Assignment 7: Time Series Analysis

Nancy Bao

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 Tuesday, March 16 at 11:59 pm.

Set up

1. Set up your session:

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

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
#Check working directory
getwd()
```

```
## [1] "/Users/Nancy/Desktop/Semester 4/ENV 872L/Environmental_Data_Analytics_2021"
```

```
#Load packages
library(tidyverse)
library(ggplot2)
library(plyr)
library(lubridate)
library(zoo)
library(trend)
#Set ggplot theme
theme07<-theme_bw(base_size=12)+
          theme(plot.title=element_text(size=12,
                                        face="bold",
                                        color="black",
                                        hjust=0.5),
```

```
                  axis.text = element_text(color = "black"),
                  axis.title = element_text(color= "black",face= "bold"),
                  legend.position = "top")
theme_set(theme07)

#2 Import Ozone_TimeSeries data using list.files
Ozone_Garinger_files = list.files(path = "./Data/Raw/Ozone_TimeSeries/",
                                  pattern="*.csv", full.names=TRUE)
Ozone_Garinger_files
```

```
##  [1] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2010_raw.csv"
##  [2] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2011_raw.csv"
##  [3] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2012_raw.csv"
##  [4] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2013_raw.csv"
##  [5] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2014_raw.csv"
##  [6] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2015_raw.csv"
##  [7] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2016_raw.csv"
##  [8] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2017_raw.csv"
##  [9] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2018_raw.csv"
## [10] "./Data/Raw/Ozone_TimeSeries//EPAair_O3_GaringerNC2019_raw.csv"
```

