Assignment 7: Time Series Analysis

Anne Harshbarger

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
getwd()
```

```
## [1] "C:/Users/Anne/Documents/ENV872/Environmental_Data_Analytics_2021/Assignments"
```

```
setwd("C:/Users/Anne/Documents/ENV872/Environmental_Data_Analytics_2021/Assignments")
#install.packages("tidyverse", "lubridate", "zoo", "trend", "ggplot2", "Kendall")
library(tidyverse)
library(lubridate)
library(zoo)
library(trend)
```

```
## Warning: package 'trend' was built under R version 4.0.4
```

```
library(ggplot2)
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.0.4
```

```
mytheme <- theme_bw() + theme( #create ggplot theme
  text = element_text(color = "black", size = 12),
  axis.text = element_text(color = "dark gray", size = 12),
  legend.position = "bottom",
  axis.line = element_line(color = "dark gray", size = 1.25))
theme_set(mytheme) #set as default theme

#2
Garinger2010 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
Garinger2011 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
Garinger2012 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
Garinger2013 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
Garinger2014 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
Garinger2015 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
Garinger2016 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
Garinger2017 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
Garinger2018 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
Garinger2019 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")

GaringerOzone <- rbind(Garinger2010, Garinger2011, Garinger2012,
                       Garinger2013, Garinger2014, Garinger2015,
                       Garinger2016, Garinger2017, Garinger2018,
                       Garinger2019) #combine yearly datasets
```

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 convert date column to date object
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

# 4 select desired columns
GaringerOzone <-
  GaringerOzone %>%
```

```
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5 create days dataframe
Days <- as.data.frame(seq(from = as.Date("2010-01-01"),
                          to = as.Date("2019-12-31"), by = 1))
colnames(Days) <- "Date" #rename column to Date

# 6 join with Days first to create a row for each day even if no data
GaringerOzone <- left_join(Days, GaringerOzone)
```

```
## 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
ozone_line <- ggplot(data = GaringerOzone,
                     aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() + geom_smooth(method = lm, color = "sky blue") +
  scale_x_date(date_breaks="1 year") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Date", y = "Ozone Concentration (ppm)",
       title = "Daily Maximum Ozone Concentrations in Garinger, NC")
print(ozone_line)
```

