

# Final Project Exploratory Data Analysis

Aidan Frederick, Ben Walter, Kyle Maher

2025-11-12

## Packages

```
library(fpp3) # Time Series Plots and Forecasting
library(corrplot) # Correlation Plot
library(forecast) # BoxCox Transformation
library(scales) # Log Scale Tick Lables
set.seed(280) # Reprodicibility
```

## Setup

### Read Data

```
data <- readRDS("data/loaded/delhi.rds")
```

### Filter to Delhi

```
delhi_city <- data %>%
  filter(file_name == "DL008.csv")
```

### Determine Parameter Counts

```
observation_counts <- delhi_city %>%
  summarise(across(everything(), ~sum(!is.na(.)))) %>%
  pivot_longer(cols = everything()) %>%
  arrange(desc(value)) %>%
  mutate(percent_available = (value / nrow(delhi_city)) * 100)

observation_counts

# A tibble: 60 x 3
  name          value percent_available
  <chr>        <int>           <dbl>
  1
```

```

1 From Date      114635      100
2 To Date       114635      100
3 file_name     114635      100
4 CO (mg/m3)   48443       42.3
5 Benzene (ug/m3) 48136       42.0
6 NOx (ppb)    48007       41.9
7 Ozone (ppb)  47576       41.5
8 Toluene (ug/m3) 46987       41.0
9 PM2.5 (ug/m3) 46610       40.7
10 Xylene (ug/m3) 46564      40.6
# i 50 more rows

```

### Drop Parameters Available For Less than 2% of Hours, Rename Columns, & Filter Date Range

```

rare_parameters <- observation_counts %>%
  filter(percent_available < 2) %>%
  pull(name)

df <- delhi_city %>%
  select(-all_of(rare_parameters), -file_name, -`From Date`, -`Ozone (ppb)`) %>%
  rename(
    "datetime" = "To Date",
    "PM2.5" = "PM2.5 (ug/m3)",
    "PM10" = "PM10 (ug/m3)",
    "NO" = "NO (ug/m3)",
    "NO2" = "NO2 (ug/m3)",
    "NOx" = "NOx (ppb)",
    "CO" = "CO (mg/m3)",
    "Benzene" = "Benzene (ug/m3)",
    "Toluene" = "Toluene (ug/m3)",
    "Xylene" = "Xylene (ug/m3)"
  ) %>%
  as_tsibble(index = datetime) %>%
  filter(datetime >= as_datetime("2017-08-31 01:00:00"))

```

### Show Hourly Data Availability

```

df %>%
  as_tibble() %>%
  summarise(across(everything(), ~sum(!is.na(.)))) %>%
  pivot_longer(cols = everything()) %>%
  arrange(desc(value)) %>%
  mutate(percent_available = (value / nrow(df)) * 100)

```

```
# A tibble: 10 x 3
  name      value percent_available
  <chr>    <int>          <dbl>
1 datetime 48936           100
2 CO        44884           91.7
3 Toluene   44635           91.2
4 Benzene   44626           91.2
5 NOx       44583           91.1
6 Xylene    44212           90.3
7 PM2.5     43227           88.3
8 PM10      42937           87.7
9 NO2       42161           86.2
10 NO        41407           84.6
```

### Aggregate to Daily Series

```
daily_df <- df %>%
  index_by(date = as.Date(datetime)) %>%
  summarise(
    PM2.5 = mean(PM2.5, na.rm = TRUE),
    PM10 = mean(PM10, na.rm = TRUE),
    NO = mean(NO, na.rm = TRUE),
    NO2 = mean(NO2, na.rm = TRUE),
    NOx = mean(NOx, na.rm = TRUE),
    CO = mean(CO, na.rm = TRUE),
    Benzene = mean(Benzene, na.rm = TRUE),
    Toluene = mean(Toluene, na.rm = TRUE),
    Xylene = mean(Xylene, na.rm = TRUE)
  )
```

### Show Daily Availability

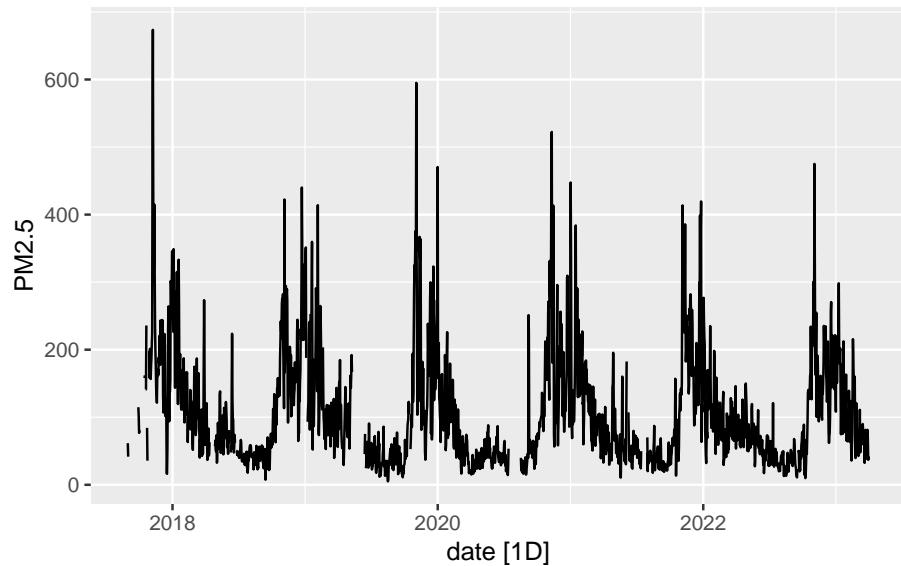
```
daily_df %>%
  as_tibble() %>%
  summarise(across(everything(), ~sum(!is.na(.)))) %>%
  pivot_longer(cols = everything()) %>%
  arrange(desc(value)) %>%
  mutate(percent_available = (value / nrow(daily_df)) * 100)

# A tibble: 10 x 3
  name      value percent_available
  <chr>    <int>          <dbl>
1 date      2040            100
2 CO        1927            94.5
3 NOx      1925            94.4
```

