# Assignment 7: Time Series Analysis

Yikai Jing

## 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 Monday, March 14 at 7:00 pm.

### 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] "/Users/me/Environmental_Data_Analytics_2022/Assignments"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- 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(Kendall)
mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
```

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
air2010 <- read.csv('~/Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC
air2011 <- read.csv('~/Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC
air2012 <- read.csv('~/Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC
air2013 <- read.csv('~/Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC
air2014 <- read.csv('~/Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC
air2015 <- read.csv('~/Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC
air2016 <- read.csv('~/Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC
air2017 <- read.csv('~/Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC
air2018 <- read.csv('~/Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC
air2019 <- read.csv('~/Environmental_Data_Analytics_2022/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC
GaringerOzone <- rbind(air2010, air2011, air2012, air2013, air2014, air2015, air2016, air2017, air2018,
```

## 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
class(GaringerOzone$Date)
```

```
## [1] "character"
```

```
GaringerOzone$Date <- as.Date(GaringerOzone$Date, "%m/%d/%Y")
# 4
GaringerOzone <- 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"), "days"))
names(Days)[1] <- "Date"
# 6
GaringerOzone <- left_join(Days, GaringerOzone)
```

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

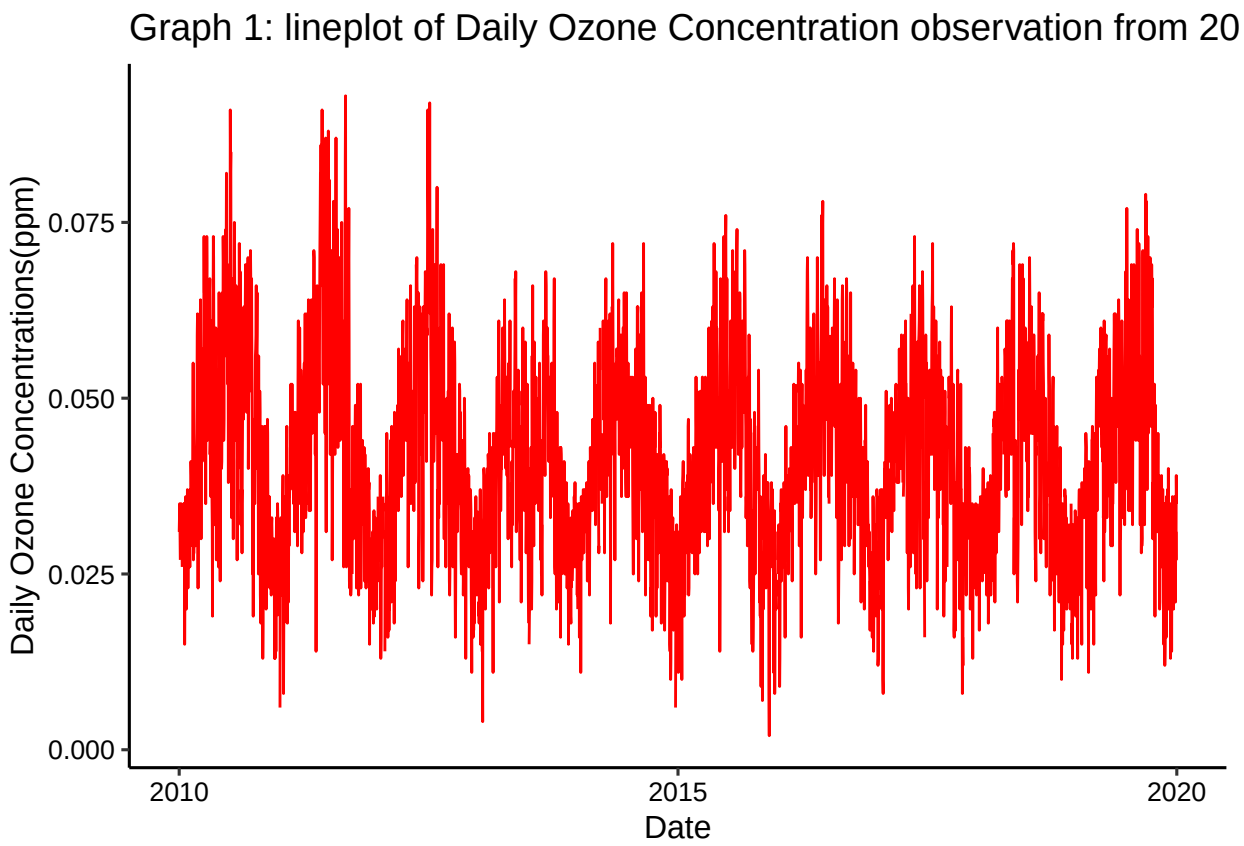
```
class(GaringerOzone$Date)
```

