Assignment 8: Time Series Analysis

Susanna Jenkins

Spring 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

```
#1

getwd()
```

```
## [1] "/home/guest/EDA-Spring2023"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.0
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(trend)
library(zoo)
```

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

```
library(Kendall)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(here)
```

```
## here() starts at /home/guest/EDA-Spring2023
```

```
here
```

```
## function (...)
## {
##     .root_env$root$f(...)
## }
## <bytecode: 0x559669863440>
## <environment: namespace:here>
```

```
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

Ozone_10 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
                     stringsAsFactors = TRUE)

Ozone_11 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
                     stringsAsFactors = TRUE)

Ozone_12 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                     stringsAsFactors = TRUE)

Ozone_13 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                     stringsAsFactors = TRUE)

Ozone_14 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                     stringsAsFactors = TRUE)

Ozone_15 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                     stringsAsFactors = TRUE)

Ozone_16 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                     stringsAsFactors = TRUE)

Ozone_17 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
                     stringsAsFactors = TRUE)

Ozone_18 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                     stringsAsFactors = TRUE)

Ozone_19 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                     stringsAsFactors = TRUE)

GaringerOzone <- rbind(Ozone_10,Ozone_11,Ozone_12,Ozone_13,Ozone_14,
                            Ozone_15,Ozone_16,Ozone_17,Ozone_18,Ozone_19)
```

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
```

```r
GaringerOzone$Date <- mdy(GaringerOzone$Date)

#4

GaringerOzone_filter <- select(GaringerOzone, Date,
                               Daily.Max.8.hour.Ozone.Concentration,
                               DAILY_AQI_VALUE)

#5

Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"),
                          by = "day"))

colnames(Days) <- c("Date")

#6

GaringerOzone <- left_join(Days, GaringerOzone_filter)
```

```
## Joining with `by = join_by(Date)`
```

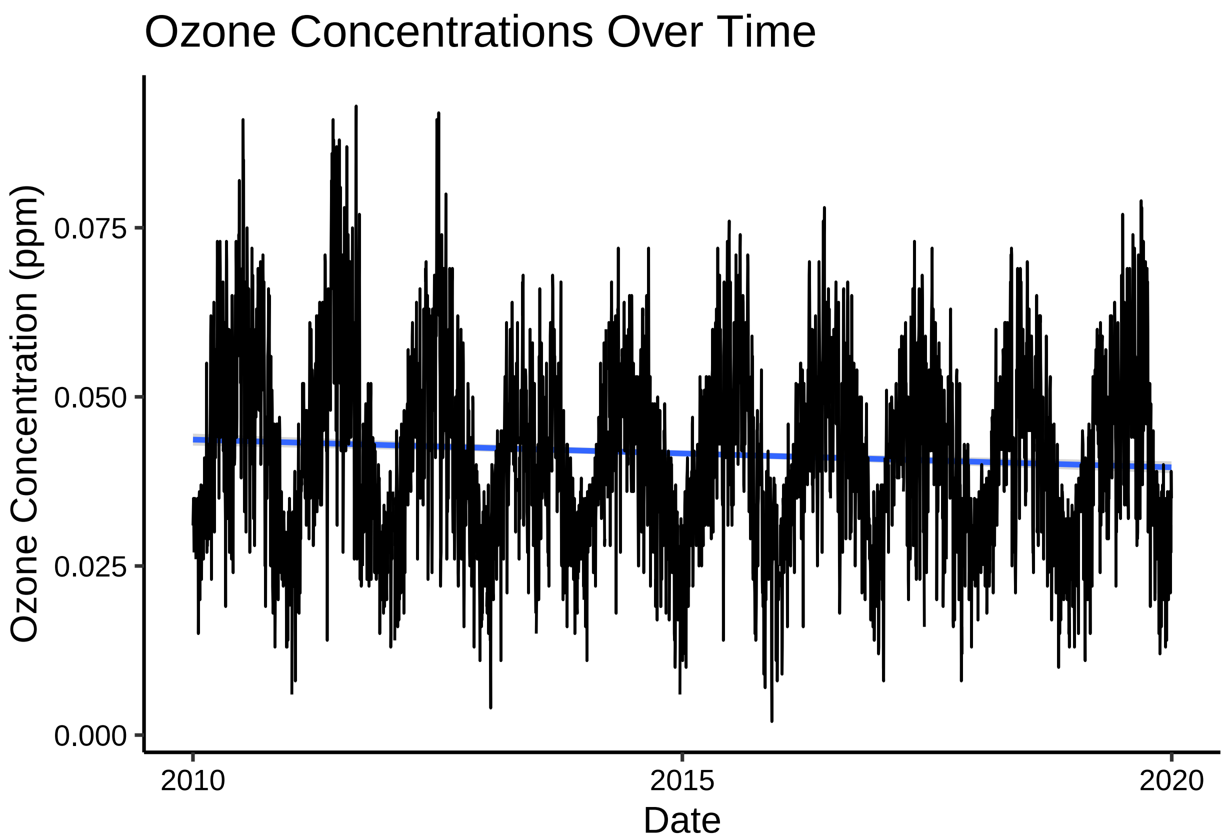
Visualize

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?

