.

For the purpose of our project, we will try to ignore a slight few missing values and use full years, which leads to the project using data from January 2015 to January 2022. Additionally, we will use monthly data and use the averages of each month for our analysis so that there are not too many observations clustered in the diagrams.

The purpose of the data is to observe air pollution in South Korea, a country that has had severe health threats in regards to its air quality. Nowon-gu, which is one of the of the districts in Seoul, the capital of South Korea, is a residential district with many people and has the highest population density within the city. Therefore, it is important for us to observe and analyze recent data of the amount of PM2.5 in Nowon-gu so that air quality can be further maintained within the area for its many residents.

Uploading the Data

```
##
## 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
```

```
##
## Attaching package: 'lubridate'
```

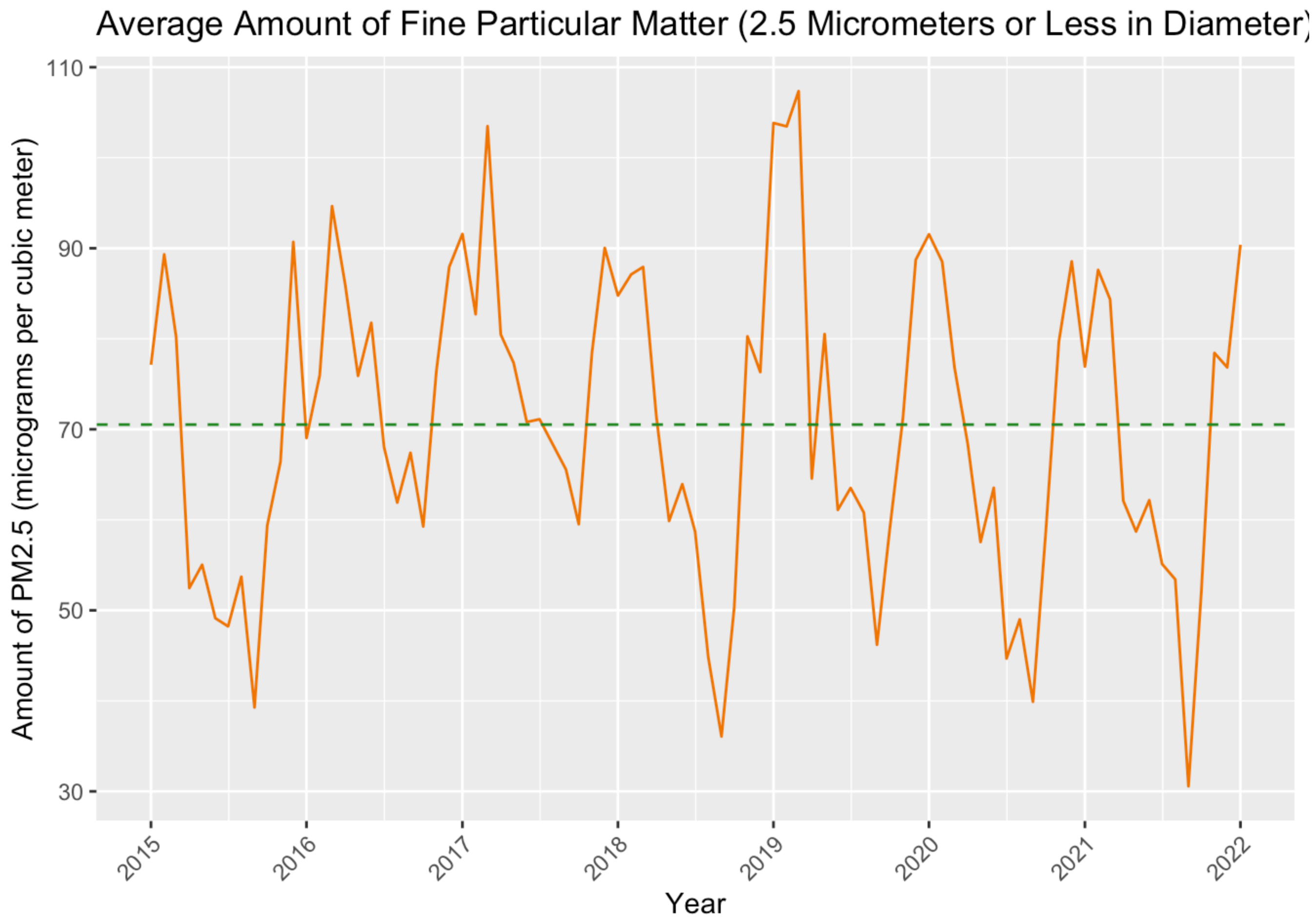
```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

II. Results

1. Modeling and Forecasting Trend

(a) Show a time-series plot of your data.

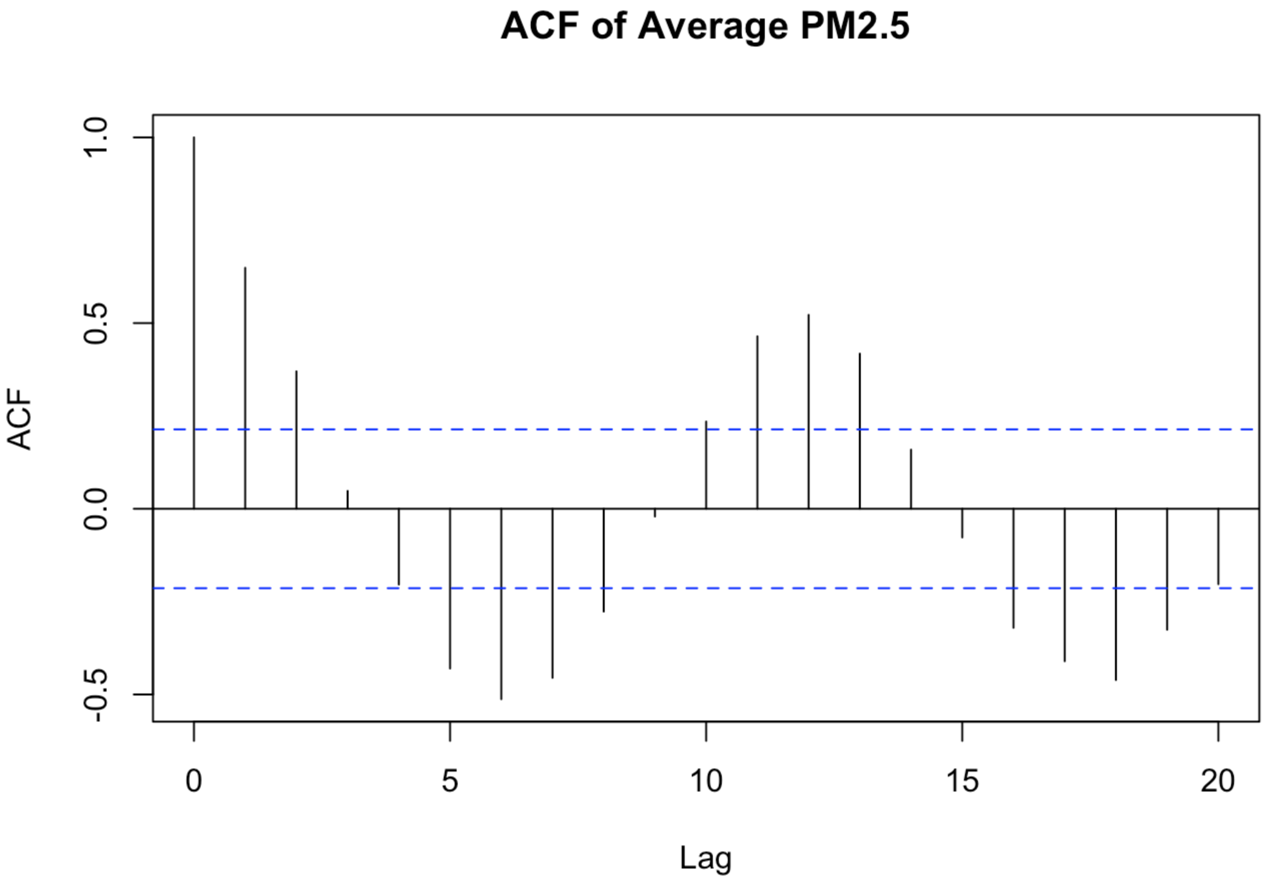
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```



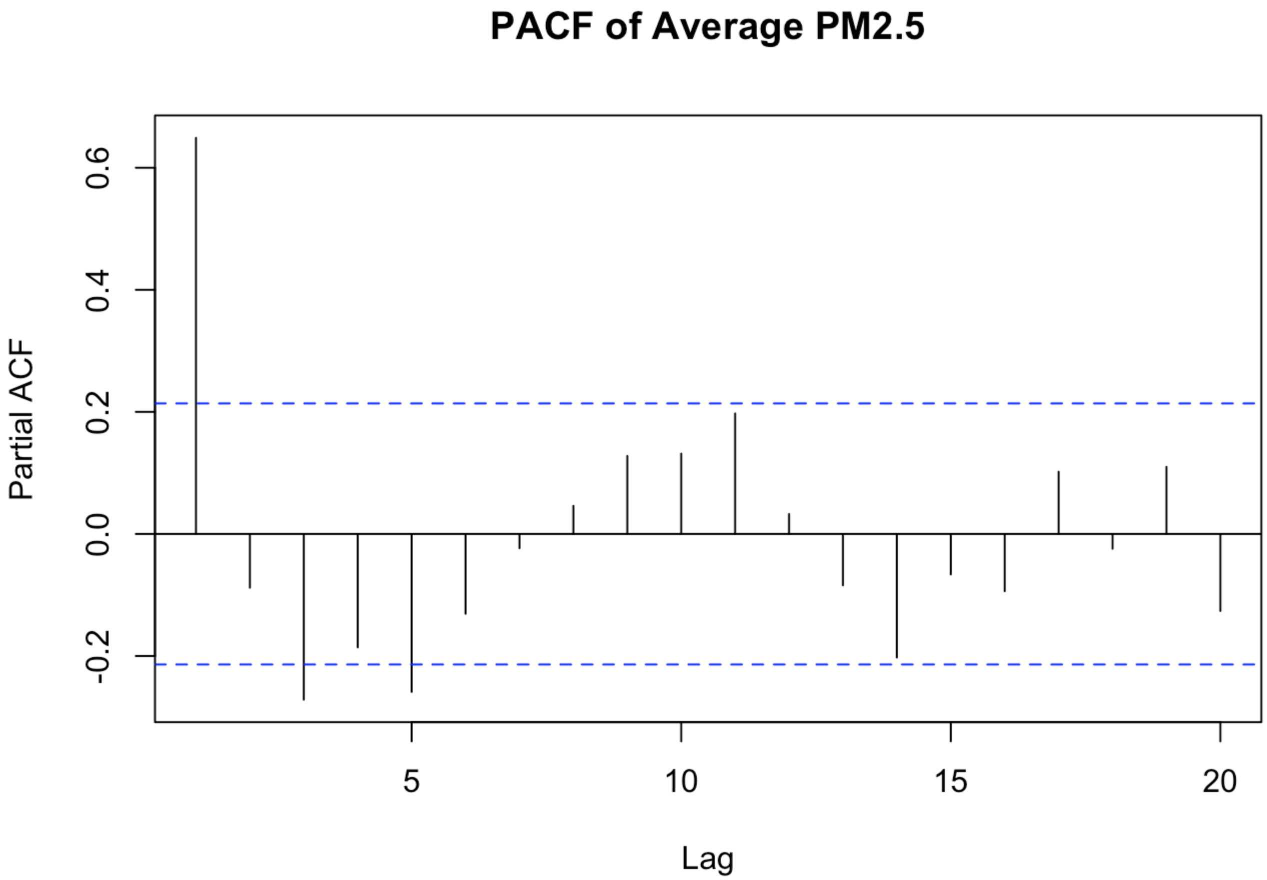
(b) Does your plot in (a) suggest that the data are covariance stationary? Explain your answer.

For the data to be covariance stationary, both the mean and variance of the series must remain fairly constant over time, along with the covariance between the observations at different points in time being only dependent on the time difference between them and not their actual positions. For the mean, we can see that the series appears to fluctuate significantly over time, with certain periods having considerably high peaks of PM2.5 levels and other periods with significantly lower PM2.5 levels. As a result, the overall mean PM2.5 levels of the monthly averages of PM2.5 between 2015-2022 in Nowon-gu appears to vary over time. Additionally, the variance seems to be inconsistent. This is because certain time chunks have brief stable periods while others have sharper fluctuations in between them. Thus, due to the mean and variance already not appearing to be constant throughout the series, it is unlikely that the data is covariance stationary.

(c) Plot and discuss the ACF and PACF of your data.



In the output above, we have an ACF plot that shows the correlation of the time series with itself at various lags. The x-axis represents the lag, or how many time periods earlier the value is, while the y-axis represents the strength of the correlation. The blue dashed lines represent the upper and lower bounds of the 95% confidence interval. If the bars extend beyond these lines, the autocorrelations are statistically significant. Using this information, we can interpret the plot as having a strong positive autocoreelation from lags 0 to 2, with a progressive decrease until it stays within the confidence interval between lags 3 to 4. Therefore, from lags 3 to 4, the is not a statistically