Final Project Exploratory Data Analysis

Aidan Frederick, Ben Walter, Kyle Maher

2025-11-14

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)  # Reproducibility
```

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 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

```r
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

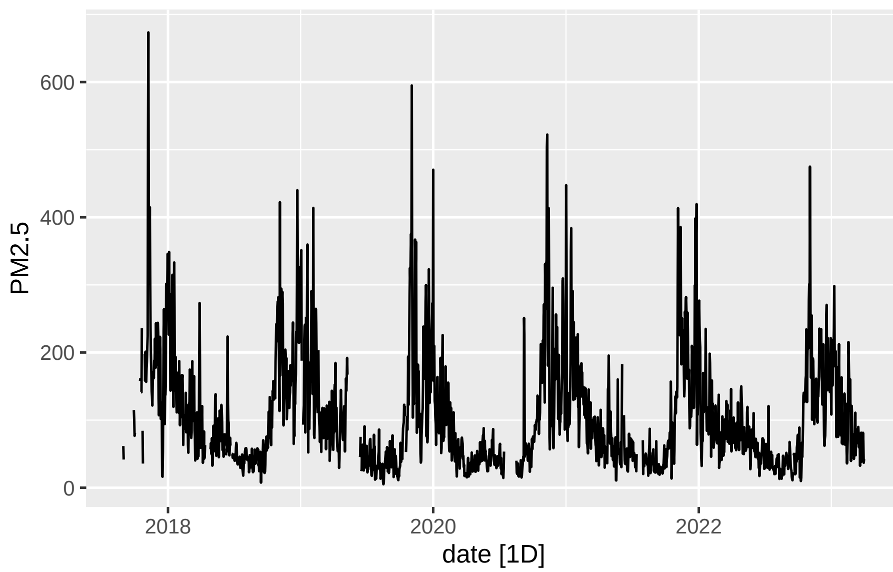
```r
daily_df %>%
  as_tibble() %>%
  summarise(across(everything(), ~sum(!is.na(.)))) %>%
  pivot_longer(cols = everything()) %>%
  arrange(desc(value)) %>%
  mutate(percent_available = (value / nrow(daily_df)) * 100)
```

```
# 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
```

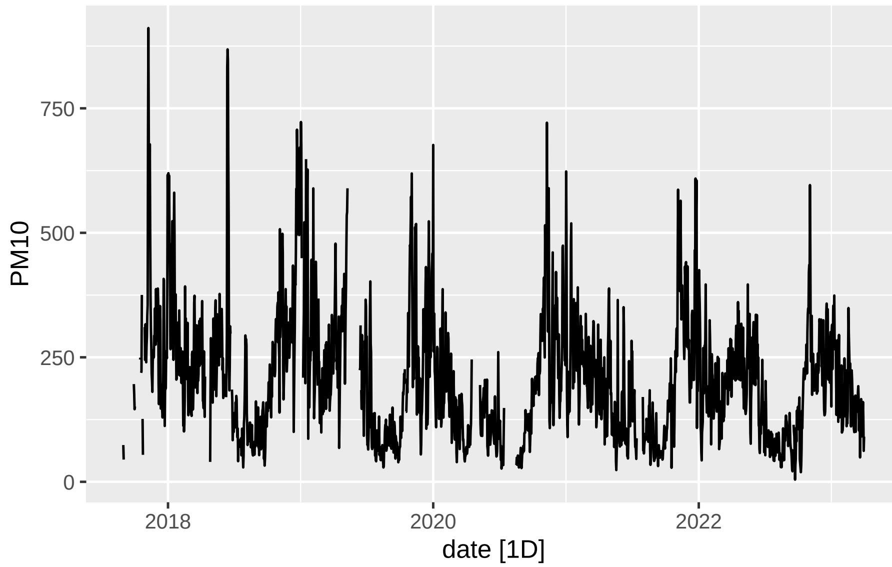
```
 4 Benzene  1911             93.7
 5 Toluene  1911             93.7
 6 Xylene   1892             92.7
 7 PM2.5    1884             92.4
 8 PM10     1867             91.5
 9 NO2      1841             90.2
10 NO       1818             89.1
```