```r
#7

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

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

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



Answer: The line is only slightly going down over time. We would need to run analyses to determine if this decreasing trend is significant.

Time Series Analysis

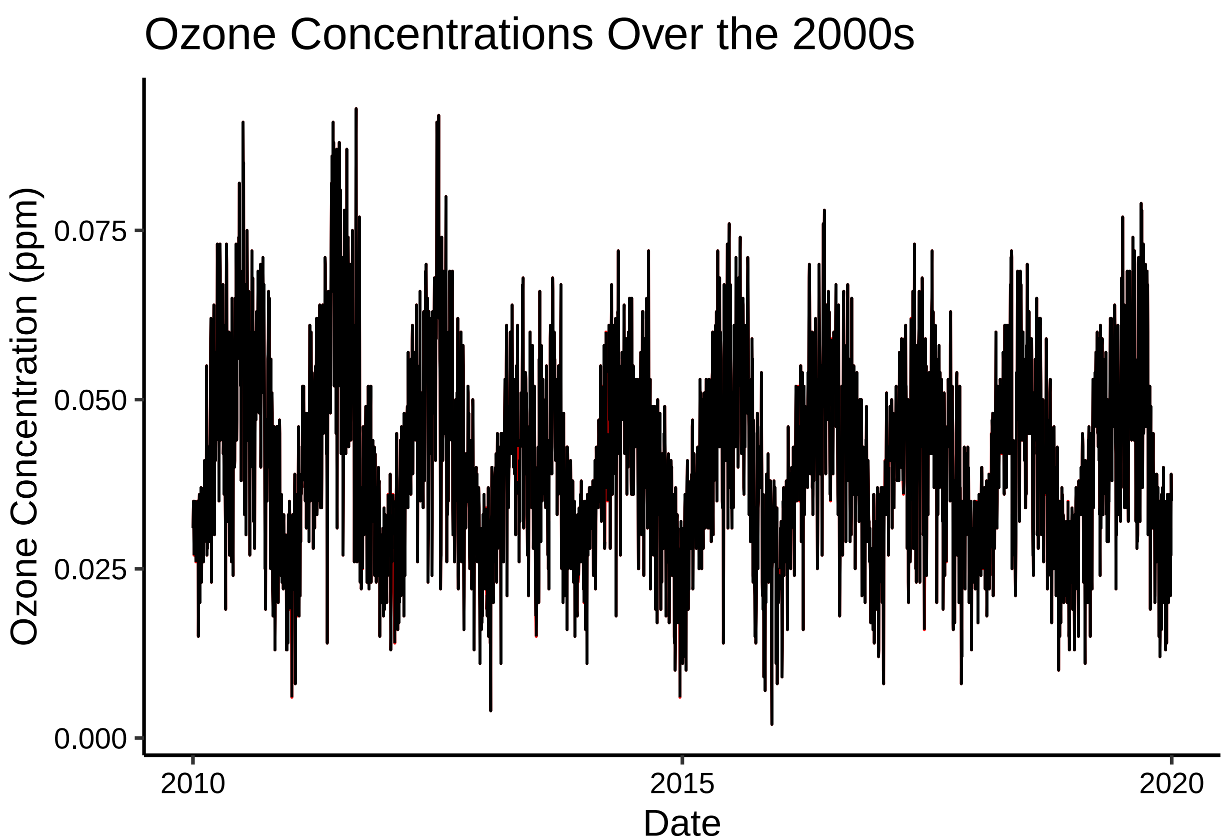
Study question: Have ozone concentrations changed over the 2010s at this station?