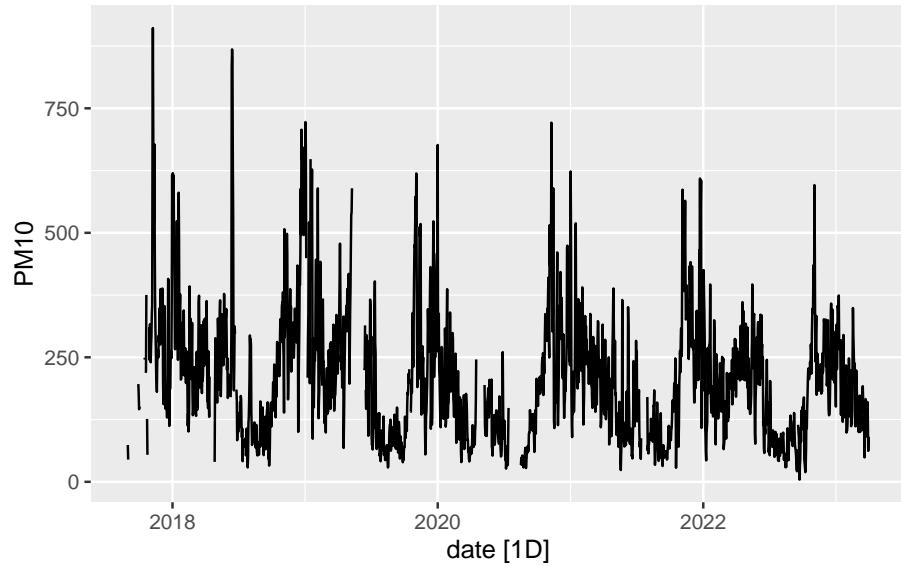
4	Benzene	1911	93.7
5	Toluene	1911	93.7
6	Xylene	1892	92.7
7	PM2.5	1884	92.4
8	PM10	1867	91.5
9	NO2	1841	90.2
10	NO	1818	89.1

