Assignment 8: Time Series Analysis

Yixin Fang

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] "D:/DKU/2023_Spring/ENV872/EDA-Spring2023"
```

```
library(tidyverse)
library(lubridate)
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.2.3
```

```
library(trend)
```

```
## Warning: package 'trend' was built under R version 4.2.3
```

```
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.2.3
```

```
Sys.setenv(LANGUAGE = "en_US.UTF-8")

mytheme <- theme_classic(base_size = 12) +
  theme(plot.background = element_rect(color = "grey"),
        plot.title =  element_text(color = "black"),
        axis.text = element_text(color = "black"),
        legend.position = "right")
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
O32010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
O32011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
O32012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
O32013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
O32014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
O32015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
O32016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
O32017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
O32018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
O32019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")

GaringerOzone <- rbind(O32010, O32011, O32012, O32013, O32014, O32015, O32016,
                       O32017, O32018, O32019)
```

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.

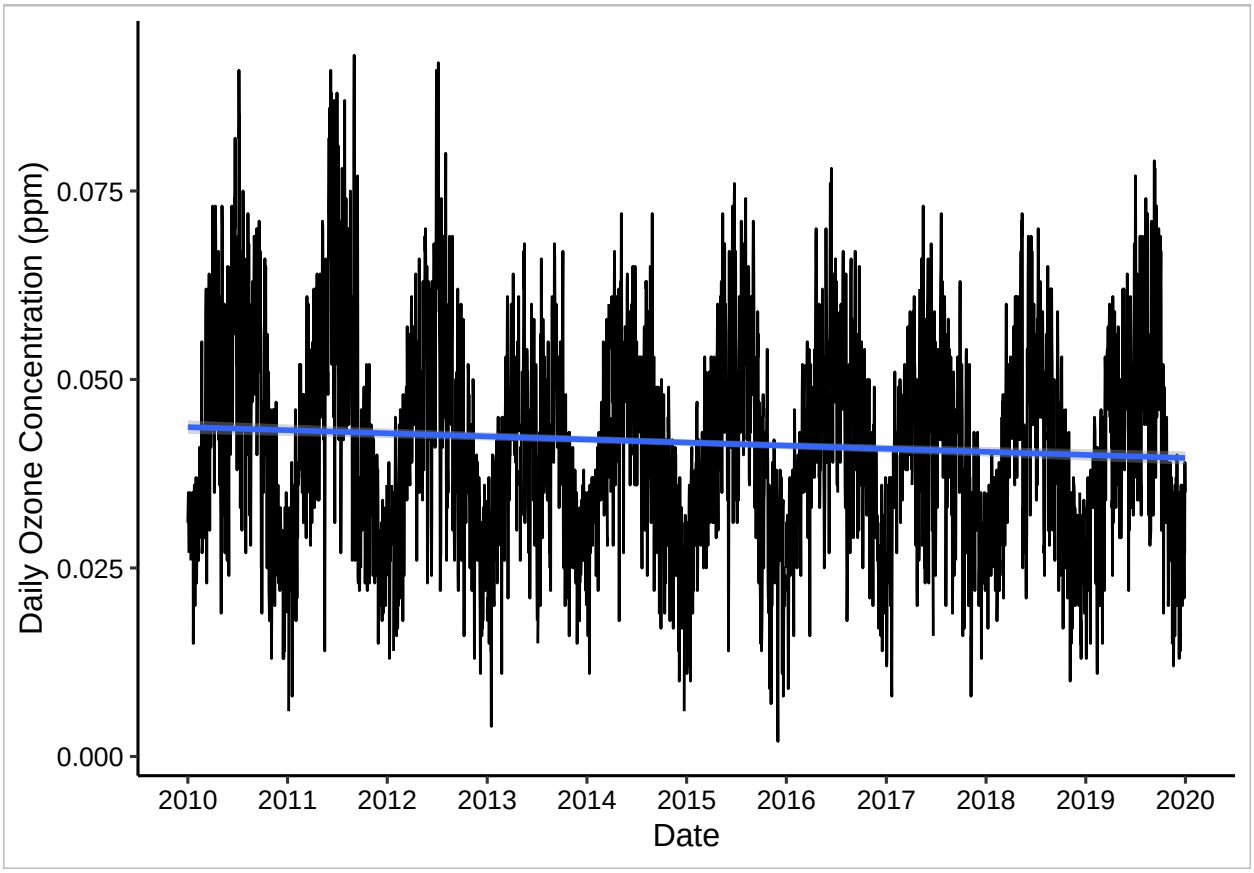
```
#3
GaringerOzone$Date <- mdy(GaringerOzone$Date)