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

GaringerOzone_clean <-
  GaringerOzone %>%
  mutate(GaringerOzone, Daily.Max.8.hour.Ozone.Concentration.clean =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration) )

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

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

```r
ggplot(GaringerOzone_clean) +
  geom_line(aes(x=Date, y = Daily.Max.8.hour.Ozone.Concentration.clean), color="red") +
  geom_line(aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration), color="black") +
  scale_x_date() +
  labs(x="Date", y="Ozone Concentration (ppm)",
       title="Ozone Concentrations Over the 2000s")
```



Answer: A linear interpolation is appropriate here because we can see that observed ozone concentrations are changing over time "between the dots" meaning any missing data would reasonably fall between the previous and next measurement when observed on a daily/monthly frequence. Thus, we are best off interpolating the data by drawing a straight line between measurements because we wouldn't want to assume the missing ozone concentration observation is equal to the nearest measurement observed (Piecewise constant). Similarly, there is nothing indicating that this is a quadratic so we do not need to interpolate by using a quadratic function (Spline).

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)

```r
#9

GaringerOzone.newdate <- GaringerOzone_clean %>%
  mutate(GaringerOzone_clean, Year = year(Date), Month = month(Date))
```

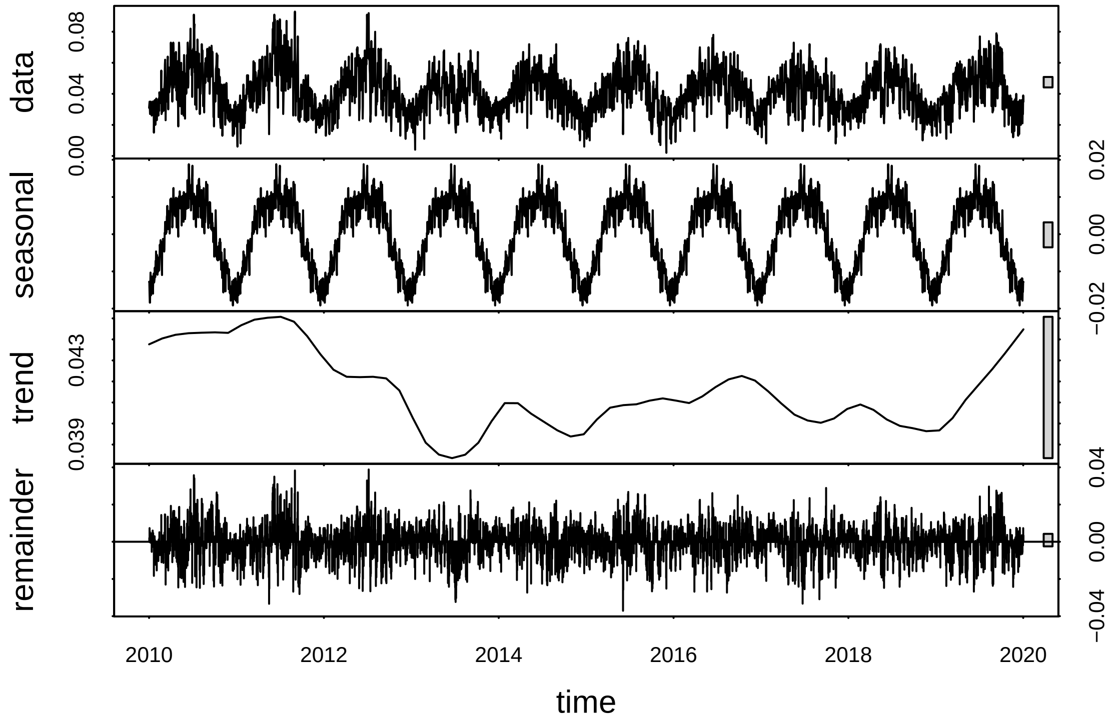
```
GaringerOzone.monthly <- GaringerOzone.newdate %>%
  mutate(GaringerOzone_clean, Date = ym(paste0(Year,"-",Month))) %>%
  group_by(Date) %>%
  summarise(Daily.Max.8.hour.Ozone.Concentration.mean =
              mean(Daily.Max.8.hour.Ozone.Concentration.clean))
```