## Plot Each Parameter

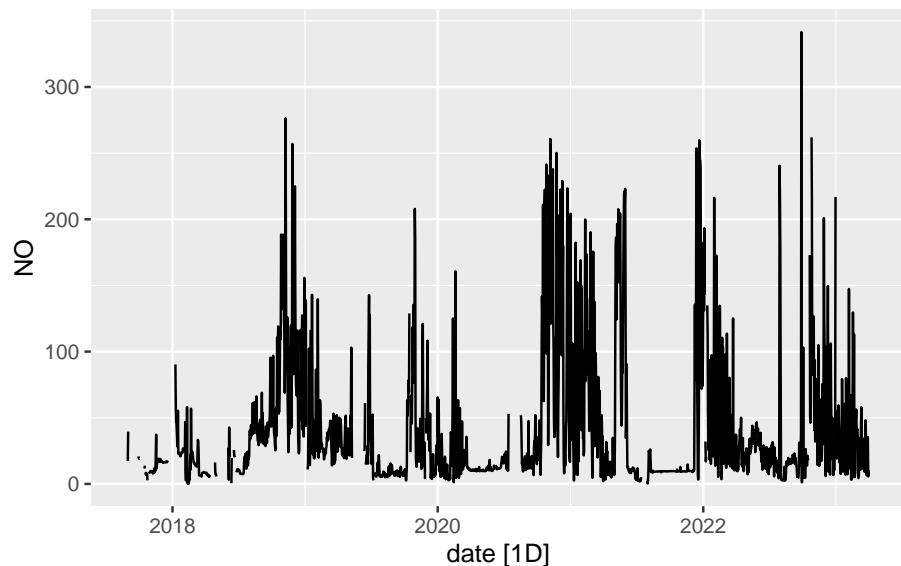
```
daily_df %>%
  autoplot(PM2.5)
```



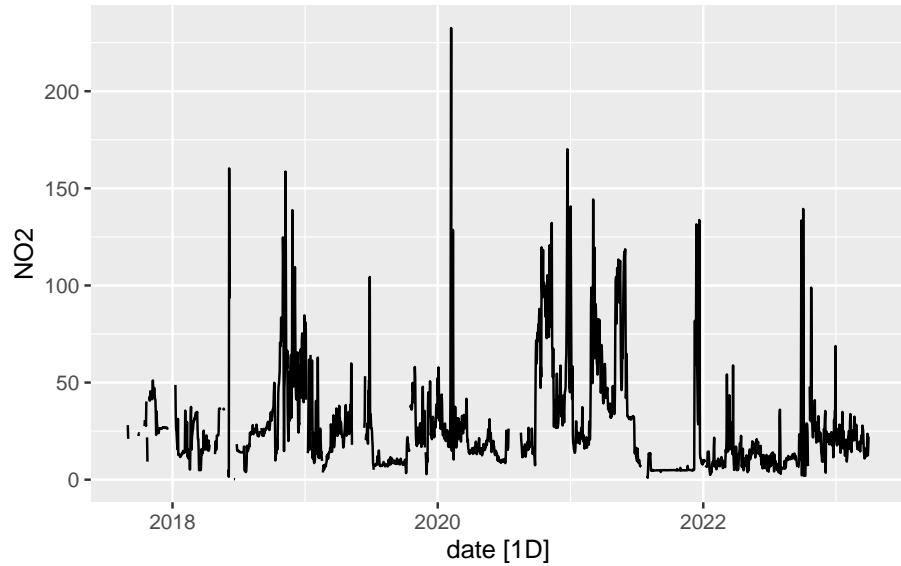
```
daily_df %>%
  autoplot(PM10)
```



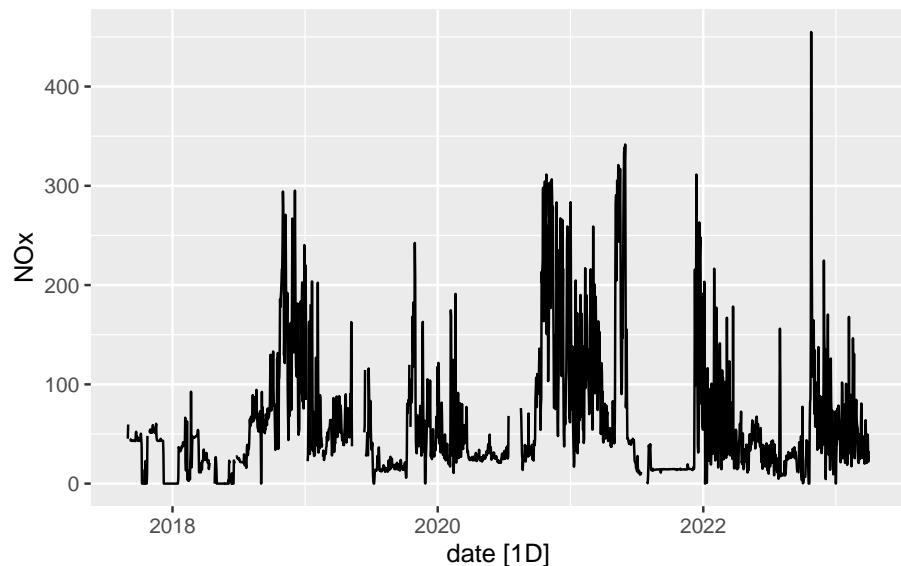
```
daily_df %>%
  autoplot(NO)
```



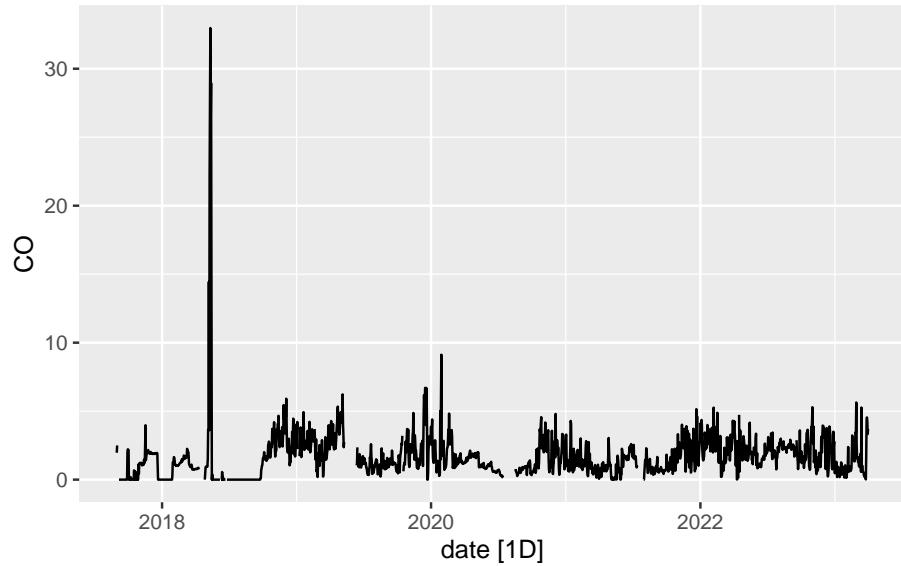
```
daily_df %>%
  autoplot(NO2)
```



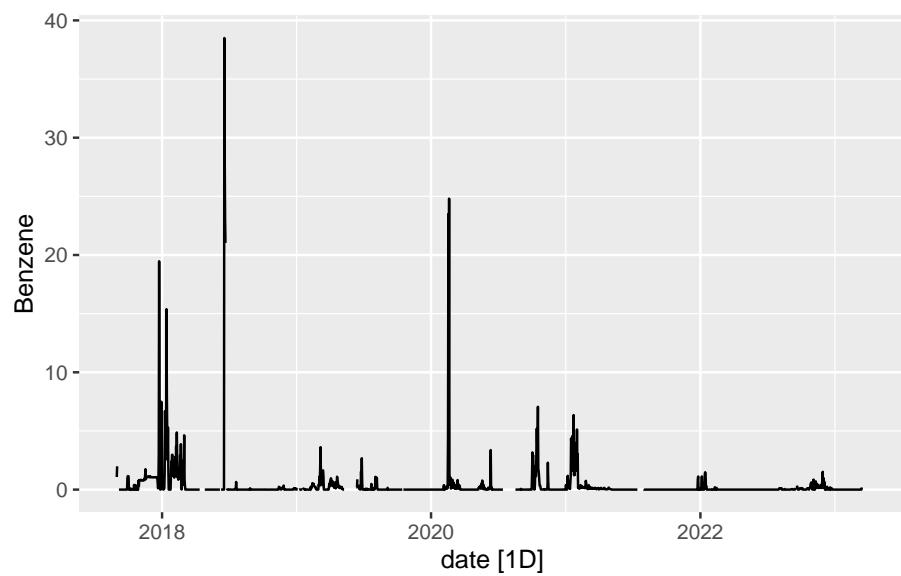
```
daily_df %>%
  autoplot(NOx)
```



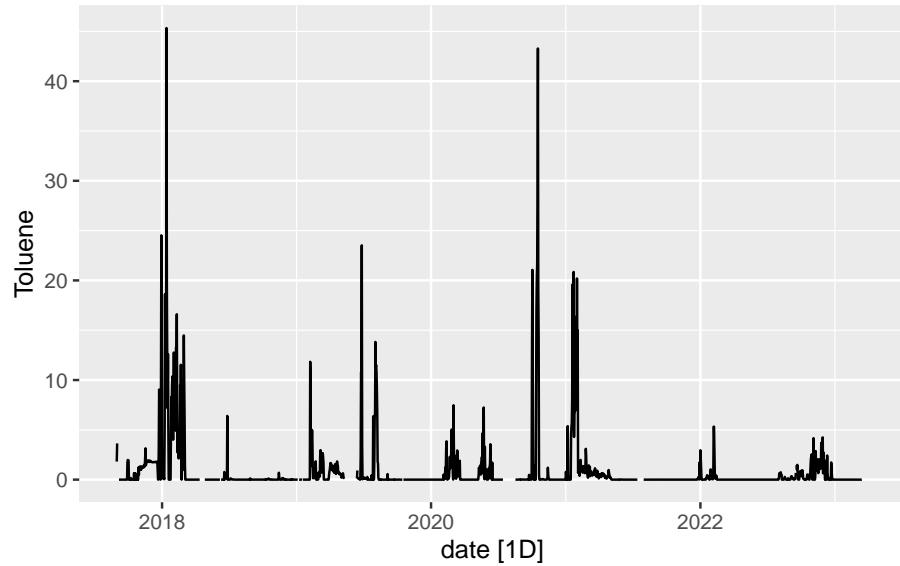
```
daily_df %>%
  autoplot(CO)
```



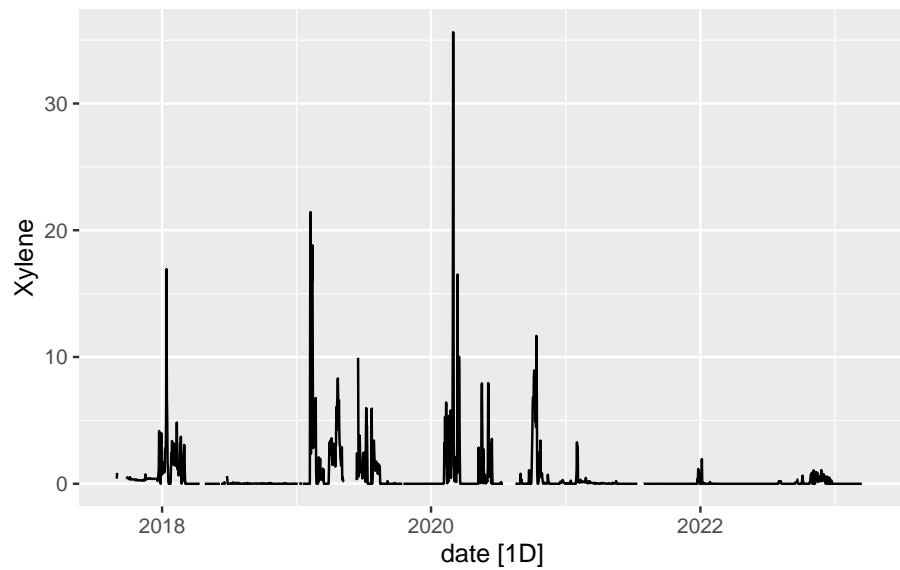
```
daily_df %>%
  autoplot(Benzene)
```



```
daily_df %>%
  autoplot(Toluene)
```



```
daily_df %>%
  autoplot(Xylene)
```

