Assignment 8: Time Series Analysis

Erin Ansbro

Fall 2023

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

```
#setting up workspace
getwd()
```

```
## [1] "C:/Users/eka19/OneDrive - Duke University/Documents/872/EDE_Fall2023"
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## 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)

#2 setting a theme
mytheme <- theme_classic(base_size = 13) +
  theme(axis.text = element_text(color = "navy"),
        legend.position = "bottom")
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.

```
#2 reading in data
Ozone2010 <-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                     stringsAsFactors=TRUE)

Ozone2011 <-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                     stringsAsFactors=TRUE)

Ozone2012 <-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                     stringsAsFactors=TRUE)

Ozone2013 <-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                     stringsAsFactors=TRUE)

Ozone2014 <-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",

                      stringsAsFactors=TRUE)

Ozone2015 <-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                     stringsAsFactors=TRUE)

Ozone2016 <-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                     stringsAsFactors=TRUE)

Ozone2017 <-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                     stringsAsFactors=TRUE)

Ozone2018 <-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                     stringsAsFactors=TRUE)
```

```
Ozone2019 <-read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                     stringsAsFactors=TRUE)


GaringerOzone <- rbind(Ozone2010, Ozone2011, Ozone2012, Ozone2013, Ozone2014, Ozone2015,
                       Ozone2016, Ozone2017, Ozone2018, Ozone2019)
```

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.

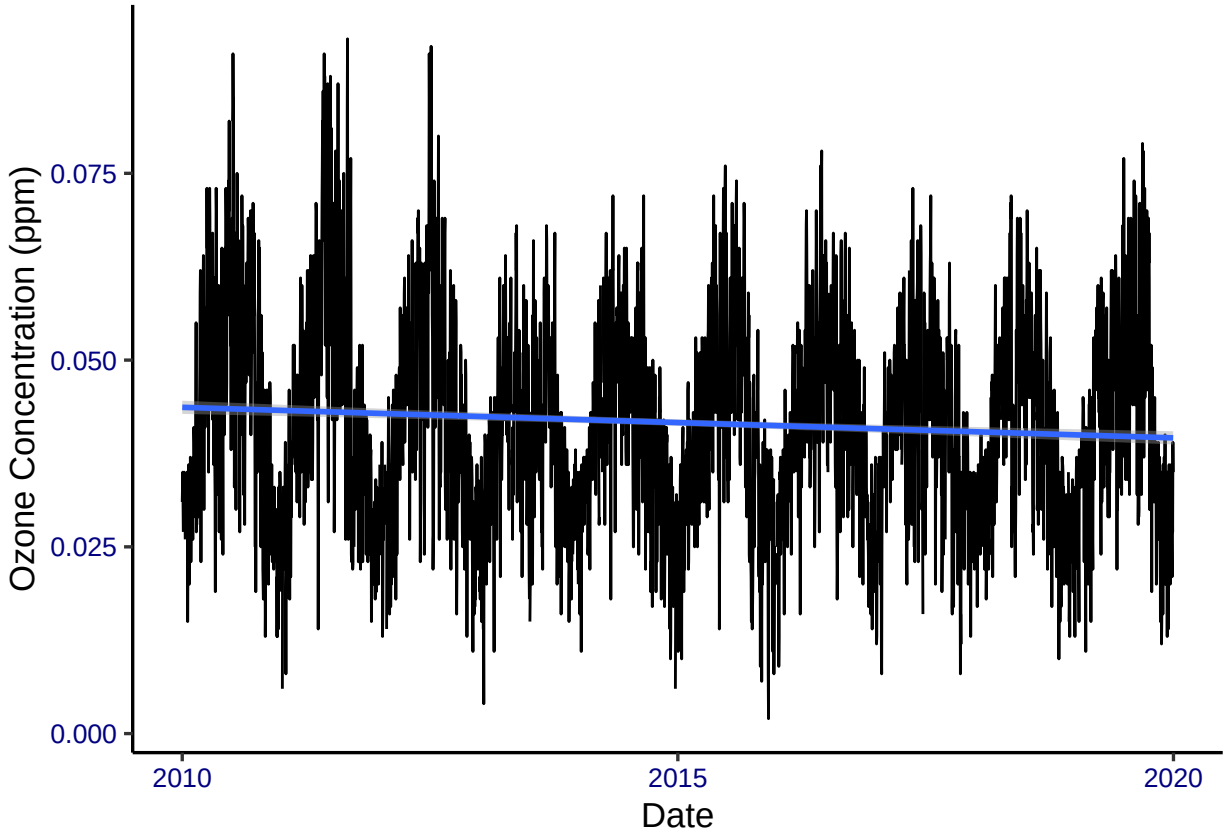
```
# 3 Setting date class
GaringerOzone$Date <-as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