Plot Each Parameter

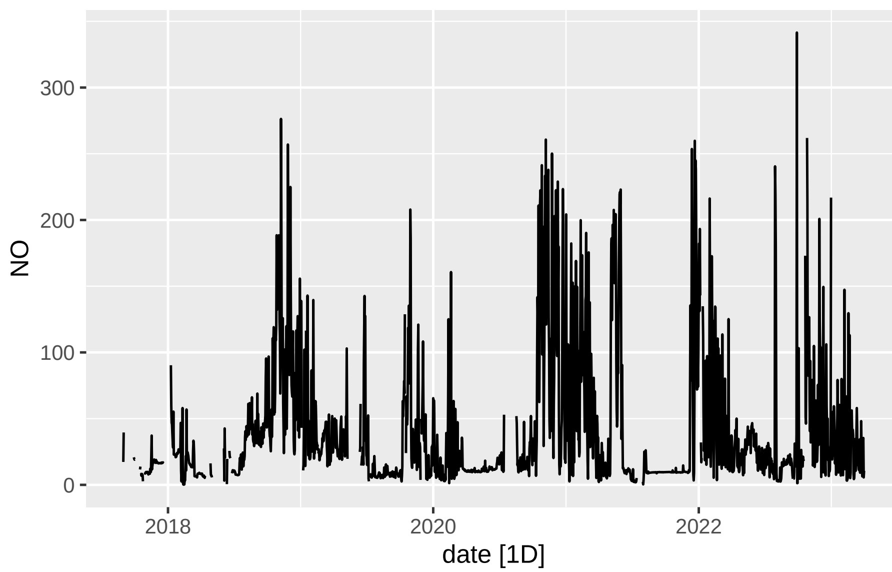
```r
daily_df %>%
  autoplot(PM2.5)
```



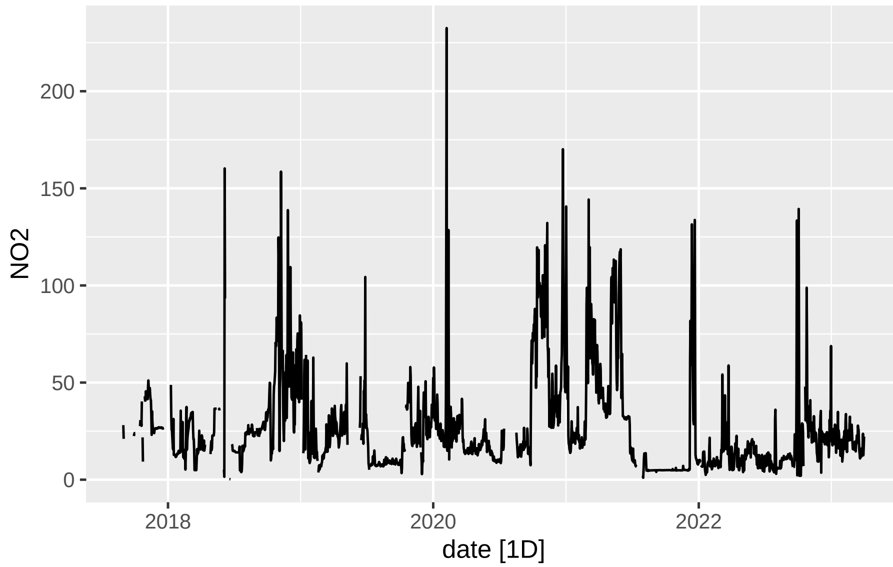

```r
daily_df %>%
  autoplot(PM10)
```



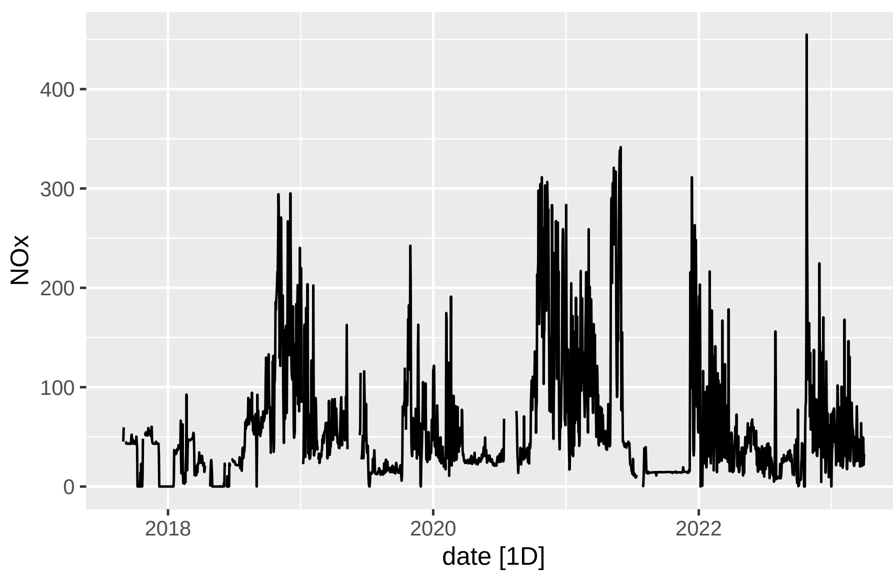

```
daily_df %>%
