Assignment 8: Time Series Analysis

Tasneem Ahsanullah

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
#checking working directory, it's correct
getwd()
```

```
## [1] "/Users/tasneemahsanullah/Desktop/Classes/EDA/DataAnalytics"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- 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
```

```
library(zoo)
```

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

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

```
## here() starts at /Users/tasneemahsanullah/Desktop/Classes/EDA/DataAnalytics
```

```
here()
```

```
## [1] "/Users/tasneemahsanullah/Desktop/Classes/EDA/DataAnalytics"
```

```
#customizing ggplot theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top",
    plot.background = element_rect(
      color='black',
      fill='plum1')
    )
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
#importing 10 Ozone datasets
EPA_2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
                         stringsAsFactors = TRUE)

EPA_2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
                    stringsAsFactors = TRUE)
```

```
EPA_2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                     stringsAsFactors = TRUE)

EPA_2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                     stringsAsFactors = TRUE)

EPA_2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                     stringsAsFactors = TRUE)

EPA_2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                     stringsAsFactors = TRUE)

EPA_2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                     stringsAsFactors = TRUE)

EPA_2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
                     stringsAsFactors = TRUE)

EPA_2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                     stringsAsFactors = TRUE)

EPA_2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                     stringsAsFactors = TRUE)

#combining the the 10 datasets into 1
GaringerOzone <- rbind(EPA_2010,EPA_2011,EPA_2012,EPA_2013,EPA_2014,
                       EPA_2015,EPA_2016,EPA_2017,EPA_2018,EPA_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.

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 column as class
GaringerOzone$Date <- mdy(GaringerOzone$Date)

#4
#wrangling dataset to only have columns: Date, Daily.Max.8.hour.Ozone.Concentration,
#and DAILY_AQI_VALUE
GaringerOzone.subset <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
```

```
#5
#new data frame with sequence of dates from 2010-01-01 to 2019-12-31,
#renamed column name in Days to "Date"
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by= "1 day"))
colnames(Days) <- "Date"

#6
#joining the Days and GaringerOzone.subset data frames into one data frame
GaringerOzone <- left_join(Days,GaringerOzone.subset)
```

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

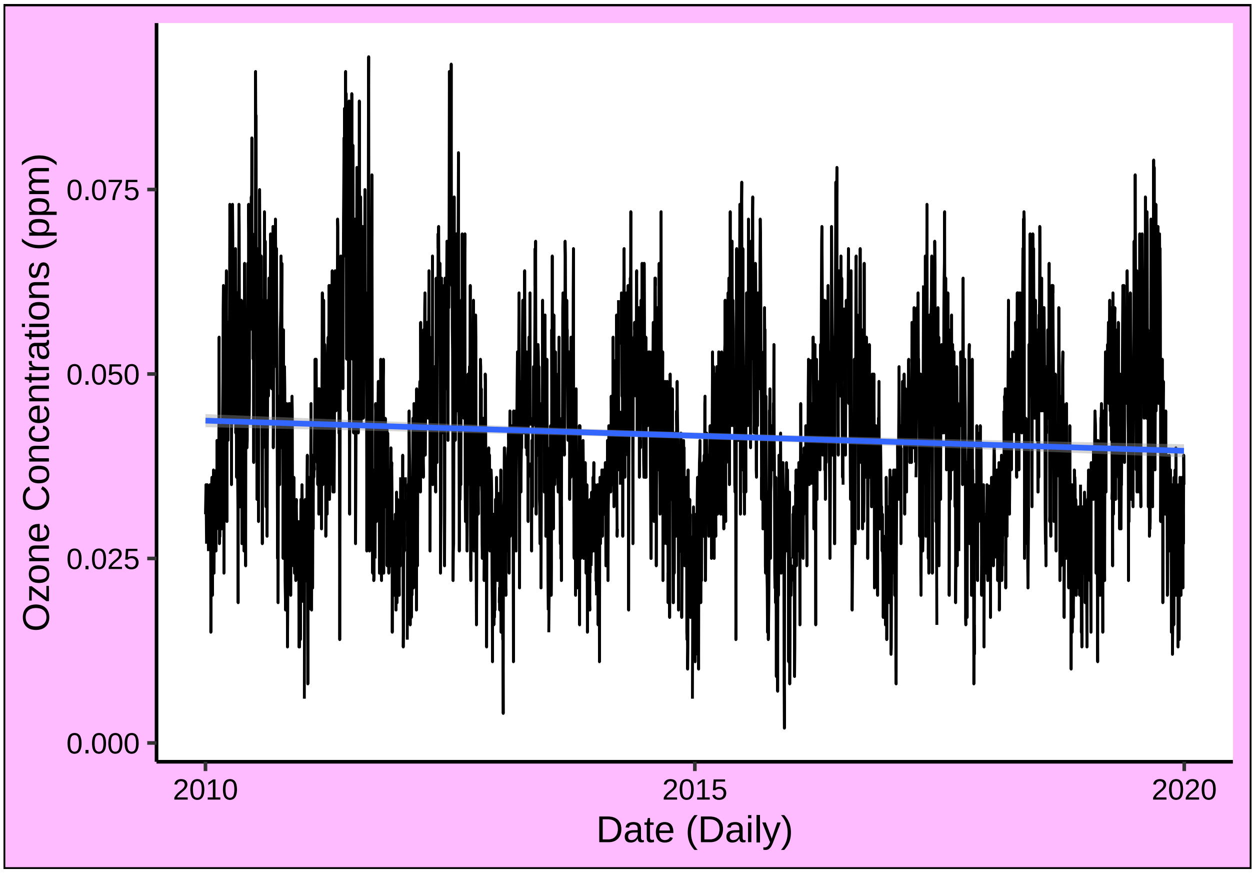
Visualize

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?