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

# 5 Filling in missing data
Days <- as.data.frame(seq(from = as.Date("2010-01-01"), to=as.Date("2019-12-31"),
                          by= "day"))
colnames(Days)<-"Date"

# 6 Joining data
GaringerOzone <- left_join(Days, GaringerOzoneSelect, 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 Ozone over time plot
GaringerPlot <- ggplot(GaringerOzone, aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  geom_smooth(method = lm)+
  ylab("Ozone Concentration (ppm)")

print(GaringerPlot)
```

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

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



Answer: The plot suggests there is a slight decrease trend over time from 2010 to 2019.

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 Linear Interpolation
head(GaringerOzone)
```

```
##         Date Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
## 1 2010-01-01                                0.031              29
## 2 2010-01-02                                0.033              31
## 3 2010-01-03                                0.035              32
## 4 2010-01-04                                0.031              29
## 5 2010-01-05                                0.027              25
## 6 2010-01-06                                   NA              NA
```

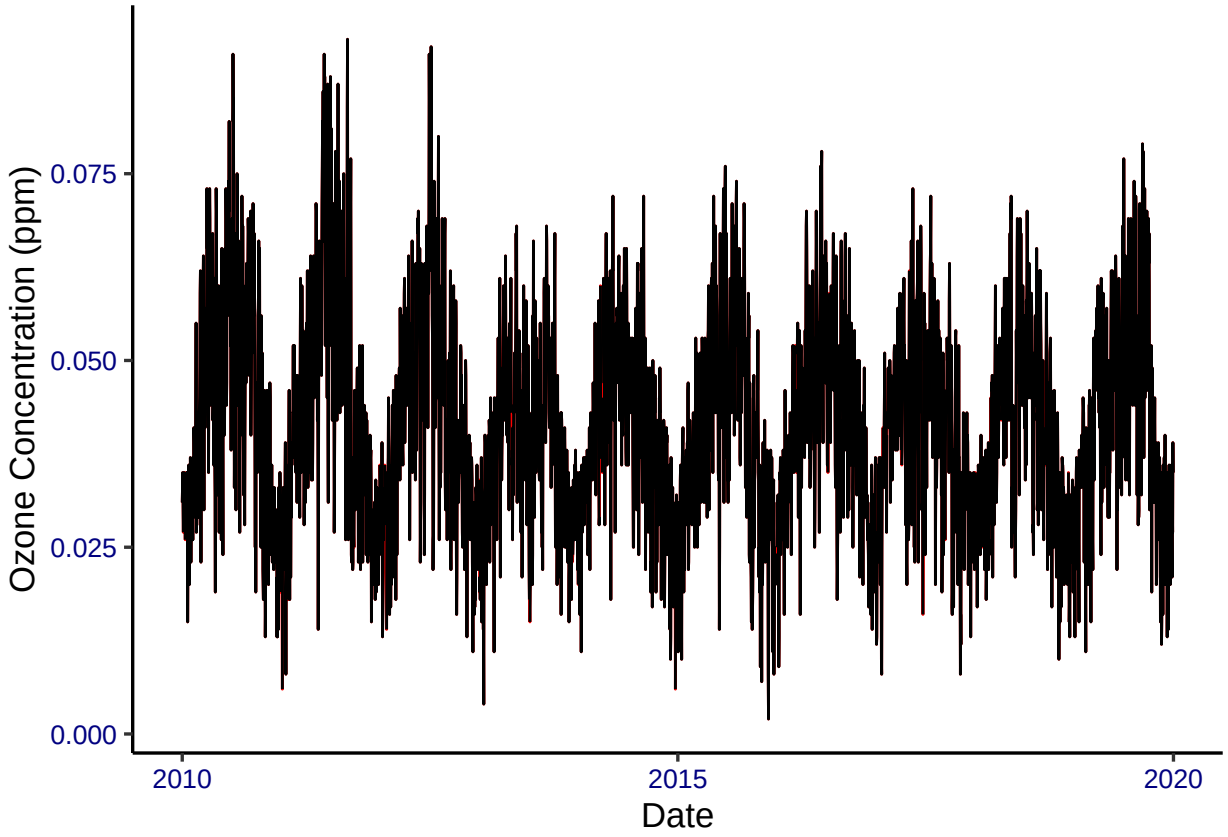
```r
summary(GaringerOzone)
```

```
##       Date             Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                       NA's   :63                           NA's   :63
```

```r
GaringerOzoneClean <- GaringerOzone %>%
  mutate(ppm.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

summary(GaringerOzoneClean$ppm.clean)
```