#4
GaringerOzone <- select(GaringerOzone, Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

#5
Days <- as.data.frame(seq(ymd("2010-01-01"), ymd("2019-12-31"), by = "day"))
names(Days) <- "Date"

#6
GaringerOzone <- left_join(Days, GaringerOzone, 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
GaringerOzone %>%
  ggplot(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year") +
  geom_smooth(method = "lm") +
  labs(y = "Daily Ozone Concentration (ppm)")
```

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



Answer: The line shows that there is a decreasing trend in the 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))
```

Answer: We use linear interpolation because the data shows a linear trend over time. If the data is constant with no graduate changes, then we would choose a piecewise constant interpolation. The spline interpolation is used when the data has a non-linear pattern.

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)

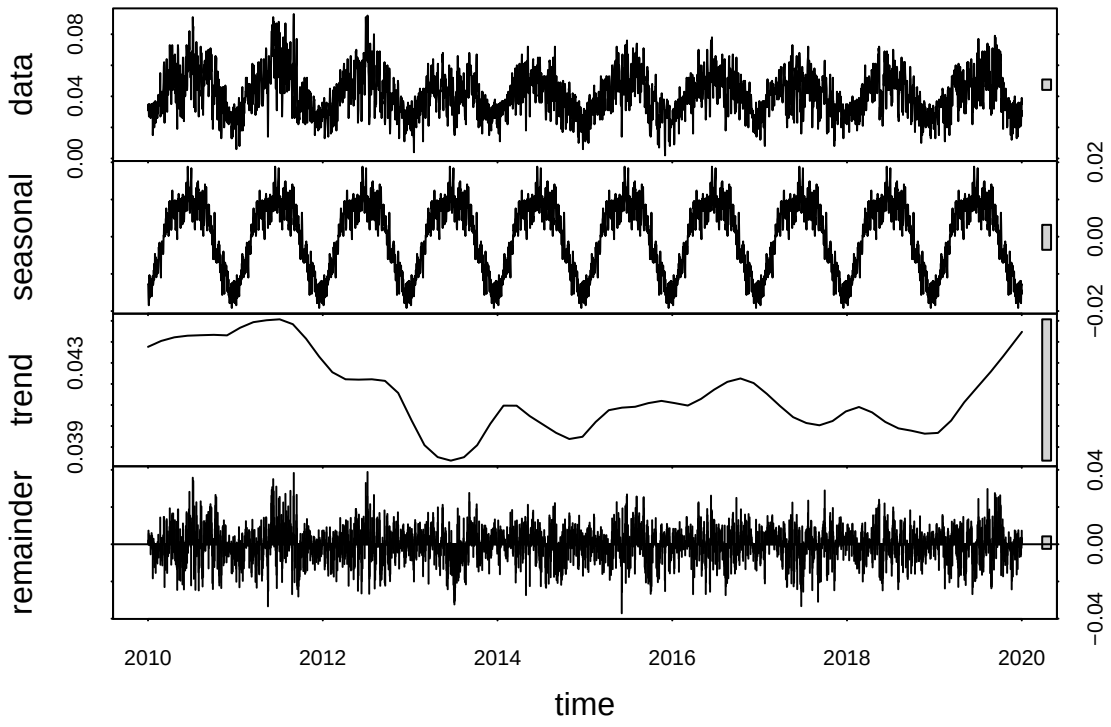
```
#9
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(month = month(Date), year = year(Date)) %>%
  group_by(year, month) %>%
  summarise(monthly.mean.concentration = mean(Daily.Max.8.hour.Ozone.Concentration))

GaringerOzone.monthly$Date <- as.Date(paste(GaringerOzone.monthly$year,
                                            GaringerOzone.monthly$month,
                                            "01",
                                            sep = "-"))
```

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.