```
#created dataframe from ldply() function in plyr package
GaringerOzone <- Ozone_Garinger_files %>%
                  ldply(read.csv)
```

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

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

```
# 4 Wrangled GaringerOzone to 3 columns and renamed new dataframe:GaringerOzone_filtered
GaringerOzone_filtered<-GaringerOzone %>%
                        select(Date, Daily.Max.8.hour.Ozone.Concentration,
                               DAILY_AQI_VALUE)
# 5 Created dataframe named Days using seq() and lubridate function ymd()
Days<-as.data.frame(seq(ymd("2010-01-01"), ymd("2019-12-31"),by="days"))
#rename column name to Date
Days<-setNames(Days,c("Date"))
# 6 used left_join to combine Days df with the GaringerOzone_filtered df
GaringerOzone<-left_join(Days,GaringerOzone_filtered)
```

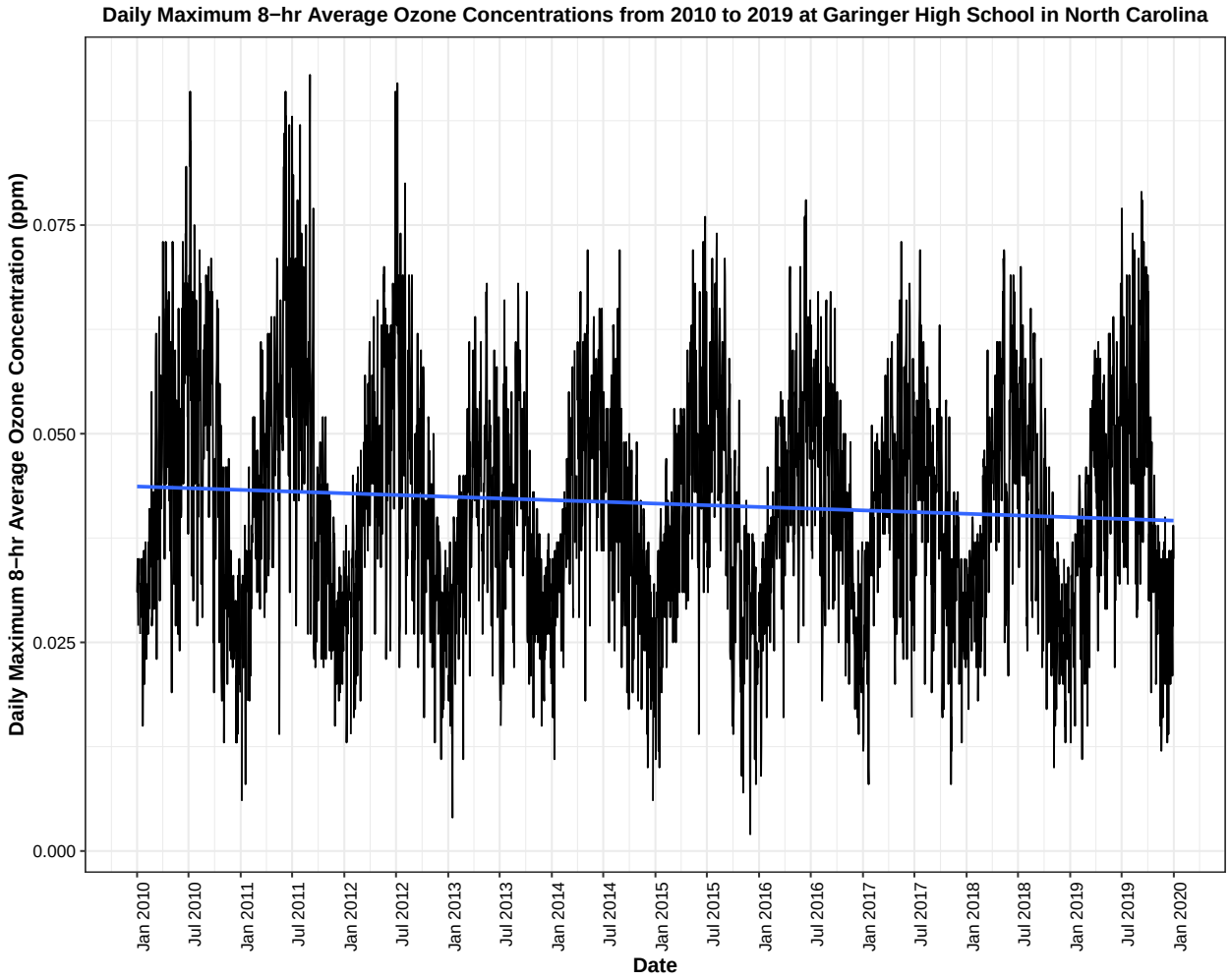
```
## Joining, by = "Date"
```

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 #created a line plot and rescaled x axis based on 3-letter month and Year
GaringerOzone_lineplot<-ggplot(GaringerOzone,aes(x = Date,
                                   y = Daily.Max.8.hour.Ozone.Concentration))+
                         geom_line(color = "black") +
                         geom_smooth(method=lm,se=FALSE)+
                         labs(y="Daily Maximum 8-hr Average Ozone Concentration (ppm)",
                              title="Daily Maximum 8-hr Average Ozone Concentrations from 2010 to 2019 a
  scale_x_date(date_breaks = "6 months",date_labels = "%b %Y")+
  theme(axis.text.x=element_text(angle=90, hjust=1))
print(GaringerOzone_lineplot)
```

