Assignment 7: Time Series Analysis

Addie Navarro

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 Monday, March 14 at 7:00 pm.

Set up

1. Set up your session:

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

```
#1
getwd()
```

```
## [1] "Z:/EDA/Environmental_Data_Analytics_2022"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

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

```r
#install.packages("zoo")
library(zoo)
```

```
##
## Attaching package: 'zoo'

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

```r
#install.packages("trend")
library(trend)

#My theme:
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.

```r
#2
EPAair_10_raw <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                          stringsAsFactors = TRUE)

EPAair_11_raw <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                          stringsAsFactors = TRUE)

EPAair_12_raw <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                          stringsAsFactors = TRUE)

EPAair_13_raw <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                          stringsAsFactors = TRUE)

EPAair_14_raw <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                          stringsAsFactors = TRUE)

EPAair_15_raw <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                          stringsAsFactors = TRUE)

EPAair_16_raw <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                          stringsAsFactors = TRUE)

EPAair_17_raw <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                          stringsAsFactors = TRUE)

EPAair_18_raw <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                          stringsAsFactors = TRUE)

EPAair_19_raw <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                          stringsAsFactors = TRUE)
```

```
#combined data frame
GaringerOzone <-
  rbind(EPAair_10_raw, EPAair_11_raw,EPAair_12_raw, EPAair_13_raw,
        EPAair_14_raw, EPAair_15_raw, EPAair_16_raw, EPAair_17_raw, EPAair_18_raw, EPAair_19_raw)
```

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.

```
# 3 Set date column as a date class
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

# 4
GaringerOzone_processed <-
  GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration,DAILY_AQI_VALUE)

# 5
Days <-
as.data.frame(seq.Date(as.Date("2010/01/01"), as.Date("2019/12/31"), "day"))
colnames(Days) <- c("Date")

# 6
GaringerOzone <- left_join(Days, GaringerOzone_processed, by = "Date")
```

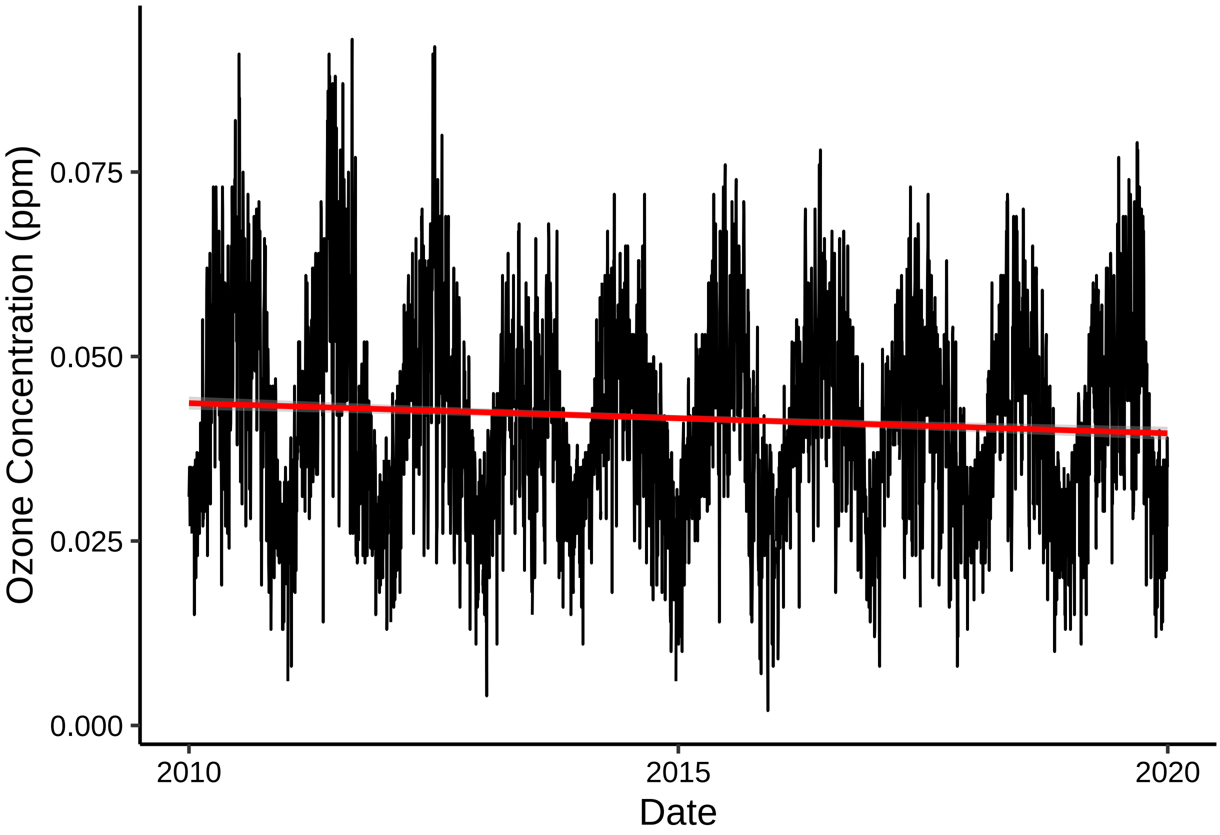
Visualize

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
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration )) +
  geom_line() +
  geom_smooth(method = "lm", color = "red") +
  labs(x = "Date", y = "Ozone Concentration (ppm)")
```