significant autocorrelation in regards to the change of the average PM2.5 levels over time. However, the autocorrelation becomes statistically significant again from lags 5 to 8, only it is negative this time. Once again, the autocorrelation is no longer statistically significant for lag 9 and then goes back to being positive and statistically significant. As a result, there seems to be a cycle where the autocorrelations change from positive to negative repeatedly and has periods where it is statistically significant and other periods where it is not. Therefore, the dependence on how much of the average PM2.5 levels change over time seems to fluctuate in a cyclic pattern. It is also important to note that lag 0 has the strongest autocorrelation, to which it weakens soon after. Also, although not considerable, the values seem to gradually decline through the oscillation seen in the ACF plot.

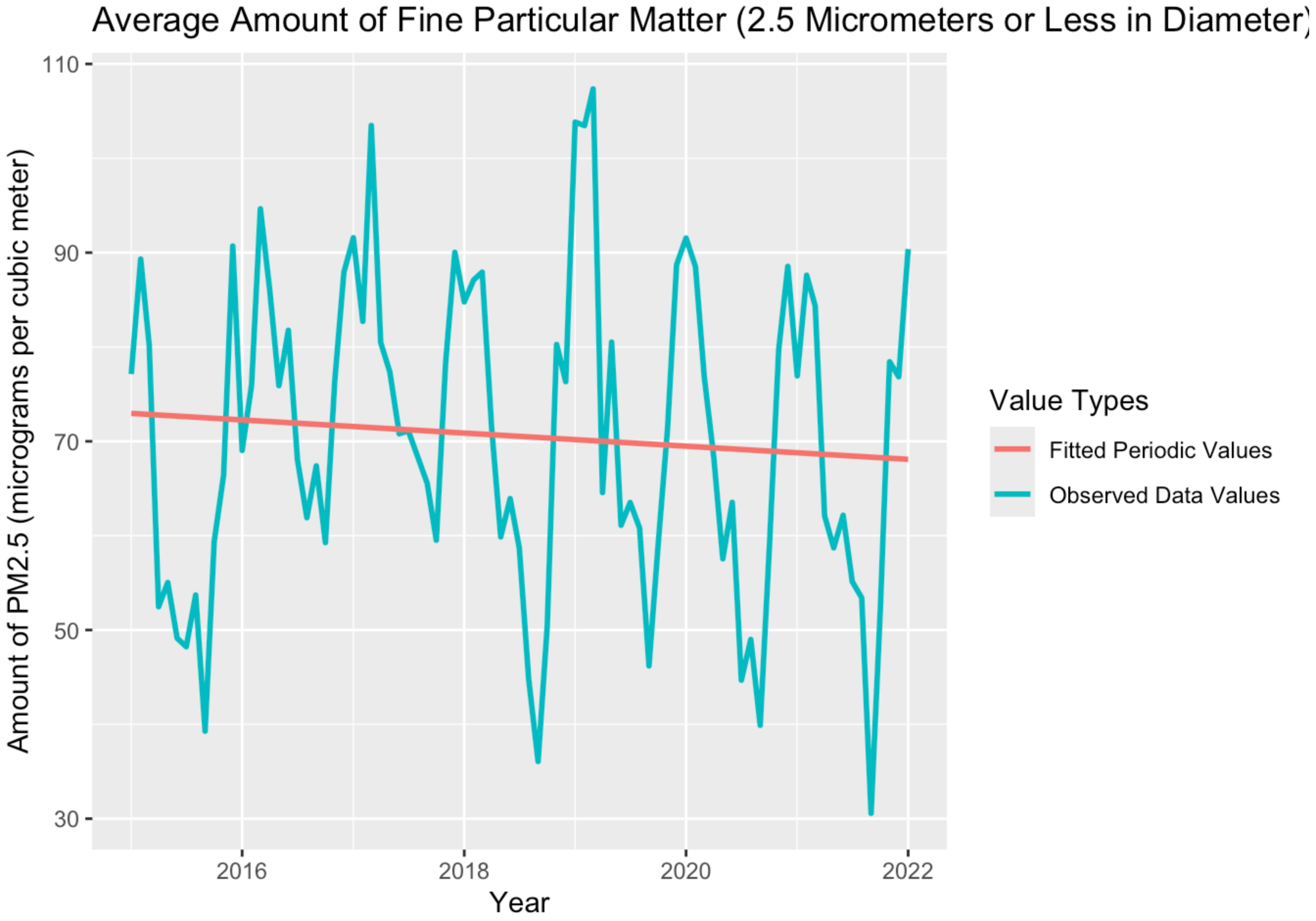


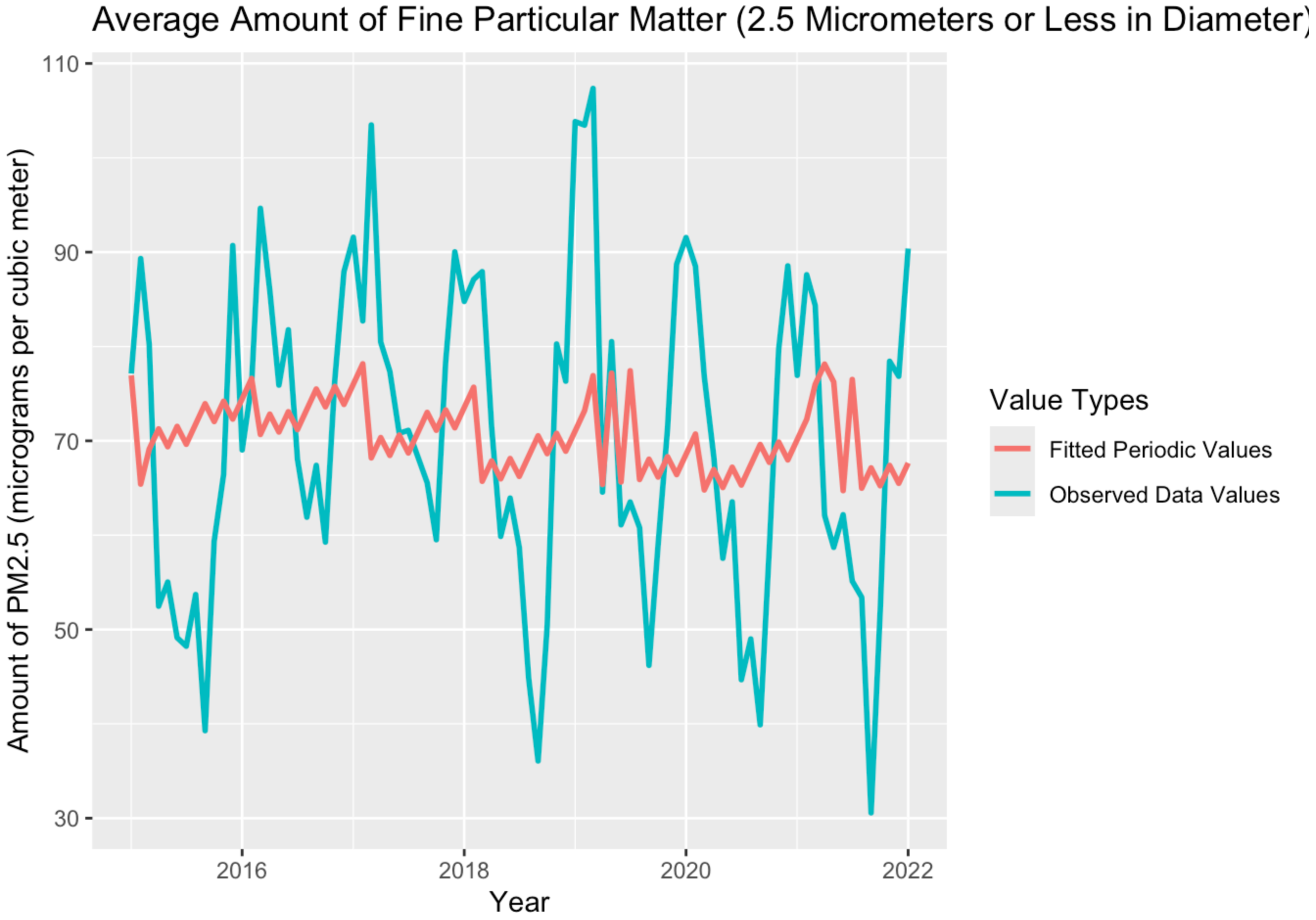
The output above depicts a PACF plot, which helps in understanding the correlation between the time series and its lag. The PACF plot also takes the effects of intermediate lags into consideration. The current value and its lagged values are shown in a PACF plot, without the influence of other intervening terms. As we can see in our PACF plot for the data, the first lag has a considerably strong partial autocorrelation at over 0.6. The bars for subsequent lags are all much shorter and are within the confidence interval, wth only lags 3 and 5 being exceptions. As a result, the plot suggests once you account for the first lag, the autocorrelation at higher lags is negligible. There is also a slight oscillation happening in the autocorrelation values within the confidence interval after the first lag.

Therefore, it can be deduced that an autoregressive model is likely fit for the data. This is because the ACF plot tails off but shows gradual decline over the lags throughout its oscillation. Additionally, the PACF plot shows that only the first lag has strong partial autocorrelation and cuts off short soon afterwards. Thus, we can further specify an AR(1) model where only the previous term in the process and the noise term have significant influence on the output.

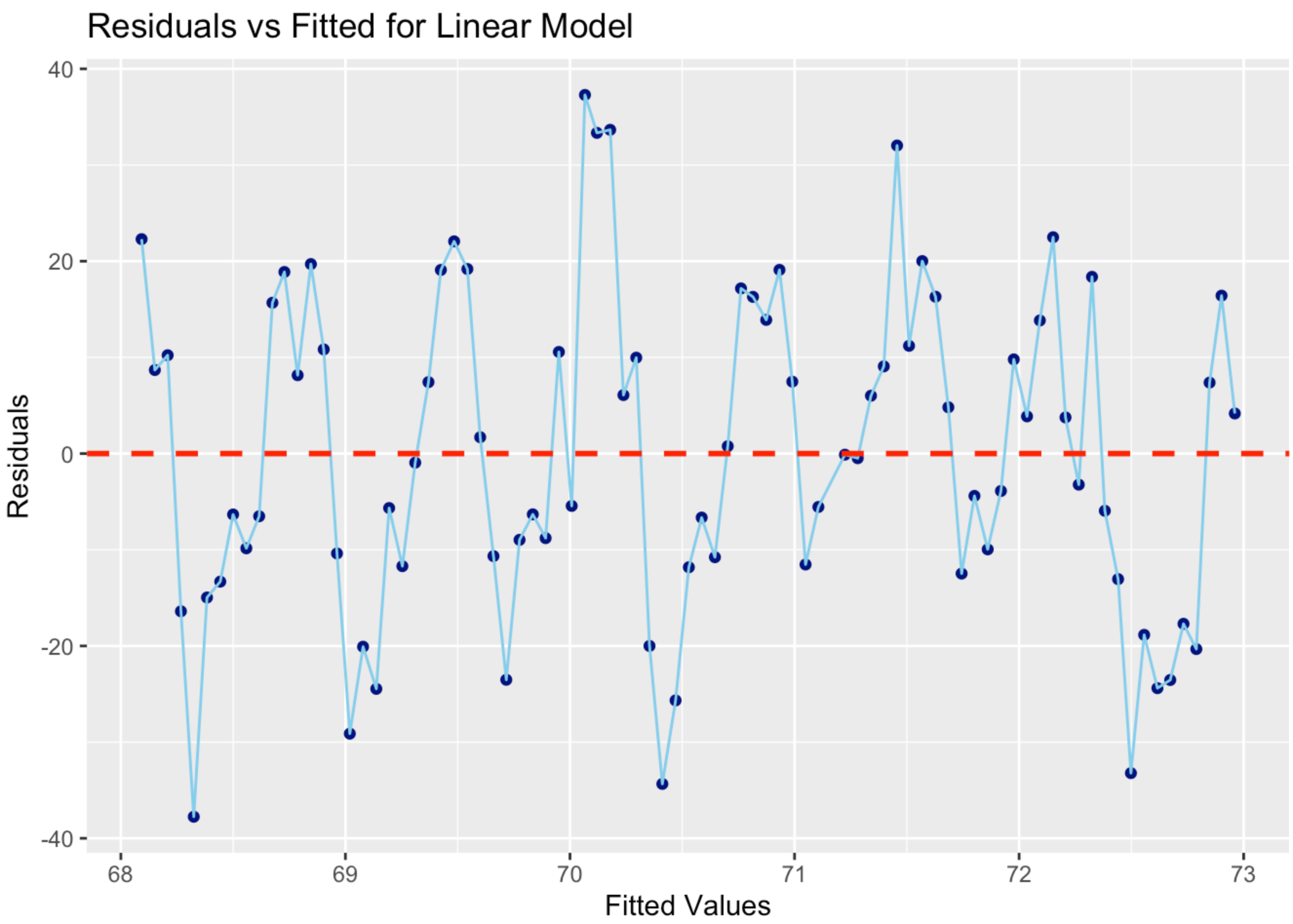
(d) it a linear and nonlinear (e.g., polynomial, exponential, quadratic + periodic, etc.) model to your series. In one window, show both figures of the original times series plot with the respective fit.

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

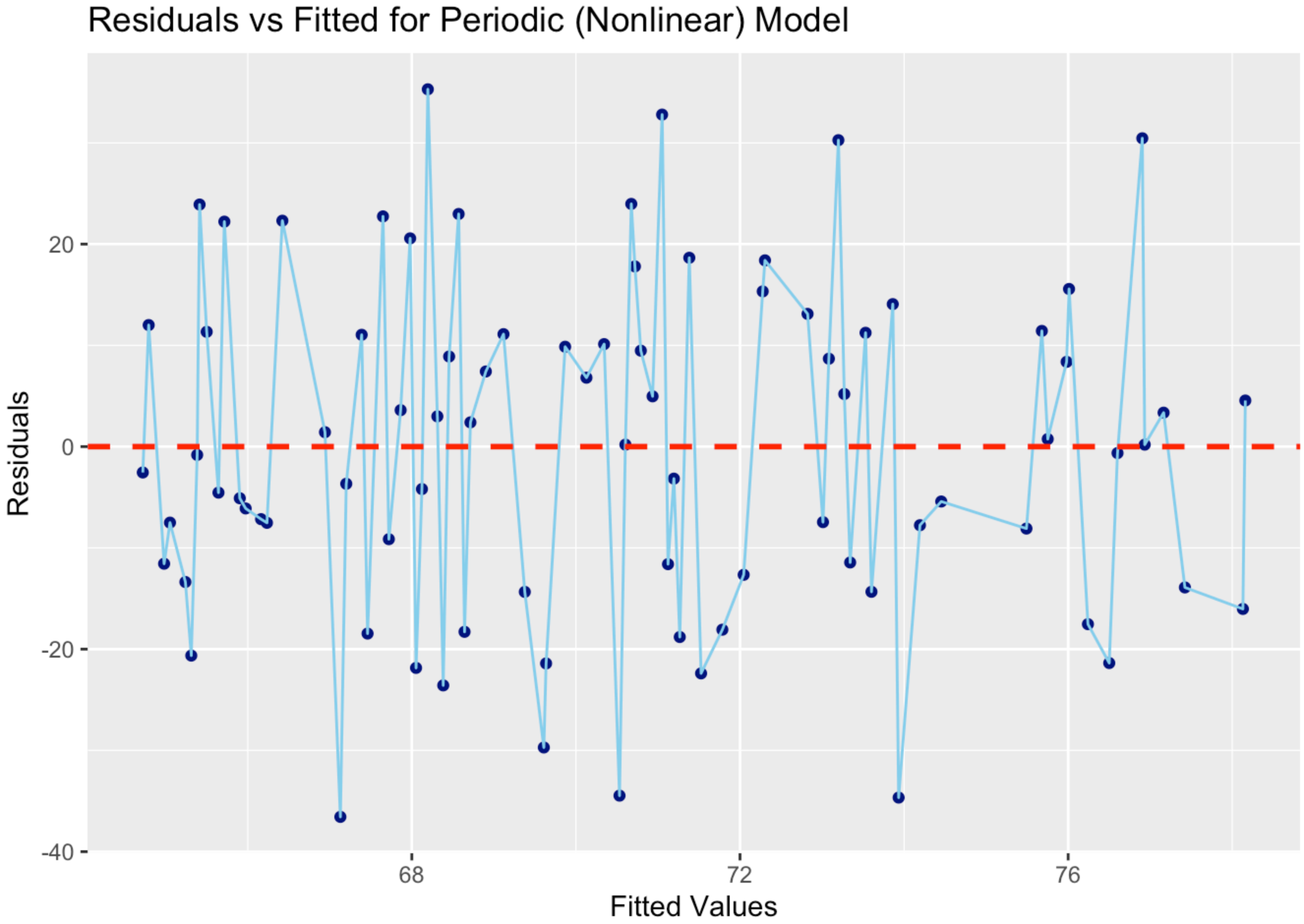

Average Amount of Fine Particular Matter (2.5 Micrometers or Less in Diameter)
Amount of PM2.5 (micrograms per cubic meter)
110
90
70
50
30
2016
2018
2020
2022
Year
Value Types
Fitted Periodic Values
Observed Data Values


Average Amount of Fine Particular Matter (2.5 Micrometers or Less in Diameter)
Amount of PM2.5 (micrograms per cubic meter)
110
90
70
50
30
2016
2018
2020
2022
Year
Value Types
Fitted Periodic Values
Observed Data Values

(e) For each model, plot the respective residuals vs. fitted values and discuss your observations.