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

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

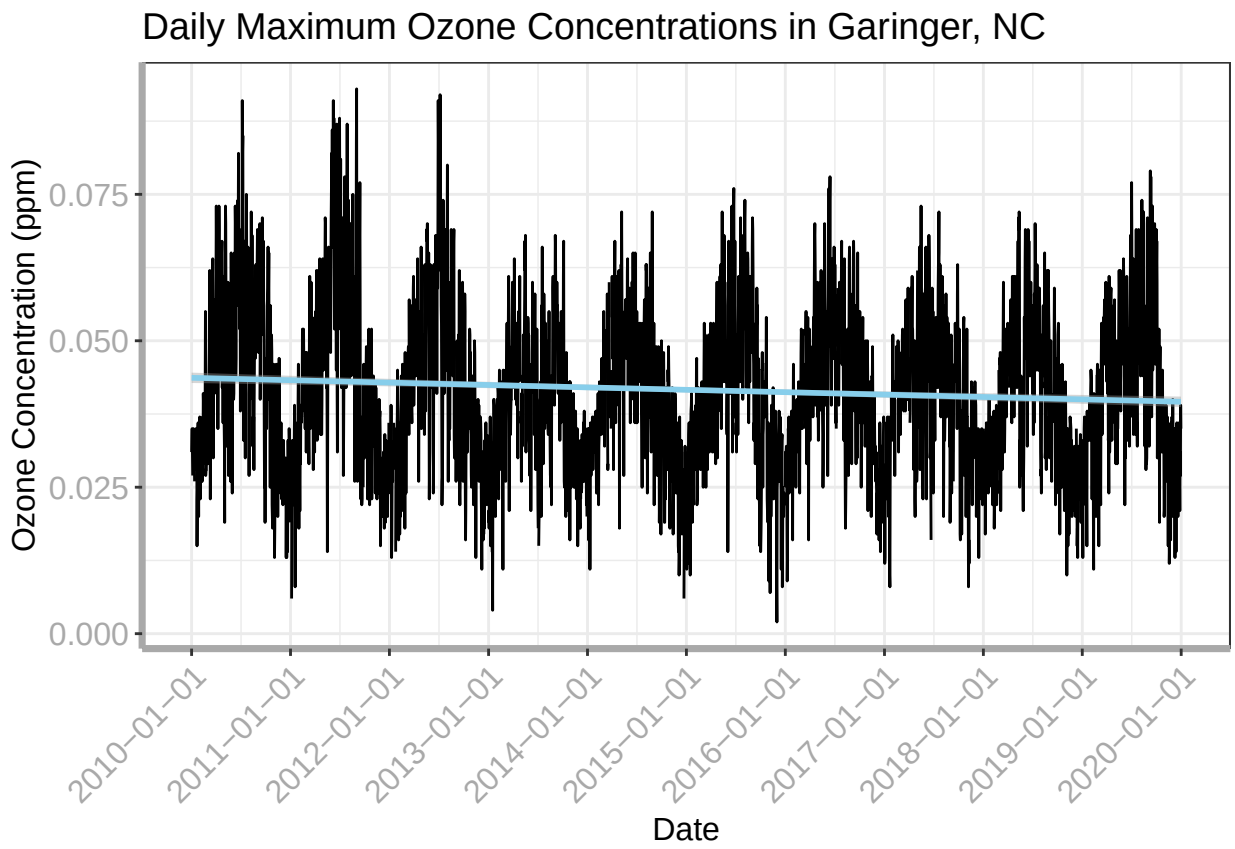

Figure 1: Daily maximum ozone concentrations (ppm) in Garinger, NC between January 1, 2010 and December 31, 2019.

> Answer: The plot suggests a strong seasonal pattern in ozone concentration, with values reaching a local maximum in the summer and a local minimum in the winter, and a slight decreasing trend in ozone concentration over time.

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 <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration)) %>% #O3 linear interpolation
  mutate(DAILY_AQI_VALUE =
           zoo::na.approx(DAILY_AQI_VALUE)) #AQI linear interpolation
```

> Answer: A piecewise constant approach would use the value of the nearest neighbor to fill in the missing data points. This would result in several days in a row having the same value of ozone

concentration. Because ozone appears to be continuously varying, this would likely introduce more errors into the data than a linear interpolation. A spline interpolation, which follows a quadratic function, would also likely not be a good fit for the data, since the seasonal increases and decreases appear to be roughly linear.

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(Year = year(Date)) %>% #add year column
  mutate(Month = month(Date)) %>% #add month column
  group_by(Year, Month) %>%
  summarise(mean.o3.concentration = mean(Daily.Max.8.hour.Ozone.Concentration),
            mean.AQI = mean(DAILY_AQI_VALUE)) #calculate mean for each month
```

```
## `summarise()` regrouping output by 'Year' (override with `.groups` argument)
```

```
GaringerOzone.monthly$Date <- as.Date(paste0(GaringerOzone.monthly$Year, "-",
                                             GaringerOzone.monthly$Month, "-01"),
                                      format = "%Y-%m-%d")
```

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) #daily ts
#did not specify end date as daily ts should have 3652 entries (same as Days)
#specifying end = (2020,1) results in 3651