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.

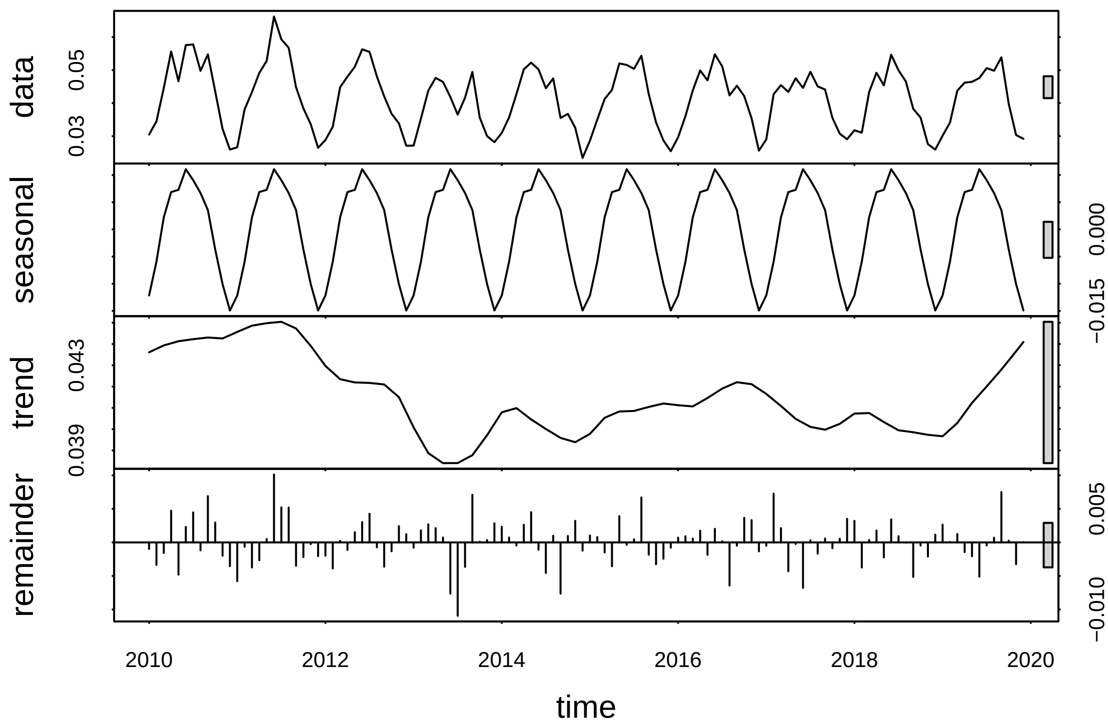
```
#10

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

GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$Daily.Max.8.hour.Ozone.Concentration.mean,
     frequency=12, start=c(2010,1))
```

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

```
#11

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




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

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

GaringerOzone.monthly.SMK
```