```
#7
#line plot of ozone concentrations over time
ggplot(GaringerOzone,aes(x=Date,y=Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method='lm') +
  xlab("Date (Daily)") +
  ylab("Ozone Concentrations (ppm)")
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

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



Answer: The plot suggest there is a negative trend in ozone concentration over time. So as time increases ozone concentrations decrease overall.

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?

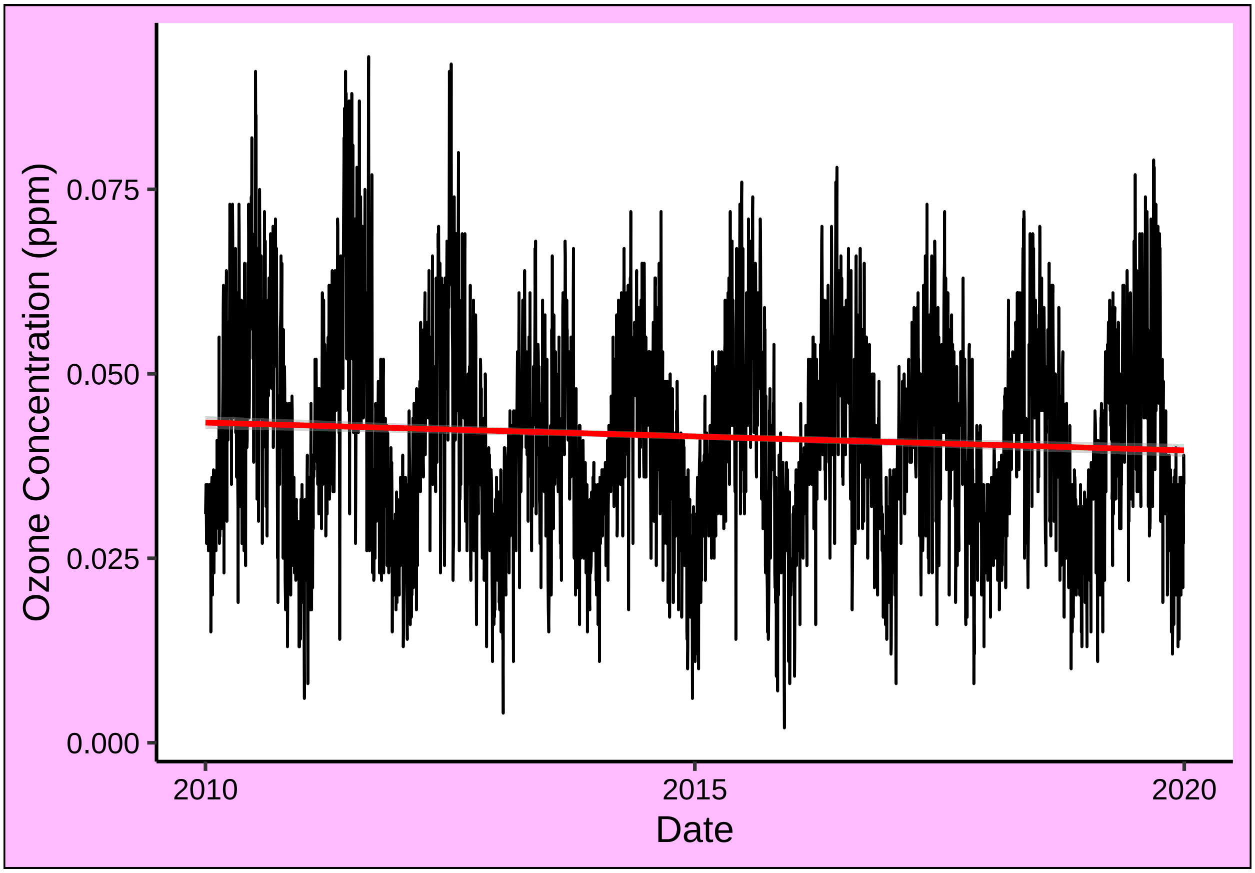
```
#8
#Using na.approx to do linear interpolation to fill in the NAs
GaringerOzone.clean <-
  GaringerOzone %>%
  mutate(Ozone.clean = na.approx(Daily.Max.8.hour.Ozone.Concentration))

#NAs are gone from Ozone concentrations
summary(GaringerOzone.clean$Ozone.clean)
```