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean.o3.concentration,
                               start = c(2010,1), end = c(2019,12),
                               frequency = 12) #monthly ts
#monthly ts has 120 entries (12 months over 10 years, ends 12/2019)
```

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

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

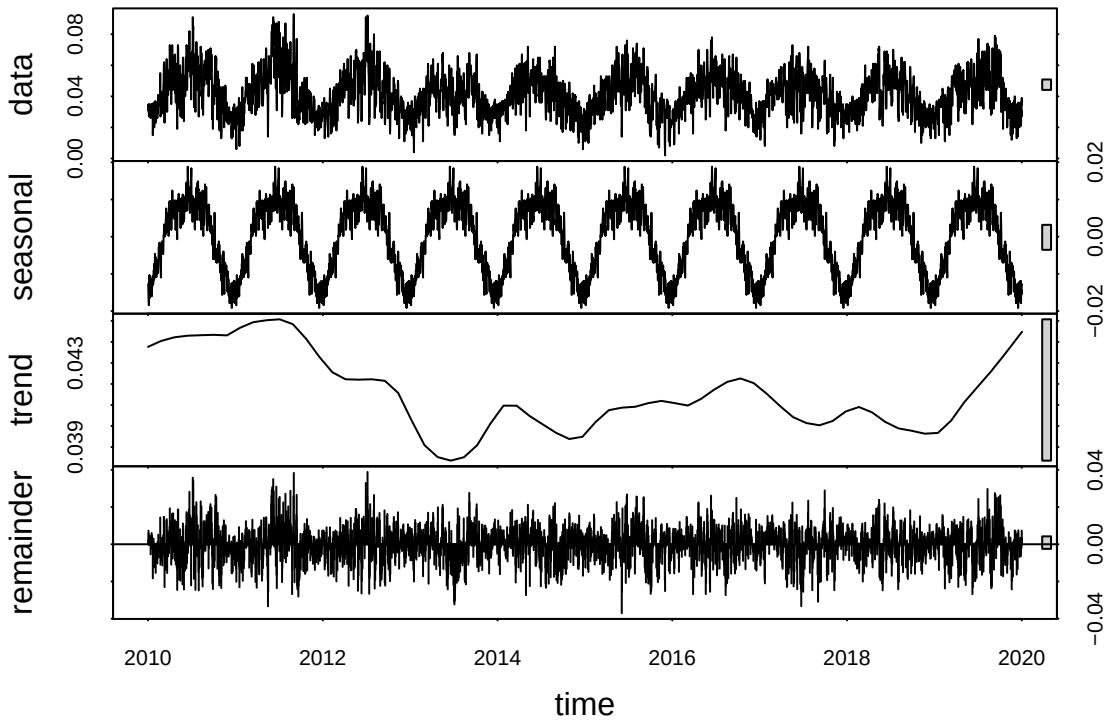

Figure 2: Components of the daily ozone time series for Garinger, NC.

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

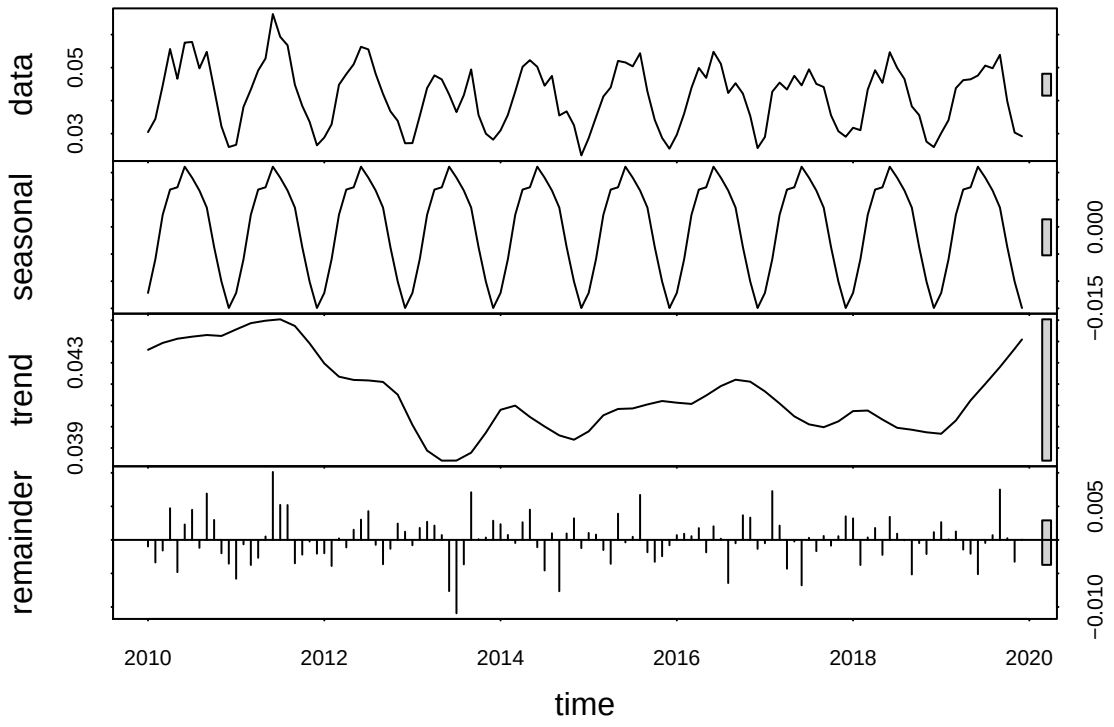

Figure 3: Components of the monthly ozone time series for Garinger, NC.

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
SeasonalMannKendall(GaringerOzone.monthly.ts)
```