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



Answer: The plot suggests that there is a slightly downward trend in daily maximum 8-hr average ozone concentration from 2010 to 2019, suggesting the ozone concentration decreases from 2010 to 2019.

Time Series Analysis

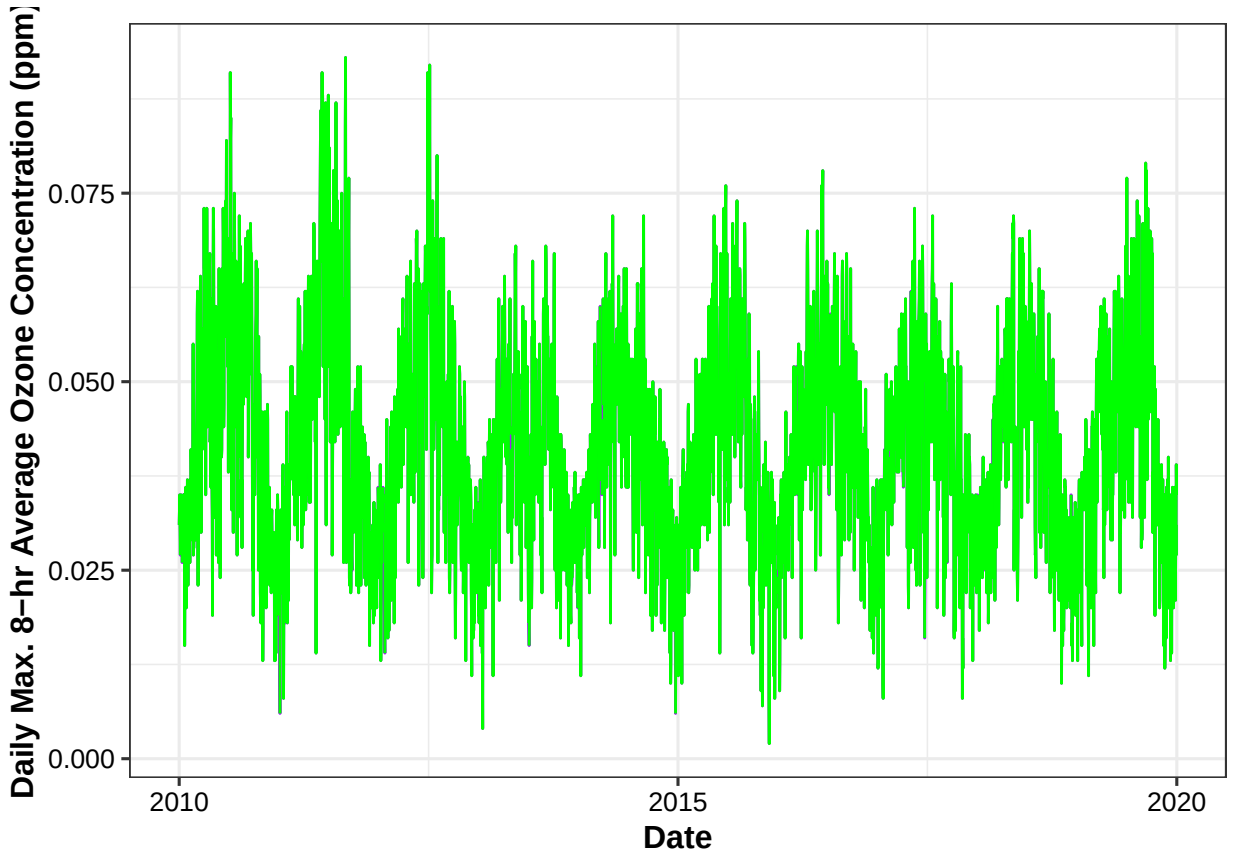
Study question: Have ozone concentrations changed over the 2010s at this station?

8. Use a linear interpolation to fill in missing daily data for ozone concentration. Why didn't we use a piece-wise constant or spline interpolation?

```
#8 Linear interpolation
GaringerOzone_cleaned <-
  GaringerOzone %>%
  mutate( Daily.Max.8hr.Ozone.Conc.Clean=
          zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration) )
#Daily.Max.8hr.Ozone.Conc.Clean variable is linear interpolation
summary(GaringerOzone_cleaned$Daily.Max.8hr.Ozone.Conc.Clean)
```

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

```
#Graphing the linear interpolation
Garinger_linear_interpolation<-ggplot(GaringerOzone_cleaned) +
                                      geom_line(aes(x = Date,
                                      y = Daily.Max.8hr.Ozone.Conc.Clean),
                                      color = "purple", alpha=1) +
                                      geom_line(aes(x = Date,
                                      y = Daily.Max.8.hour.Ozone.Concentration),
                                      color = "green",alpha=1) +
                    ylab("Daily Max. 8-hr Average Ozone Concentration (ppm)")
print(Garinger_linear_interpolation)
```



Answer: We didn't use a spline interpolation because the data does not follow a quadratic trend. Since the spline interpolation uses a quadratic function to fill in the missing values, that could potentially overestimate the missing daily ozone concentrations, since quadratic functions are based on a polynomial order of 2. We didn't use a piecewise constant interpolation because our data because we are trying to see how ozone concentrations change over a continuous time period from 2010 to 2019 and piecewise constant could use values from an earlier day or later day that may underestimate or overestimate the interpolation. You do not know if that nearest neighbor was potentially an outlier measurement (e.g. super cloudy or rainy day could alter daily measurements) that could dramatically change the trend of the data.