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

```
#plot of interpolated Ozone concentration and Date
ggplot(GaringerOzone.clean) +
  geom_line(aes(x = Date, y = Ozone.clean), color = "black") +
  geom_smooth(aes(x = Date, y = Ozone.clean), color = "red", method='lm')  +
  ylab("Ozone Concentration (ppm)")
```

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



Answer: The Ozone concentrations have decreased over the 2010s at this station. We used linear interpolation instead of peicewise constant because the data is not locally constant.

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)

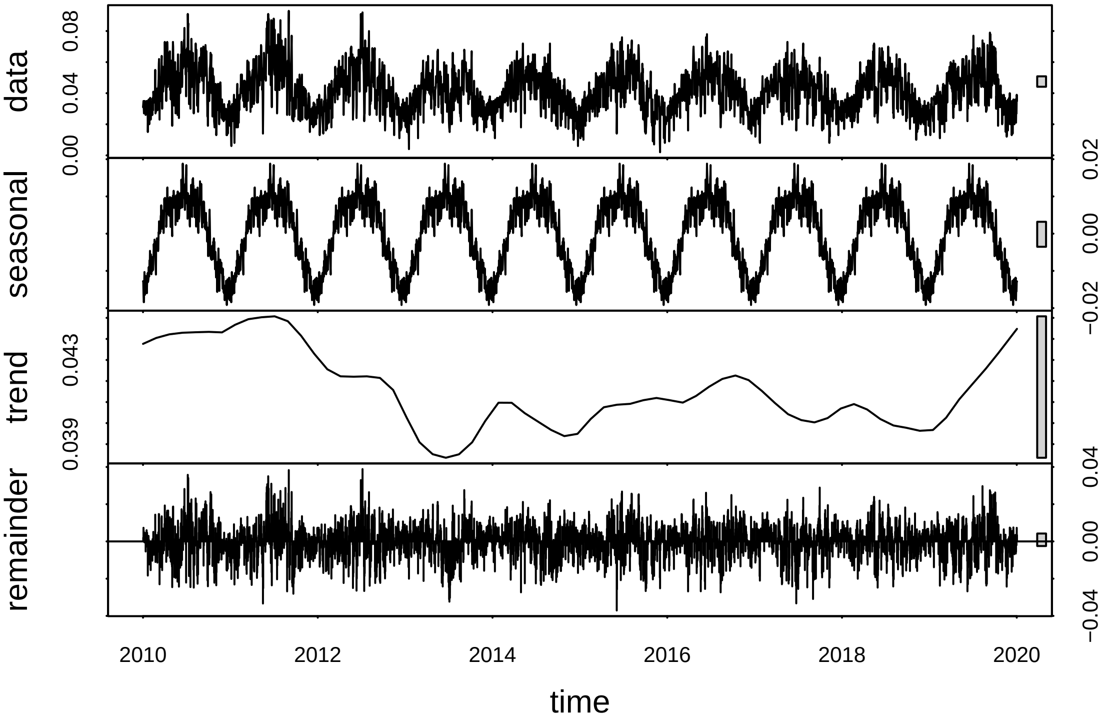
```
#9
#aggregated data by year then month creating new year and month columns, new date column
GaringerOzone.monthly <-
  GaringerOzone.clean %>%
  mutate(GaringerOzone.clean, year = year(Date)) %>%
  mutate(GaringerOzone.clean, month = month(Date)) %>%
  separate(Date,c("Null","Month","Null2")) %>%
  rename(Year = Null) %>%
  mutate(Date= my(paste0(Month,"-", Year))) %>%
  select(Date, Ozone.clean, DAILY_AQI_VALUE)
```

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.