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

Answer: From the plot of the components, there appears to be a strong seasonal pattern in the ozone data, which makes the seasonal Mann-Kendall a better fit than the linear regression, Mann-Kendall, or Spearman Rho (all non-seasonal). Since the other tests are not equipped to handle seasonal data, performing them on seasonal data might give us erroneous results.

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
monthly_ozone <- ggplot(GaringerOzone.monthly,
                        aes(x = Date, y = mean.o3.concentration)) +
  geom_point() +
  geom_line() +
  geom_smooth(method = lm) +
```

```
  scale_x_date(date_breaks="1 year") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Date", y = "Mean Ozone Concentration (ppm)",
       title = "Monthly Mean Ozone Concentrations in Garinger, NC")
print(monthly_ozone)
```

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

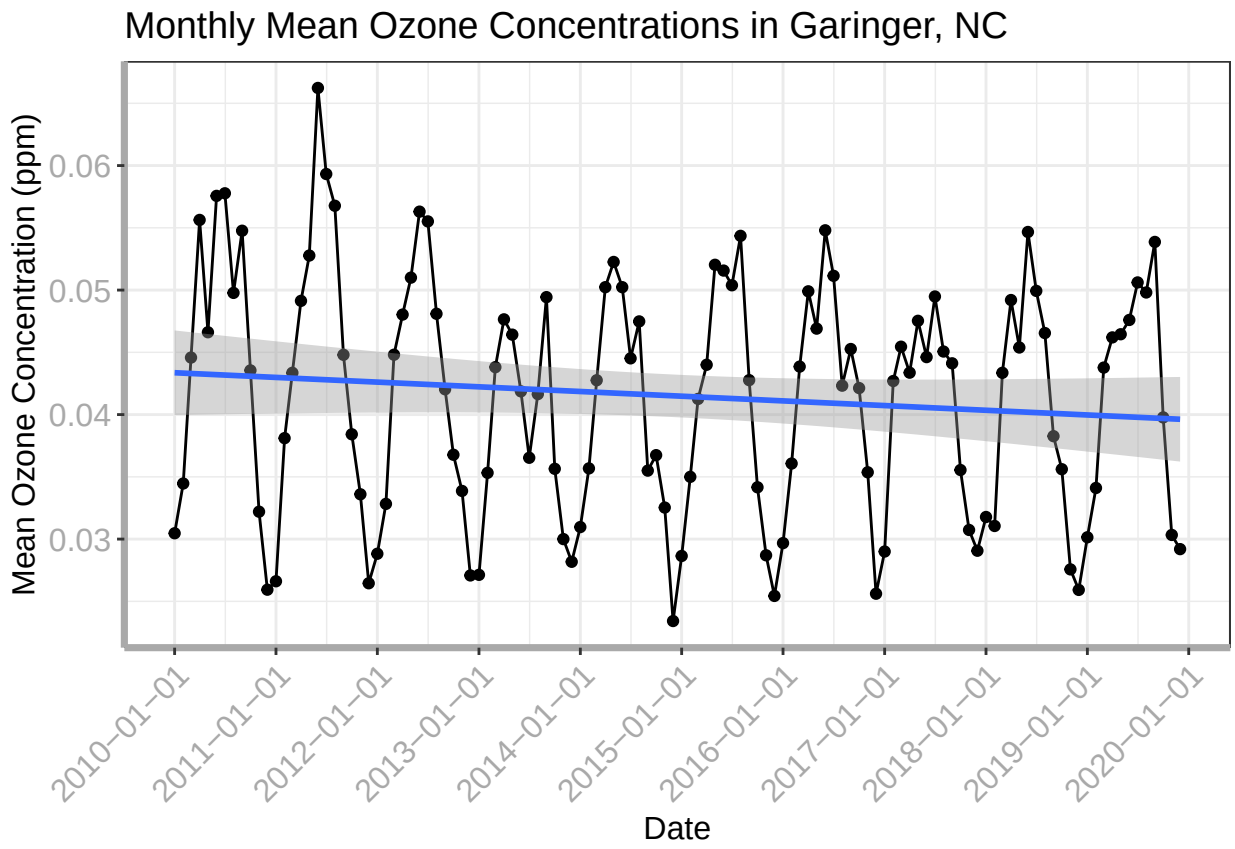

Figure 4: Monthly mean ozone concentrations (ppm) for Garinger, NC between January 2010 and December 2019.

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: Each year, the ozone concentration measured in Garinger, NC fluctuates seasonally, reaching a local minimum in the winter and a local maximum in the summer. Over time, the mean ozone concentrations measured at this location are generally decreasing. Fitting a linear trend line to the data reveals a slight decrease in ozone over time, with higher uncertainty than the line of best fit for the daily ozone (Figure 1) due to a smaller dataset (one value for each month instead of ~30 values for each month). A seasonal Mann-Kendall test confirms that there is a slight decreasing trend in mean monthly ozone over time between the beginning of 2010 and the end of 2019 (tau = -0.143, p = 0.0467).

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

