Assignment 7: Time Series Analysis

Molly Bruce

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A07_TimeSeries.Rmd") prior to submission.

The completed exercise is due on Tuesday, March 16 at 11:59 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

2. Import the ten datasets from the Ozone_TimeSeries folder in the Raw data folder. These contain ozone concentrations at Garinger High School in North Carolina from 2010-2019 (the EPA air database only allows downloads for one year at a time). Import these either individually or in bulk and then combine them into a single dataframe named `GaringerOzone` of 3589 observation and 20 variables.

```
#1

getwd()
```

```
## [1] "C:/Users/mmb88/Desktop/Environmental_Data_Analytics_2021/Assignments"
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3
## Warning: package 'ggplot2' was built under R version 4.0.3
## Warning: package 'tibble' was built under R version 4.0.3
## Warning: package 'tidyr' was built under R version 4.0.3
## Warning: package 'readr' was built under R version 4.0.3
## Warning: package 'purrr' was built under R version 4.0.3
## Warning: package 'dplyr' was built under R version 4.0.3
## Warning: package 'stringr' was built under R version 4.0.3
```

```
## Warning: package 'forcats' was built under R version 4.0.3
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.3
```

```
library(trend)
```

```
## Warning: package 'trend' was built under R version 4.0.4
```

```
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.0.4
```

```
library(agricolae)
```

```
## Warning: package 'agricolae' was built under R version 4.0.3
```

```
library(ggplot2)
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.0.3
```

```
library(scales)
```

```
## Warning: package 'scales' was built under R version 4.0.3
```

```
# I could add more to this theme but will leave it more simple for now
mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right",
        panel.grid.major = element_line(colour = "gray"))
theme_set(mytheme)

#2

# Providing full paths b/c RStudio on my VM won't process the relative paths
# I recognize that relative paths are preferable

O3_2010 <- read.csv("C:/Users/mmb88/Desktop/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/
O3_2011 <- read.csv("C:/Users/mmb88/Desktop/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/
O3_2012 <- read.csv("C:/Users/mmb88/Desktop/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/
O3_2013 <- read.csv("C:/Users/mmb88/Desktop/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/
O3_2014 <- read.csv("C:/Users/mmb88/Desktop/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/
O3_2015 <- read.csv("C:/Users/mmb88/Desktop/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/
O3_2016 <- read.csv("C:/Users/mmb88/Desktop/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/
O3_2017 <- read.csv("C:/Users/mmb88/Desktop/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/
O3_2018 <- read.csv("C:/Users/mmb88/Desktop/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/
O3_2019 <- read.csv("C:/Users/mmb88/Desktop/Environmental_Data_Analytics_2021/Data/Raw/Ozone_TimeSeries/

Ozone <- rbind(O3_2010, O3_2011, O3_2012, O3_2013, O3_2014,
               O3_2015, O3_2016, O3_2017, O3_2018, O3_2019)
```

Wrangle

3. Set your date column as a date class.
4. Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.

```
# 3

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

# 4

OzoneShort <-
  select(Ozone,
         Date,
         Daily.Max.8.hour.Ozone.Concentration,
         DAILY_AQI_VALUE)
```

5. Notice there are a few days in each year that are missing ozone concentrations. We want to generate a daily dataset, so we will need to fill in any missing days with NA. Create a new data frame that contains a sequence of dates from 2010-01-01 to 2019-12-31 (hint: `as.data.frame(seq())`). Call this new data frame Days. Rename the column name in Days to "Date".

6. Use a `left_join` to combine the data frames. Specify the correct order of data frames within this function so that the final dimensions are 3652 rows and 3 columns. Call your combined data frame GaringerOzone.

```
# 5

Days <- as.data.frame(seq(as.Date("2010/1/1"),as.Date("2019/12/31"),"day"))
names(Days)[1] <- "Date"

#6 Days is left so that we retain all dates (3652 observations)

GaringerOzone <- left_join(Days, OzoneShort, c("Date"="Date"))
```