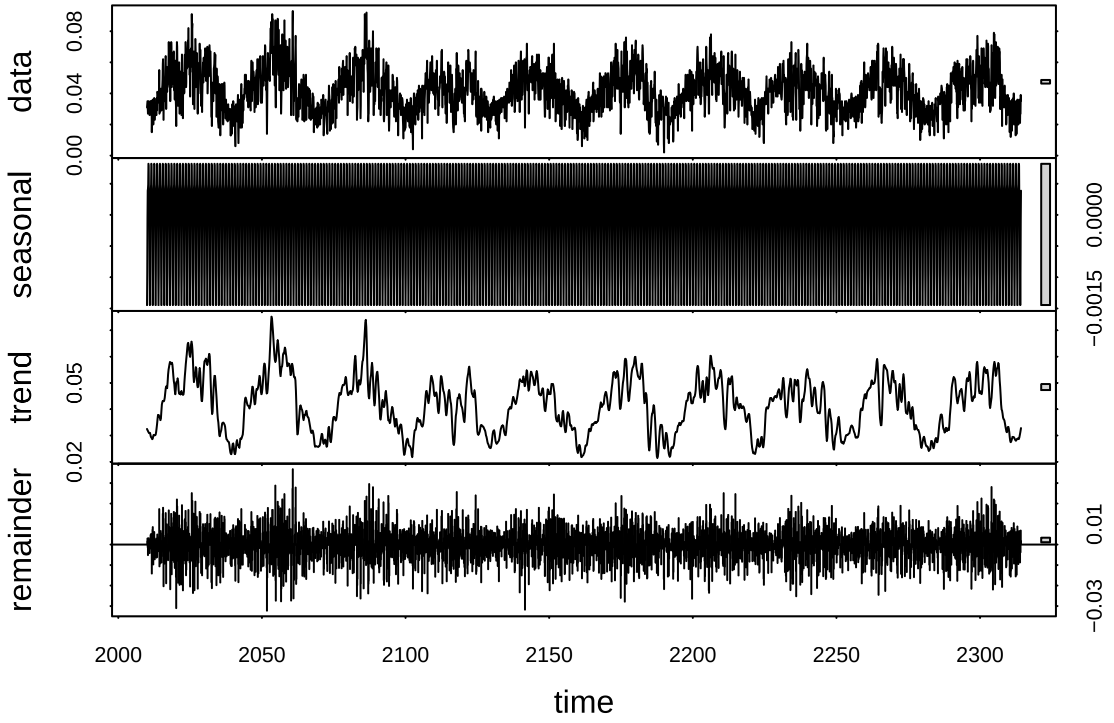
```
#10
#2 time series for daily and monthly average ozone values
GaringerOzone.daily.ts <- ts(GaringerOzone.clean$Ozone.clean,start=c(2010,1),frequency=365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Ozone.clean,start=c(2010,1),frequency=12)
```

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

```
#11
#decomposing the daily and monthly time series and plotting the components
Daily_decomp <- stl(GaringerOzone.daily.ts,s.window="periodic")
plot(Daily_decomp)
```