```
## [1] "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
ggplot(GaringerOzone) +
  geom_line(aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration), color = "red") +
  ylab("Daily Ozone Concentrations(ppm)") +
  ggtitle("Graph 1: lineplot of Daily Ozone Concentration observation from 2010 to 2019")+
  mytheme
```




Answer: Yes, the line plot suggest a seasonal trend that shows higher Ozone concentration during the summers and lower concentration during the winters for each year.

## 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
summary(GaringerOzone)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                       NA's   :63                           NA's   :63
```

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

Answer: Because we are assuming a simple linear relationship with our time series data. The piecewise constant or spline interpolation is based on a polynomial calculation, so we are not using them in this case.

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)

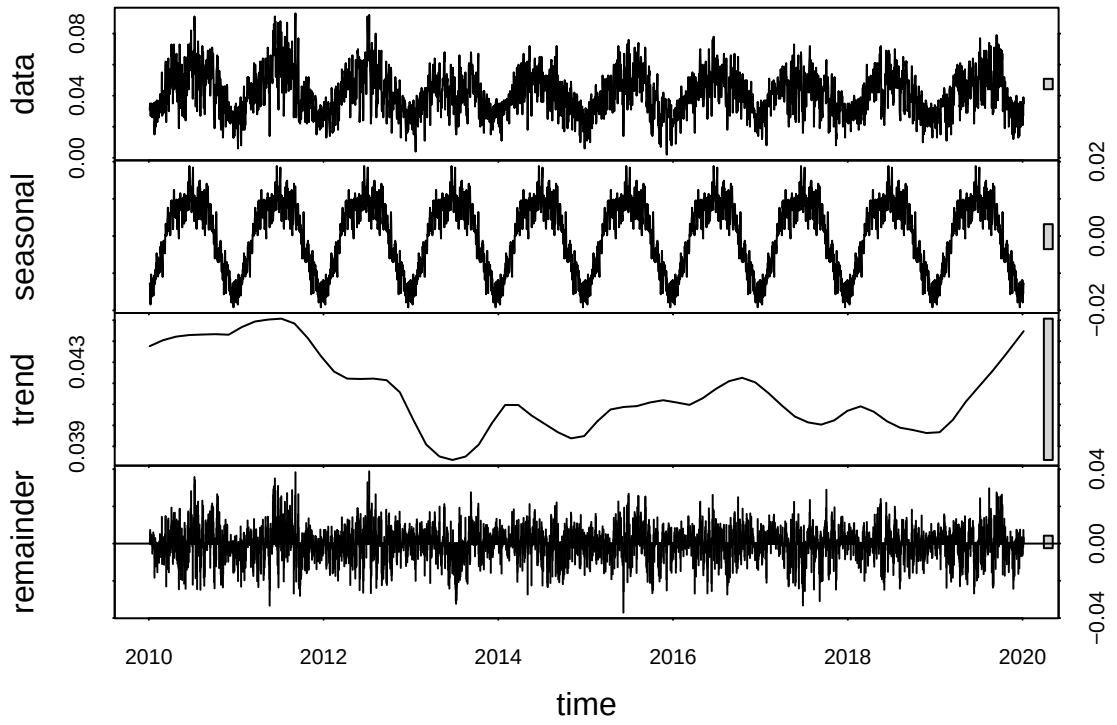
```
#9
GaringerOzone.monthly <- GaringerOzone_clean %>%
  mutate(Year = year(Date), Month = month(Date)) %>%
  mutate(Date = as.Date(paste(Year, Month, 1, sep = "-"))) %>%
  group_by(Date) %>%
  summarise(MonthlyOzone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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.

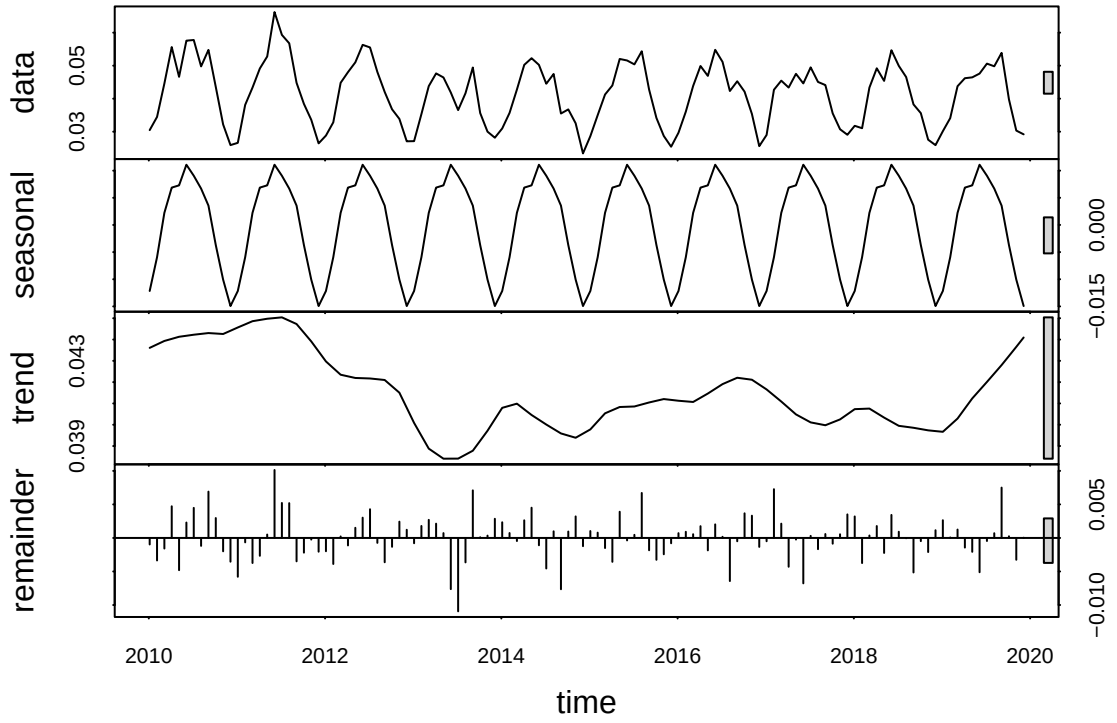
```
#10
GaringerOzone.daily.ts<- ts(GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration,
                            start = c(2010.01), frequency = 365)
GaringerOzone.monthly.ts<- ts(GaringerOzone.monthly$MonthlyOzone,
                              start = c(2010.01), frequency = 12)
```

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

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




```
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
monthly_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
monthly_trend1
```

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