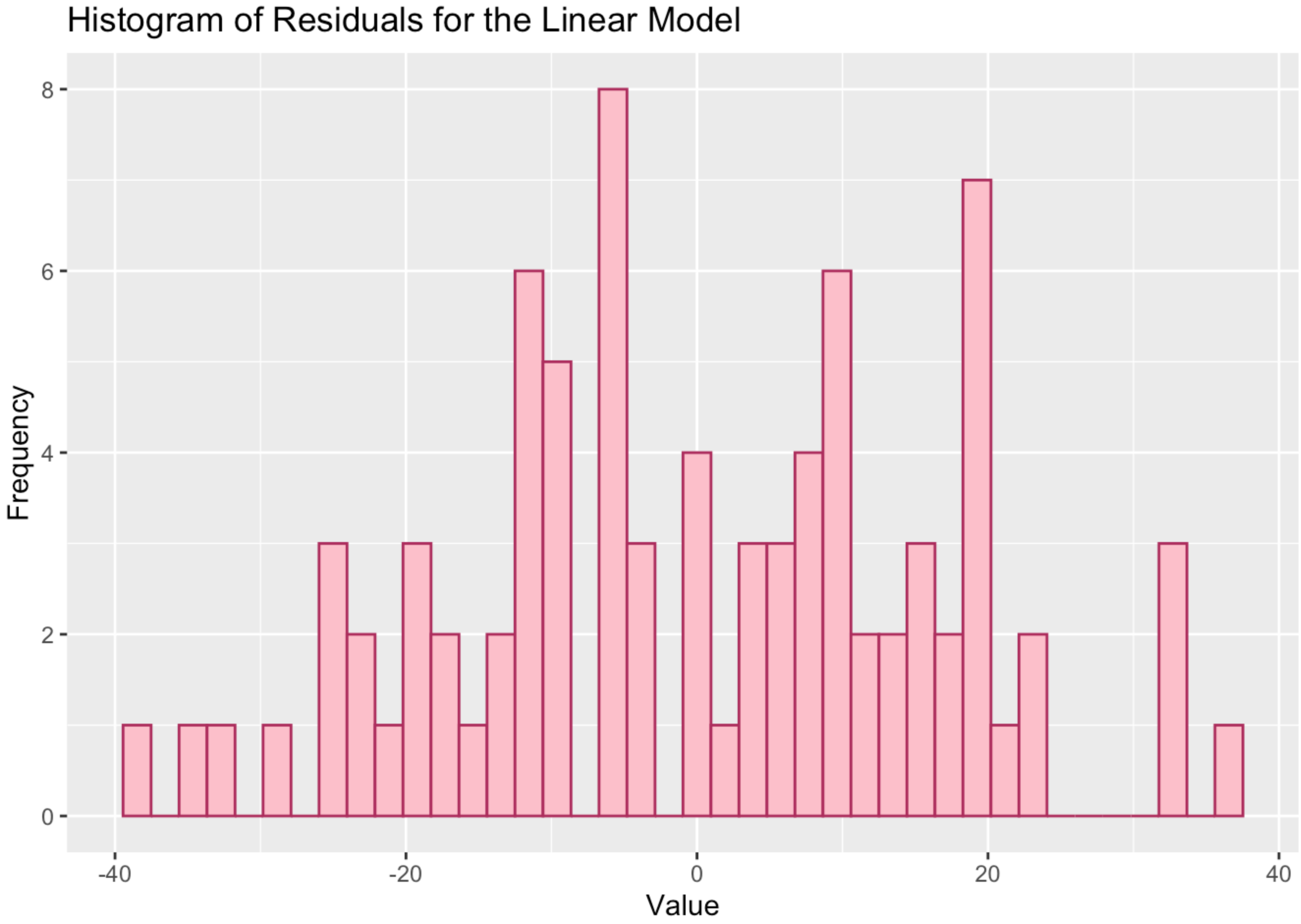
The range of the residuals for the linear model is between -40 to 40 and the mean of zero can be depicted on the dashed red horizontal line. Looking at the linear residual plot further, we can see that the points seem to be generally randomly scattered from left to right, with no distinct pattern. Therefore, the predictors seem to not be missing any information when capturing the response variable. Overall, the residual plot show that the linear model is a good fit for the data due to its random pattern throughout the plot.



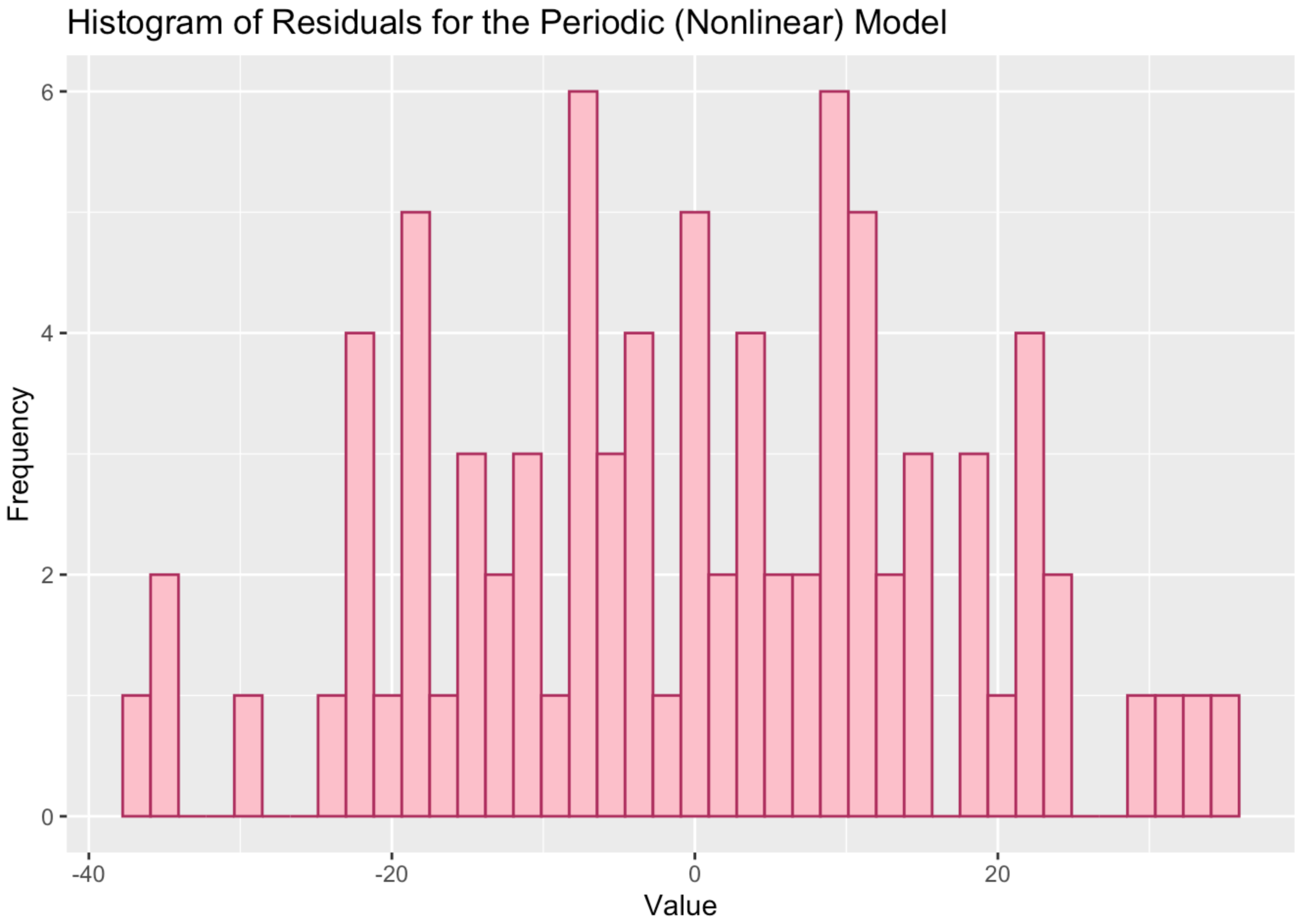
The range of the residuals for our chosen nonlinear model is also between is between -40 to 40 and the mean of zero can be depicted on the dashed red horizontal line. Looking at the nonlinear residual plot further, we can see that the points seem to be generally randomly scattered from left to right, with no distinct pattern. Therefore, the predictors seem to not be missing any information when capturing the response variable. Overall, the residual plot show that the nonlinear model is also a good fit for the data due to its random pattern throughout the plot.

Therefore, we may need further analysis to determine which one of the two models fit the data better than the other.

(f) For each model, plot a histogram of the residuals and discuss your observations.



For the histogram of the residuals for the linear model, the distribution is relatively spread out and does not appear to follow a normal distribution as there seems to be a slight skew to the right due to a little more frequent positive values. There are noticeable spikes around the values of -30 and 0, suggesting that the model may be consistently under-predicting or over-predicting values in some cases The residuals are more frequent near 0, but the tails extend in both directions, with values as extreme as -40 and 30. The symmetry of the residuals is not perfect, which may indicate that the linear model is not capturing all the underlying patterns in the data. The spread and irregular peaks suggest that the linear model may not be an ideal fit for the data, as it does not produce normally distributed residuals.

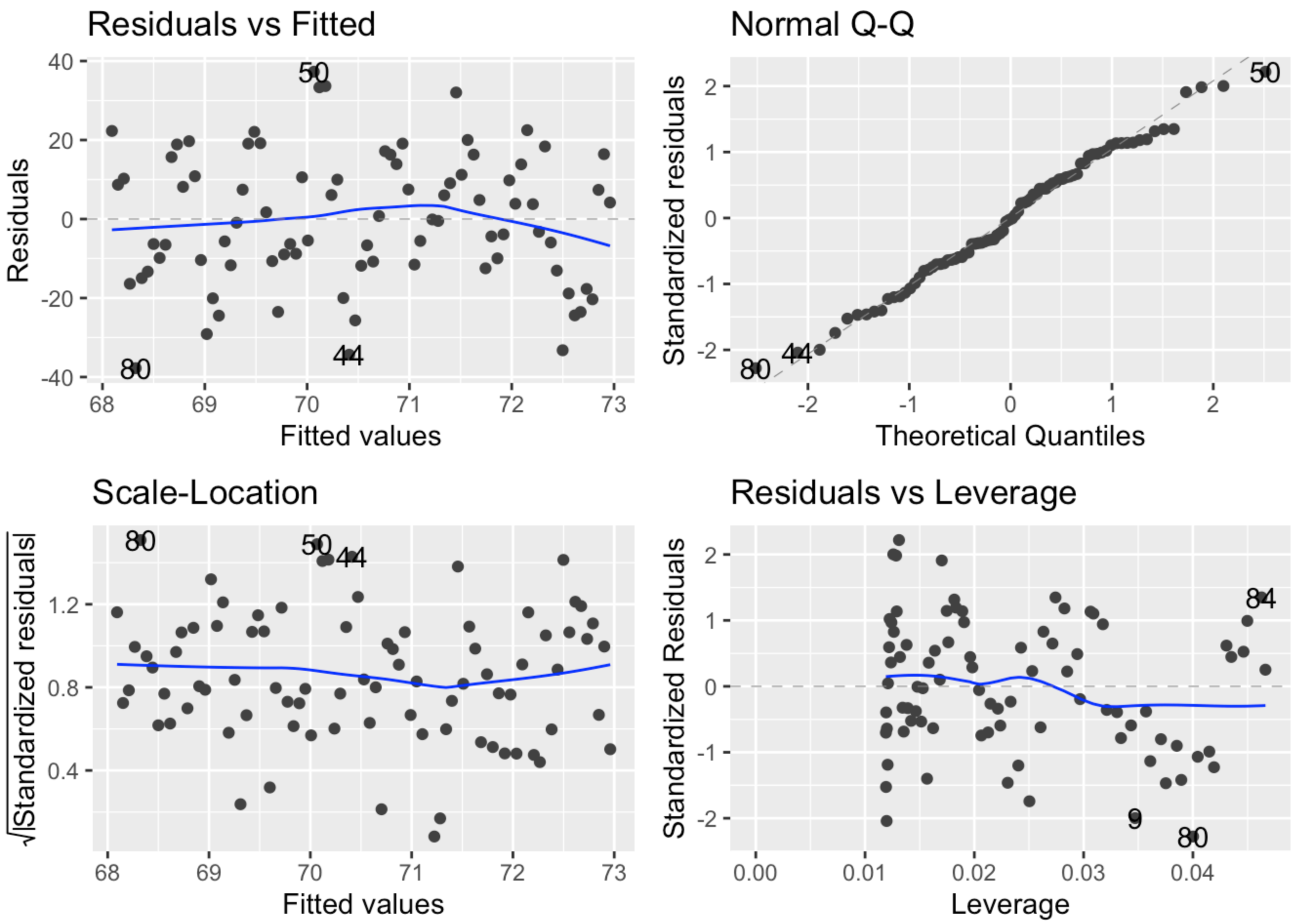


For the histogram of the residuals for the nonlinear model, the residuals show a wider spread, but the peaks are less pronounced than in the linear model's residuals. The distribution is less skewed, with more frequency centered around zero as we can see more clusters around the center rather than at the ends. This implies that the nonlinear model may have a better fit, though there are still residuals as extreme as -40 and 30. There are fewer sharp peaks and dips, suggesting a smoother residual distribution. Although it has outliers in the extreme points, the distribution looks more balanced in terms of under-predictions and over-predictions, which might indicate a better fit than the linear model for the data we are working with.

(g) For each model, discuss the associated diagnostic statistics (R2, t–distribution, F –distribution, etc.)