  autoplot(NO)
```



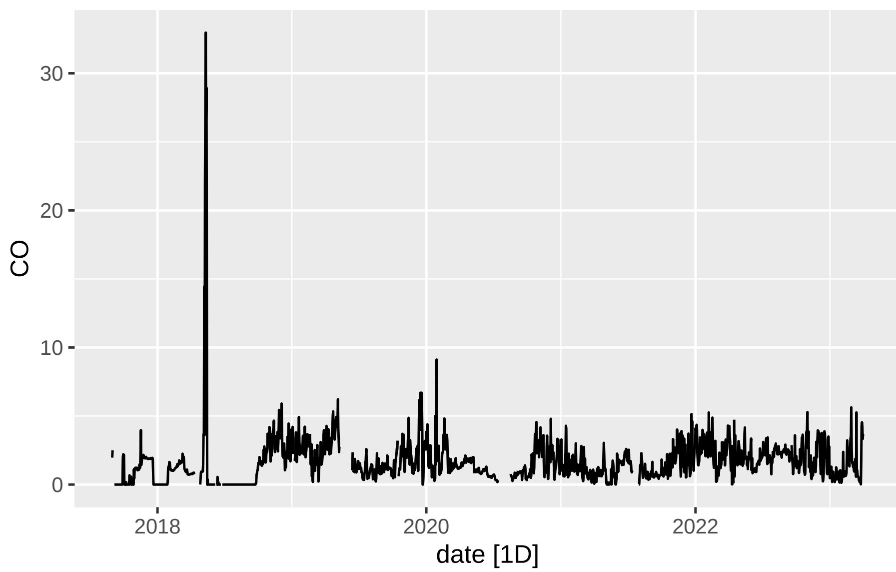

```
daily_df %>%
  autoplot(NO2)
```



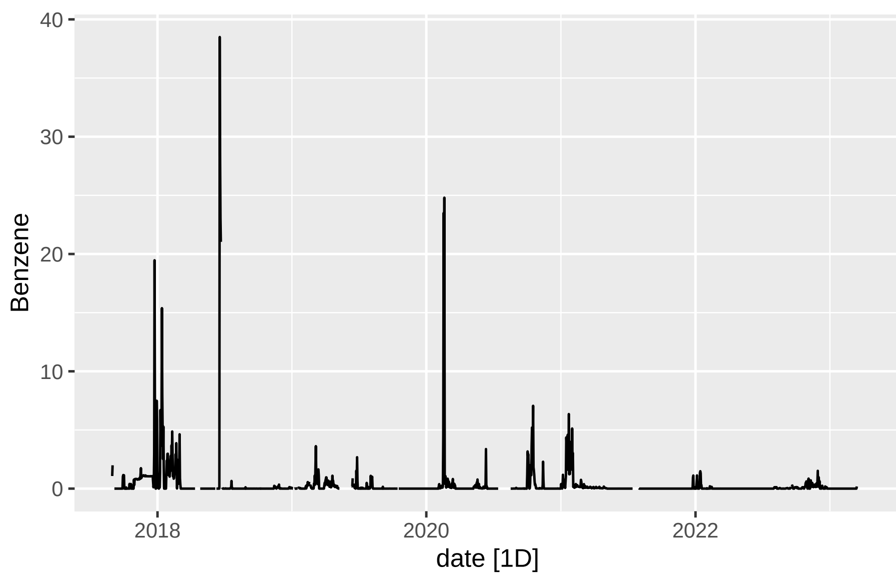

```r
daily_df %>%
  autoplot(NOx)
```



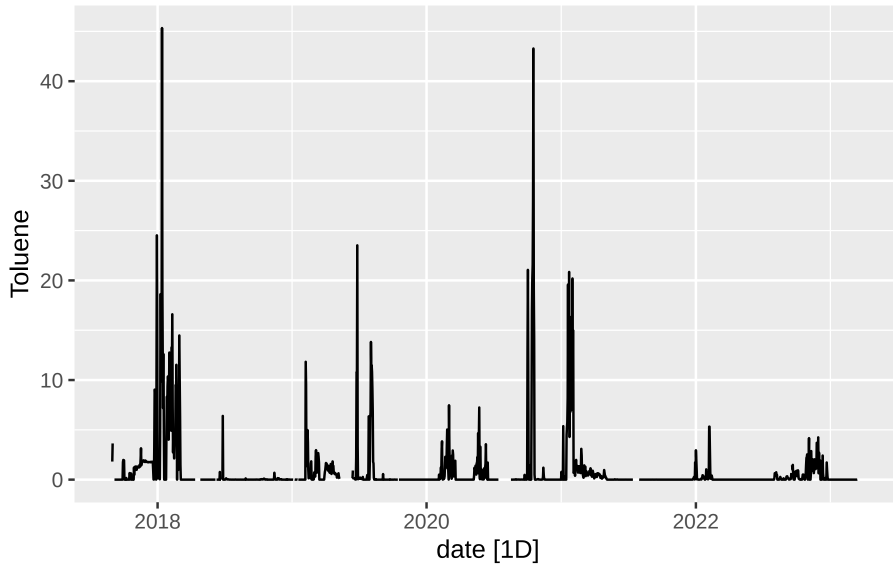

```r
daily_df %>%