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

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



Answer: The plot suggests a cyclical trend in ozone concentration over time with ozone concentrations repeatedly rising and falling. It does not show a linear trend.

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)Checking number of NA's
GaringerOzone <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))%>%
  mutate(DAILY_AQI_VALUE = zoo::na.approx(DAILY_AQI_VALUE))
#summary(GaringerOzone)NA's are interpolated
```

Answer: We used a linear interpolation instead of a piecewise constant or spline interpolation because the missing data points were few and close together, so the linear interpolation would be most accurate. The piecewise constant would not be accurate for ozone concentration as it would assume the missing data to be equal to the measurements nearest to it. The Spline interpolation would use a quadratic formula instead of a linear interpolation and therefore may be less accurate than the linear interpolation as the missing data points are assumed to be within the previous and next measurements.

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 <-
  GaringerOzone %>%
  mutate(Month = month(Date),
         Year = year(Date))%>%
  mutate(Date = my(paste0(Month, "-", Year)))%>%
  dplyr::group_by(Date, Month, Year)%>%
  dplyr::summarise(mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration))%>%
  select(mean_Ozone, Date)
```

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

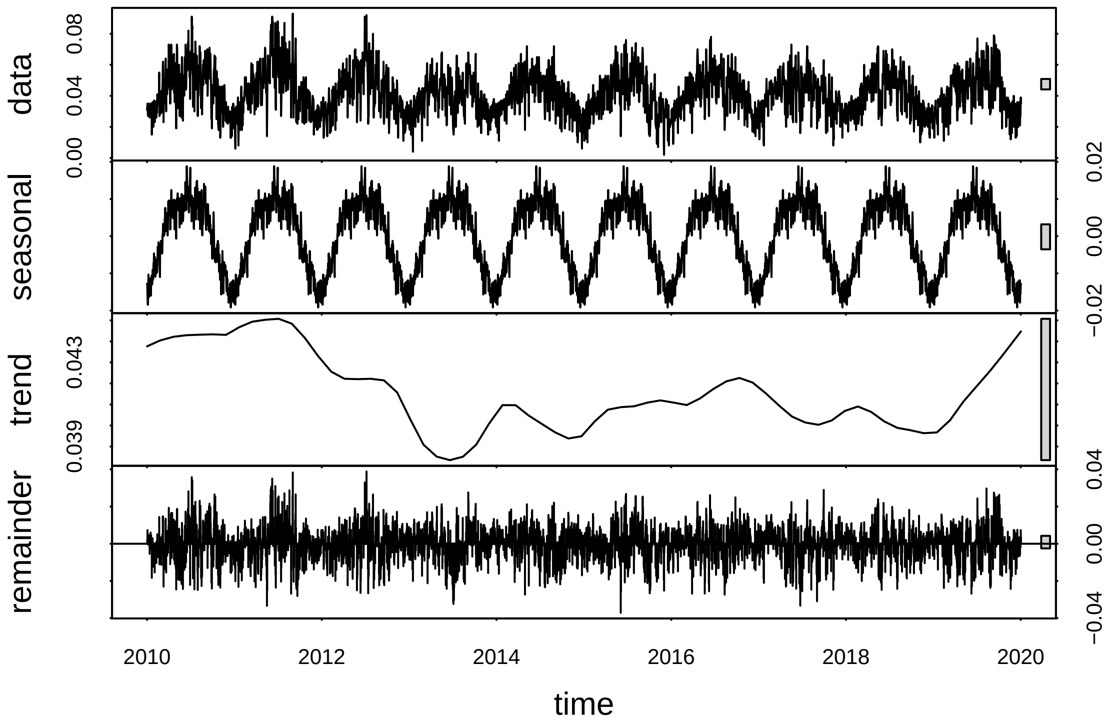
```
## Adding missing grouping variables: `Month`
```

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.