```
## Registered S3 methods overwritten by 'ggfortify':
##   method                 from
##   autoplot.Arima         forecast
##   autoplot.acf           forecast
##   autoplot.ar            forecast
##   autoplot.bats          forecast
##   autoplot.decomposed.ts forecast
##   autoplot.ets           forecast
##   autoplot.forecast      forecast
##   autoplot.stl           forecast
##   autoplot.ts            forecast
##   fitted.ar              forecast
##   fortify.ts             forecast
##   residuals.ar           forecast
```

```
##
## Call:
## lm(formula = `Average pm2.5` ~ Date, data = nowon_pollution_monthly_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.758 -11.576  -0.297  11.864  37.288
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 104.252662  43.636733   2.389   0.0192 *
## Date         -0.001904   0.002461  -0.774   0.4413
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.92 on 82 degrees of freedom
## Multiple R-squared:  0.007248,   Adjusted R-squared:  -0.004859
## F-statistic: 0.5986 on 1 and 82 DF,  p-value: 0.4413
```

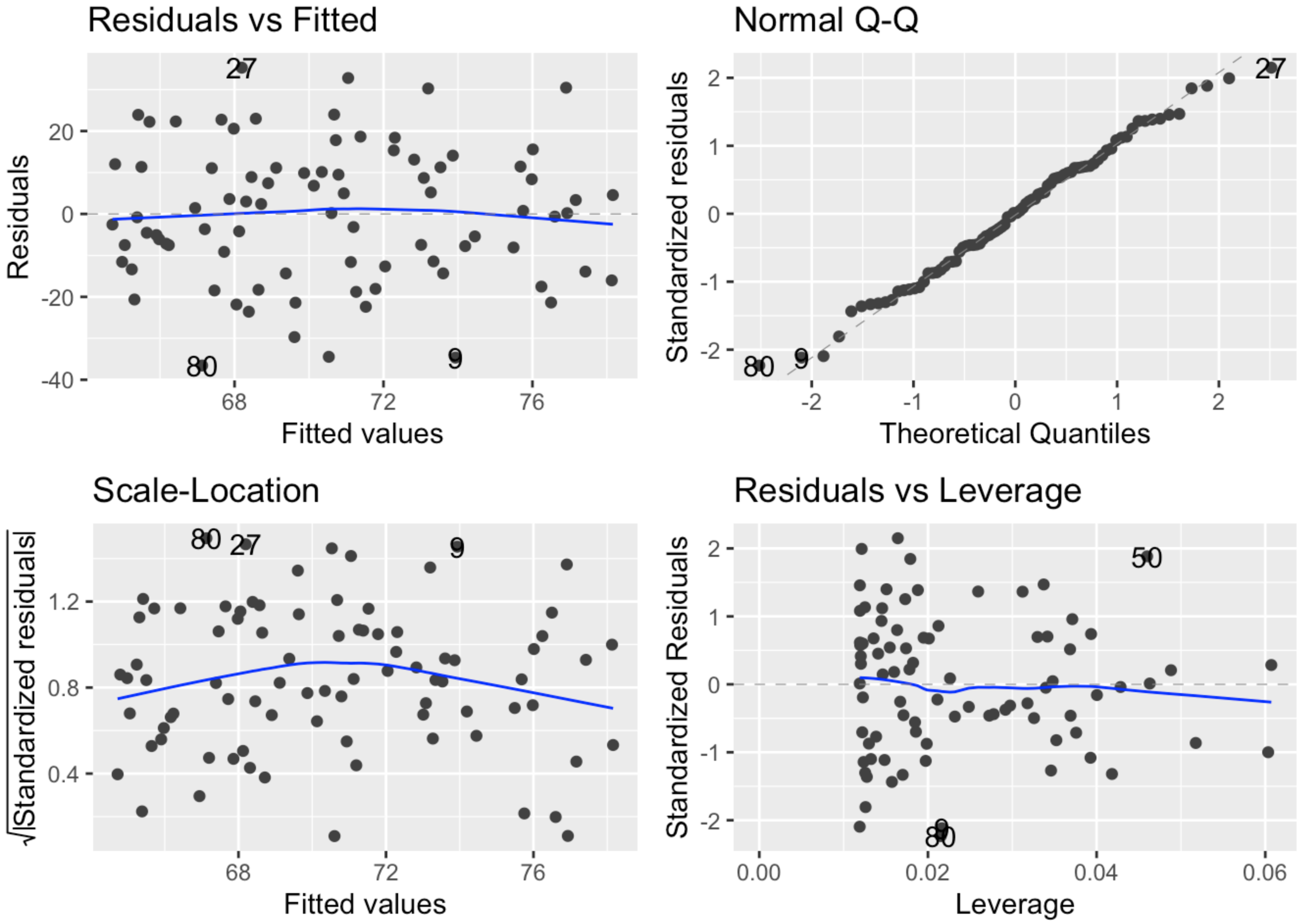


Looking at the summary statistics, we can see that the intercept is statistically significant due to its low p-value of 0.0192, where we note that the α value for statistical significance is usually 0.05. So, we can see that 0.0192 is less than this number. However, the p-value for the Date variable is 0.4413, which is greater than 0.05. Therefore, the Date variable is not statistically significant. We get an R-squared value of 0.007248, which is extremely insufficient as this means that only 0.7248% of the variability in the PM2.5 levels can be explained by the linear model’s predictor variable. The F-statistic is 0.5986, which is also not ideal as higher F-statistic values typically indicate statistically significant variables. Additionally, the residual standard error is 16.92, which is also somewhat high.

Looking at the residual plots, there seems to be a random scatter for both the Residuals vs. Fitted plot and the Scale-Location plot, which indicates constant variance. The Q-Q plot shows the points following a fairly straight linear line, meaning the assumption of normality in the data is satisfied. The Residuals vs. Leverage plot also depicts no points being in the top right or bottom areas, meaning there are no strong influential points affecting the data.

Thus, most of the setbacks of the linear model seems to be seen in the summary statistics, even with the plot diagnostics showing very few problems.

```
## 
## Call:
## lm(formula = `Average pm2.5` ~ `Sine Term`, data = nowon_pollution_monthly_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -36.562 -11.869   0.196  11.267  35.289 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 7.578e+01  3.097e+00  24.471   <2e-16 ***
## `Sine Term` 9.393e+11  4.496e+11   2.089   0.0398 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 16.55 on 82 degrees of freedom
## Multiple R-squared:  0.05054,    Adjusted R-squared:  0.03896 
## F-statistic: 4.364 on 1 and 82 DF,  p-value: 0.0398
```



Looking at the summary statistics, we can see that the intercept is statistically significant due to its low p-value of less than 2e-16, where we note that the α value for statistical significance is usually 0.05. So, we can see that a value almost near zero is satisfactory. Additionally, the p-value for the Sine term is 0.0398, which is also elss than 0.05. Therefore, the Sine term is also statistically significant. We get an R-squared value of 0.05054, which is extremely insufficient as this means that only 5.054% of the variability in the PM2.5 levels can be explained by the linear model's predictor variable. However, this is still an improvement from the linear model's extremely low R-squared value. The F-statistic is 4.364, which is also an improvement from the linear model as it was stated earlier that higher F-statistic values indicate statistically significant variables. Additionally, the residual standard error is 16.55, which is still somewhat high but a slight decrease from the residual standard error seen in the linear model's summary statistics.

Looking at the residual plots, there seems to be a random scatter for both the Residuals vs. Fitted plot and the Scale-Location plot, which indicates constant variance. The Q-Q plot shows the points following a fairly straight linear line, meaning the assumption of normality in the data is satisfied for the nonlinear model, like the linear model. The Residuals vs. Leverage plot also depicts no points being in the top right or bottom areas, meaning there are no strong influential points affecting the data.

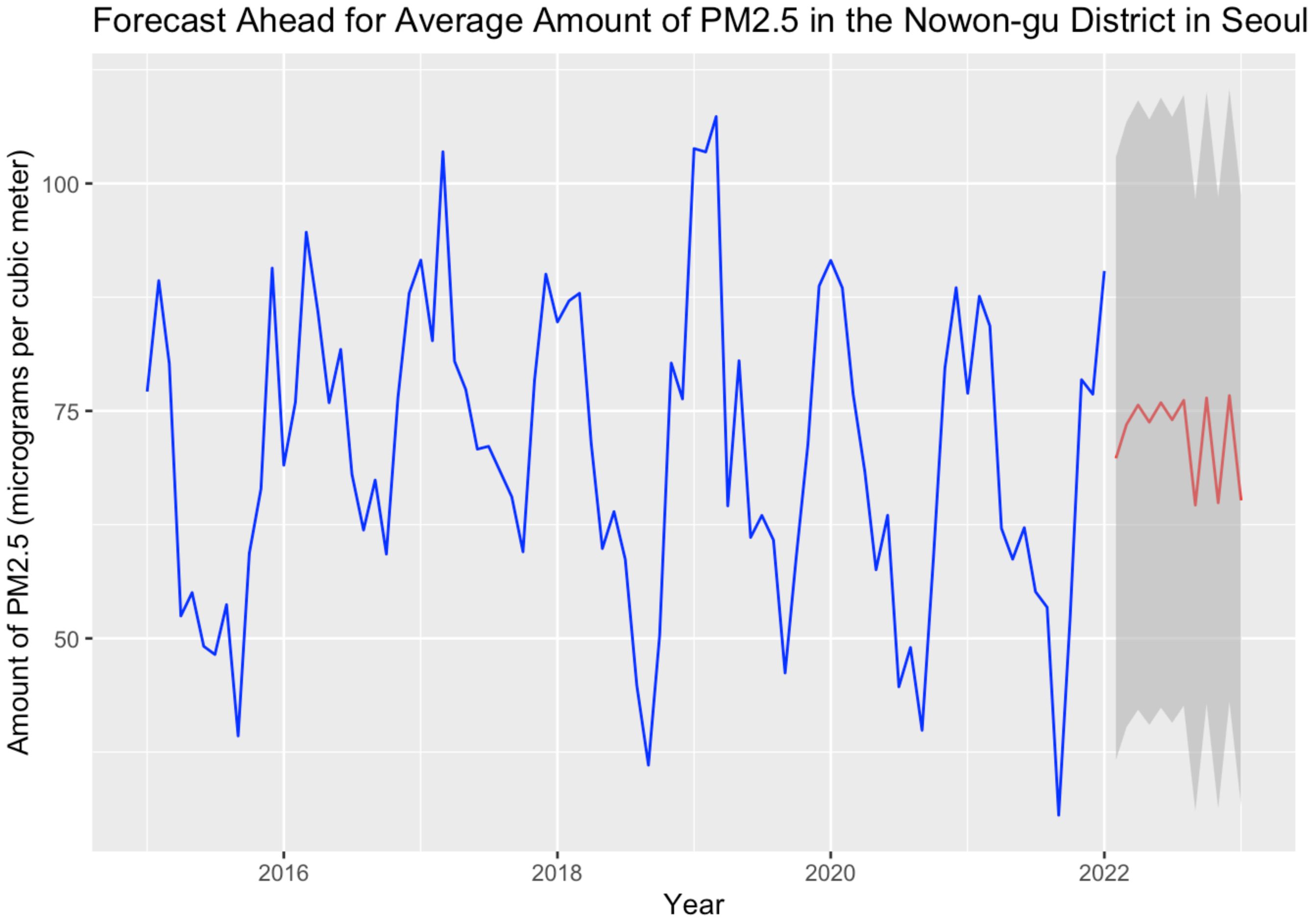
Thus, only the R-squared value of of the nonlinear model seems to be seen in the summary statistics seem to be concerning due to its low value, even with the plot diagnostics showing very few problems. Even then, all of the values in the summary statistics for the nonlinear model can be seen as an improvement from the linear model. Due to the diagnostic plots for the nonlinear model being similar to the linear model, the imporvements seen in the summary statistics can indicate that the nonlinear model is a better fit for the data compared to the linear model.

(h) Select a trend model using AIC and one using BIC (show the values obtained from each criterion). Do the selected models agree?