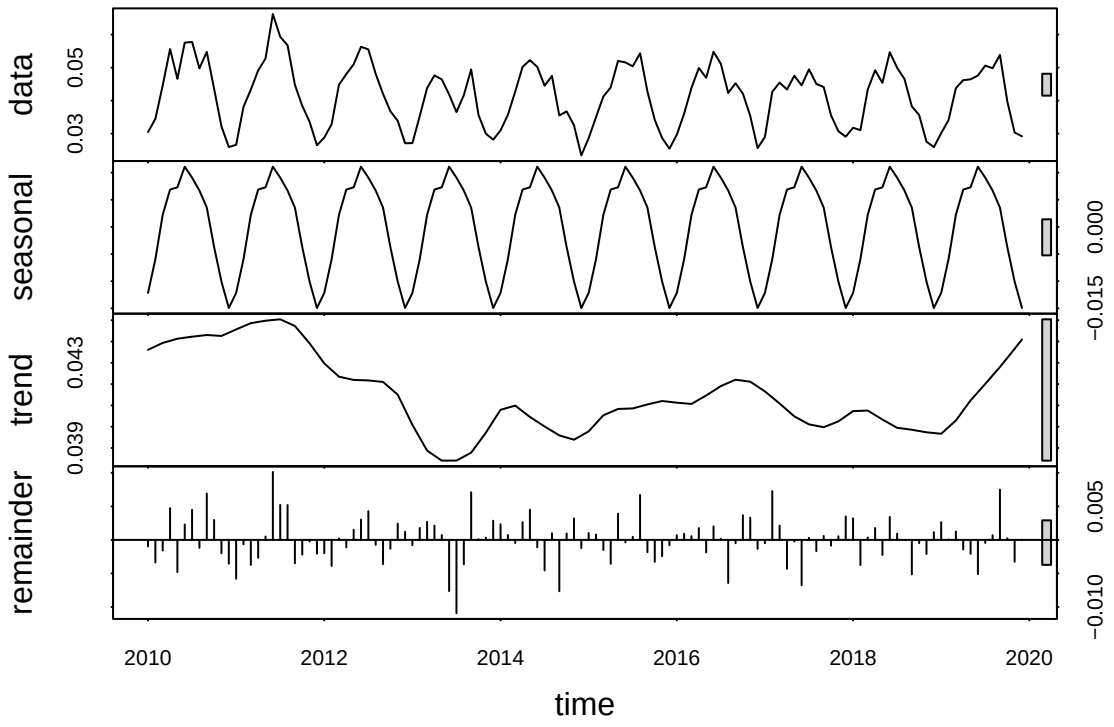
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010,1), frequency = 365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$monthly.mean.concentration,
                               start = c(2010,1), frequency = 12)
```

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

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