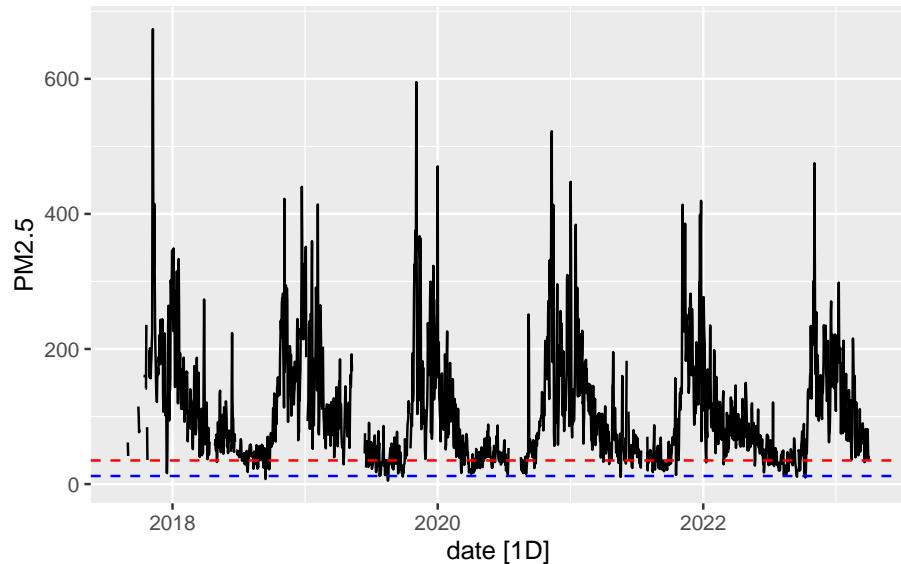


## Recomended Limit for PM2.5

```

df %>%
  index_by(date = as.Date(datetime)) %>%
  summarise(PM2.5 = mean(PM2.5, na.rm = TRUE)) %>%
  autoplot(PM2.5) +
  geom_hline(yintercept = 35, color = "red", linetype = "dashed") +
  geom_hline(yintercept = 12, color = "blue", linetype = "dashed")

```



The World Health Organization recommends PM2.5 levels below  $35\text{ug}/\text{m}^3$  for a daily average and below  $12\text{ug}/\text{m}^3$  for a yearly average. The daily and yearly recommended levels are plotted in red and blue respectively. Only during the summer months are the PM2.5 levels below the WHO recommended daily limit.

