Assignment 8: Time Series Analysis

Lauren Shohan

Fall 2024

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on generalized linear models.

Directions

1. Rename this file `<FirstLast>_A08_TimeSeries.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up

1. Set up your session:

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

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(trend)
library(here)
```

```
## here() starts at /home/guest/EDE_Fall2024
```

```
getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
here()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

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

EPAair10 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
                   stringsAsFactors = TRUE)
EPAair11 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
                   stringsAsFactors = TRUE)
EPAair12 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                   stringsAsFactors = TRUE)
EPAair13 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                   stringsAsFactors = TRUE)
EPAair14 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                   stringsAsFactors = TRUE)
EPAair15 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                   stringsAsFactors = TRUE)
EPAair16 <-
```

```r
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                       stringsAsFactors = TRUE)
EPAair17 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
                       stringsAsFactors = TRUE)
EPAair18 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                       stringsAsFactors = TRUE)
EPAair19 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                       stringsAsFactors = TRUE)

GaringerOzone <- rbind(EPAair10,EPAair11,EPAair12,EPAair13,EPAair14,EPAair15,
                       EPAair16,EPAair17,EPAair18,EPAair19)
```

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.
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.

```r
# 3
GaringerOzone$Date <- mdy(GaringerOzone$Date)
class(GaringerOzone$Date)
```

```
## [1] "Date"
```

```r
# 4
Wrangled_GaringerOzone <- select(GaringerOzone, Date,
                                 Daily.Max.8.hour.Ozone.Concentration,
                                 DAILY_AQI_VALUE)
# 5
Days <- as.data.frame(seq(from = as.Date('2010-01-01'),
                          to = as.Date('2019-12-31'), by = 'day'))
colnames(Days) <- 'Date'

# 6
GaringerOzone <- left_join(Days, Wrangled_GaringerOzone, by = 'Date')
```

Visualize

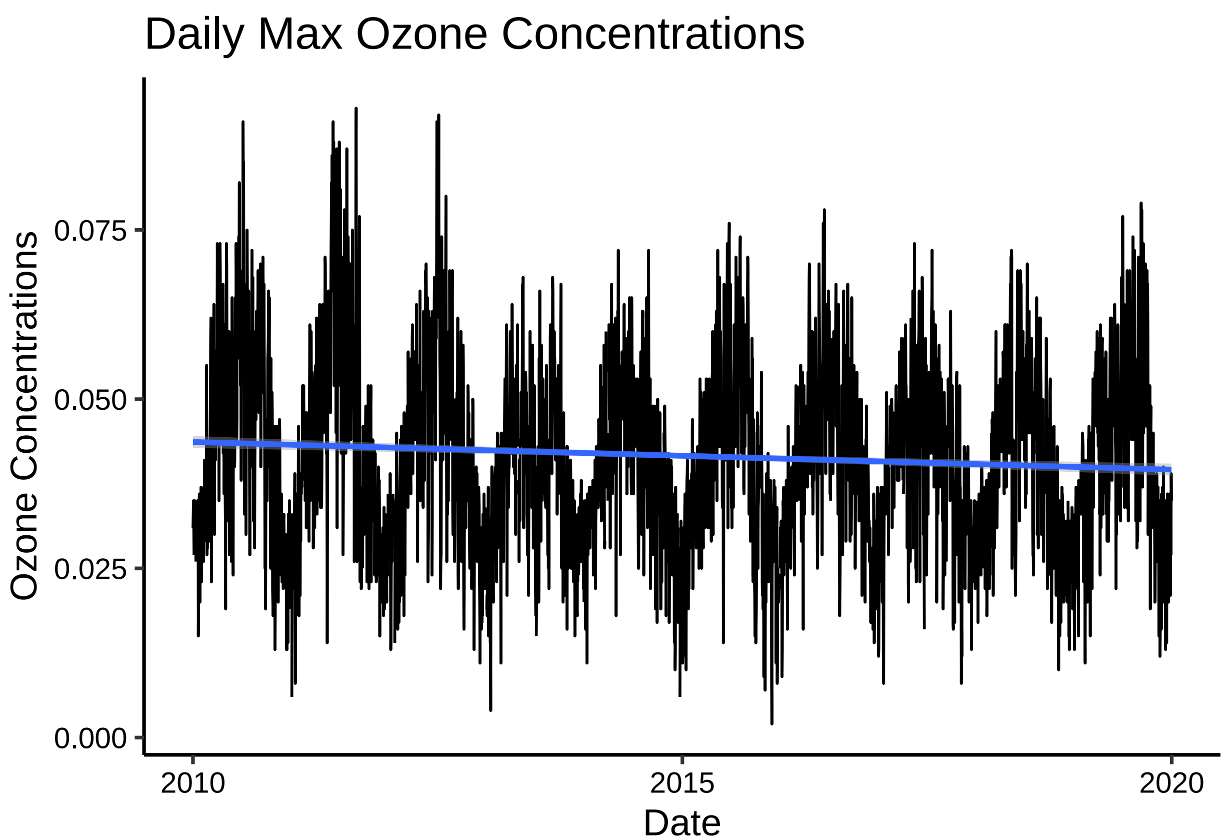
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7

GaringerOzone_LinePlot <- ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration))
  geom_line() +
  geom_smooth(method = 'lm') +
  labs(y = 'Ozone Concentrations',
       title = 'Daily Max Ozone Concentrations')
GaringerOzone_LinePlot
```