```
GaringerOzone.monthly.ts_Decomposed <- stl(GaringerOzone.monthly.ts,
                                           s.window = "periodic")
plot(GaringerOzone.monthly.ts_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
monthly.ozone_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(monthly.ozone_trend)
```

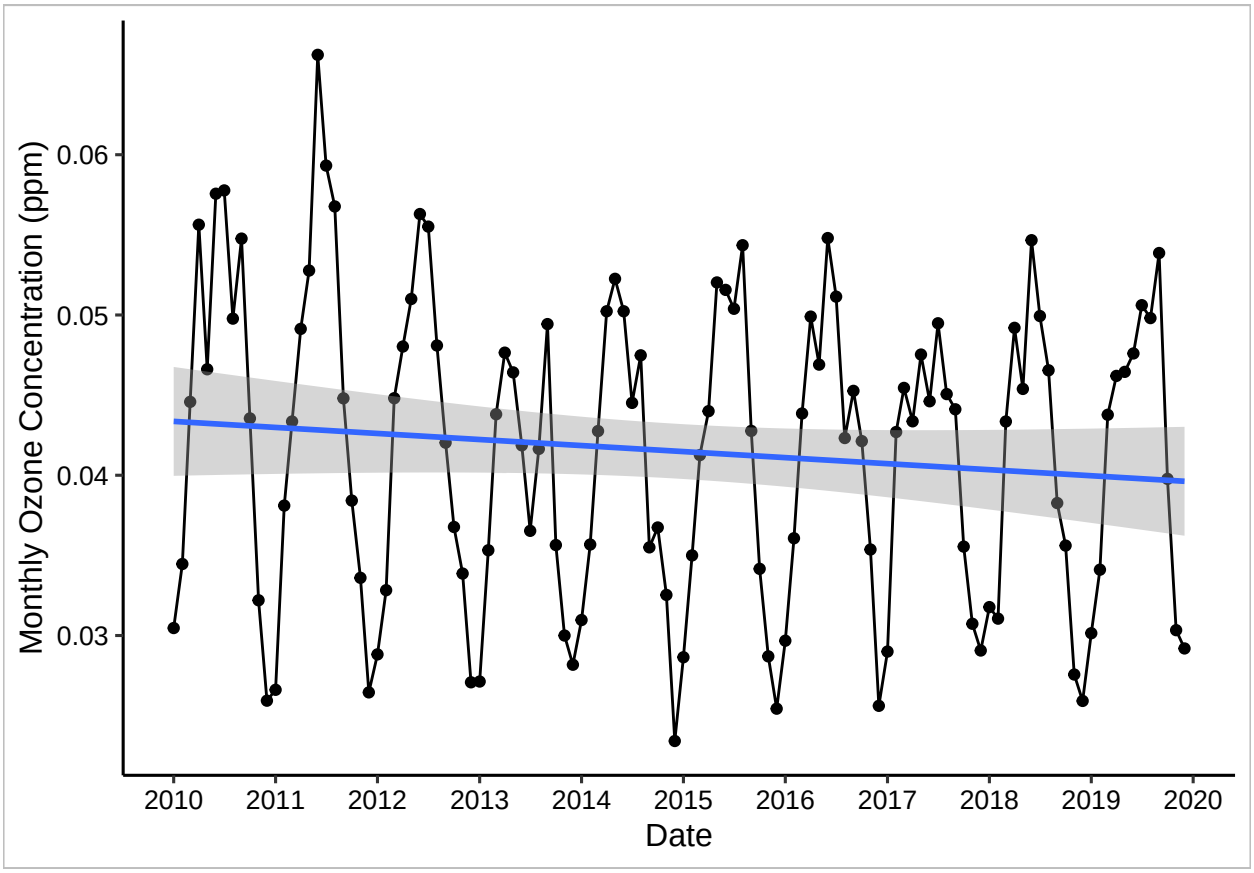
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: Because the data has a seasonal pattern, and the seasonal Mann-Kendall method is the one that can analyze it. Other methods, the linear regression, Mann-Kendall, and Spearman Rho are not suited for 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
monthly.ozone_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = monthly.mean.concentration)) +
  geom_point() +
  geom_line() +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year") +
```

```
  ylab("Monthly Ozone Concentration (ppm)") +
  geom_smooth( method = lm )
print(monthly.ozone_plot)
```



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 pattern at this station, therefore, by using the seasonal Mann-Kendall analysis, the result shows that the ozone concentrations have a decreasing trend since the 2010s (Score = -77, P = 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
GaringerOzone.monthly.ts_Component <- GaringerOzone.monthly.ts -
  GaringerOzone.monthly.ts_Decomposed$time.series[,"seasonal"]

#16
Noseasonal.trend <- MannKendall(GaringerOzone.monthly.ts_Component)
summary(Noseasonal.trend)
```

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

Answer: The Mann Kendall test still shows that the ozone concentrations have a decreasing trend. But the result is more significant (p = 0.0075402) compared to the result from the seasonal test (P = 0.046724). The trend is also stronger in the Mann Kendall test with an absolute value of 1179, while the absolute score value in the seasonal Mann Kendall test is only 77.