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_cleaned %>%
                        mutate(Month = month(Date),
                               Year=year(Date)) %>%
                        mutate(Date_combined = my(paste0(Month,"-",Year)))%>%
                        group_by(Date_combined)%>%
                        dplyr::summarise(Mean.monthly.Ozone = mean(Daily.Max.8hr.Ozone.Conc.Clean))
#I used paste0 to combined Month and Year into a new column called Date_combined
#I used group_by to summarize mean montly ozone concentrations
#Separate pipe to create new Date column using lubridate package function: make_date()
GaringerOzone.monthly<-GaringerOzone.monthly %>%
                        mutate(Date=as.Date(Date_combined))
```

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
#checking first and last dates of the observations
first(GaringerOzone_cleaned$Date) #first date is in January 2010
```

```
## [1] "2010-01-01"
```

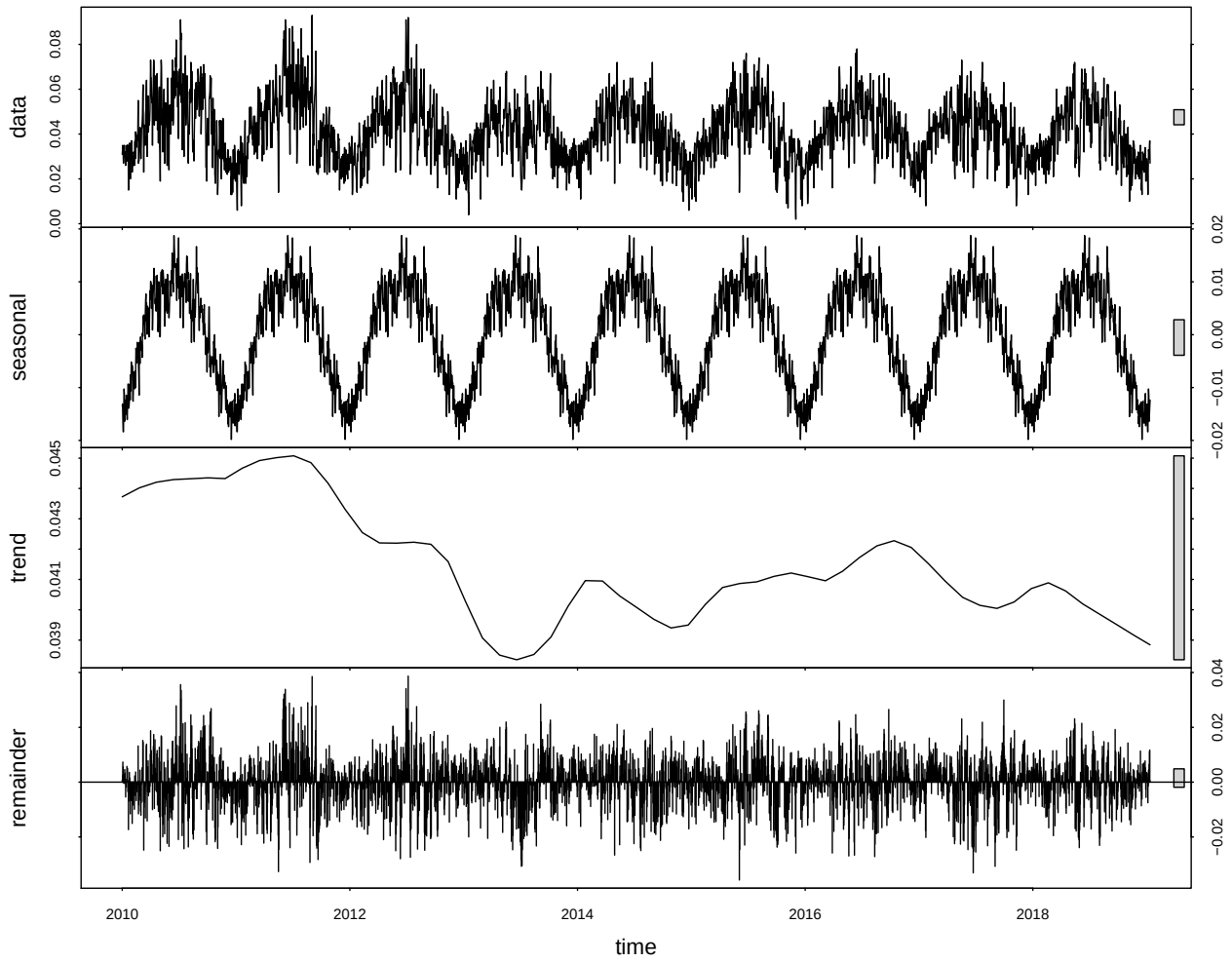
```
last(GaringerOzone_cleaned$Date) #last date is in December 2019
```

```
## [1] "2019-12-31"
```