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

## Warning: Removed 63 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```



Answer: Yes it does, there is a very slight decrease in my linear trend line over time.

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) #63 NAs
```

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

```
Garinger.Cleaned <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentrati
  select(Date, Daily.Max.8.hour.Ozone.Concentration.clean, DAILY_AQI_VALUE)

summary(Garinger.Cleaned$Daily.Max.8.hour.Ozone.Concentration.clean)#no more NAs
```

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

Answer: We used the linear interpolation 'connect the dots' approach to fill in any missing data because we assume it to fall between the previous and next measurement. Piecewise would've assumed it to be equal to the measurement nearest to that date and spline would've used a quadratic function to interpolate. Linear makes the most sense with this data since we are going by daily values so we are just continuing the data line in a sense and assuming it to be continous.

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 <- Garinger.Cleaned %>%
  mutate( Year = format(Date, "%Y"),
          Month = format(Date, '%m'))

GaringerOzone.monthly <- GaringerOzone.monthly %>%
  group_by(Year, Month) %>%
  summarise(Mean_Ozone_Concentrations_each_month =
              mean(Daily.Max.8.hour.Ozone.Concentration.clean)) %>%
  mutate( Date = as.Date(paste(Year, Month, '01', sep = '-'))) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

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.

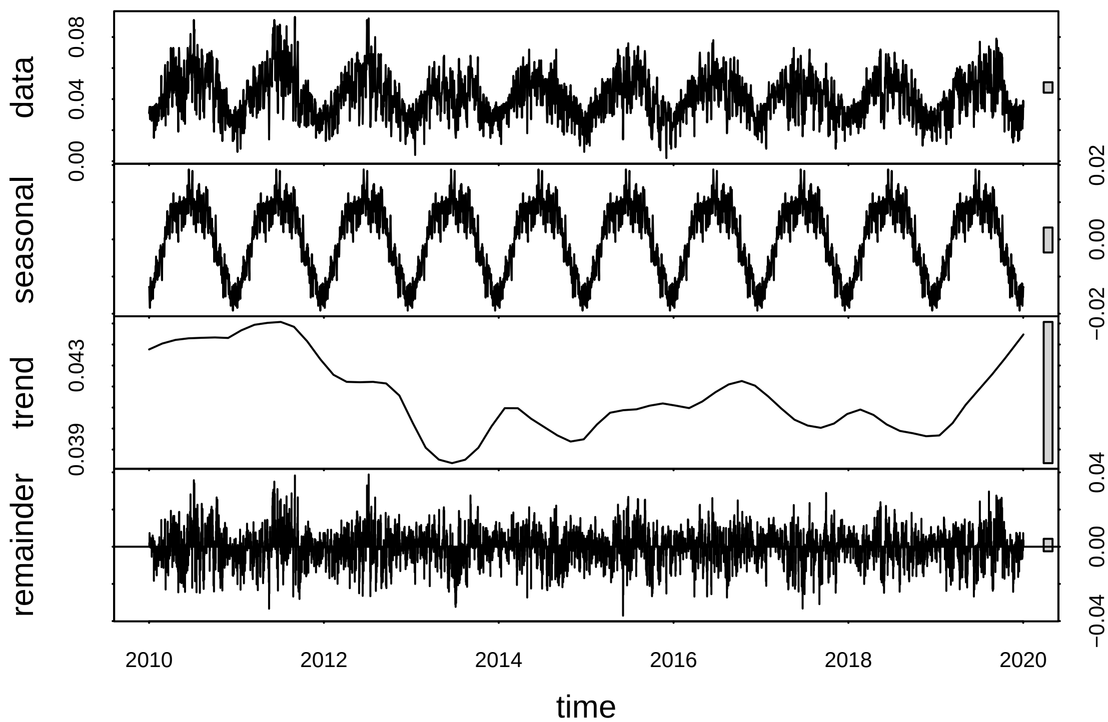
```r
#10
GaringerOzone.daily.ts <-
  ts(Garinger.Cleaned$Daily.Max.8.hour.Ozone.Concentration.clean,
                             start = c(2010,1),
                             frequency = 365)

GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$Mean_Ozone_Concentrations_each_month,
                               start = c(2010,1),
                               frequency = 12)
```

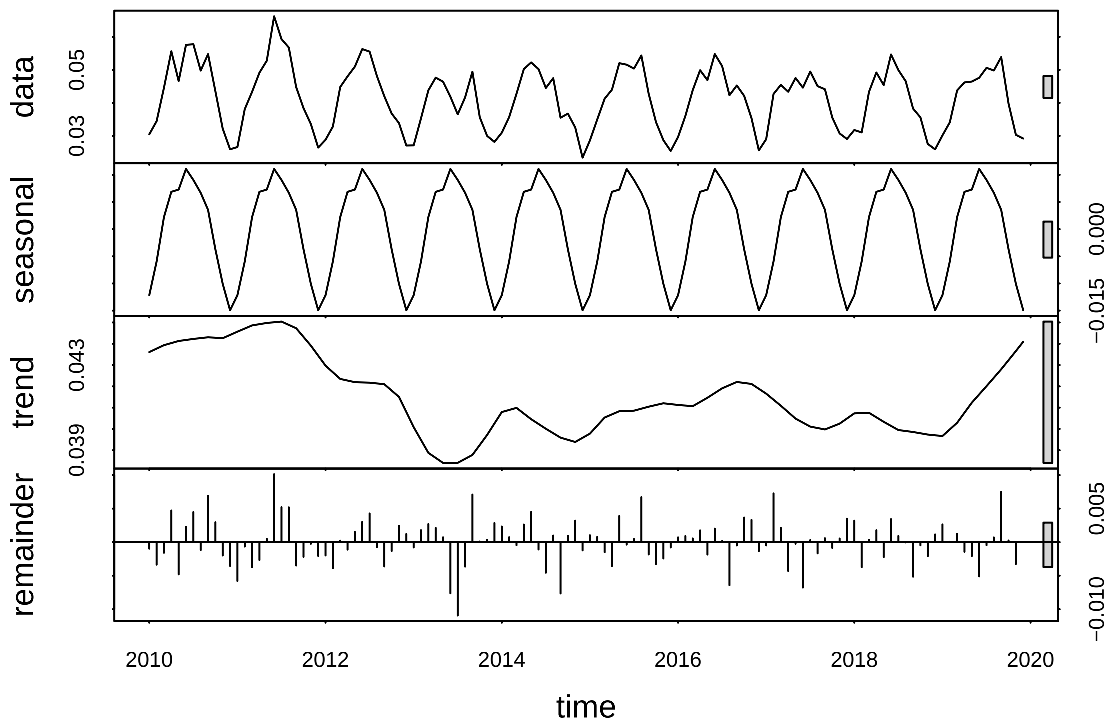
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```r
#11

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




```r
Ozone_monthly_decomposed <- stl(GaringerOzone.monthly.ts, s.window = 'periodic')
plot(Ozone_monthly_decomposed)
```



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

Ozone_monthlydata_trend1 <-
  Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
Ozone_monthlydata_trend1
```