```
summary(monthly_trend1)
```

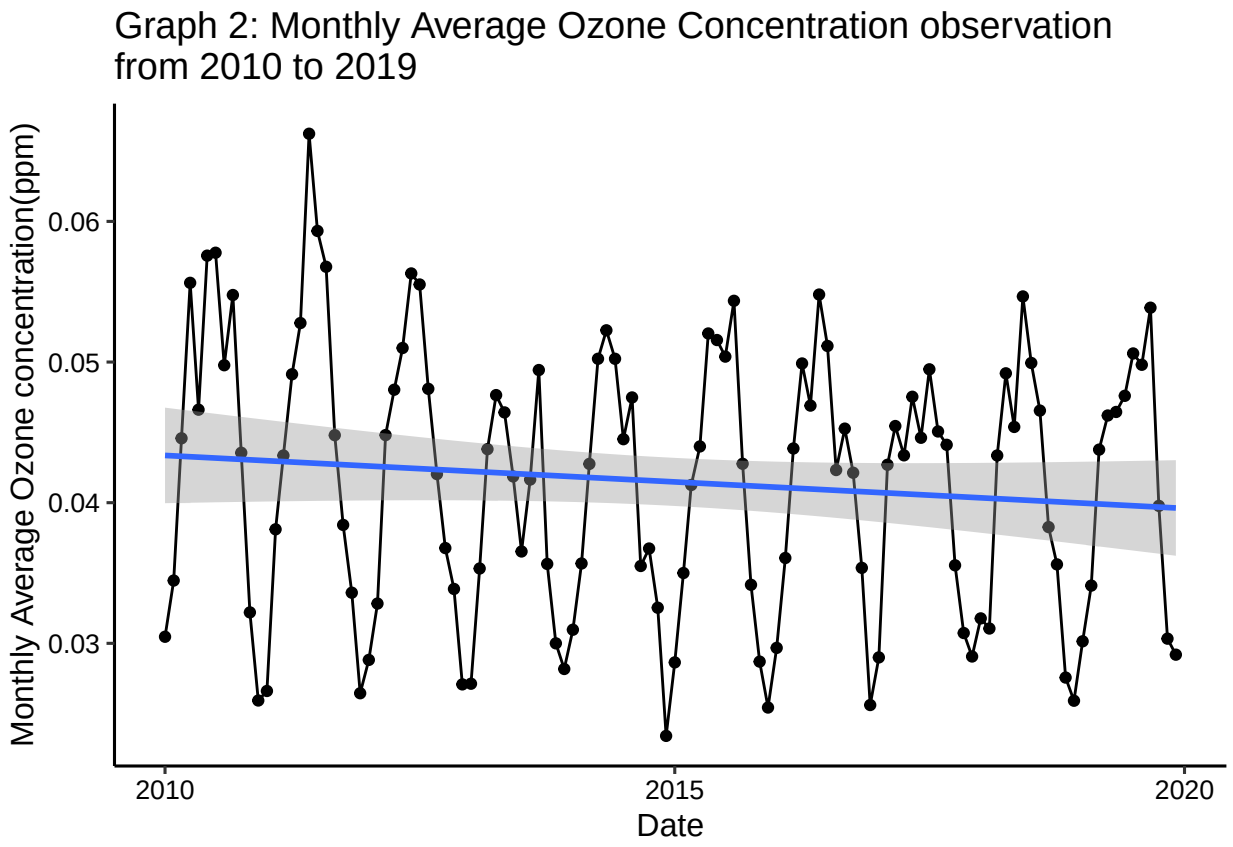
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: Because our data has a clear pattern of seasonality based on the plot that we generated on Q11.

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
G_monthly_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = MonthlyOzone)) +
  geom_point() +
  geom_line() +
  ylab("Monthly Average Ozone concentration(ppm)") +
  geom_smooth( method = lm ) +
  ggtitle("Graph 2: Monthly Average Ozone Concentration observation
from 2010 to 2019")+
  mytheme
print(G_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: Based on result of the Seasonal Mann-Kendall test, we can successfully reject the null hypothesis and state that the monthly avarage Ozone concentrations at Garinger High School in North Carolina are not stationary and follow a decreasing monotonic trend from 2010 to 2019 (tau = -0.143, 2-sided pvalue =0.046724, Score = -77). As the graph is showing, we can see a slightly negative trend of the monthly detected Ozone level from 2010 to 2019.

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
Garinger_seasonal_substracted <- as.data.frame(GaringerOzone.monthly_Decomposed$time.series[,1:3])
Garinger_seasonal_substracted <- mutate(Garinger_seasonal_substracted,
                                        Date = GaringerOzone.monthly$Date,
                                        Ozone = GaringerOzone.monthly$MonthlyOzone)
#16
monthly_trend2 <- Kendall::MannKendall(Garinger_seasonal_substracted$Ozone-Garinger_seasonal_substracte
monthly_trend2
```

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

```
summary(monthly_trend2)
```

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

Answer: By removing the seasonality and conducting the Mann Kendall test, we received a test result that has a much smaller 2-sided pvalue and a more significant implication that our data has a monotonic trend.