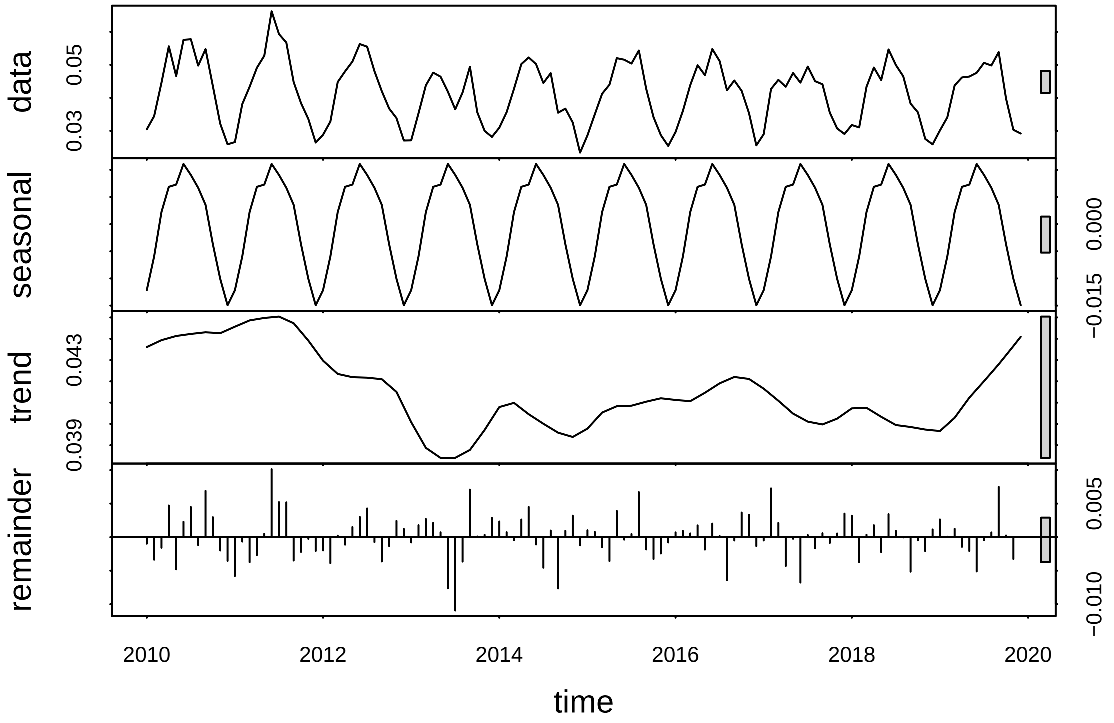
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010,1), frequency = 365)

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

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

```
#11
GaringerOzone.daily.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.decomposed)
```



```
GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GaringerOzone.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
monthly_ozone_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

#Looking at results
monthly_ozone_trend
```

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

```
summary(monthly_ozone_trend)
```