  autoplot(CO)
```



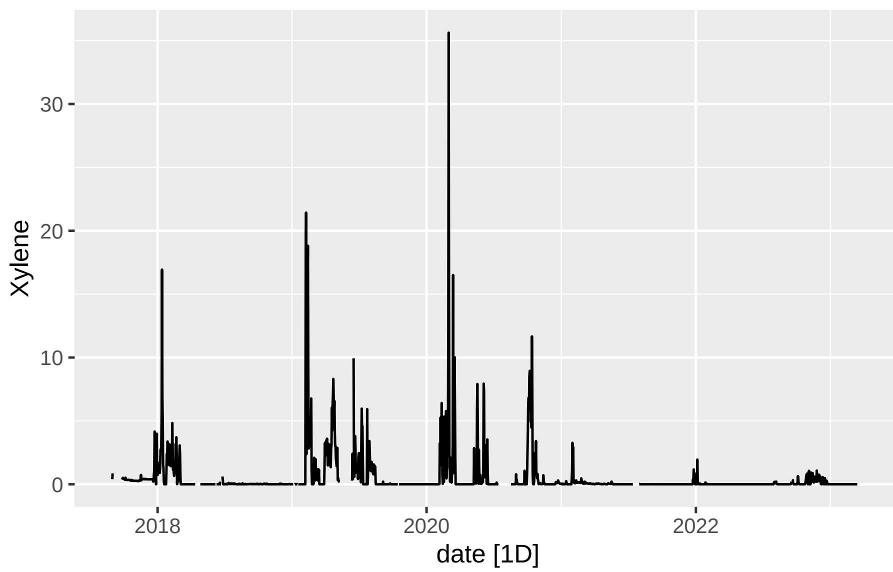

```r
daily_df %>%
  autoplot(Benzene)
```




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

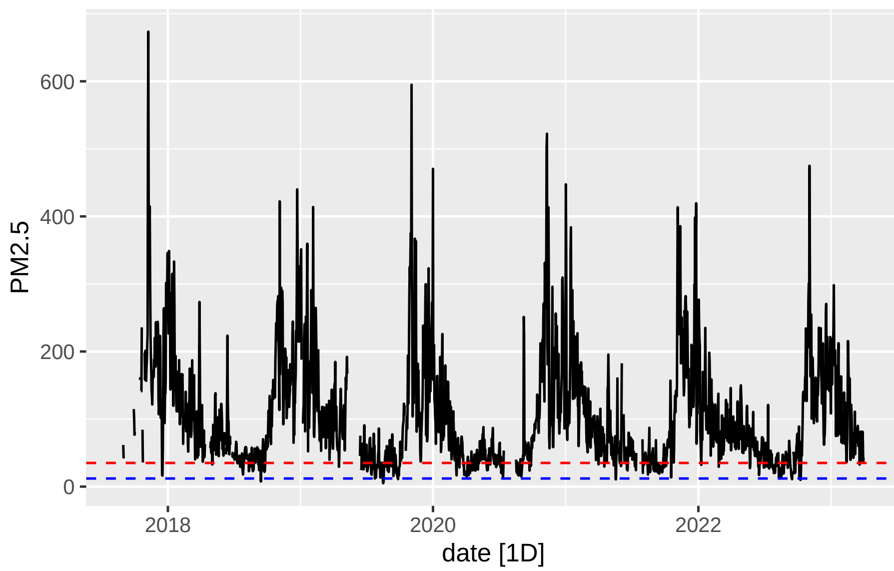



```
daily_df %>%
  autoplot(Xylene)
```