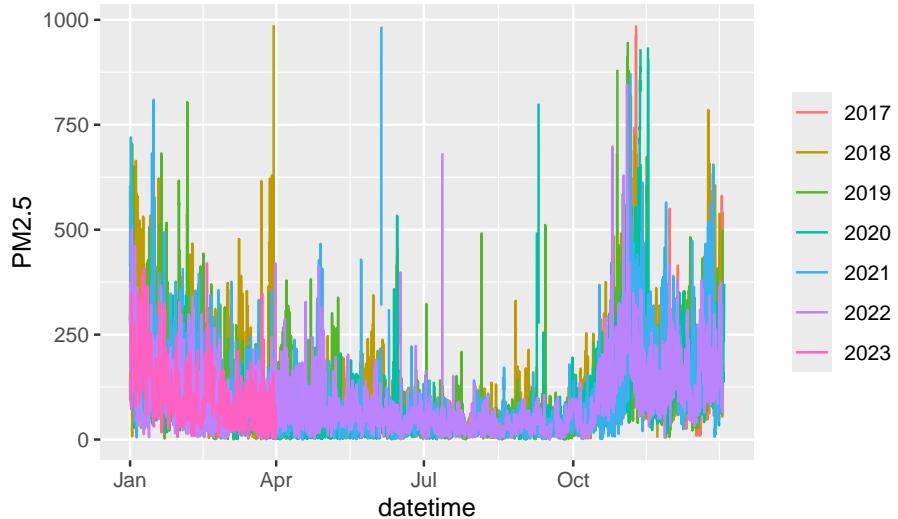
## Seasonality

```

gg_season(df, PM2.5, period = "1 year") +
  labs(title = "Yearly Seasonality")

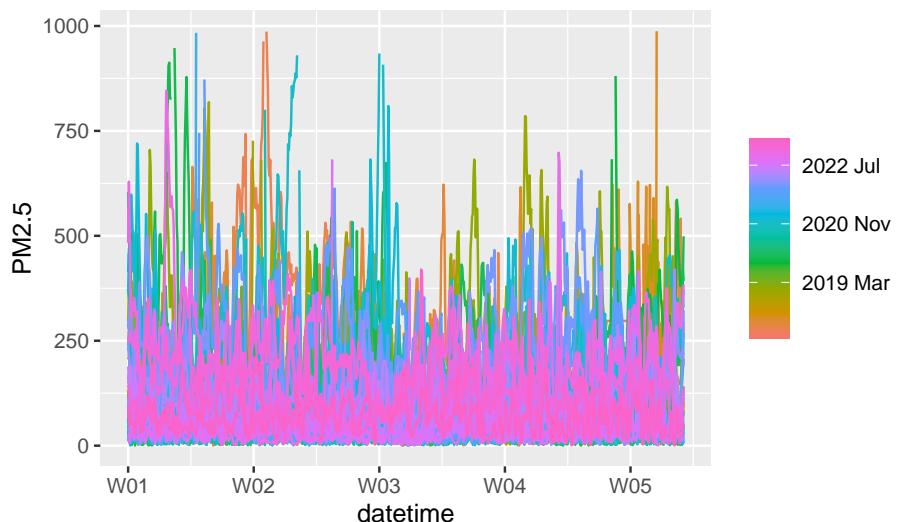
```

### Yearly Seasonality



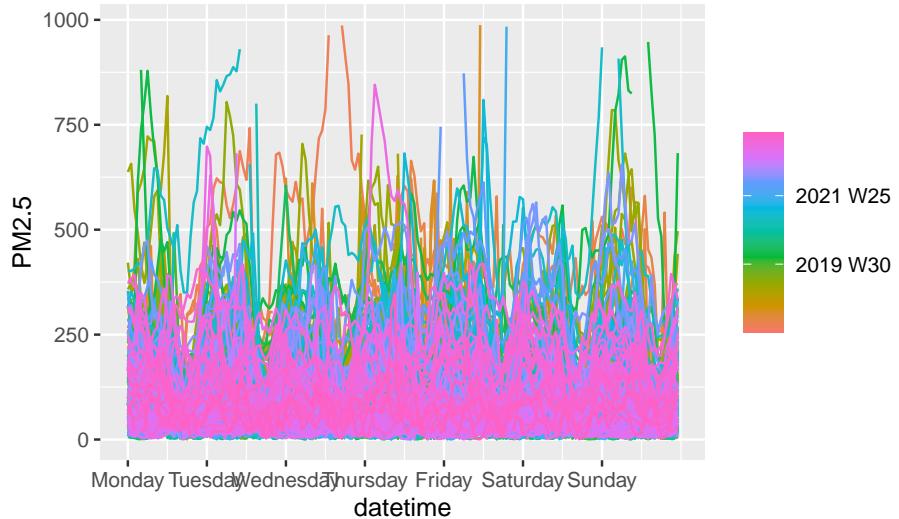
```
gg_season(df, PM2.5, period = "1 month") +  
  labs(title = "Monthly Seasonality")
```