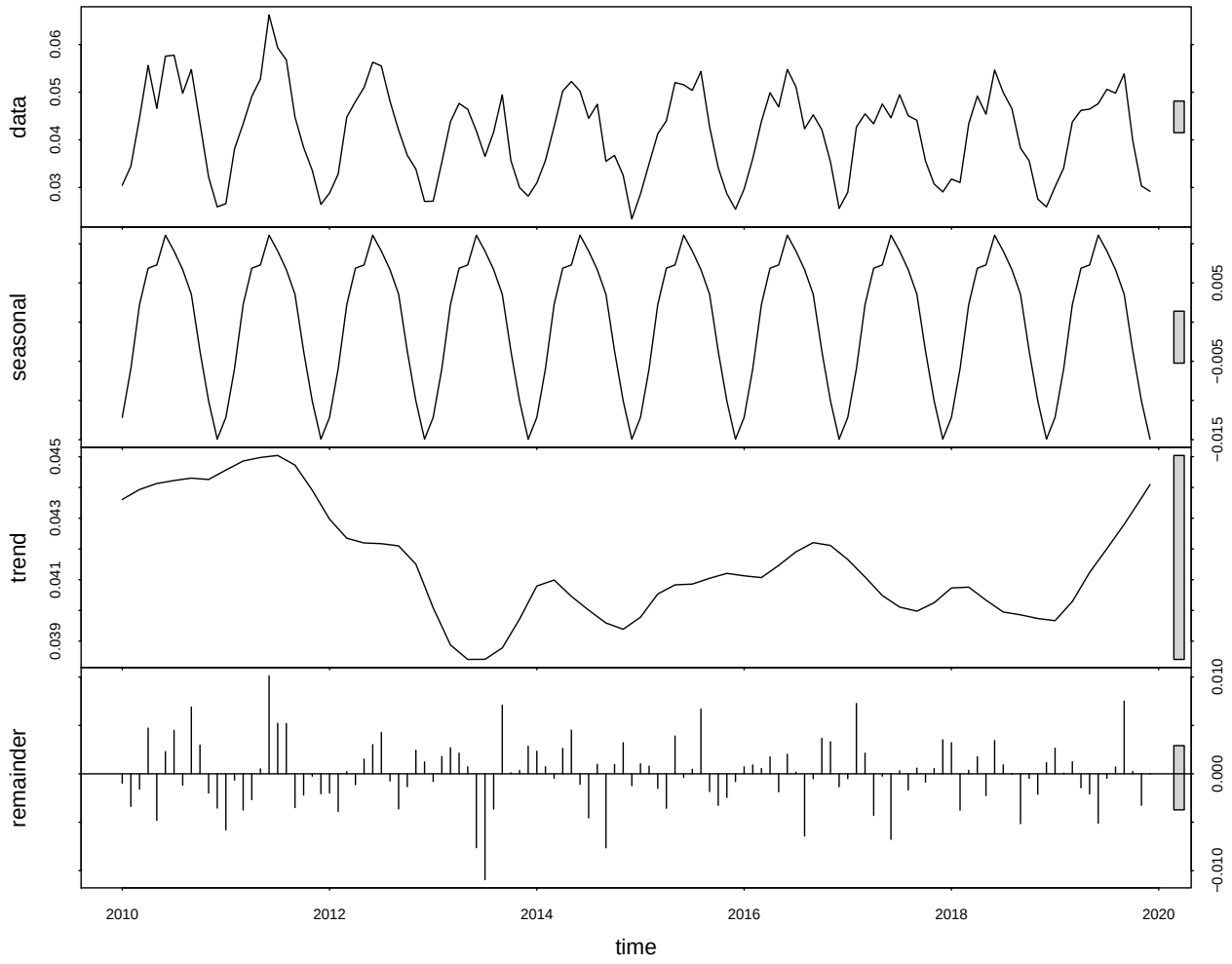
```
#First time series object: daily observations
GaringerOzone.daily.ts<-ts(GaringerOzone_cleaned$Daily.Max.8hr.Ozone.Conc.Clean,
                           start=c(2010,1),
                           end=c(2019,12),
                           frequency=365)
#Second time series object: monthly average ozone values
GaringerOzone.monthly.ts<- ts(GaringerOzone.monthly$Mean.monthly.Ozone,
                              start=c(2010,1),
                              end=c(2019,12),
                              frequency=12)
```