Recomended Limit for PM2.5

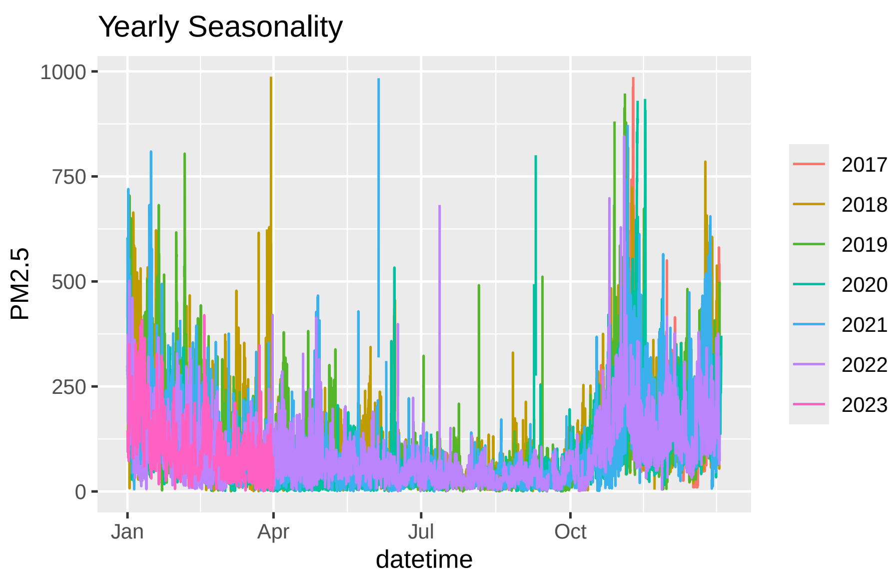
```r
df %>%
  index_by(date = as.Date(datetime)) %>%
  summarise(PM2.5 = mean(PM2.5, na.rm = TRUE)) %>%
  autoplot(PM2.5) +
  geom_hline(yintercept = 35, color = "red", linetype = "dashed") +
  geom_hline(yintercept = 12, color = "blue", linetype = "dashed")
```



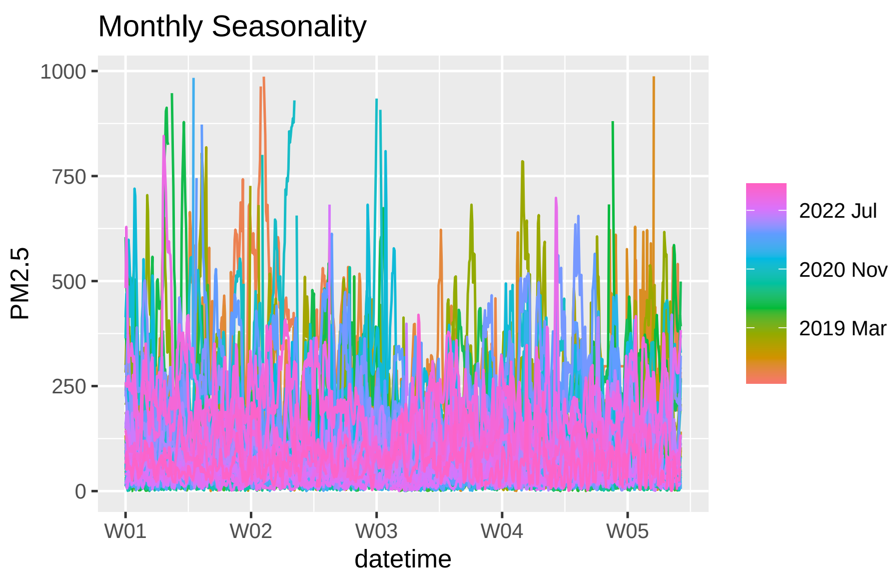
The World Health Organization recomends PM2.5 levels below $35ug/m^3$ for a daily average and below $12ug/m^3$ for a yearly average. The daily and yearly recomended levels are plotted in red and blue respectively. Only during the summer months are the PM2.5 levels below the WHO recomended daily limit.

Seasonality

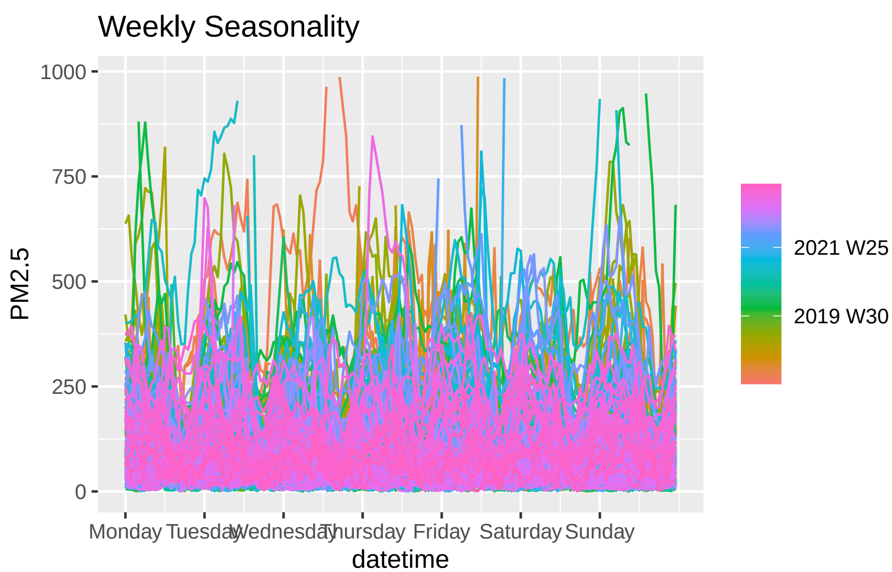
```r
gg_season(df, PM2.5, period = "1 year") +
  labs(title = "Yearly Seasonality")
```