```
##                            AIC      BIC
## Linear              717.5834 724.8759
## Periodic (Nonlinear) 713.8384 721.1309
```

In terms of AIC and BIC scores, we generally want models with lower values in both cases because lower values indicate a better trade-off between the fit of the model and its complexity. As we can see in the data frame above, the periodic (nonlinear) model has lower values than the linear model in both AIC and BIC scores. Therefore, we would select the nonlinear model over the linear model in this case. Thus, the selected models for both AIC and BIC scores agree.

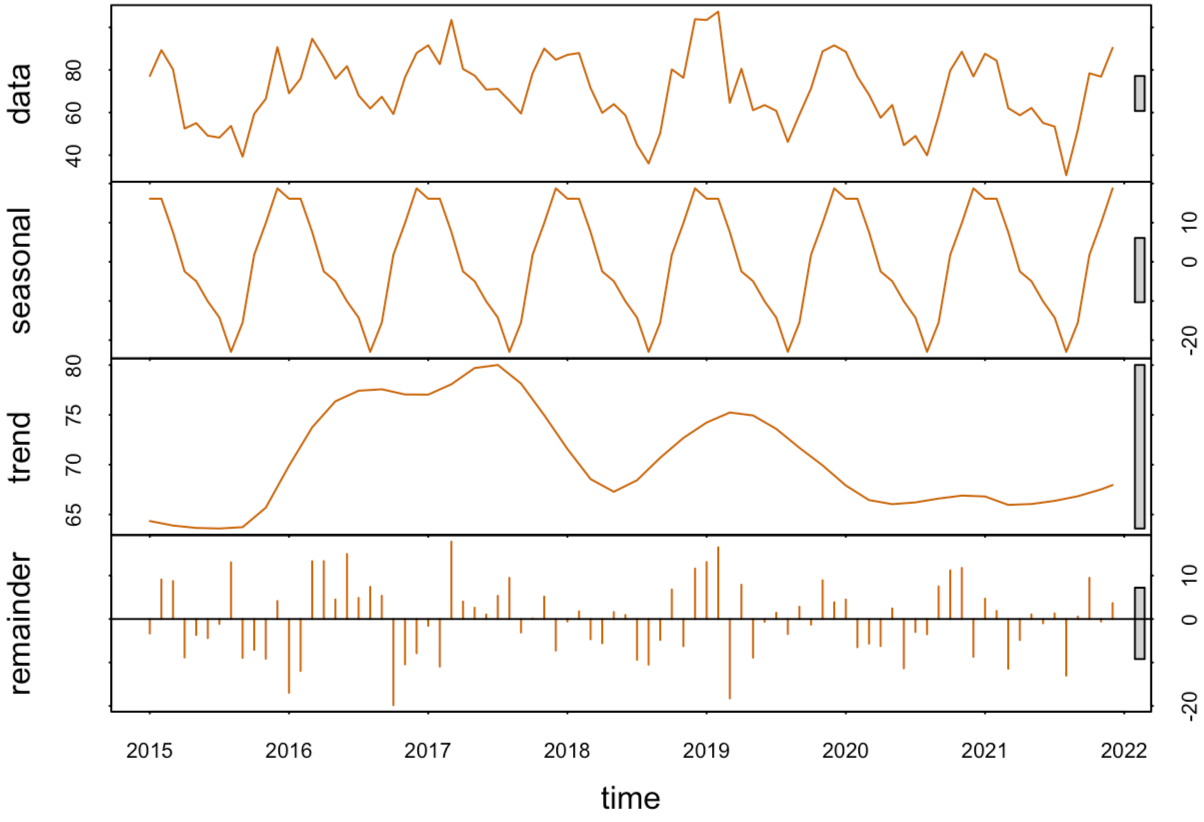
(i) Use your preferred model to forecast h-steps (at least 12) ahead. Your forecast should include the respective uncertainty prediction interval. Depending on your data, h will be in days, months, years, etc.

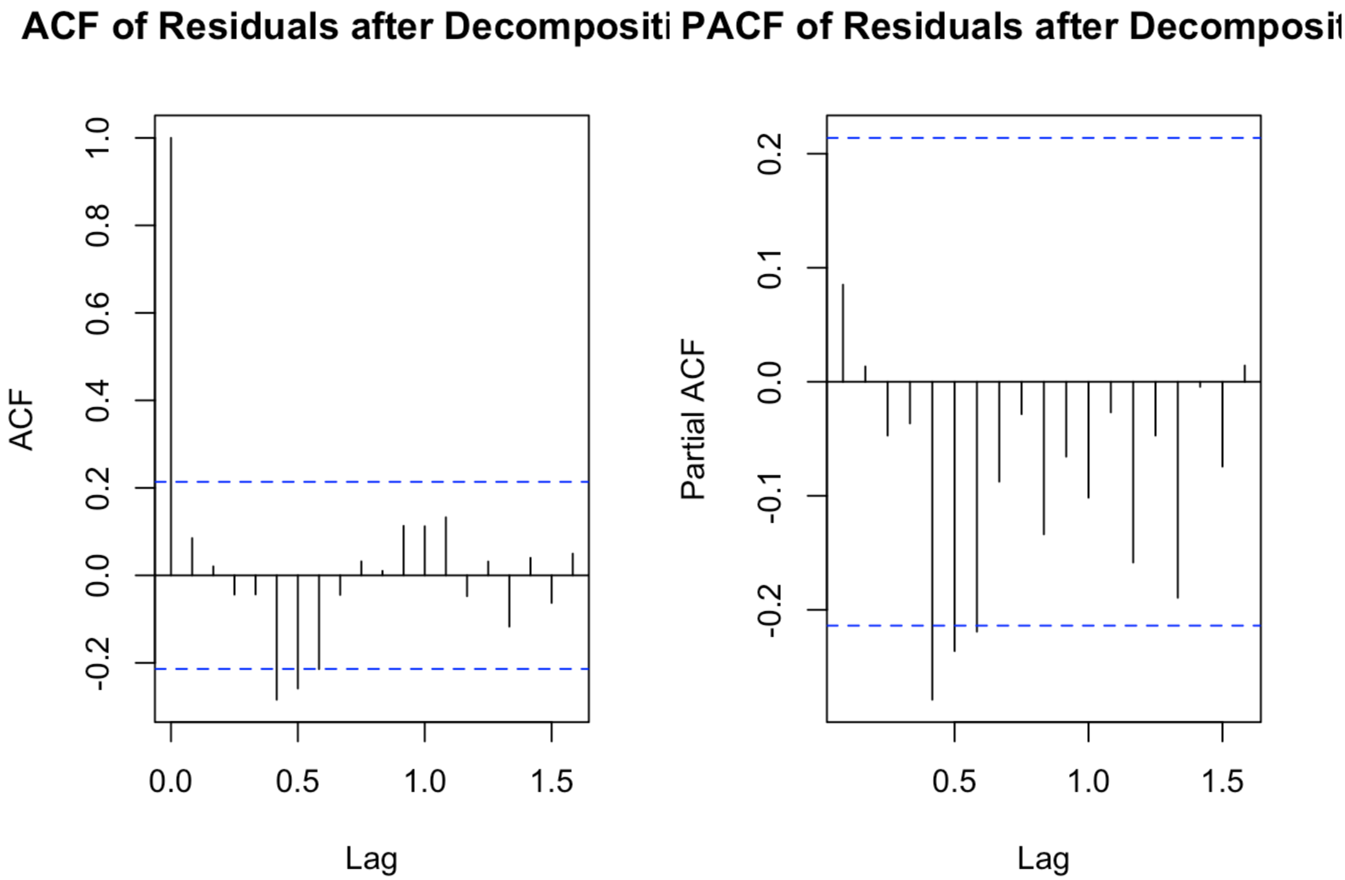


For a brief description, the output depicts our forecast plot with the periodic (nonlinear) model’s monthly forecasts for the next year. In this case, since the last complete month of the original data set was January 2022, the forecast data contains forecast for up to January 2023. As we can see, the forecast follows a nonlinear path with a spikes and dips, indicating the line to follow some type of periodic path. The prediction interval also has dips and spikes and extends towards the a similar range to that of the line depicting the values from the original data set. The boundaries of the prediction interval also oscillates, similar to what is happening in both the line corresponding with the observed data and the line corresponding with the forecast data.

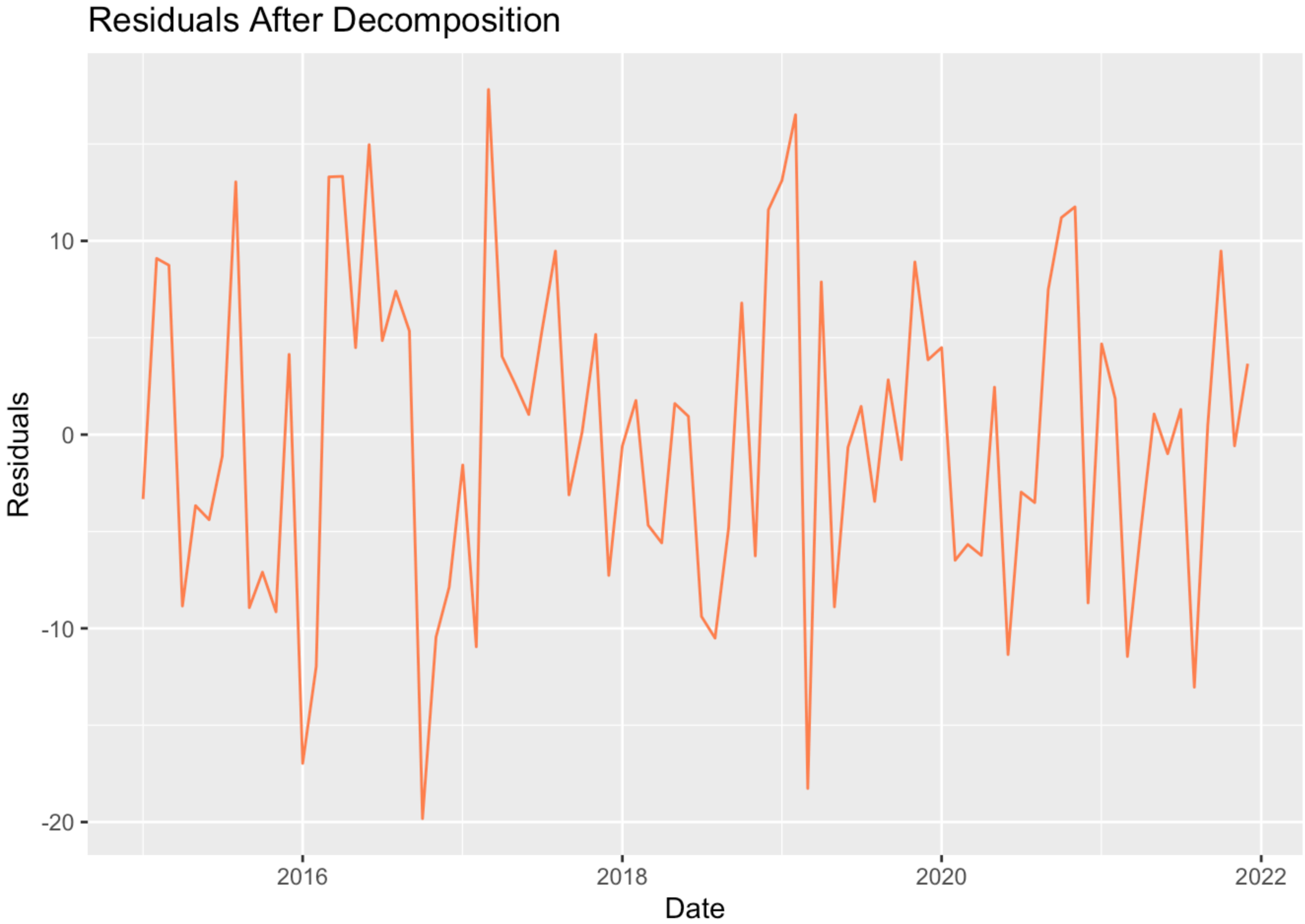
2. Trend and Seasonal Adjustments

(a) Perform an additive decomposition of your series. Remove the trend and seasonality, and comment on the ACF and PACF of the residuals (i.e., what is left after detrending and seasonally adjusting the series). Comment on the results.