Visualize

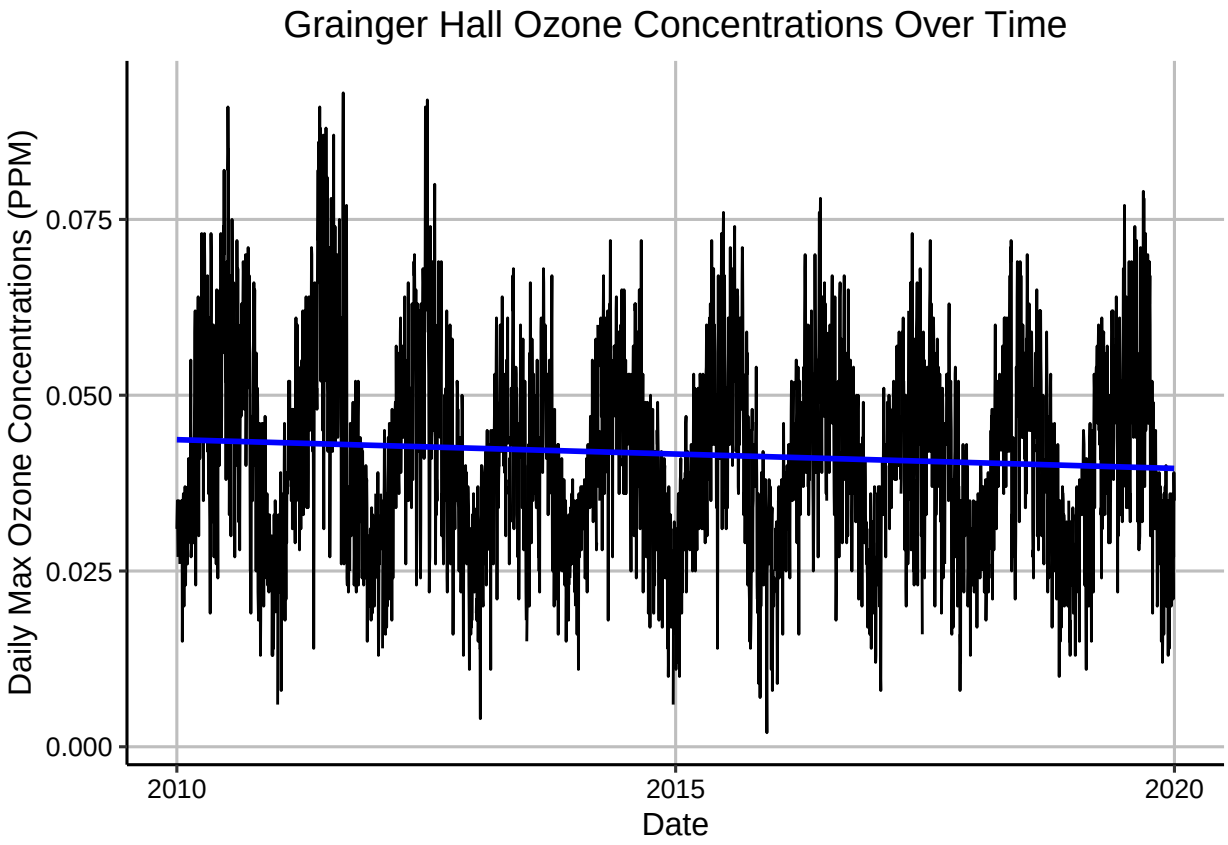
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7

OzonePlot <- ggplot(GaringerOzone,
                    aes(x = Date,
                        y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = lm, se = FALSE, color = "blue") +
  labs(title = "Grainger Hall Ozone Concentrations Over Time",
       y = "Daily Max Ozone Concentrations (PPM)",
       x = "Date") +
  theme(plot.title = element_text(hjust=0.5))
OzonePlot
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 63 rows containing non-finite values (stat_smooth).
```



Answer: Based on visual analysis, the plot suggests that, though mild, ozone levels at Grainger Hall have been decreasing over the 10 year period from 2010 to 2020. Specifically, it looks as though the levels have fallen from ~.044 to ~.040 ppm.

Time Series Analysis

Study question: Have ozone concentrations changed over the 2010s at this station?

8. Use a linear interpolation to fill in missing daily data for ozone concentration. Why didn't we use a piecewise constant or spline interpolation?