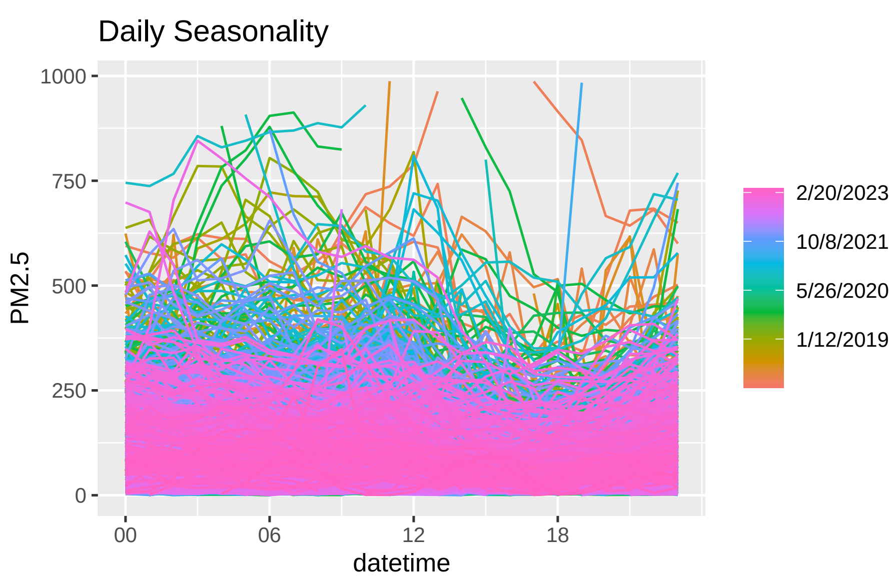

```r
gg_season(df, PM2.5, period = "1 month") +
  labs(title = "Monthly Seasonality")
```




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



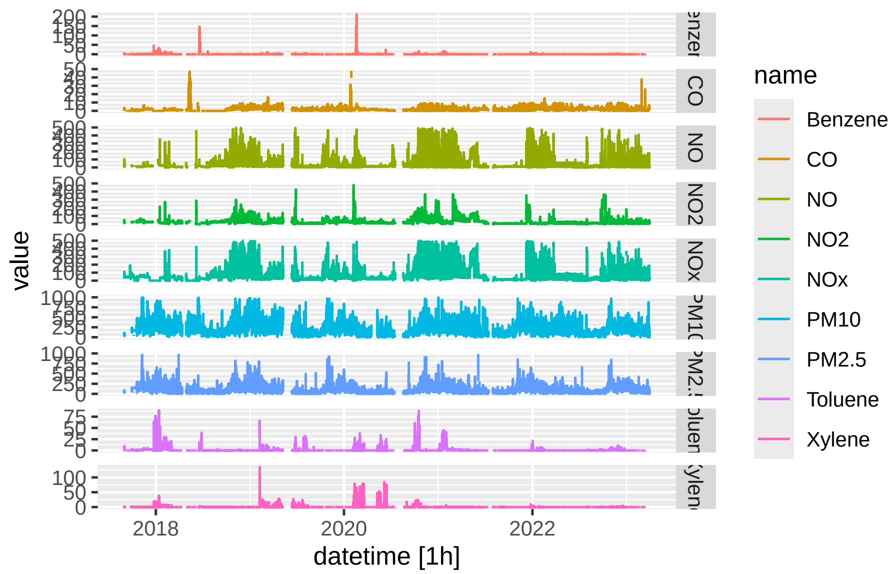

```
gg_season(df, PM2.5, period = "1 day") +
  labs(title = "Daily Seasonality")
```