```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```

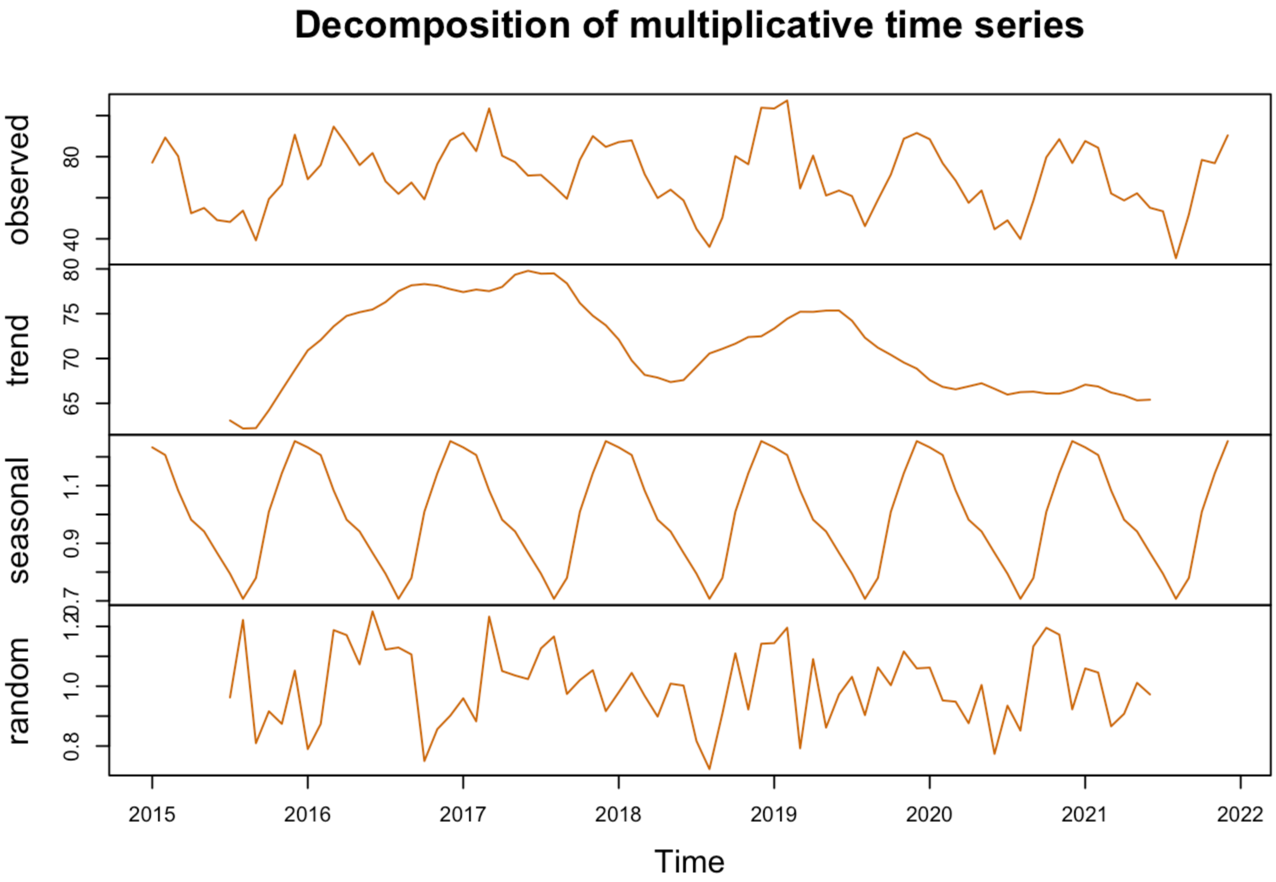


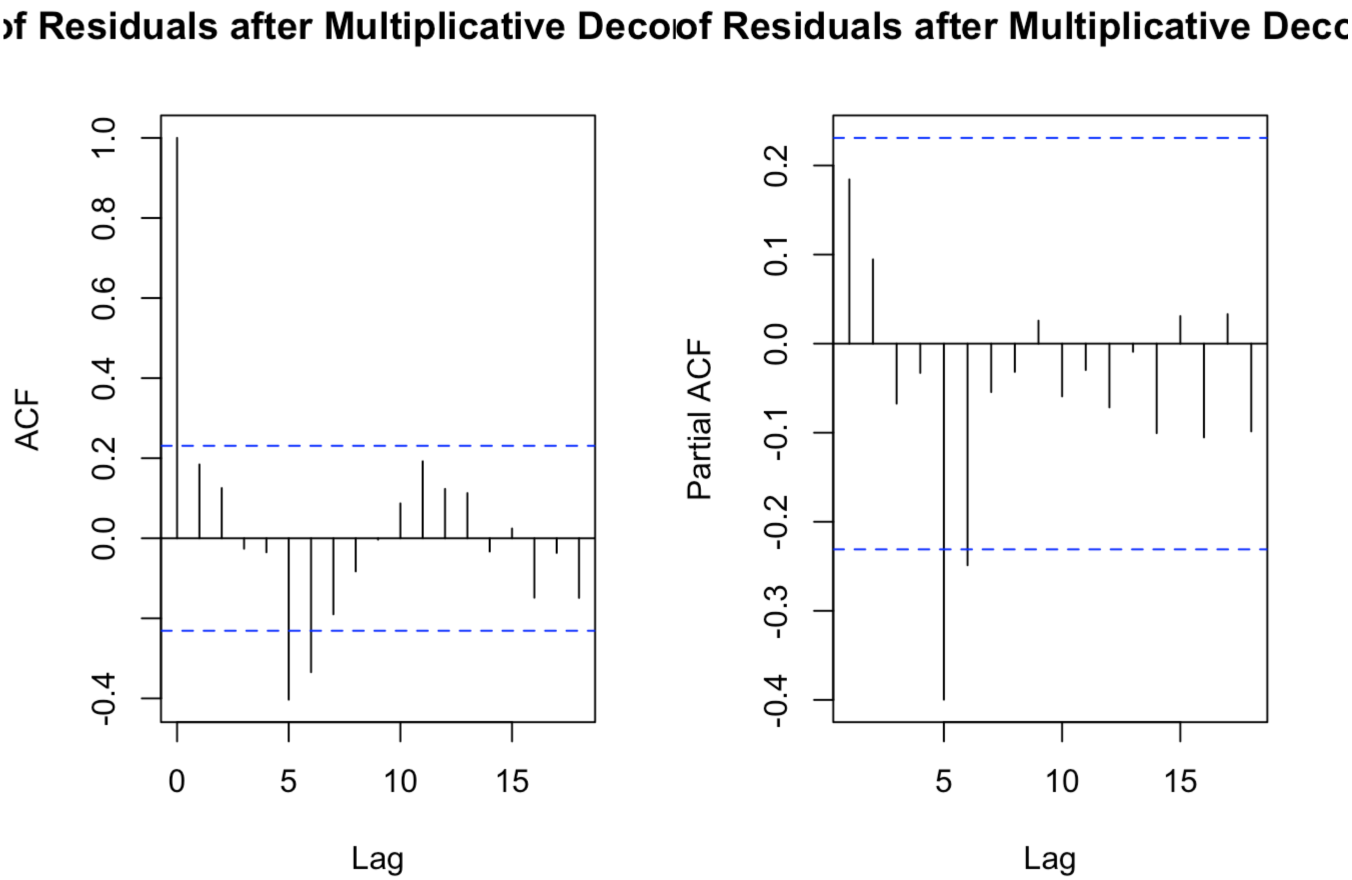
The ACF plot shows the autocorrelation at various lags for the residuals after decomposition. The spikes represent the correlation between observations at different lags, with the dashed blue line marking the significance threshold. Most of the spikes fall within the significance bounds, suggesting that the residuals do not exhibit significant autocorrelation at most lags. However, a few minor spikes at small lags indicate a bit of weak autocorrelation. Therefore, this ACF plot suggests that the model has done a reasonable job in capturing the dependencies in the data, though some minor autocorrelation might still exist at lower lags.

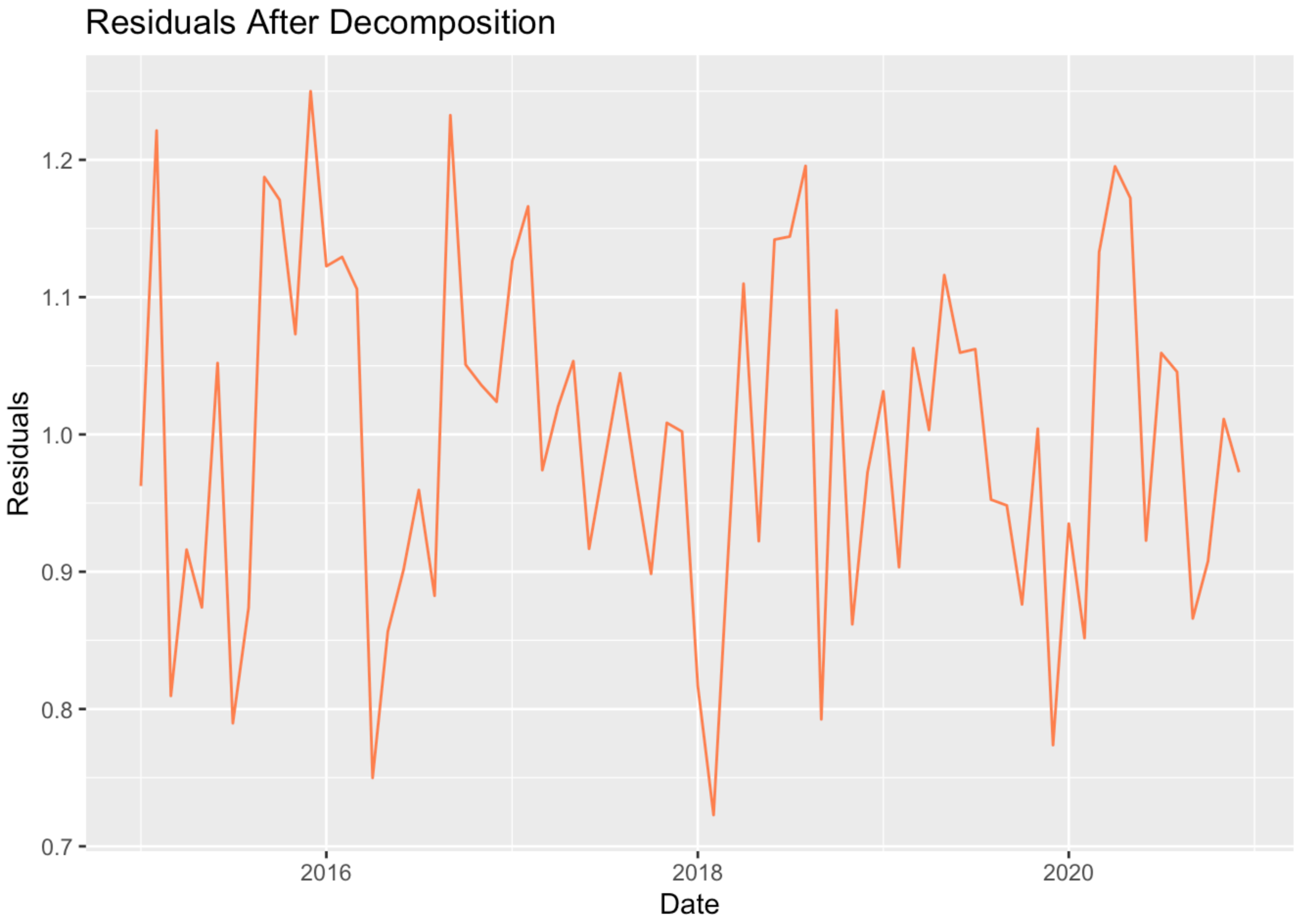
The PACF plot indicates the autocorrelation of residuals while controlling for the effects of previous lags. Similar to the ACF, most spikes fall within the significance bounds, suggesting no significant partial autocorrelation except at some very small lags. As a result, the PACF plot confirms that after decomposition, the residuals show little correlation, implying that most of the structure in the data has been captured by the additive decomposition.

Looking at the plot of the residuals after the additive decomposition, this time series plot of the residuals over time depicts how the residuals fluctuate up and down after STL decomposition. There is some variability that can be seen in the residuals, with peaks and troughs, but they generally appear to oscillate around zero. We know that with residuals after decomposition, there should be no pattern clearly observable, like white noise. In our plot, although fluctuations exist, there are no clear pattern in regard to seasonality or trend, suggesting that the decomposition has been fairly successful. However, the high variability indicates that there may still be some variance that has not been accounted for.

(b) Perform a multiplicative decomposition of your series. Remove the trend and seasonality, and comment on the ACF and PACF of the residuals (i.e., what is left after detrending and seasonally adjusting the series). Comment on the results.


Decomposition of multiplicative time series
observed
40
80
trend
65
70
75
80
seasonal
0.7
0.9
1.1
random
0.8
1.0
1.2
2015
2016
2017
2018
2019
2020
2021
2022
Time


of Residuals after Multiplicative Deco
of Residuals after Multiplicative Decc
ACF
1.0
0.8
0.6
0.4
0.2
0.0
-0.4
0
5
10
15
Lag
Partial ACF
0.2
0.1
0.0
-0.1
-0.2
-0.3
-0.4
5
10
15
Lag


Residuals After Decomposition
Residuals
1.2
1.1
1.0
0.9
0.8
0.7
2016
2018
2020
Date

The ACF plot shows that most of the residuals’ autocorrelation values lie within the confidence intervals, indicated by the blue dashed lines, which means that there is no significant autocorrelation left in the residuals. The first lag has a significant positive autocorrelation with it being close to 1, which is expected. After that, however, the autocorrelations mostly fall within the noise level. The absence of significant autocorrelation in the residuals suggests that the model captures the seasonal and trend components effectively.

The PACF plot shows the partial autocorrelation at different lags, and most of the values also fall within the confidence intervals, except for some spikes. A few partial autocorrelations, especially within the first few lags, slightly go beyond the bounds, indicating minor correlations that are not completely accounted for. However, the residuals seem to show that the model captures season and trend components more effectively for the most part.

Thus, the fact that no significant patterns seem to appear in both the ACF plot and the PACF plot means that the residuals can be considered as white noise, further validating the multiplicative decomposition.

The plot of the residuals after the multiplicative decomposition displays the residuals over time, which fluctuate randomly around the mean. The residuals do not show any apparent trends, which suggests that the multiplicative decomposition effectively removed the trend and seasonality. The variation in the residuals appears to be random, without any observable pattern, which further vaidates the performance of the model.

(c) Which decomposition is better, additive or multiplicative? Why?

The ACF plot for the additive decomposition shows significant autocorrelation at lag 1, but most residual autocorrelations fall within the confidence intervals for subsequent lags. This suggests that most of the patterns have been captured by the additive decomposition, although the residuals may still contain some structure. The PACF plot for the additive decomposition also shows a significant spike at lag 1 and some mild partial autocorrelations at other lags, but these are relatively small.

On the other hand, in regards to the multiplicative decomposition,the ACF plot has a similar number of significant lags, with about 3 specifically. Therefore, this suggests that in terms of the ACF and PACF plots, the additive decomposition and multiplicative decomposition have similar effectiveness in capturing the model. However, a different result can be seen when looking at the residuals of each decomposition.

For the residuals of each decomposition, the additive decomposition shows the residuals fluctuating more widely around 0, ranging from approximately -20 to +20. This suggests that the additive model is leaving behind some large variations that have not been fully explained by the trend and seasonal components. The residuals in the multiplicative model fluctuate within a much smaller range (around 0.7 to 1.2). This indicates that the multiplicative model has removed more of the variation in the data and produced more consistent residuals. In other words, since the range of residuals is much smaller, it suggests that the multiplicative decomposition has made a larger improvement in capturing the dynamics, leaving less room for significant unexplained variability.

Thus, from the output and our observations, it seems that the multiplicative decomposition is better.

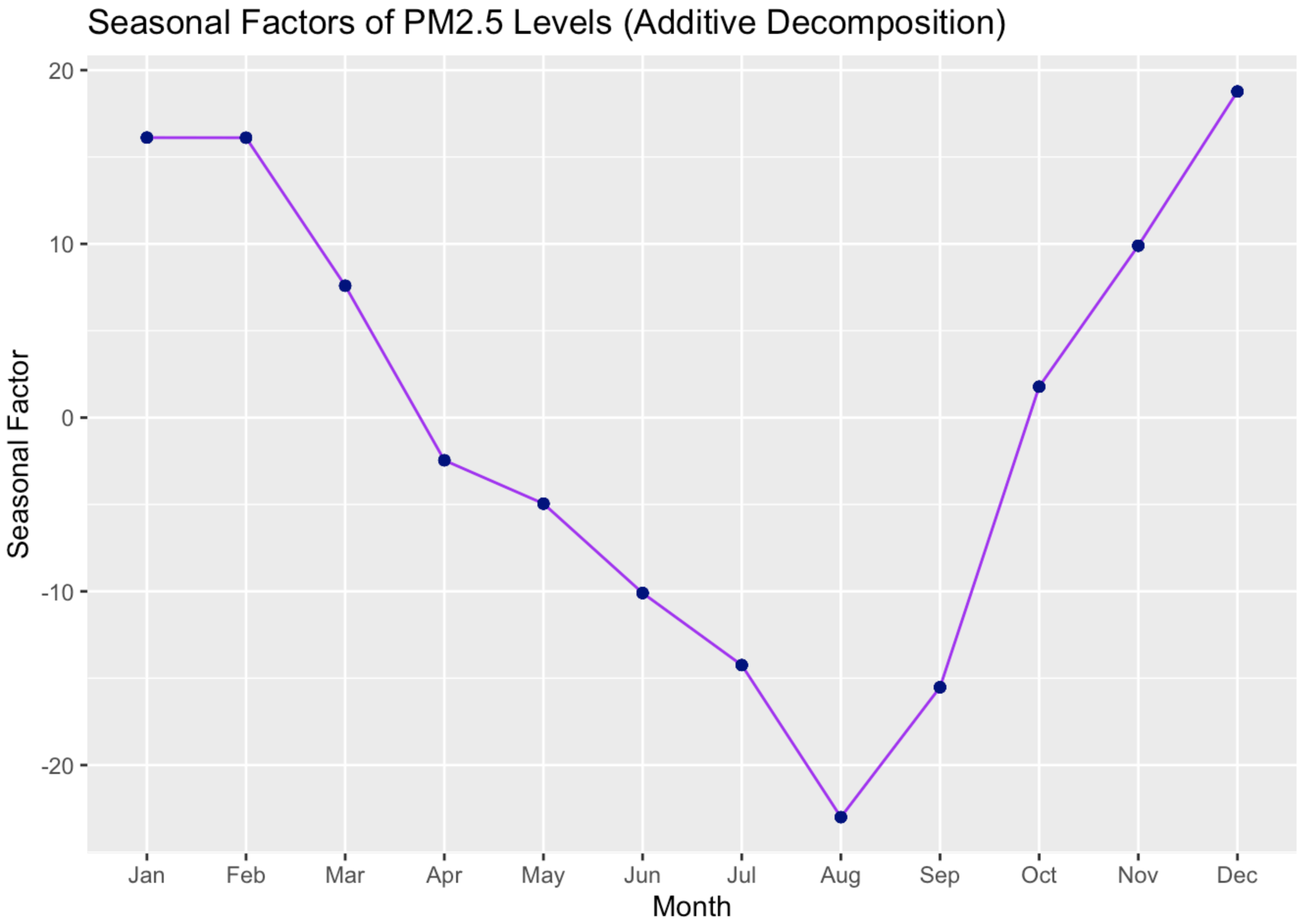
(d) Based on the two decompositions, and interpretation of the random components, would your models for the cycles be similar (additive vs. multiplicative) or very different? Why?