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

```r
OzoneInterpolation <- ggplot(GaringerOzoneClean) +
  geom_line(aes(x = Date, y = ppm.clean), color = "red") +
  geom_line(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration), color = "black") +
  ylab("Ozone Concentration (ppm)")

print(OzoneInterpolation)
```



Answer: It does appear ozone concetrations have changed over time. In the early 2010s there were higher levels, reaching above .075 ppm. Overtime, those highs have lowered to right at or under .075 ppm. A linear interpolation was used because we want to know the missing values between the dots. Spline was not used because we are not using a quadratic equation to find the distances, just a straight line. Piecewise was not used because we are connecting by the past and previous data points, we do not need a peicewise function that has different values for different steps.

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)

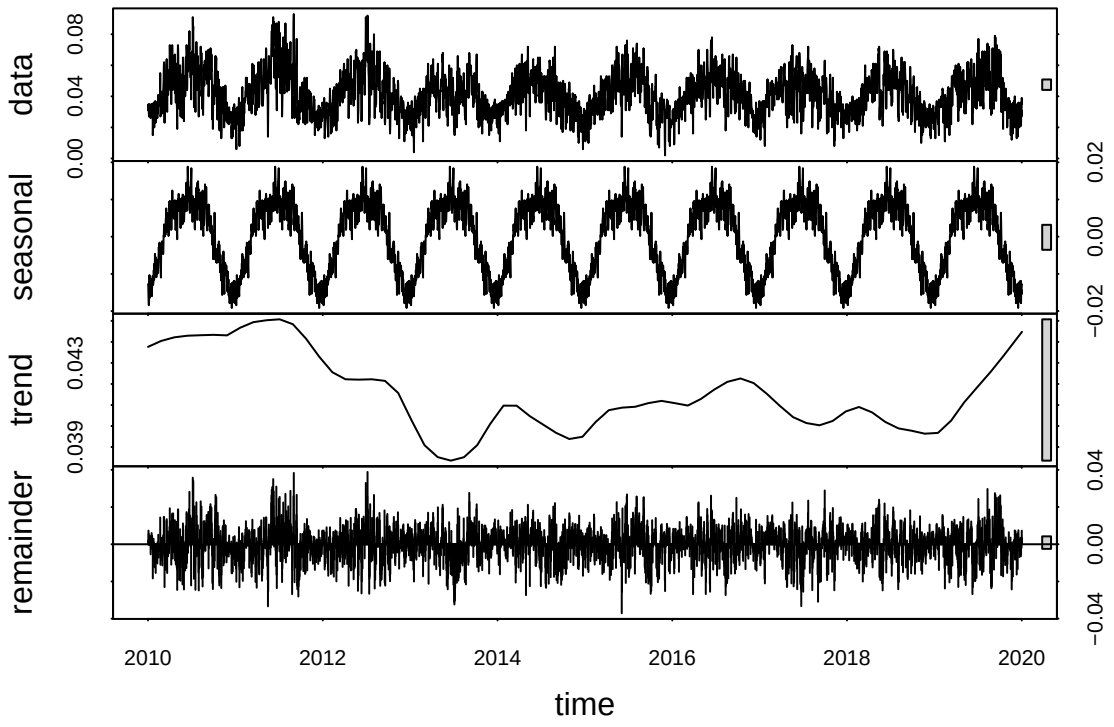
```
#9 Creating monthly average data
GaringerOzone.monthlypreliminary <- GaringerOzoneClean%>%
  mutate(GaringerOzoneClean, Month=month(Date))%>%
  mutate(GaringerOzoneClean, Year=year(Date))%>%
  mutate(CleanDate = my(paste0(Month,"-",Year)))

GaringerOzone.monthly <-aggregate(GaringerOzone.monthlypreliminary$ppm.clean,
                                  by=list(GaringerOzone.monthlypreliminary$CleanDate),
                                  FUN=mean)

colnames(GaringerOzone.monthly) <- c("Date", "MeanPPM")
```

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.