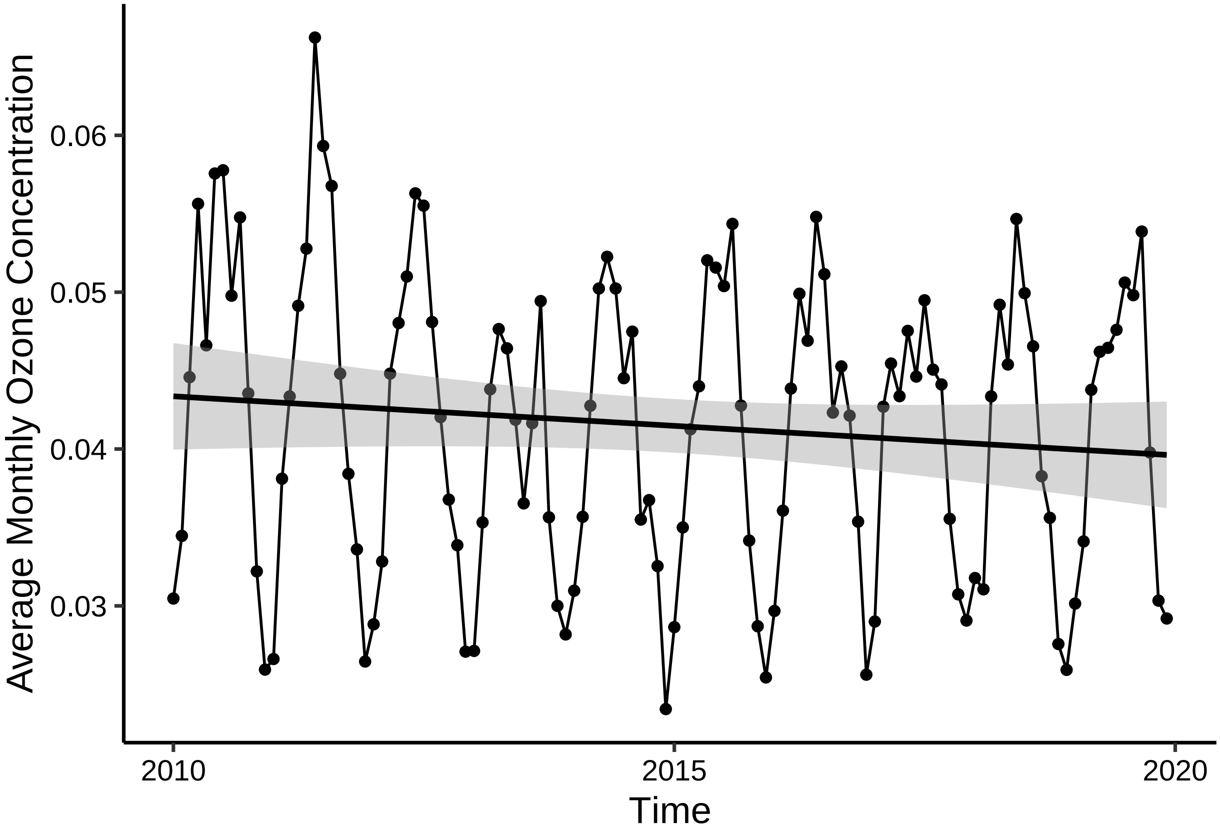
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: the Seasonal Mann-Kendall is the only monotonic trend analysis test that can handle seasonality, so it is the most appropriate test for this data. The result of this test gives us a pvalue less than .05, therefore we can reject the null hypothesis that the data is stationary.

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
mean_monthly_ozone_plot <-
  ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_Ozone)) +
  geom_point()+
  geom_line()+
  ylab("Average Monthly Ozone Concentration")+
  xlab("Time")+
  geom_smooth(method = lm, color = "black")
print(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.

Research Question: Have ozone concentrations changed over the 2010s at this station?

> Answer: The Null Hypothesis for the Mann-Kendall test states that the data is stationary and ozone concentrations have not changed over time. The result of the Seasonal Mann-Kendall test is that the pvalue is less than .05, so we can reject the null hypothesis and state that there is a trend and ozone concentrations have changed over the 2010s at this station. According to the Seasonal Mann-Kendall test, there is a slight negative trend, so ozone concentration levels have slightly decreased over the 2010s at this station (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
#subtracting seasonal component
GaringerOzone.monthly.ts.components <-
  as.data.frame(GaringerOzone.monthly.decomposed$time.series[,2:3])
#added trend and remainder into observed column
GaringerOzone.monthly.ts.components <-
  mutate(GaringerOzone.monthly.ts.components,
         Observed = GaringerOzone.monthly.ts.components$trend
         + GaringerOzone.monthly.ts.components$remainder,
         Date = GaringerOzone.monthly$Date)

#16
#Non-seasonal Mann-Kendall
fmonth <- month(first(GaringerOzone.monthly.ts.components$Date))
fyear <- year(first(GaringerOzone.monthly.ts.components$Date))

GaringerOzone.monthly.trend2.ts <-
  ts(GaringerOzone.monthly.ts.components$Observed,
     start = c(fmonth, fyear), frequency = 12)

GaringerOzone.monthly.trend2 <-
  Kendall::MannKendall(GaringerOzone.monthly.trend2.ts)
summary(GaringerOzone.monthly.trend2)
```

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

Answer: The results of the non-seasonal Mann-Kendall show a lower pvalue and a stronger negative trend. The pvalue for this test is .0075402, less than .05, therefore we can reject the null hypothesis that the Ozone concentration is stationary over time. This test shows a Score of -1179, a stronger negative trend than the seasonal Mann-Kendall, showing that the ozone concentration is decreasing over the 2010s.