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

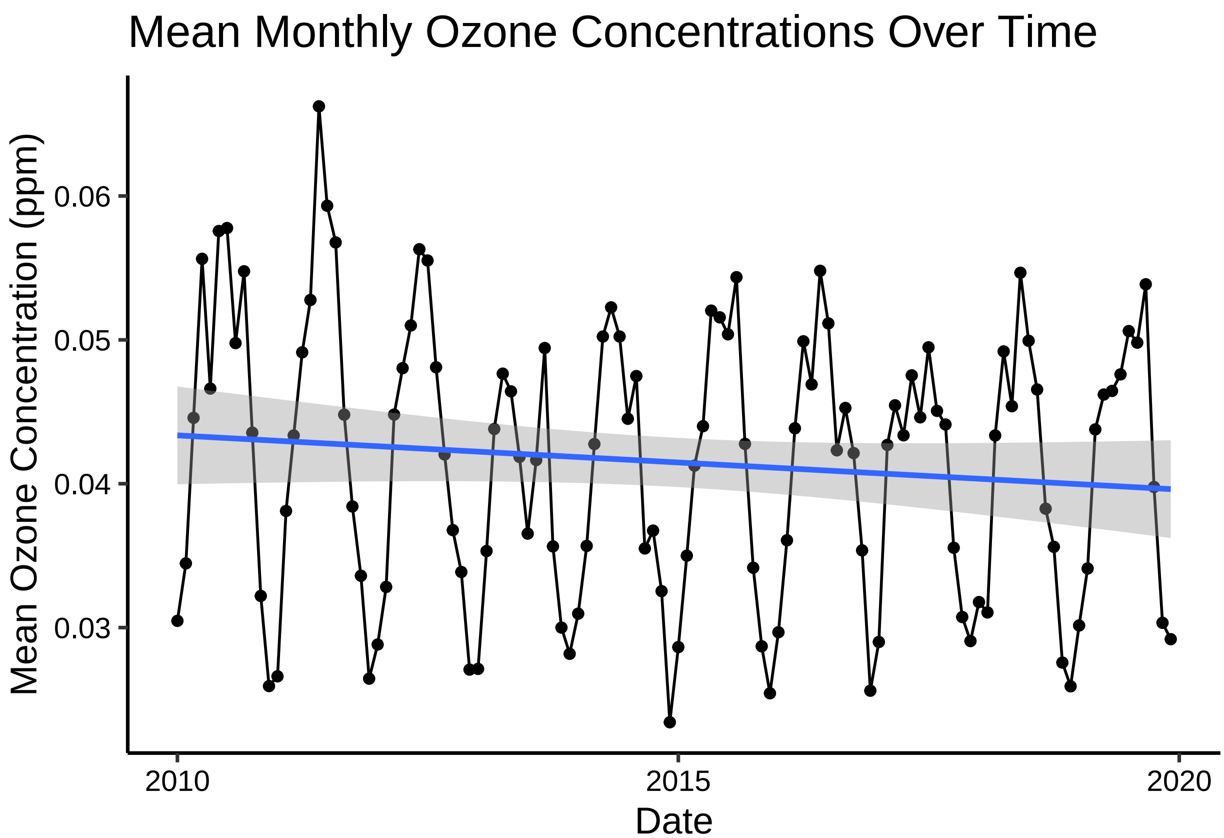
Answer: A Seasonal Mann-Kendall is appropriate here because ozone concentrations are affected by seasonality and its non-parametric. You don't use the regular Mann Kendall because there are seasonal trends in the data.

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

GaringerOzone.monthly.plot <-
  ggplot(GaringerOzone.monthly,
         aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration.mean)) +
  geom_point() +
```