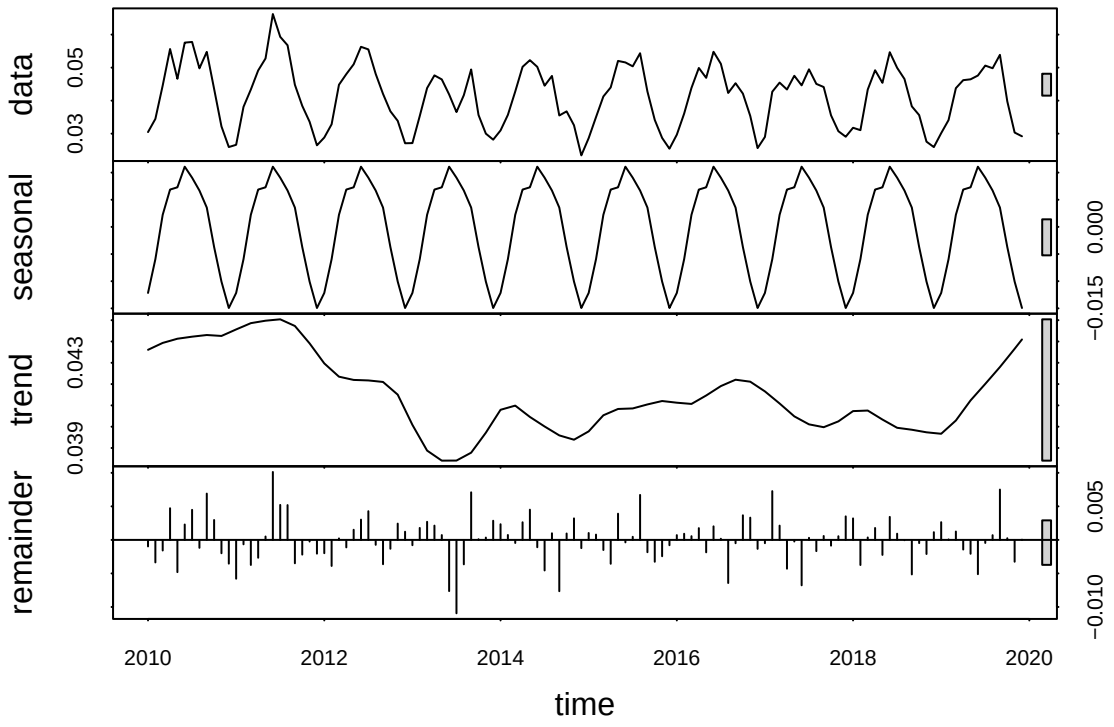
```
#10 Generate time series daily
f_monthdaily <- month(first(GaringerOzoneClean$Date))
f_yeardaily <- year(first(GaringerOzoneClean$Date))

GaringerOzone.daily.ts <- ts(GaringerOzoneClean$ppm.clean,
                             start=c(f_yeardaily,f_monthdaily),
                             frequency=365)

# Generate time series monthly
f_monthmonthly <- month(first(GaringerOzone.monthly$Date))
f_yearmonthly <- year(first(GaringerOzone.monthly$Date))

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$MeanPPM,
                   start=c(f_yearmonthly,f_monthmonthly),
                   frequency=12)
```

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

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




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



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 Run SMK test monthly
Ozone_Monthly_SMK <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