### Monthly Seasonality



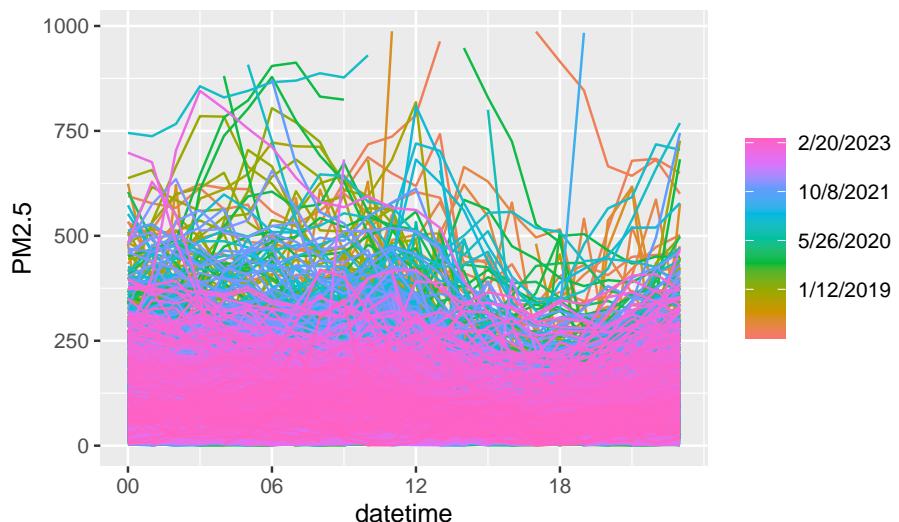
```
gg_season(df, PM2.5, period = "1 week") +  
  labs(title = "Weekly Seasonality")
```

### Weekly Seasonality



```
gg_season(df, PM2.5, period = "1 day") +  
  labs(title = "Daily Seasonality")
```

### Daily Seasonality



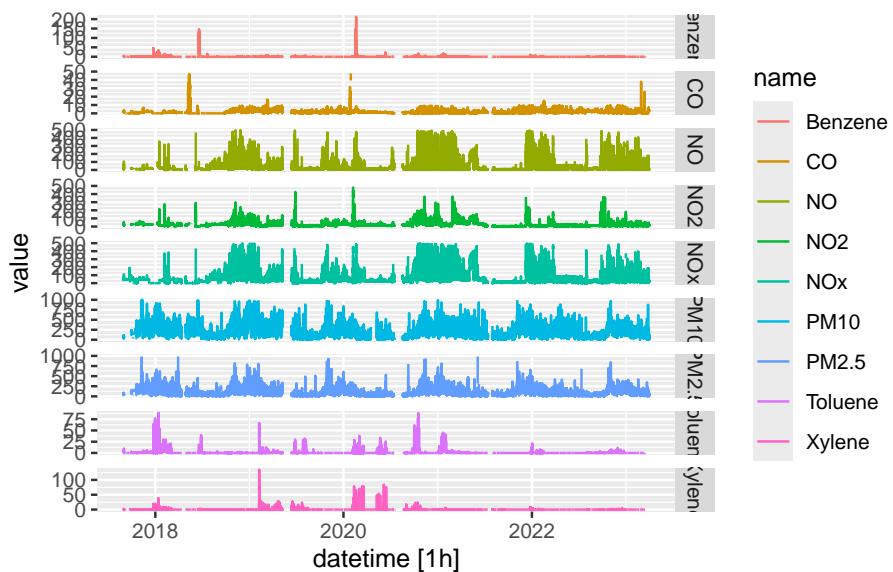
## Plot Parameters Together

```

df_long <- df %>% pivot_longer(c(2:10))

df_long %>% autoplot(value) +
  facet_grid(rows = vars(name), scales = "free_y")

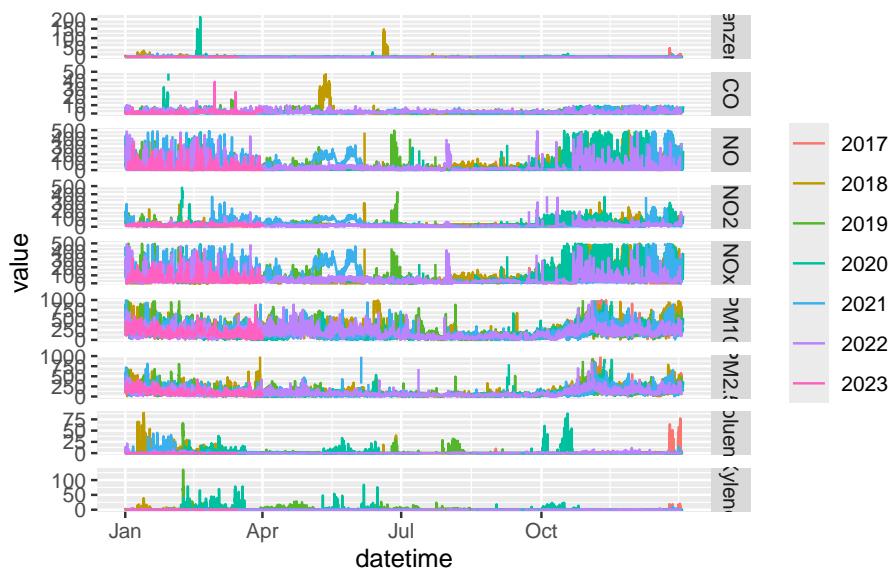
```



```

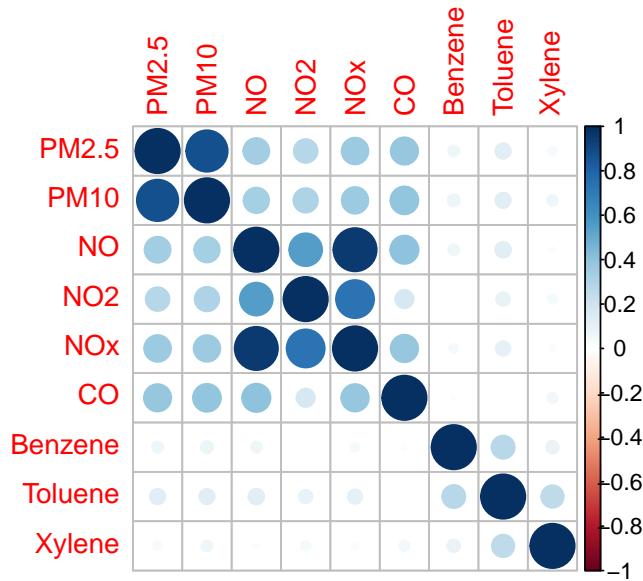
df_long %>% gg_season(value) +
  facet_grid(rows = vars(name), scales = "free_y")

```



## Parameter Correlations

```
M <- cor(x = df[, c(2:10)],
           y = df[, c(2:10)],
           use = "na.or.complete")
corrplot(M)
```



High concentrations of CO, NO, NO<sub>2</sub>, NOx, PM2.5, and PM10 generally occur at the same time. High concentrations of Benzene, Toluene, and Xylene generally occur at the same time.