```
#8

summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```
GaringerOzoneClean <-
  GaringerOzone %>%
  mutate(Ozone.Concentration.Clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration)) %>%
  select(Date,
         Ozone.Concentration.Clean,
         DAILY_AQI_VALUE)
summary(GaringerOzoneClean$Ozone.Concentration.Clean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

Answer: A spline interpolation applies a polynomial fit to interpolate missing values rather than a linear fit. Meanwhile, a piecewise constant picks the nearest data and assigns that data's value to the missing value. In our situation, there are not that many days with missing values over our 10 year span. As such, rather than assigning the same value to missing days as is assigned to either the previous day or the next day, it makes more sense to assign a value somewhere in the middle (either linear or spline). However, because we aren't missing multiple days in a row and because we have no reason to believe that O3 concentrations aren't linear, a linear interpolation makes more sense and is easier to accomplish.

9. Create a new data frame called `GaringerOzone.monthly` that contains aggregated data: mean ozone concentrations for each month. In your pipe, you will need to first add columns for year and month to form the groupings. In a separate line of code, create a new Date column with each month-year combination being set as the first day of the month (this is for graphing purposes only)

```
#9

GaringerOzone.monthly <- GaringerOzoneClean %>%
  separate(Date, c("Year","Month","Day"), "-") %>%
  mutate(Date = my(paste0(Month,"-",Year))) %>%
  group_by(Date) %>%
  summarize(OzoneMonthlyAverage = mean(Ozone.Concentration.Clean))
```

10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

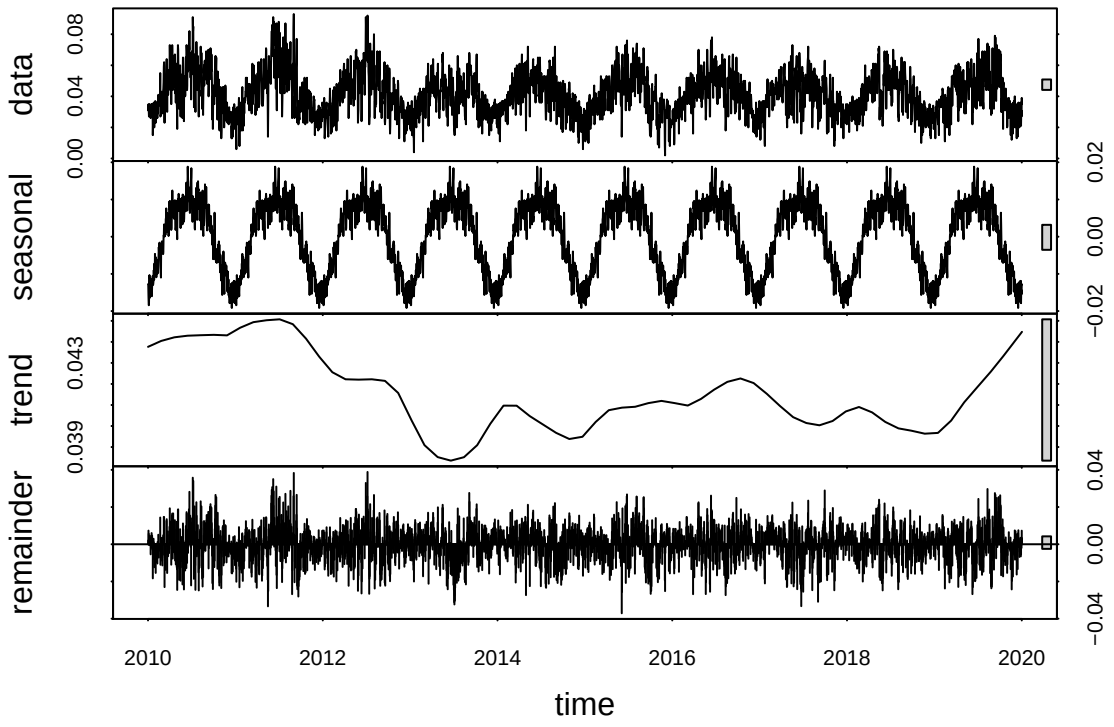
```
#10

GaringerOzone.daily.ts <- ts(GaringerOzoneClean$Ozone.Concentration.Clean,
                             start=c(2010,01,01),
                             frequency=365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$OzoneMonthlyAverage,
                               start=c(2010,01),
                               frequency=12)
```