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

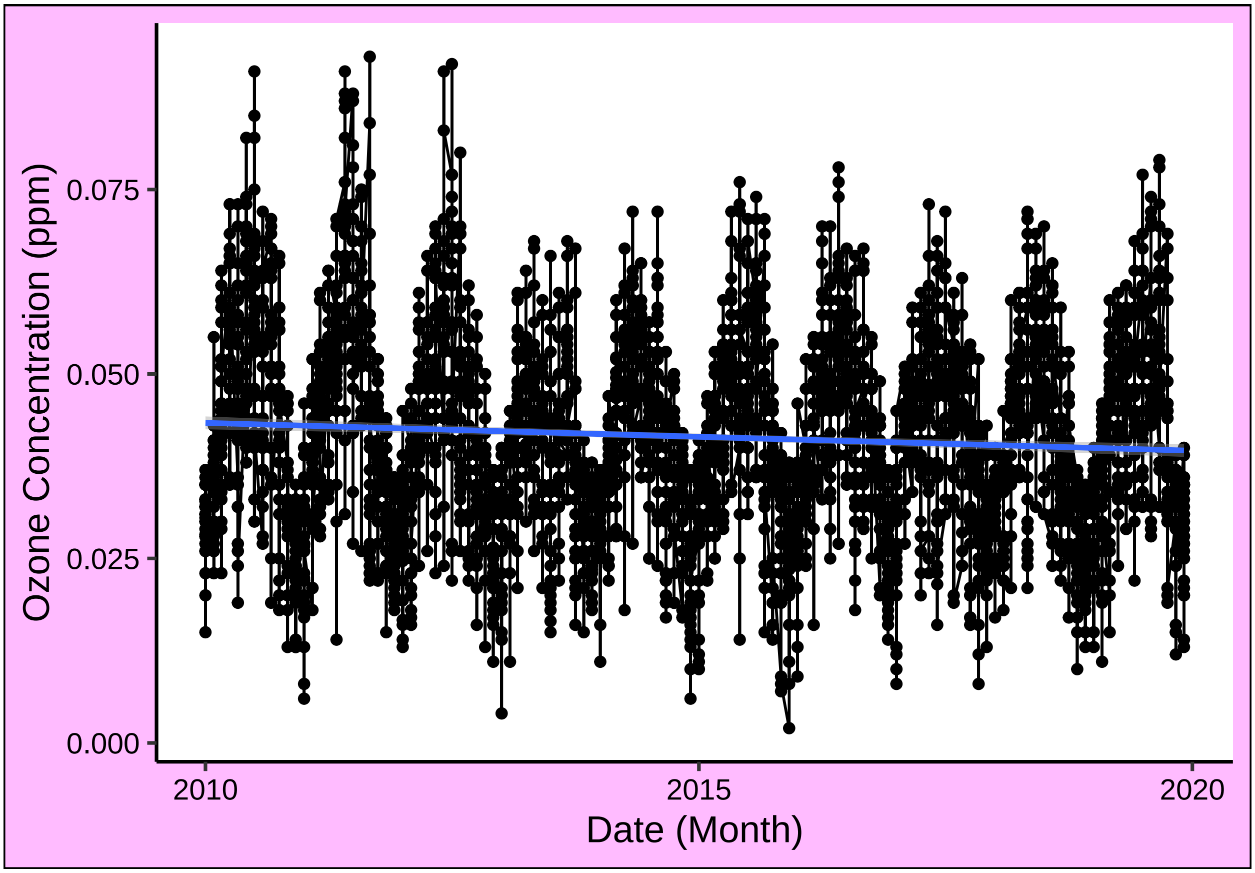
```
#12
#seasonal Mann-Kendall trend analysis
Monthly.Ozone.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
```

Answer: The seasonal Mann-Kendall is most appropriate because this data has seasonality differences in the data (the up and down pattern 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
#plot of mean monthly ozone concentrations over time
Monthly.ozone.plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = Ozone.clean)) +
  geom_point() +
  geom_line() +
  xlab("Date (Month)") +
  ylab("Ozone Concentration (ppm)") +
  geom_smooth( method = lm )
print(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: The average Ozone concentrations are decreasing overall from 2010 to 2020. The S value is -22362 which support this negative correlation.

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 the seasonal component from the monthly time series
Garinger_Components <- as.data.frame(GaringerOzone.monthly.ts)

Garinger.Ozone_Components <- mutate(Garinger_Components,
        Observed = GaringerOzone.monthly$Ozone.clean,
        Date = GaringerOzone.monthly$Date)

#non-seasonal time series of component
Garinger.components.ts <- ts(Garinger.Ozone_Components$Observed,start=c(2010,1),frequency=12)

#16
```

```r
#non-seasonal Mann-Kendall trend analysis
Nonseasonal.Ozone <- Kendall::MannKendall(Garinger.components.ts)
```

> Answer: The average Ozone concentrations are decreasing overall from 2010 to 2020. The S value is -264863 which is more negative than the seasonal S value of -22362 so it shows a stronger negative correlation.