Based on the two decompositions and the interpretation of the random components, we can see that the large variations in the residuals for the additive model suggest that it may not adequately capture cycles if they exist. Additionally, the ACF and PACF plots suggest some remaining autocorrelations, particularly at higher lags, implying that some long-term cycles are not fully captured. Additive models are better suited when the magnitude of seasonal or cyclical patterns remains constant over time. On the other hand, the residuals from the multiplicative model show tighter fluctuations, indicating that the model has better captured the time series’ structure. As a result However, once again, the ACF and PACF plots suggest some remaining autocorrelations, particularly at higher lags, implying that some long-term cycles are not fully captured. But, this can be negligible due to the fact that the additive decomposition faces a similar problem for this case.

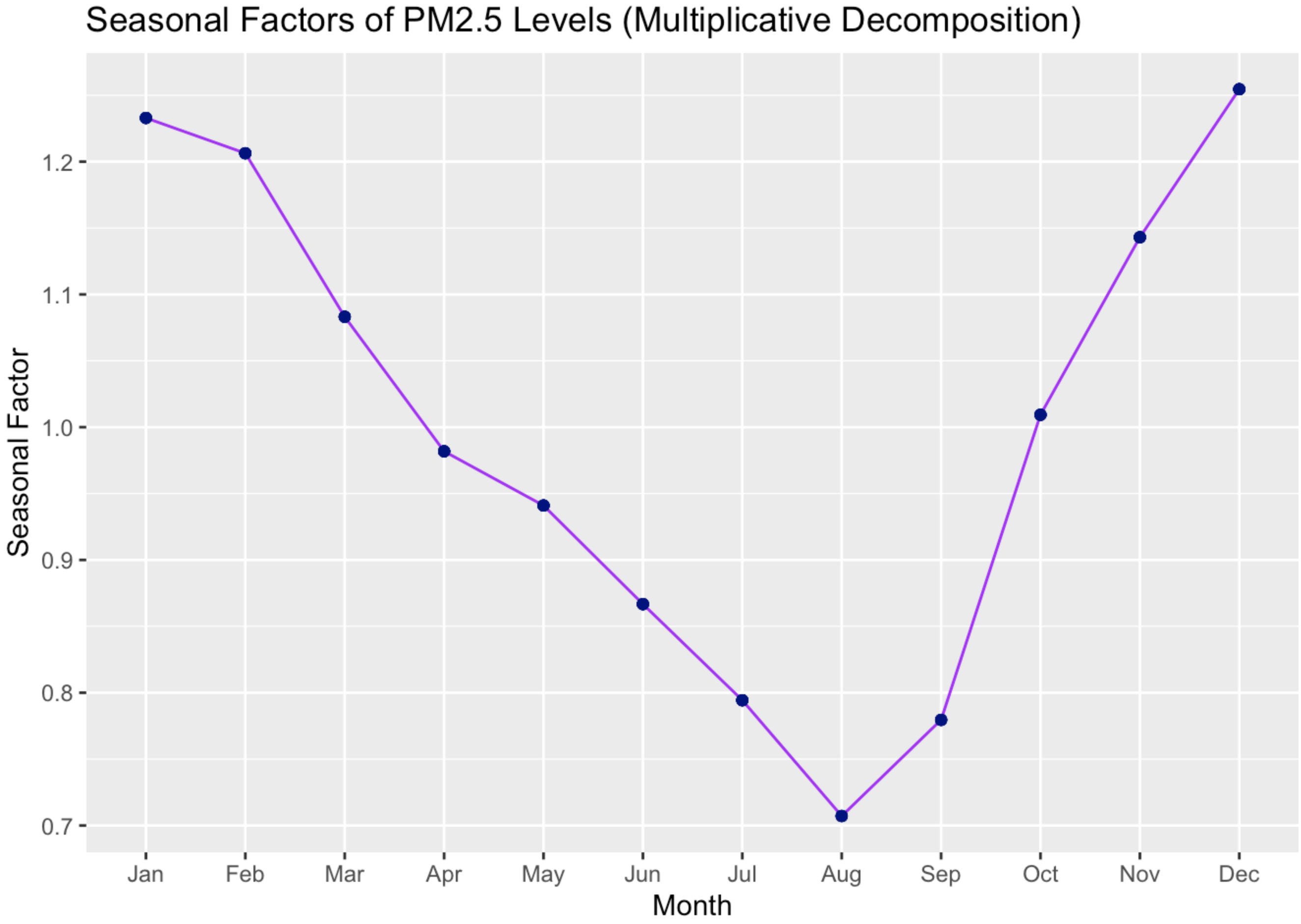
Overall, the additive decomposition struggles to capture cycles if the their magnitude changes over time, which can be seen in our plot of residuals suggesting unexplained cyclical variation. The model associatd with the multiplicative decomposition captures the cyclical behavior better. The smaller range of residuals and the observations seen in the ACF and PACF plots indicate that the multiplicative model is more appropriate for data with proportional cycles.

(e) Plot the seasonal factors and comment on the plot.

Let’s plot the seasonal factors for both the additive decomposition model and the multiplicative decomposition model.



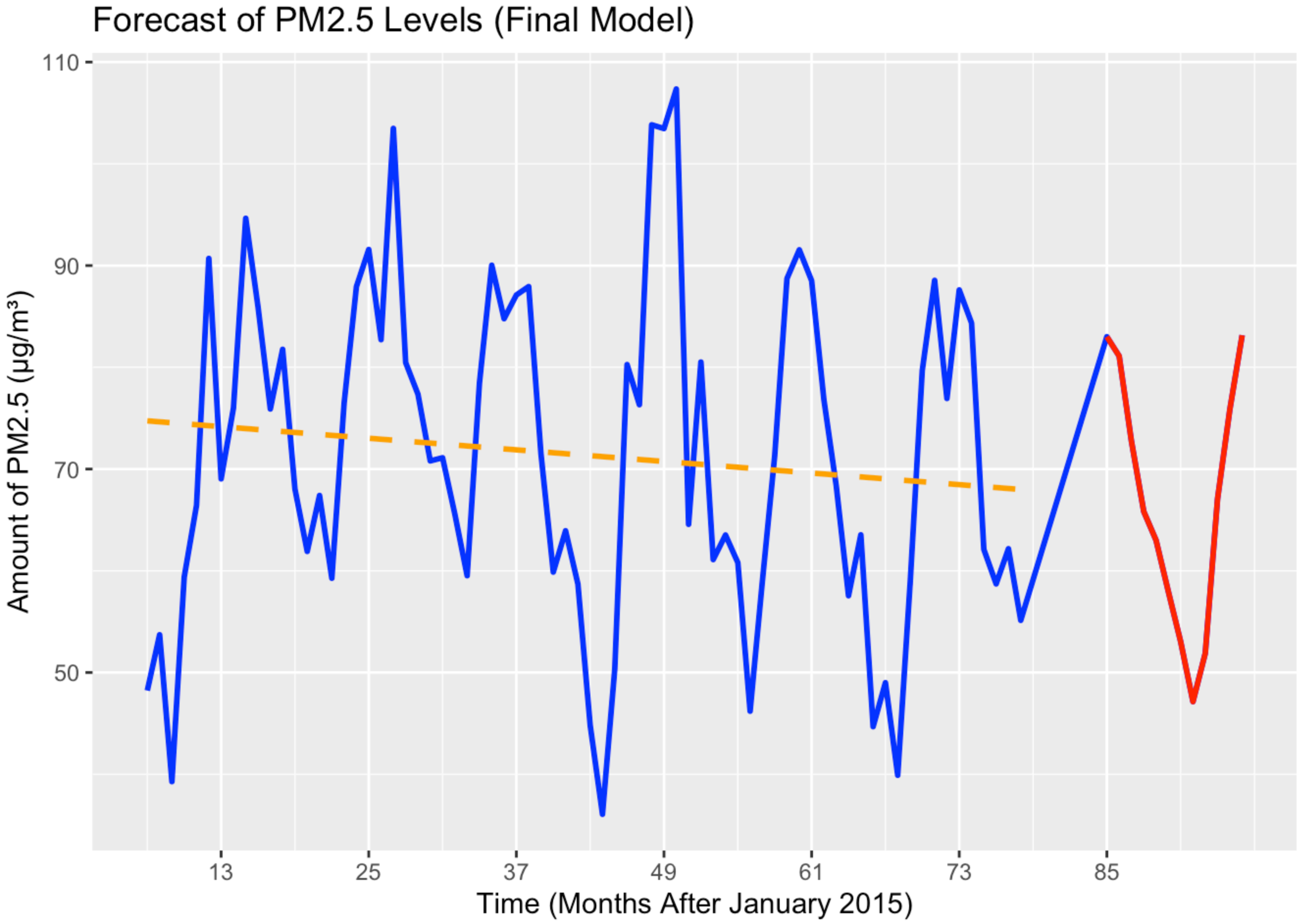
In the plot of the seasonal factors for additive decomposition above, positive factors in January, February, and December (around 20) suggest that PM2.5 levels are significantly higher during these winter months. They are at their lowest values during the summer months between June to September. The lowest value is in August, with a value of less than -20. Overall, there is a clear U-shaped pattern, with pollution levels dropping through summer and rising again towards the winter months. It is also important to note that for this plot being associated with the additive decomposition model, these values are expressed as the absolute values rather than any sort of relative values.



In the plot of the seasonal factors for multiplicative decomposition above, the seasonal factors are expressed as multipliers. Values above 1 indicate months with higher PM2.5 levels than average, while values below 1 represent months with lower levels. Factors above 1.2 indicate a strong increase in PM2.5 during winter, specifically for the monthso f January, February, and December. August has the lowest factor at approximately 0.75, implying that PM2.5 levels are around 25% below the average in that month. The seasonal factor plot for the multiplicative decomposition also has a trend that follows a similar U-shape, with pollution dipping in summer and rising during winter.

Therefore, both models indicate that winter months experience higher pollution, while summer months, especially August, show the lowest pollution.

(f) Based on your analysis thus far, choose a model that includes your preferred trend a seasonal model to forecast 12-steps ahead, and show the plot of the data, respective fit, and forecast.



III. Conclusions and Future Work

As we can see as a result from our analysis, the monthly averages of PM2.5 levels seem to have been oscillating over the years from 2015 to 2022. Therefore, there seems to be no linear trend but rather some type of periodic trend.

When using the full model that uses multiplicative decomposition, we can also see that there is seasonality in our data as winter months often have higher amounts of PM2.5 compared to the summer months, especially August. The full model also seems to improve the R^2 values that were seen to be extremely low in our original linear and nonlinear models.

Looking at the forecast plot, the oscillating pattern seems to continue with deeps and spikes here and there. As a result, it is likely that the full model accounts for the data sufficiently and shows a continuing pattern of less PM2.5 levels in certain periods of time (summer) and more PM2.5 levels in other periods of time (winter).

Even though the full model fits the data well, there is still improvements that could be done to further enhance it. We can see that from the ACF and PACF plots for the multiplicative decomposition that there were still significant lag values that can contribute towards certain periods of time not being explained by the model well enough. Therefore we should figure out ways such as accounting for even more lags in order to help improve the ACF and PACF plots. Additionally, recall the prediction interval from our original forecast plot being extremely large in range. We can improve this by allowing for further sampling and attempting to see if data from previous years can be obtained to increase the number of observations. This increase of the number of observations will help decrease the issue of the large range of the prediction interval.

IV. References

AirKorea, (https://www.airkorea.or.kr/eng/ (https://www.airkorea.or.kr/eng/))

Kaggle.com, (https://www.kaggle.com/datasets/calebreigada/south-korean-pollution (https://www.kaggle.com/datasets/calebreigada/south-korean-pollution))

California Air Resources Board, (https://ww2.arb.ca.gov/resources/inhalable-particulate-matter-and-health (https://ww2.arb.ca.gov/resources/inhalable-particulate-matter-and-health))

The International Trade Administration, (https://www.trade.gov/country-commercial-guides/south-korea-air-pollution-control (https://www.trade.gov/country-commercial-guides/south-korea-air-pollution-control))

City of Irvine, (https://www.cityofirvine.org/multicultural-and-international-affairs/nowon-gu (https://www.cityofirvine.org/multicultural-and-international-affairs/nowon-gu))