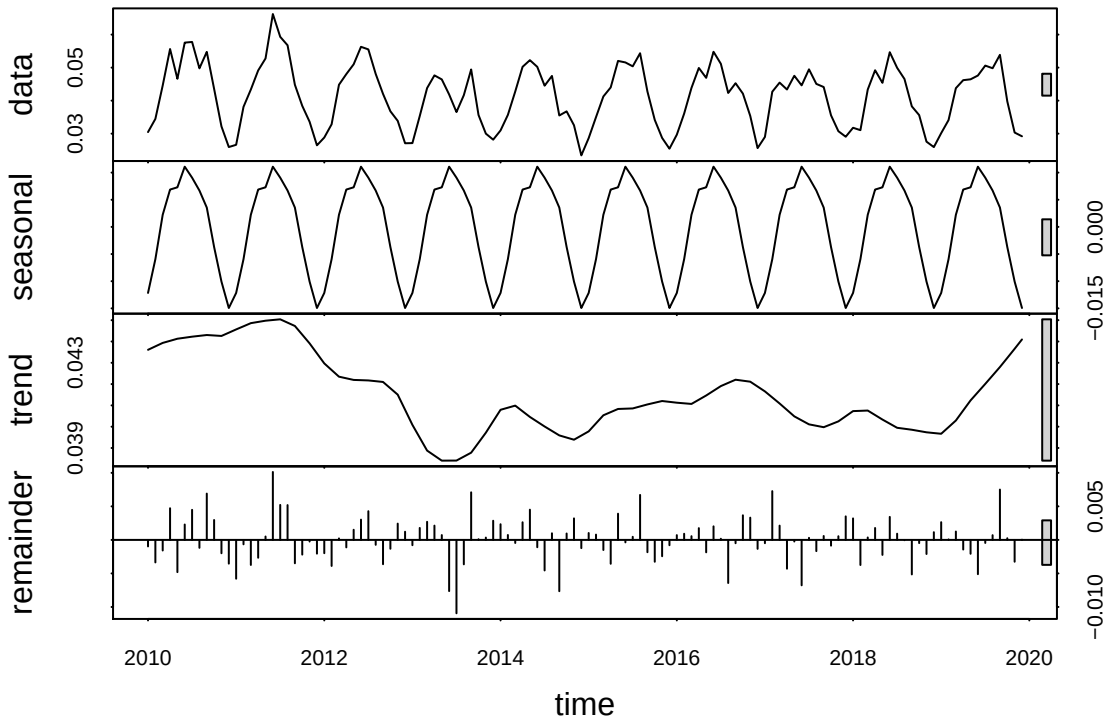
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11

OzoneDailyDecomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(OzoneDailyDecomp)
```




```
OzoneMonthlyDecomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(OzoneMonthlyDecomp)
```



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

```
#12 SMK test

OzoneMonthlyTrend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
OzoneMonthlyTrend
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