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

```r
#11
#Decompose the time series objects
#I used period for s.window, because I see a seasonal component from the lineplot in #7
#Daily
GaringerOzone.daily.decomposed<-stl(GaringerOzone.daily.ts, s.window="periodic")
plot(GaringerOzone.daily.decomposed)
```




```r
#Monthly
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 Monotonic trend analysis for monthly Ozone series
trend.monthly.Ozone<- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
trend.monthly.Ozone
```

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

```
summary(trend.monthly.Ozone)
```

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

```
#I also ran the smk test to look at differences in each season
trend.monthly.Ozone.smk <- trend::smk.test(GaringerOzone.monthly.ts)
summary(trend.monthly.Ozone.smk)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
```

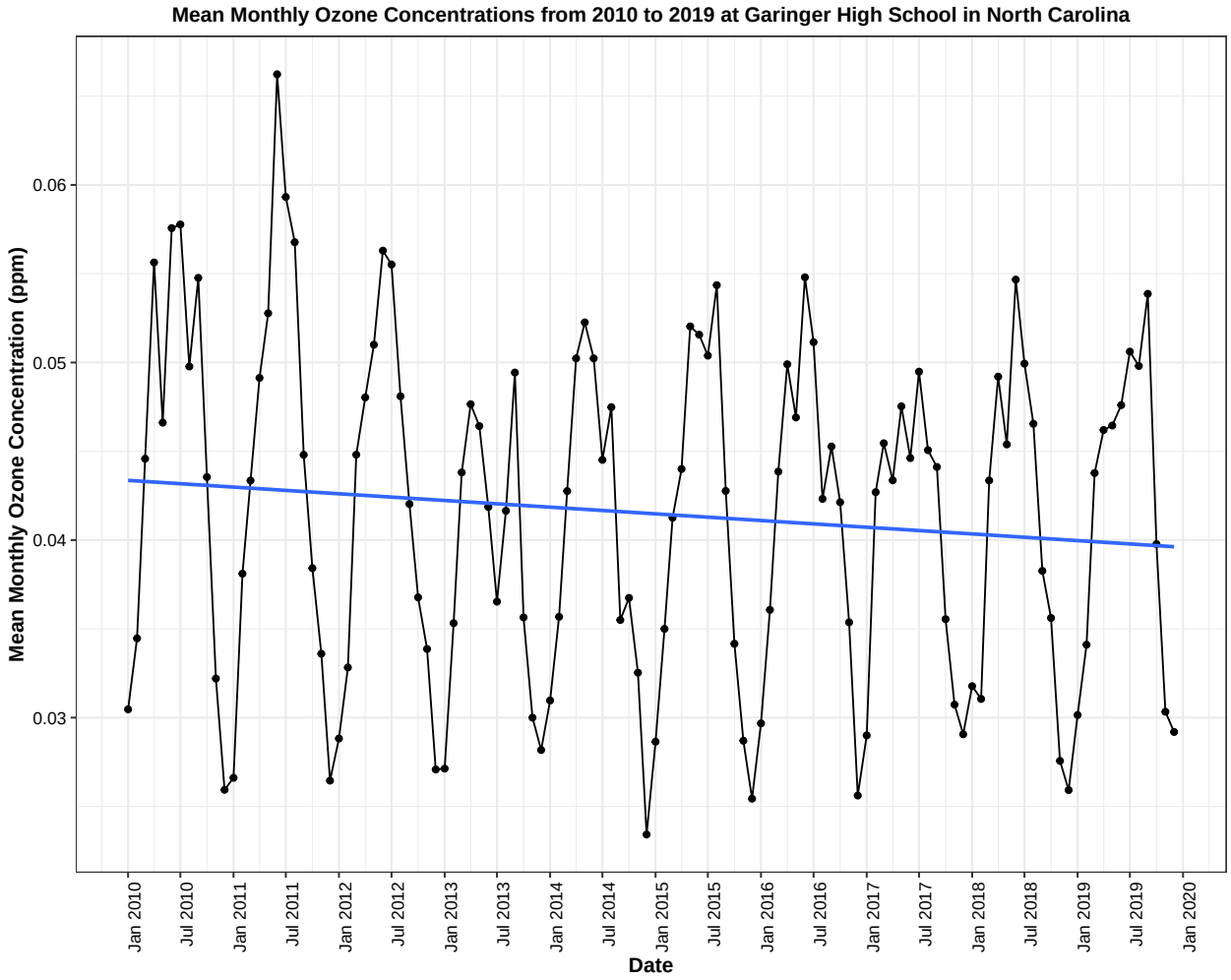
```
## Statistics for individual seasons
##
## H0
##                       S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 2:   S = 0   -1  125 -0.022  0.000  1.00000
## Season 3:   S = 0   -4  124 -0.090 -0.269  0.78762
## Season 4:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 5:   S = 0  -15  125 -0.333 -1.252  0.21050
## Season 6:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 7:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 8:   S = 0   -7  125 -0.156 -0.537  0.59151
## Season 9:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 10:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 11:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 12:   S = 0  11  125  0.244  0.894  0.37109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The seasonal Mann-Kendall is the most appropriate because I see with the daily and monthly ozone data that there is seasonality in the data. When I plotted the decomposed daily and monthly time series objects I see that in the seasonal vs. time graph that there is a strong seasonal component. So, when running the Mann-Kendall we need to take into consideration seasonality.

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.mean.monthly.plot<-ggplot(GaringerOzone.monthly,
                                        aes(x=Date,y=Mean.monthly.Ozone))+
                                  geom_point(alpha=1,size=1.5)+
                                  geom_line()+
                                  geom_smooth(method=lm, se=FALSE)+
                                  labs(y="Mean Monthly Ozone Concentration (ppm)", x="Date",
                                  title=
          "Mean Monthly Ozone Concentrations from 2010 to 2019 at Garinger High School in North Carolina
  scale_x_date(date_breaks = "6 months",date_labels = "%b %Y")+
  theme(axis.text.x=element_text(angle=90, hjust=1))
plot(GaringerOzone.mean.monthly.plot)
```