## PM2.5 as a Stationary Time Series

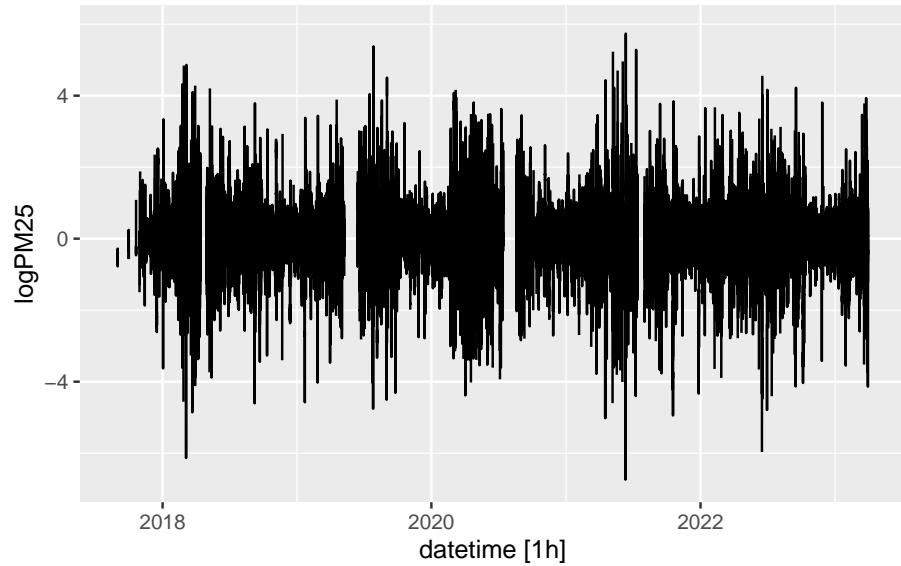
```
df$logPM25 <- log(df$PM2.5) %>%
  difference(24)

# Box-Cox Transformation
BoxCox.lambda(df$PM2.5)

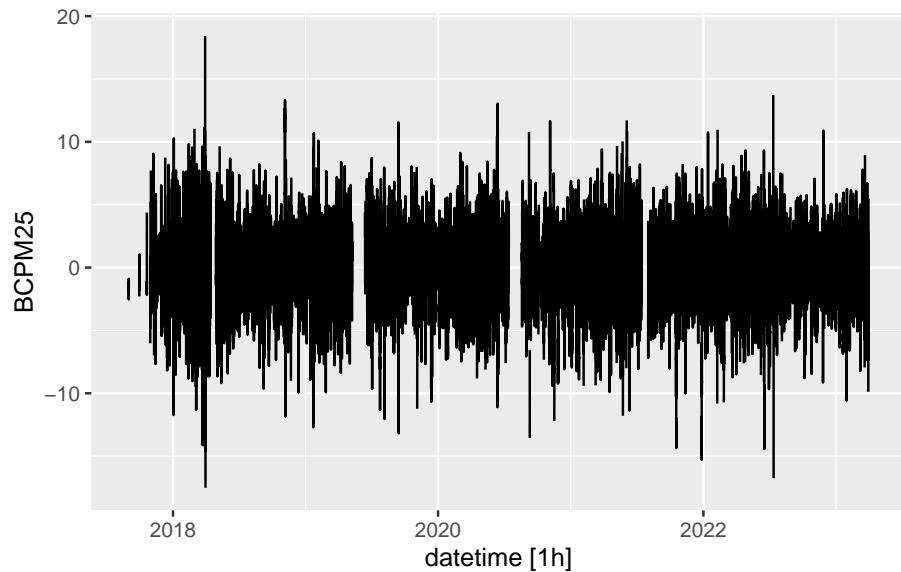
[1] 0.2959454

df$BCPM25 <- BoxCox(df$PM2.5, 0.2931797) %>%
  difference(24)

df %>% autoplot(logPM25)
```



```
df %>% autoplot(BCPM25)
```



Differencing PM2.5 centers the data, and using a log transform centers the data, giving a stationary time series.

## Check for Constant Variance

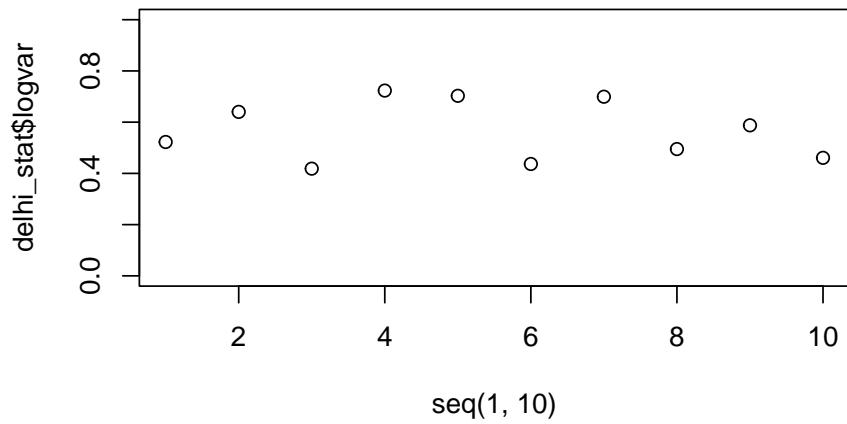
```
delhi_stat <- data.frame(logvar = numeric(10),
                           boxvar = numeric(10))
for (i in 1:10){
  delhi_stat[i,1] <- var(df$logPM25[
    floor(nrow(df)*0.1*(i-1)):floor(nrow(df)*0.1*i)
  ], na.rm = TRUE)

  delhi_stat[i,2] <- var(df$BCPM25[
    floor(nrow(df)*0.1*(i-1)):floor(nrow(df)*0.1*i)
  ], na.rm = TRUE)
}

delhi_stat
```