```
  geom_line() +
  geom_smooth( method = lm ) +
  labs(x = "Date", y = "Mean Ozone Concentration (ppm)",
       title = "Mean Monthly Ozone Concentrations Over Time")

print(GaringerOzone.monthly.plot)
```

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



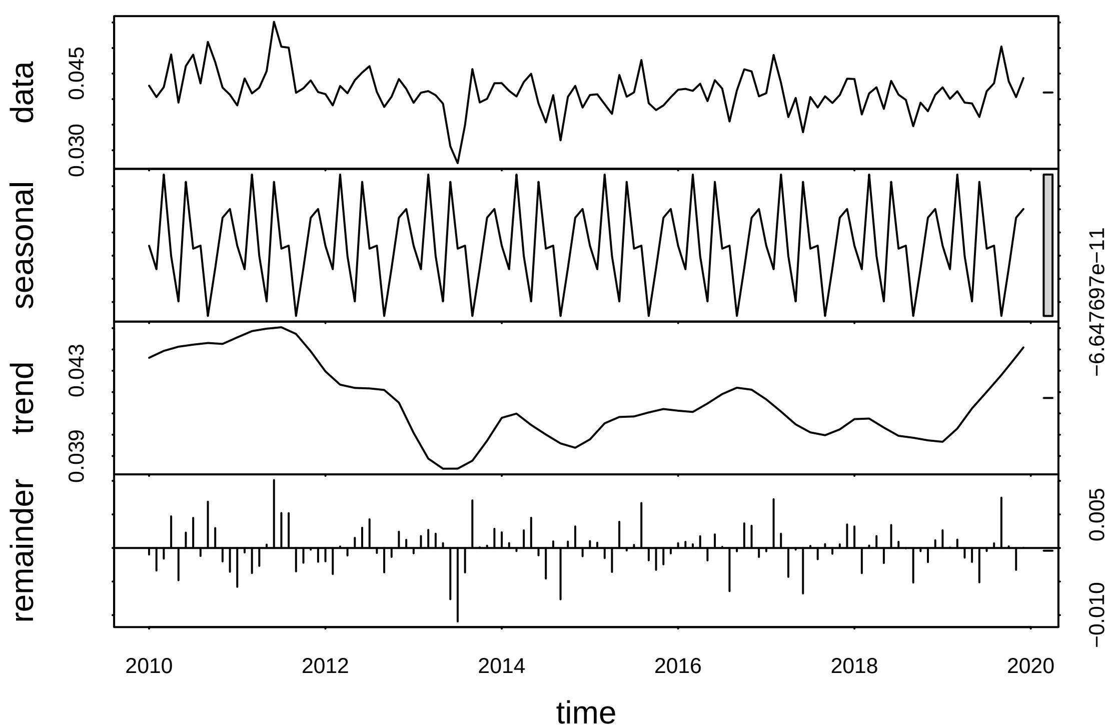
14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

    Answer: The p-value = 0.046724. Seen in the graph, the results show there are seasonal oscillations in ozone concentration over time, and the trend is slightly significantly decreasing over time.

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

#CORRECT ALTERNATIVE WAY FROM OFFICE HOURS:

#GaringerOzone.monthly.redo <- GaringerOzone.monthly.ts - GaringerOzone.monthly.decomp$time.series[,1]

#GaringerOzone.monthly.nonseasonality.MKredo <- Kendall::MannKendall(GaringerOzone.monthly.redo)
#GaringerOzone.monthly.nonseasonality.MKredo

GaringerOzone.monthly.extract <- as.data.frame(GaringerOzone.monthly.decomp$time.series[,1:3])

GaringerOzone.monthly.extract <- mutate(GaringerOzone.monthly.extract,
                          Observed=GaringerOzone.monthly$Daily.Max.8.hour.Ozone.Concentration.mean,
                          Date=GaringerOzone.monthly$Date)

GaringerOzone.monthly.noseasonality <- GaringerOzone.monthly.extract[,4] - GaringerOzone.monthly.extrac

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

GaringerOzone.monthly.noseasonality.decomp <- stl(GaringerOzone.monthly.noseasonality.ts,s.window="peri
plot(GaringerOzone.monthly.noseasonality.decomp)
```




```
#16
```

```
GaringerOzone.monthly.nonseasonality.MK <-
  Kendall::MannKendall(GaringerOzone.monthly.noseasonality.ts)

GaringerOzone.monthly.nonseasonality.MK
```

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

Answer: The initial p-value = 0.046724 which is very slightly significant. Once you remove the seasonality, the p-value = 0.0075402, meaning the trend becomes much more significant.