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




```
#I used theme(axis.text.x=element_text(angle=90, hjust=1)) to rotate the x axis labels
#scale_x_date(date_breaks = "6 months",date_labels = "%b %Y") used to rename time intervals
#I added the geom_smooth to better visualize the monotonic downward trend.
```

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 ozone concentrations have a seasonal component at the Garinger High School Station in North Carolina. As seen in the figure above, the monthly mean ozone concentrations (ppm) peak during the summmer months and dip in the winter months. From the Seasonal Mann-Kendall test, with a p-value =0.046 $<$ 0.05 alpha level, we reject the null hypothesis that the overall mean monthly ozone concentrations are stationary at Garinger High School Station in NC from 2010 to 2019 (tau = -0.143, p-value =0.046724). We can conclude that there is a monotonic trend where mean monthly ozone concentrations are slightly decreasing from 2010 to 2019. From the smk test, we see that between each season the decreases is not significant (p-value$>$0.05); however, the overall downward trend shows that ozone concentrations have changed across the 2010s at Garinger Station (Score=-77, tau = -0.143, p-value =0.046724).This downward trend is also shown in the blue linear trendline on the graph above.

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 Subtract seasonal component form GaringerOzone.monthly.ts
#I turned the monthly decomposed object into a dataframe
GaringerOzone.monthly.components<-as.data.frame(GaringerOzone.monthly.decomposed$time.series[,1:3])
# I used [,1:3] to get the seasonal, trend, and remainder columns
#Now I need to subtract the seasonal column (from the GaringerOzone.monthly.components)
#in order to look at the monthly.ts without seasonality
GaringerOzone.month.no.season.ts<-(GaringerOzone.monthly.ts-GaringerOzone.monthly.components$seasonal)

#16 Mann Kendall test on non-seasonal Ozone monthly series
non.seasonal.Ozone.monthly.trend<-Kendall::MannKendall(GaringerOzone.month.no.season.ts)
non.seasonal.Ozone.monthly.trend
```

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

```
summary(non.seasonal.Ozone.monthly.trend)
```

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

Answer: Similar to the Seasonal Mann-Kendall test, the non-seasonal Mann Kendall also found that the mean monthly ozone concentrations at Garinger Station in North Carolina changes from 2010 to 2019. We reject the null hypothesis that the mean monthly ozone concentrations are stationary and conclude that there is a downward trend in mean monthly ozone concentrations from 2010 to 2019 (tau=-0.165, p-value=0.0075< 0.05-alpha level). The Mann Kendall p-value is smaller than the seasonal mann-kendall p-value. When seasonal component is taken out the Score is -1179 which is more negative than the Score from the Seasonal Mann Kendall test which had a score of -77.The Mann Kendall non-seasonal Score has a higher variance (V(Score)=194365) than the seasonal Mann Kendall score (Var(Score)=1499).