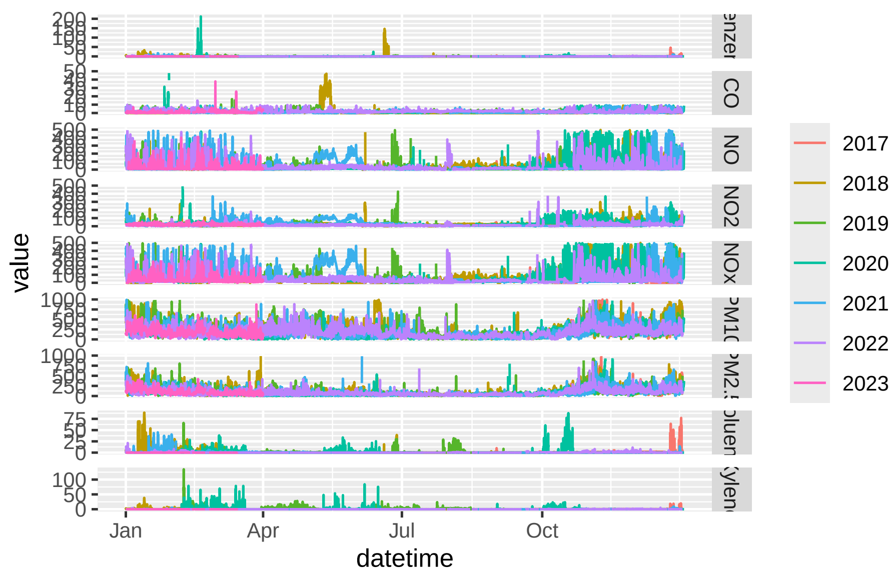
Plot Parameters Together

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

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

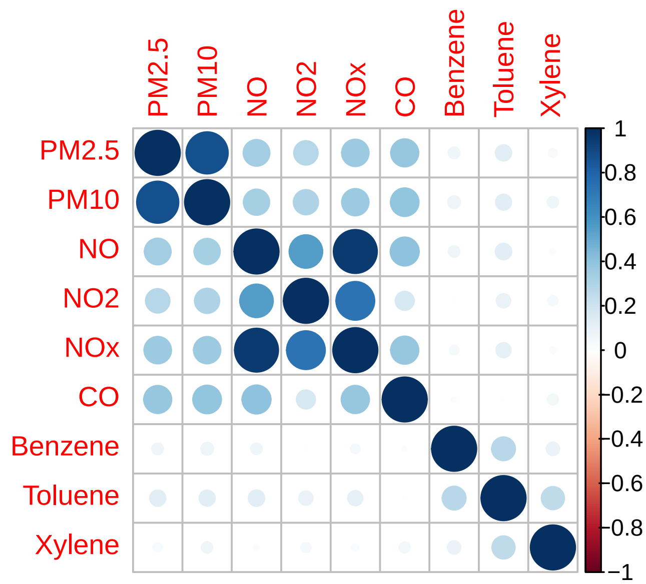



```r
df_long %>% gg_season(value) +
    facet_grid(rows = vars(name), scales = "free_y")
```



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, NO2, 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

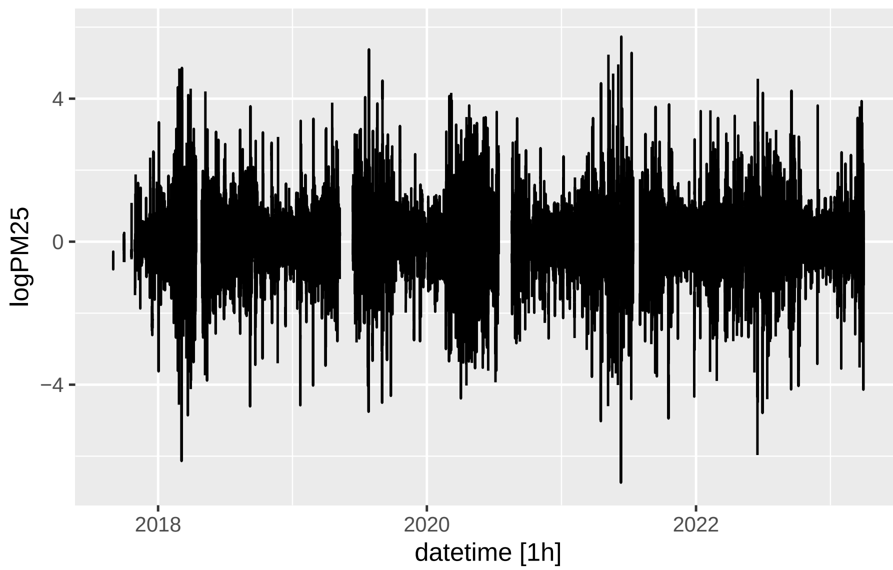
```
df$logPM25 <- log(df$PM2.5) %>%
    difference(24)

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

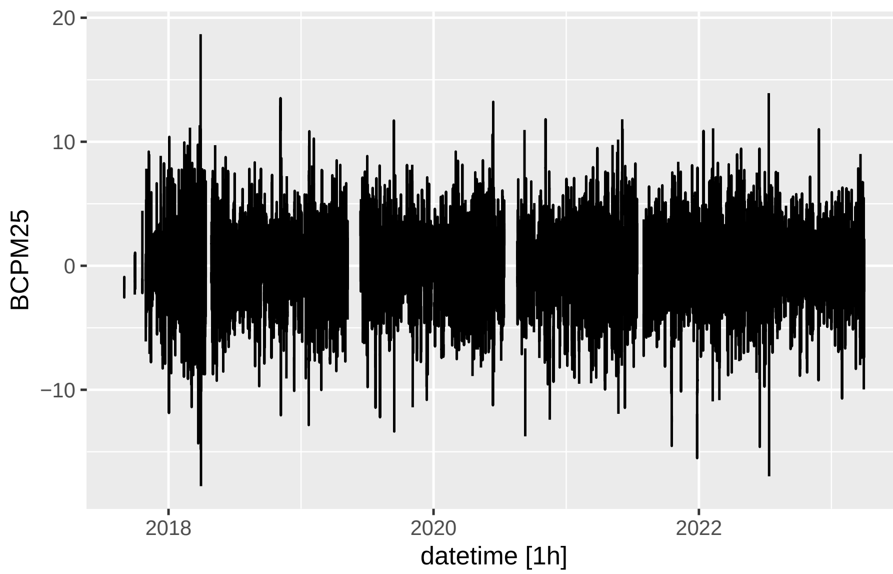
```
[1] 0.2959454
```