GaringerOzone.monthly_Components <- mutate(GaringerOzone.monthly_Components,
        Ozone = GaringerOzone.monthly$mean.o3.concentration,
        Date = GaringerOzone.monthly$Date,
        NoSeasonalOzone = (GaringerOzone.monthly$mean.o3.concentration-
                        GaringerOzone.monthly_Components$seasonal))

GaringerOzone.monthly.no.season.ts <- ts(GaringerOzone.monthly_Components$NoSeasonalOzone,
                                         start = c(2010,1),frequency = 12)

#16

MannKendall(GaringerOzone.monthly.no.season.ts)
```

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

Answer: For the complete time series, we found a significant result for the seasonal Mann-Kendall test (tau = -0.143, p = 0.0467), leading us to reject the null hypothesis that the data were stationary in favor of the alternative hypothesis that there was a trend in the data. Similarly, for the time series with the seasonal component removed, we found a significant result for the non-seasonal Mann-Kendall test (tau = -0.165, p = 0.00754). Again, we reject the hypothesis that the time series is stationary in favor of the alternative hypothesis that the data follow a trend. Because we accounted for seasonal changes in the first test and removed the seasonal component of this time series in the second, we know the trend must be due to some other factor contributing to variation in the data. In fact, the seasonal pattern in mean ozone concentration may even be slightly masking the trend, since the value of tau was more negative for the non-seasonal Mann-Kendall test than the seasonal Mann-Kendall test. Overal, both tests support evidence of a decrease in mean monthly ozone concentrations in Garinger, NC between 2010 and 2019.

Duke Community Standard affirmation: I have adhered to the Duke Community Standard in completing this assignment. -Anne Harshbarger