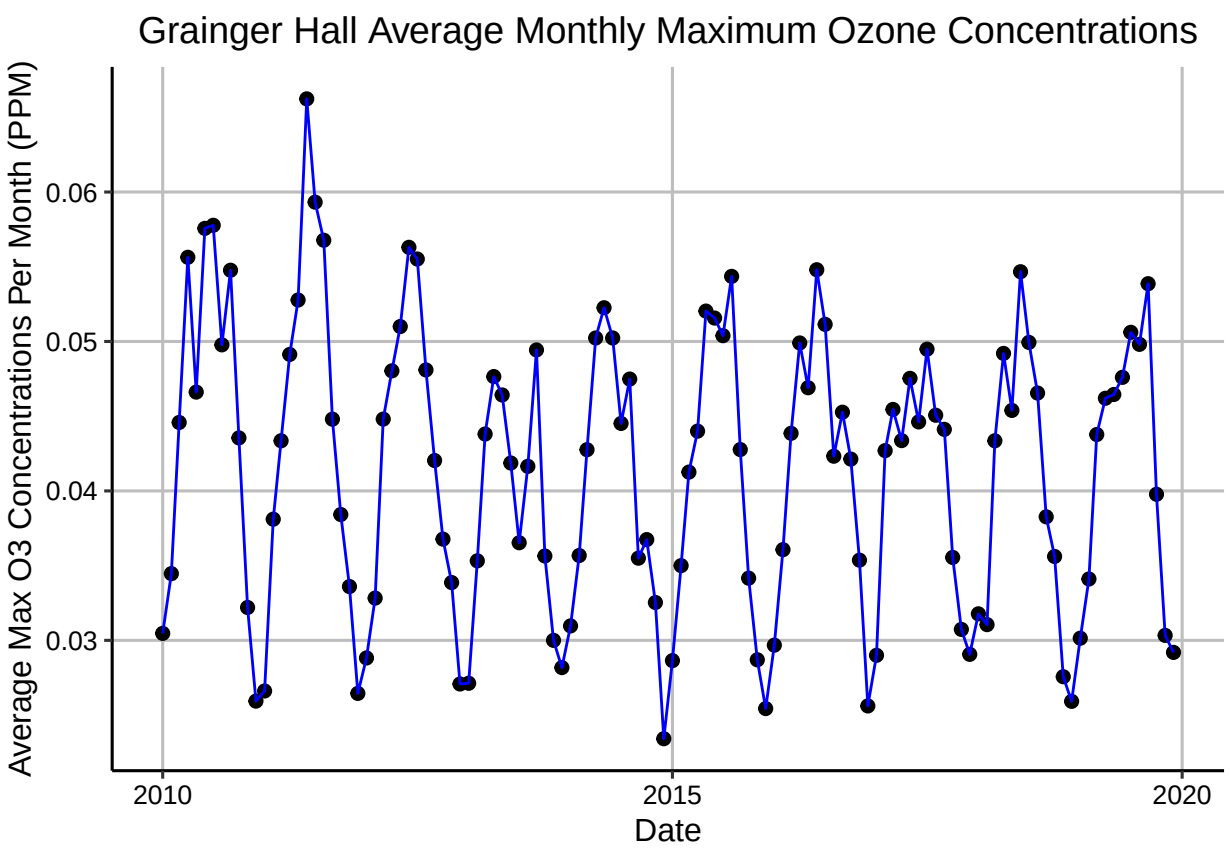
Answer: We haven't yet removed seasonality's impact on our dataset; therefore, the seasonal Mann-Kendall is most appropriate as it acknowledges the dataset's seasonality.

13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```
# 13

OzoneMonthlyPlot <-
  ggplot(GaringerOzone.monthly,
         aes(x = Date,
             y = OzoneMonthlyAverage)) +
  geom_point(size = 2.0) +
  geom_line(color = "blue") +
  labs(title = "Grainger Hall Average Monthly Maximum Ozone Concentrations",
       y = "Average Max O3 Concentrations Per Month (PPM)",
       x = "Date") +
  theme(plot.title = element_text(hjust=0.5))
```

```
OzoneMonthlyPlot
```



14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

    Answer: The null hypothesis of the seasonal Mann-Kendall is that that the data doesn't have a trend. Here, that means that the null hypothesis is that average monthly maximum Ozone levels at Grainger Hall remain statistically stationary over our 10 year time period. Because our p value is less than 0.05 (it's 0.046724), we can reject our null hypothesis. This in turn means that we have a trend in our data–that average monthly maximum ozone levels are NOT stationary.

15. Subtract the seasonal component from the `GaringerOzone.monthly.ts`. Hint: Look at how we extracted the series components for the EnoDischarge on the lesson Rmd file.

16. Run the Mann Kendall test on the non-seasonal Ozone monthly series. Compare the results with the ones obtained with the Seasonal Mann Kendall on the complete series.

```
#15

OzoneMonthlyComponents <- as.data.frame(OzoneMonthlyDecomp$time.series[,1:3]) %>%
  mutate(Observed = GaringerOzone.monthly$OzoneMonthlyAverage,
         Date = GaringerOzone.monthly$Date)

OzoneMonthlyNonseasonal <-
  ts(data = OzoneMonthlyComponents$Observed - OzoneMonthlyComponents$seasonal,
                        start = c(2010,1),
                        frequency = 12)
```

```r
# 16 MK test

OzoneMonthlyComponentsTrend <- Kendall::MannKendall(OzoneMonthlyNonseasonal)
OzoneMonthlyComponentsTrend
```

```
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer: In our Mann Kendall test, after we had removed the seasonality from our data, our p value decreased to 0.0075402 meaning that we are even more confident that the null hypothesis–that average monthly maximum Ozone levels at Grainger Hall remain stationary over our 10 year time period–is false. Put another way, after removing the seasonality, we feel even more confident that there is a trend in average monthly maximum ozone levels.