V. R Source Code

```
# LOAD DATA
# Load necessary libraries
library(tidyr)
library(dplyr)
library(lubridate)

# Read the data
pollution_data <- read.csv("south-korean-pollution-data.csv")

# Create a new data frame variable that we will later subset to account only for the Nowon-gu district in Seoul, and then switch Around the names of the City and District variables
copy_pollution_data <- pollution_data
colnames(copy_pollution_data)[11:12] = c("District", "City")
colnames(copy_pollution_data)[2:3] = c("Date", "pm2.5")

# Checking only the Nowon-Gu district in Seoul
nowon_pollution_data <- copy_pollution_data %>%
  filter(District == "Nowon-Gu", City == "Seoul")

# Changing the date format
nowon_pollution_data$Date <- as.Date(nowon_pollution_data$Date, format = "%Y/%m/%d")
nowon_pollution_data$Date <- format(nowon_pollution_data$Date, format = "%Y/%m")

# Use only monthly for cleaner analysis later
nowon_pollution_monthly_data <- nowon_pollution_data %>%
  group_by(Date) %>%
  summarize(`Average pm2.5` = mean(pm2.5, na.rm = TRUE))

# Use monthly data from January 2015 to the most recent data complete observation for the Nowon-gu District in Seoul (January 2022)
# The entire month of February 2022 was not completed at the time the data set was uploaded
nowon_pollution_monthly_data <- nowon_pollution_monthly_data[14:97,]

# Modify the date format one last time to translate into the time series plot later on
nowon_pollution_monthly_data$Date <- as.Date(paste0(nowon_pollution_monthly_data$Date,
"-01"), format = "%Y/%m-%d")

#1a: Model Forecast
library(ggplot2)
library(tseries)
library(forecast)

ggplot(nowon_pollution_monthly_data, aes(x = Date, y = `Average pm2.5`)) +
  geom_line(color = "darkorange2") +
  geom_hline(yintercept = mean(nowon_pollution_monthly_data$`Average pm2.5`), color = 'forestgreen', linetype = 'dashed') +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year") +  # Format x-axis to show only Years
  labs(title = "Average Amount of Fine Particular Matter (2.5 Micrometers or Less in Diameter) for the Month in the Nowon-gu District in Seoul",
       x = "Year",
       y = "Amount of PM2.5 (micrograms per cubic meter)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels

#1c: ACF and PACF
# Extracting the response variable
pm_data <- nowon_pollution_monthly_data$`Average pm2.5`

# ACF Plot
acf(pm_data, main = "ACF of Average PM2.5", lag.max = 20)

# 1d: Plots and Their Respective Fit
nowon_pollution_monthly_data$NumericDate <- as.numeric(nowon_pollution_monthly_data$Date - min(nowon_pollution_monthly_data$Date)) # Using Number of Days since the minimum date (January 1st, 2015), to be more exact

# Linear Model:
nowon_linear_model <- lm(`Average pm2.5` ~ Date, data = nowon_pollution_monthly_data)

# Add a sine term for periodic modeling
nowon_pollution_monthly_data$`Sine Term` <- sin(2 * pi * as.integer(nowon_pollution_monthly_data$Date))

# Fitting the Periodic Model
nowon_periodic_model <- lm(`Average pm2.5` ~ `Sine Term`, data = nowon_pollution_monthly_data)

# Plots and their respective fit

# Plot With Linear Fit
ggplot(nowon_pollution_monthly_data, aes(x = Date)) +
  geom_line(aes(y = `Average pm2.5`, color = "Observed Data Values"), size = 1) +  # Original data points
  geom_line(aes(y = nowon_linear_model$fitted.values, color = "Fitted Periodic Values"), size = 1, linetype = "solid") +  # Fitted periodic model line
  labs(title = "Average Amount of Fine Particular Matter (2.5 Micrometers or Less in Diameter) for the Month in the Nowon-gu District in Seoul",
       x = "Year",
       y = "Amount of PM2.5 (micrograms per cubic meter)",
       color = "Value Types")

# Plot With Periodic Fit
ggplot(nowon_pollution_monthly_data, aes(x = Date)) +
  geom_line(aes(y = `Average pm2.5`, color = "Observed Data Values"), size = 1) +  # Original data points
  geom_line(aes(y = nowon_periodic_model$fitted.values, color = "Fitted Periodic Values"), size = 1, linetype = "solid") +  # Fitted periodic model line
  labs(title = "Average Amount of Fine Particular Matter (2.5 Micrometers or Less in Diameter) for the Month in the Nowon-gu District in Seoul",
       x = "Year",
       y = "Amount of PM2.5 (micrograms per cubic meter)",
       color = "Value Types")

# 1e: Residuals
# Linear Model
ggplot(nowon_pollution_monthly_data, aes(x = nowon_linear_model$fitted.values, y = nowon
_linear_model$residuals)) +
  geom_point(color = "navy") +
  geom_line(color = "skyblue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red", lwd = 1) +
  labs(title = "Residuals vs Fitted for Linear Model",
       x = "Fitted Values",
       y = "Residuals")

# Nonlinear Model
ggplot(nowon_pollution_monthly_data, aes(x = nowon_periodic_model$fitted.values, y = now
on_periodic_model$residuals)) +
  geom_point(color = "navy") +
  geom_line(color = "skyblue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red", lwd = 1) +
  labs(title = "Residuals vs Fitted for Periodic (Nonlinear) Model",
       x = "Fitted Values",
       y = "Residuals")

# 1f: Histograms
# Linear Model
ggplot(nowon_pollution_monthly_data) +
  geom_histogram(aes(x = nowon_linear_model$residuals), bins = 40, fill = "pink", color
= "maroon") +
  labs(title = "Histogram of Residuals for the Linear Model",
      x = "Value",
      y = "Frequency")

# Nonlinear Model
ggplot(nowon_pollution_monthly_data) +
  geom_histogram(aes(x = nowon_periodic_model$residuals), bins = 40, fill = "pink", colo
r = "maroon") +
  labs(title = "Histogram of Residuals for the Periodic (Nonlinear) Model",
      x = "Value",
      y = "Frequency")

# 1g: Summary Statistics

library(ggfortify)

# Linear Model
summary(nowon_linear_model)
autoplot(nowon_linear_model)

# Nonlinear Model
summary(nowon_periodic_model)
autoplot(nowon_periodic_model)
```

```
#1h: AIC and BIC Values
# Data Frame
nowon_aic <- AIC(nowon_linear_model, nowon_periodic_model)
nowon_bic <- BIC(nowon_linear_model, nowon_periodic_model)
nowon_aic_bic_df <- data.frame(nowon_aic[,2], nowon_bic[,2], row.names = c('Linear', 'Periodic (Nonlinear)'))
colnames(nowon_aic_bic_df) <- c("AIC", "BIC")
nowon_aic_bic_df

#1i: Forecast Plot
# Forecast h-steps (12 months ahead)
h <- 12  # Set the forecast horizon to 12 months

# Create future time points for forecasting (months since the last observed date)
future_months <- date(c("2022-02-01", "2022-03-01", "2022-04-01", "2022-05-01", "2022-06-01", "2022-07-01",
                        "2022-08-01", "2022-09-01", "2022-10-01", "2022-11-01", "2022-12-01", "2023-01-01"))

# Generate sine and cosine terms for future months (to fit the periodic model)
future_sine_term <- sin(2 * pi * as.integer(future_months))

# Create a data frame for future months
nowon_future_data <- data.frame(Date = future_months, `Sine Term` = future_sine_term)
colnames(nowon_future_data) <- c("Date", "Sine Term")

# Forecast the future values using the periodic model
forecasted_values <- predict(nowon_periodic_model, newdata = nowon_future_data, interval = "predict", level = 0.95)

# Create a new data frame for the forecasted results (with prediction intervals)
nowon_forecast_df <- data.frame(
  Date = seq.Date(from = max(nowon_pollution_monthly_data$Date) + months(1),
                  by = "month", length.out = h),
  `Average pm2.5` = NA,
  `Forecast` = forecasted_values[, "fit"],
  Lower = forecasted_values[, "lwr"],
  Upper = forecasted_values[, "upr"]
)

# Combine with the original data for visualization
nowon_combined_data <- rbind(
  data.frame(Date = nowon_pollution_monthly_data$Date,
             `Average pm2.5` = nowon_pollution_monthly_data$`Average pm2.5`, Forecast = NA, Lower = NA, Upper = NA),
  nowon_forecast_df
)
colnames(nowon_combined_data)[2] <- "Average pm2.5"

# Plot the observed data and forecast with prediction intervals
ggplot(nowon_combined_data, aes(x = Date)) +
  geom_line(aes(y = `Average pm2.5`), color = "blue", na.rm = TRUE) +  # Observed data
  geom_line(aes(y = Forecast), color = "red", linetype = "solid", na.rm = TRUE) +  # For
ecasted values
  geom_ribbon(aes(ymin = Lower, ymax = Upper), fill = "darkgray", alpha = 0.5, na.rm = T
RUE) +  # Prediction Intervals
  labs(title = "Forecast Ahead for Average Amount of PM2.5 in the Nowon-gu District in S
eoul for 12 Months Ahead with Prediction Intervals",
       x = "Year",
       y = "Amount of PM2.5 (micrograms per cubic meter)")

#2a: Additive Decomposition
# Convert the data to a time series object with monthly frequency
pm25_ts <- ts(nowon_pollution_monthly_data$`Average pm2.5`,
              frequency=12,
              start=c(year(min(nowon_pollution_monthly_data$Date)), month(min(nowon_poll
ution_monthly_data$Date))))

# Perform STL decomposition (Seasonal-Trend Decomposition using LOESS)
pm25_decomp_add <- stl(pm25_ts, s.window = "periodic")

# Plot the decomposition
plot(pm25_decomp_add, col = c("darkorange3"))

# Extract the remainder (residuals) after removing trend and seasonality
ts_residuals_decomp <- pm25_decomp_add$time.series[, "remainder"]

# ACF and PACF of the residuals
par(mfrow = c(1, 2))  # Set up plotting area for side-by-side plots
acf(ts_residuals_decomp, main = "ACF of Residuals after Decomposition")
pacf(ts_residuals_decomp, main = "PACF of Residuals after Decomposition")

# Reset plotting area
par(mfrow = c(1, 1))

# Plot residuals to inspect
ggplot(data.frame(Date = seq.Date(from = min(nowon_pollution_monthly_data$Date),
                                  by = "month",
                                  length.out = length(ts_residuals_decomp)),
                  Residuals = ts_residuals_decomp), aes(x = Date, y = Residuals)) +
  geom_line(color = "coral") +
  labs(title = "Residuals After Decomposition",
       x = "Date",
       y = "Residuals")

#2b: Multiplicative Decomposition
# Convert the data to a time series object with monthly frequency
pm25_ts <- ts(nowon_pollution_monthly_data$`Average pm2.5`,
              frequency=12,
              start=c(year(min(nowon_pollution_monthly_data$Date)), month(min(nowon_poll
ution_monthly_data$Date))))

# Perform Multiplicative Decomposition
```

```
pm25_decomp_mult <- decompose(pm25_ts, type = "multiplicative")

# Plot the decomposition
plot(pm25_decomp_mult, col = "darkorange3")

# Extract the remainder (residuals) after removing trend and seasonality
residuals_mult <- pm25_decomp_mult$random
residuals_mult <- residuals_mult[!is.na(residuals_mult)]

# ACF and PACF of the residuals
par(mfrow = c(1, 2))  # Set up plotting area for side-by-side plots
acf(residuals_mult, main = "ACF of Residuals after Multiplicative Decomposition")
pacf(residuals_mult, main = "PACF of Residuals after Multiplicative Decomposition")

# Reset plotting area
par(mfrow=c(1, 1))

# Plot residuals to inspect
ggplot(data.frame(Date = seq.Date(from = min(nowon_pollution_monthly_data$Date),
                                by = "month",
                                length.out = length(residuals_mult)),
                  Residuals = residuals_mult), aes(x = Date, y = Residuals)) +
  geom_line(color = "coral") +
  labs(title = "Residuals After Decomposition",
       x = "Date",
       y = "Residuals")

#2e: Seasonal Plots
# Additive Decomposition Model
# Extract the seasonal component
nowon_add_seasonal_factors <- pm25_decomp_add$time.series[, "seasonal"]

# Prepare the data for plotting
nowon_add_seasonal_factors_data <- data.frame(
  Month = factor(rep(month.abb, length.out = length(nowon_add_seasonal_factors)), levels
= month.abb),  # Get month abbreviations
  Seasonal = as.numeric(nowon_add_seasonal_factors)
)

# Plot the seasonal factors (STL Additive Decomposition)
ggplot(nowon_add_seasonal_factors_data, aes(x = Month, y = Seasonal)) +
  geom_line(group = 1, color = "purple") +
  geom_point(color = "navy") +
  labs(title = "Seasonal Factors of PM2.5 Levels (Additive Decomposition)",
       x = "Month",
       y = "Seasonal Factor")

# Multiplicative Decomposition Model
# Extract the seasonal component
nowon_mult_seasonal_factors <- pm25_decomp_mult$seasonal

# Plot the seasonal factors
```

```
nowon_mult_seasonal_plot_data <- data.frame(
  Month = factor(rep(month.abb, length.out = length(nowon_mult_seasonal_factors)), levels = month.abb),  # Get month abbreviations
  Seasonal = as.numeric(nowon_mult_seasonal_factors)
)

ggplot(nowon_mult_seasonal_plot_data, aes(x = Month, y = Seasonal)) +
  geom_line(group = 1, color = "purple") +
  geom_point(color = "navy") +
  labs(title = "Seasonal Factors of PM2.5 Levels (Multiplicative Decomposition)",
       x = "Month",
       y = "Seasonal Factor")

#2f: Final Forecast Plot
# Remove NA values from the time series data
pm25_ts_clean <- na.omit(pm25_ts)

# Decompose the time series (multiplicative)
pm25_decomp_mult2 <- decompose(pm25_ts_clean, type = "multiplicative")

# Extract components
trend_component_mult <- pm25_decomp_mult$trend
seasonal_component_mult <- pm25_decomp_mult$seasonal

# Create a data frame for the trend component
trend_data_mult <- data.frame(
  Time = 1:length(trend_component_mult),
  Trend = trend_component_mult
)

# Fit a linear model to the trend component (removing NA values)
trend_model_mult <- lm(Trend ~ Time, data = na.omit(trend_data_mult))

# Forecast future trend values for 12 periods ahead
future_time_mult <- (length(trend_component_mult) + 1):(length(trend_component_mult) + 12)
future_trend_mult <- predict(trend_model_mult, newdata = data.frame(Time = future_time_mult))

# Extract seasonal factors for the last year (to forecast)
seasonal_factors_mult <- seasonal_component_mult[(length(seasonal_component_mult) - 11):length(seasonal_component_mult)]

# Ensure the seasonal factors have the same length as forecast periods
if (length(seasonal_factors_mult) < 12) {
  seasonal_factors_mult <- rep(seasonal_factors_mult, length.out = 12)  # Replicate if less than needed
}

# Create forecast by multiplying future trend by seasonal factors
forecast_values_mult <- future_trend_mult * seasonal_factors_mult
```

```
# Combine historical and forecasted data for plotting
# Remove NA values (NA values are present due to the multiplicative nature)
forecast_data_mult <- data.frame(
  Time = c(7:78, future_time_mult),
  PM2.5 = c(pm25_ts_clean[7:78], forecast_values_mult),
  Fitted = c(trend_model_mult$fitted.values, rep(NA, 12))  # Fitted values with NA for f
orecast period
)

# Plot the original data, the fitted trend, and the forecast
ggplot(forecast_data_mult, aes(x = Time)) +
  geom_line(aes(y = PM2.5), color = "blue", size = 1, na.rm = TRUE) +  # Original data
  geom_line(aes(y = Fitted), color="orange", linetype = "dashed", size = 1, na.rm = TRU
E) +  # Fitted trend
  geom_line(aes(y = c(rep(NA, length(pm25_ts_clean[7:78])), forecast_values_mult)),
            color = "red", size = 1, na.rm = TRUE) +  # Forecast values
  labs(title = "Forecast of PM2.5 Levels (Final Model)",
       x = "Time (Months After January 2015)",
       y = "Amount of PM2.5 (µg/m³)") +
  scale_x_continuous(breaks = seq(1, length(pm25_ts_clean) + 12, by = 12),
                          labels = seq(1, length(pm25_ts_clean) + 12, by = 12))
```