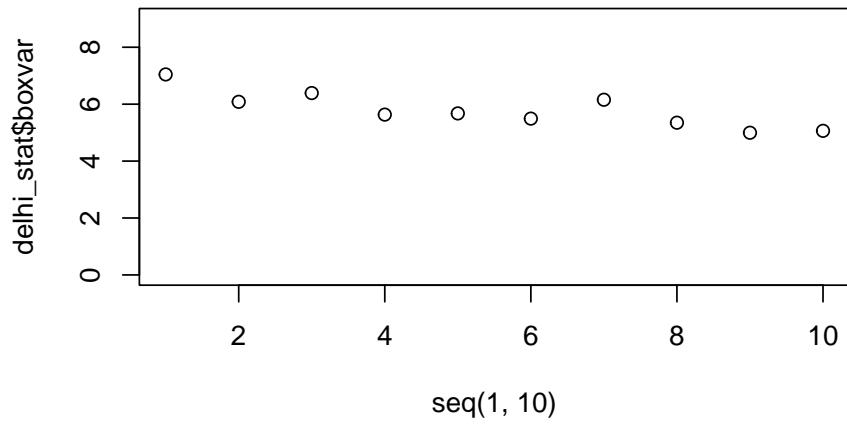
	logvar	boxvar
1	0.5225954	7.044642
2	0.6401363	6.082767
3	0.4184593	6.390390
4	0.7233174	5.632785
5	0.7029033	5.675282
6	0.4367190	5.492182
7	0.6993186	6.155172
8	0.4949893	5.349204
9	0.5876888	4.995905
10	0.4608093	5.062568

```
plot(seq(1,10),delhi_stat$logvar, ylim = c(0,1))
```



The Variance changes without transformation.

```
plot(seq(1,10),delhi_stat$boxvar, ylim = c(0,9))
```



The BoxCox transformation results in stable variances.

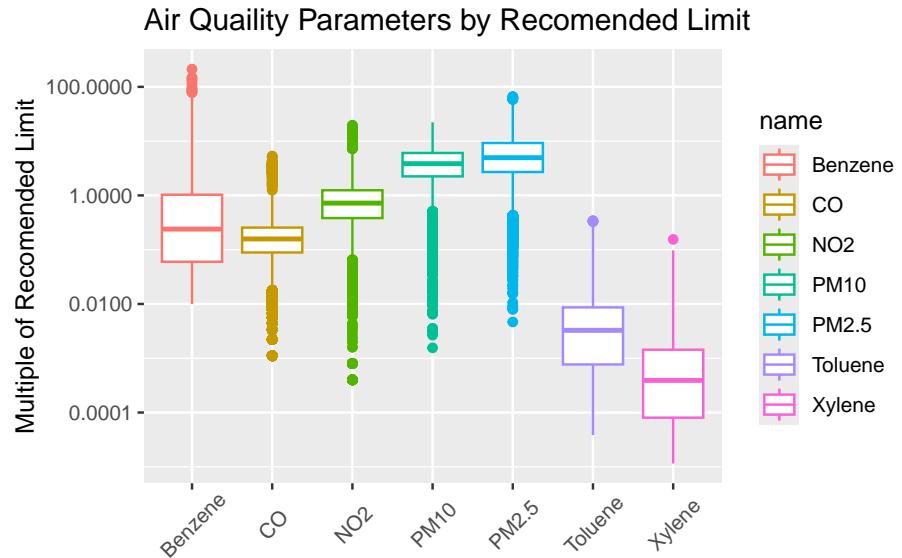
## World Health Organization Recomended Limits:

CO: 9 ppm  
PM<sub>10</sub>: 15 µg/m<sup>3</sup>  
PM<sub>2.5</sub>: 45 µg/m<sup>3</sup>  
NO : 25 µg/m<sup>3</sup>  
Benzene: As low as possible (<5 µg/m<sup>3</sup>)  
Toluene: 260 µg/m<sup>3</sup>  
Xylene: 870 µg/m<sup>3</sup>

## Scaled Parameters by Recomended Limits

```
df_scaled <- df %>%
  select(-NOx, -NO) %>%
  mutate(
    PM2.5 = PM2.5 / 15,
    PM10 = PM10 / 45,
    CO = CO / 9,
    NO2 = NO2 / 25,
    Benzene = Benzene / 1,
    Toluene = Toluene / 260,
    Xylene = Xylene / 870
  ) %>%
  pivot_longer(c(2:8))

ggplot(data = df_scaled) +
  aes(x = name, y = value, color = name) +
  geom_boxplot() +
  scale_y_log10(labels = label_number()) +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.6)) +
  labs(
    x = NULL,
    y = "Multiple of Recomended Limit",
    title = "Air Quaility Parameters by Recomended Limit"
  )
```



## Summary

Focus will be given to PM2.5 as it is a good indicator of overall air quality. It was found that PM2.5 is typically much above the recommended limits set by the World Health Organization. There is seasonality present, particularly at the yearly, daily, and hourly levels. Presence of PM2.5 is correlated with PM10, NO, NO<sub>2</sub>, NO<sub>x</sub>, and CO. While presence of Benzene, Toluene, and Xylene are not. It was found that a BoxCox transformation was able to transform the PM2.5 series to have constant variance. Future work will focus on creating various forecasting models for the PM2.5 series.