```
df$BCPM25 <- BoxCox(df$PM2.5, BoxCox.lambda(df$PM2.5)) %>%
    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. However, using a Box-Cox transform creates a more stationary time series than the log transform. This method of transformation is likely the better option if we were to move forward with a model choice that requires stationarity.

Check for Constant Variance

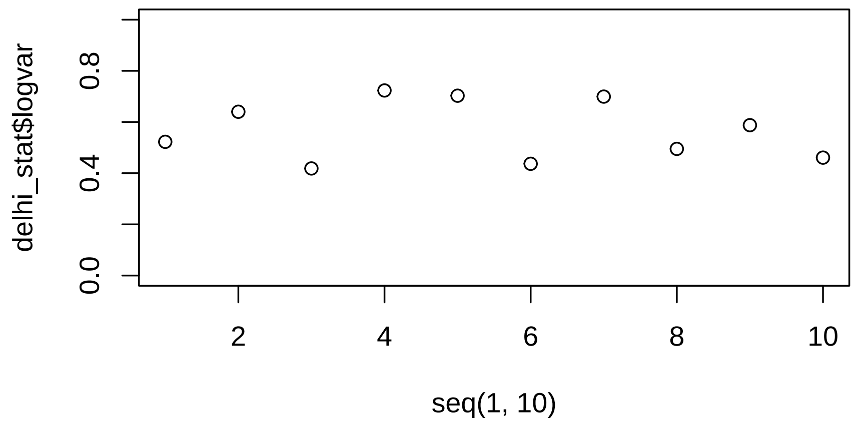
```r
delhi_stat <- data.frame(logvar = numeric(10),
                         boxvar = numeric(10))
for (i in 1:10){
    delhi_stat[i,1] <- var(df$logPM25[
        floor(nrow(df)*0.1*(i-1)):floor(nrow(df)*0.1*i)
        ], na.rm = TRUE)

    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
```

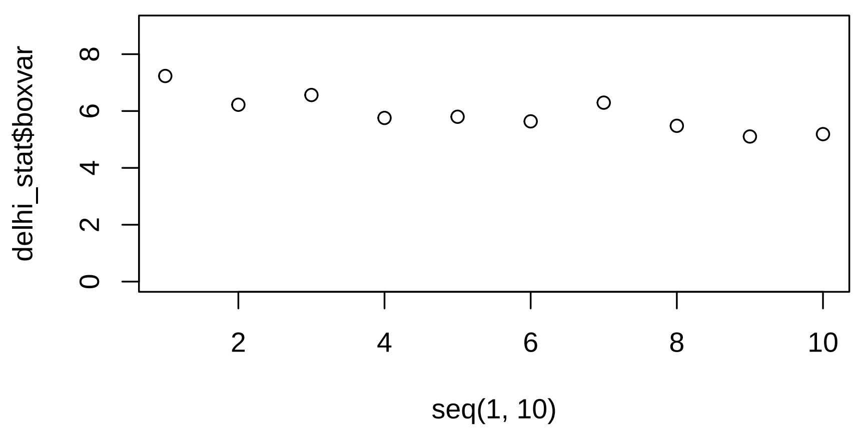
```
      logvar   boxvar
1  0.5225954 7.232752
2  0.6401363 6.220892
3  0.4184593 6.564328
4  0.7233174 5.757312
5  0.7029033 5.798939
6  0.4367190 5.636711
7  0.6993186 6.294659
8  0.4949893 5.480331
9  0.5876888 5.103950
10 0.4608093 5.187683
```

```r
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 Box-Cox transformation results in stable variances.

World Health Organization Recomended Limits:

CO: 9 ppm
PM . : 15 µg/m^3
PM : 45 µg/m^3
NO : 25 µg/m^3
Benzene: As low as possible (<5 µg/m^3)
Toluene: 260 µg/m^3
Xylene: 870 µg/m^3

Scaled Parameters by Recomended Limits

```r
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 Quality 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 recomended 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, NO2, NOx, 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.