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

```
summary(Ozone_monthlydata_trend1)
```

```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

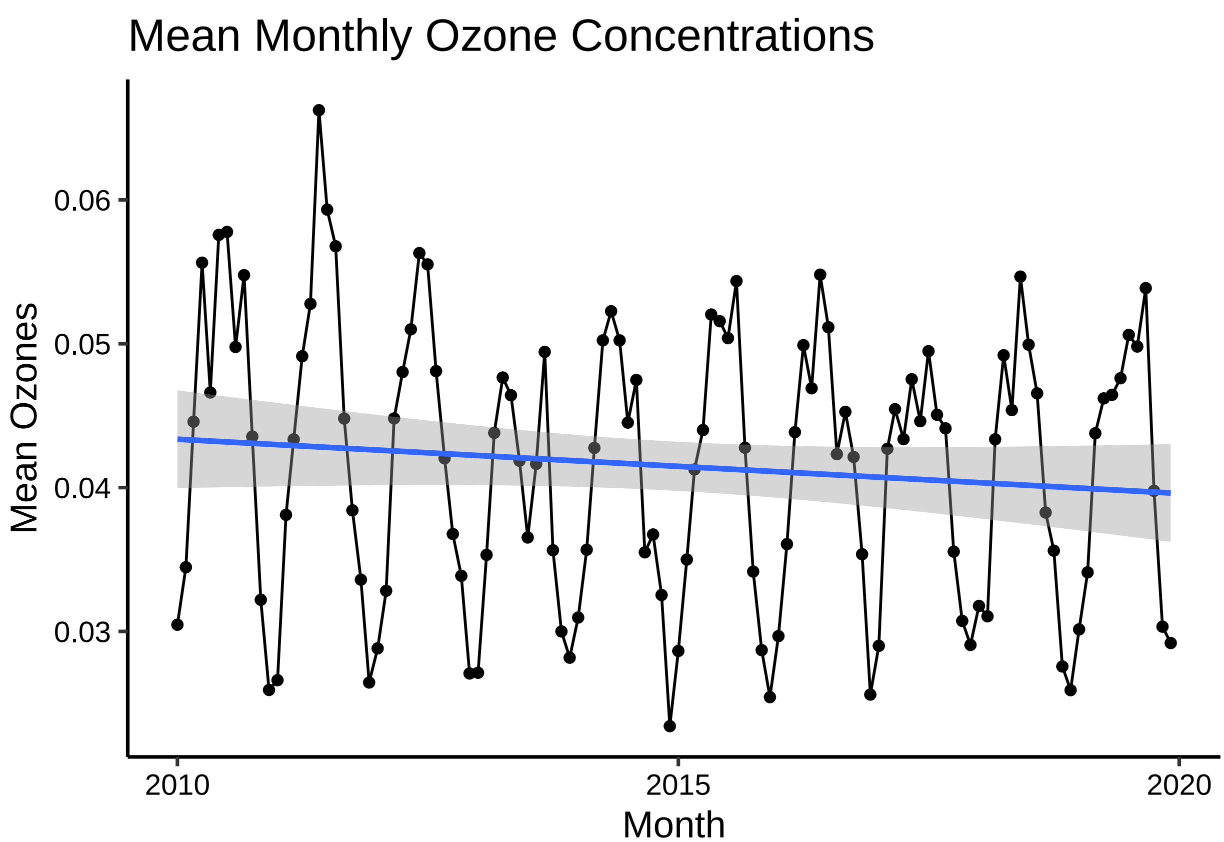
Answer: It is most appropriate because you can see a clear seasonal trend in the data and thus we would use the seasonal Mann-Kendall which uses seasonality in its monotonic trend analysis.

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.

```r
# 13
Mean_monthly_ozone_plot <- GaringerOzone.monthly %>%
  ggplot(aes(x = Date, y = Mean_Ozone_Concentrations_each_month)) +
  geom_point() +
  geom_line() +
  labs(y = 'Mean Ozones',
       title = 'Mean Monthly Ozone Concentrations',
       x = 'Month') +
  geom_smooth(method = 'lm')

Mean_monthly_ozone_plot
```

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



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: Have ozone concentrations changed over the 2010s at this station? The graph shows a slight downwards decrease in the mean ozones over this time period. Our pvalue of 0.0467, which is below our 0.05 threshold and therefore we can reject our null hypothesis and say there is significance between mean ozones over the 2010s. Statistical test output: Score = -77 , Var(Score) = 1499 denominator = 539.4972 tau = -0.143, 2-sided pvalue =0.046724

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

Ozone_monthly_Components <-
  as.data.frame(Ozone_monthly_decomposed$time.series[,1:3])

Ozone_monthly_Components <- Ozone_monthly_Components %>%
  mutate(ObservedMeans = GaringerOzone.monthly$Mean_Ozone_Concentrations_each_month,
         Date = GaringerOzone.monthly$Date) %>%
  mutate(Nonseasonal = ObservedMeans - seasonal)

nonseasonal_ozone.ts <- ts(Ozone_monthly_Components$Nonseasonal,
   start = c(2010,1),
   frequency = 12)

#16

nonozone_seasonal_data <- Kendall::MannKendall(nonseasonal_ozone.ts)
summary(nonozone_seasonal_data)
```

```
## Score =  -1179 , Var(Score) = 194365.7
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer: The nonseasonal data has a significant pvalue of 0.00754 and thus we can reject our null hypothesis that ozone levels and seasonality don't have signficance. Both the nonseasonal and seasonal show signficant pvalues and so we can deduce a significant relationship between ozone and seasonality and ozone and other nonseasonal aspects.