Ozone_Monthly_SMK
```

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

```
summary(Ozone_Monthly_SMK)
```

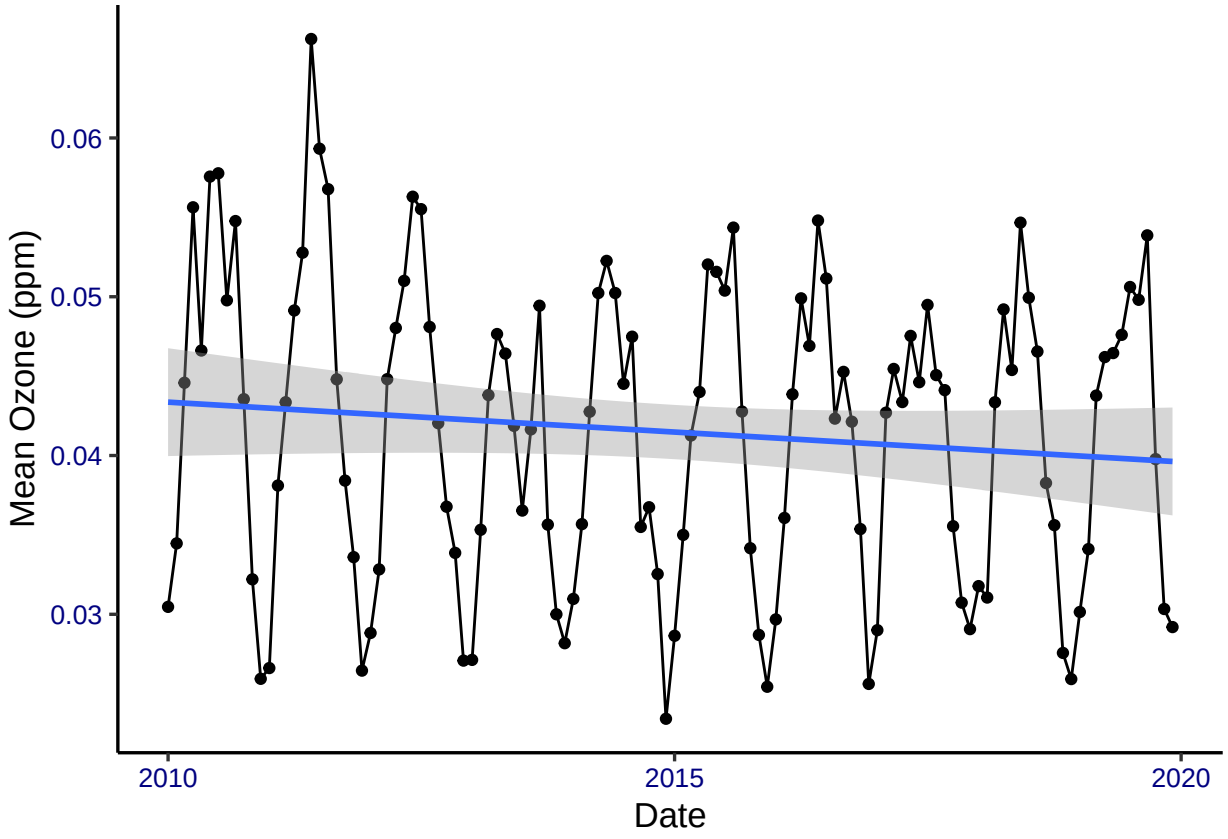
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: From the Decompose Plots, I see there is strong seasonality. Therefore, I need a trend test that includes seasonality in its calculations. In addition, missing data is allowed in a SM test.

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 Visualization
Ozone_Plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = MeanPPM)) +
  geom_point() +
  geom_line() +
  ylab("Mean Ozone (ppm)") +
  geom_smooth( method = lm )

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

    Ozone concentrations have changed over the 2010s at this station. Visually, you can see the linear regression line decrease moderately from 2010 to 2019. Looking at the summary statistics, the p-value is <.05 at about .04, meaning the null hypothesis that ozone does not change should be rejected in favor that ozone does change. (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
# Removing seasonality
Ozone_NoSeasonal <-GaringerOzone.monthly.ts.decomp$time.series[,2:3]

#16 Run MK test monthly
Ozone_Monthly_MK <- Kendall::MannKendall(Ozone_NoSeasonal)

Ozone_Monthly_MK
```

```
## tau = -0.568, 2-sided pvalue =< 2.22e-16
```

```
summary(Ozone_Monthly_MK)
```

```
## Score =  -16300 , Var(Score) = 1545533
## denominator =  28680
## tau = -0.568, 2-sided pvalue =< 2.22e-16
```

Answer: The results appear to be stronger in regards that ozone has decreased over time. The p-value is now at <2.22e-16 instead of about .04. This means the null hypothesis should be rejected. (Score = -16300 , Var(Score) = 1545533 denominator = 28680 tau = -0.568, 2-sided pvalue =< 2.22e-16)