Assignment 7: Time Series Analysis

Jiyoung Park

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.

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/hyjgp/Documents/EDA-Fall2022"
```

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

```
## Warning: package 'zoo' was built under R version 4.2.2
```

```
library(trend)
```

```
## Warning: package 'trend' was built under R version 4.2.2
```

```
my_theme <-theme_classic(base_size = 15)+
  theme(legend.position = "bottom")
theme_set(my_theme)

#2
ozone10 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFactors = T
ozone11 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFactors = T
ozone12 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFactors = T
ozone13 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFactors = T
ozone14 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFactors = T
ozone15 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFactors = T
ozone16 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFactors = T
ozone17 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFactors = T
ozone18 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFactors = T
ozone19 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFactors = T

GaringerOzone <- rbind(ozone10,ozone11,ozone12, ozone13, ozone14, ozone15, ozone16, ozone17, ozone18, oz
```

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$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

# 4
GaringerOzone_Processed <-
  GaringerOzone %>%
  select(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"), row.names = NULL, optional =FALSE)
colnames(Days)= "Date"

# 6
GaringerOzone <- left_join(Days,GaringerOzone_Processed,by="Date")
```

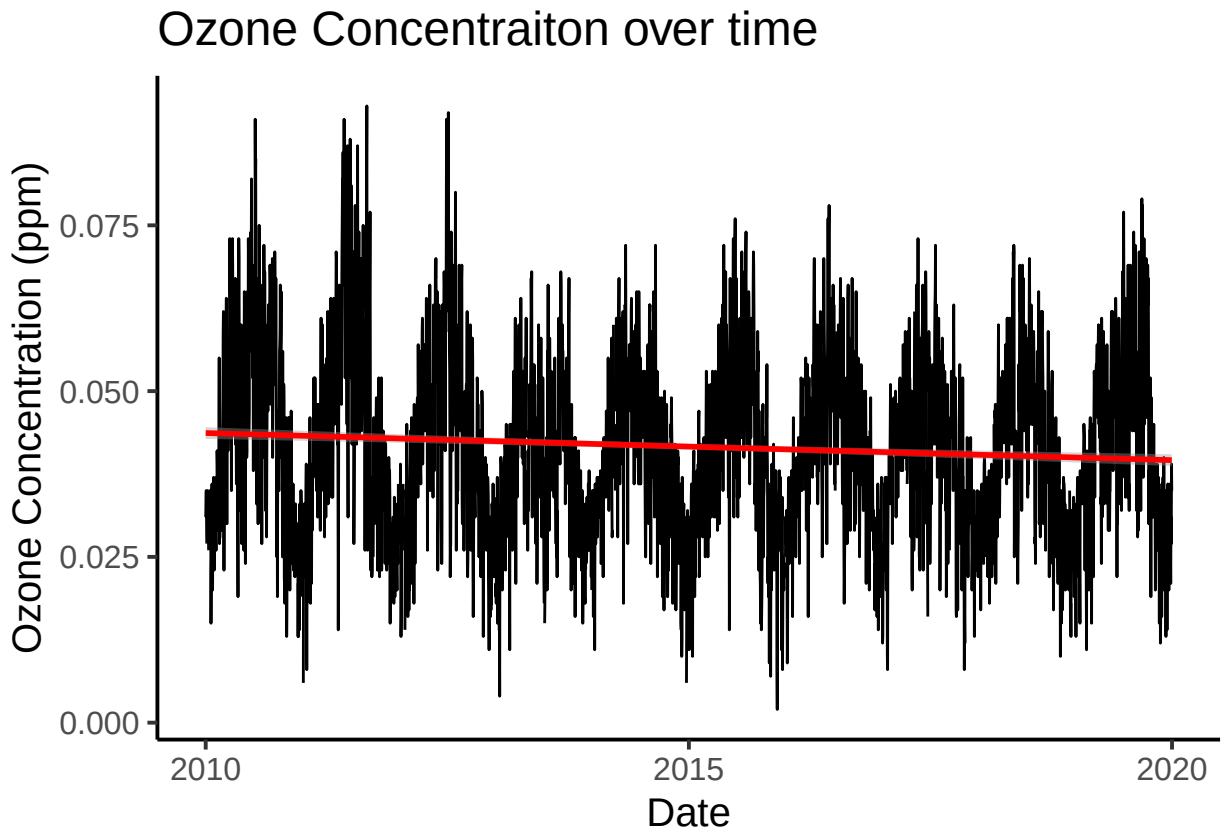
Visualize

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
ggplot(GaringerOzone) +
  geom_line(aes(x=Date,y= Daily.Max.8.hour.Ozone.Concentration)) +
  geom_smooth(aes(x=Date,y= Daily.Max.8.hour.Ozone.Concentration), method = lm, color ="red")+
  labs(title ="Ozone Concentraiton over time", y = "Ozone Concentration (ppm)")
```

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

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



Answer: there is an overall declining trend of ozone concentration over the years. Also, there is a seasonal pattern of ozone concentration observed.

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

Answer: Linear interpolation was used to fill in missing daily data because linear trend was observed from question 7. therefore,linear interpolation would give better estimate of missing values than piecewise constant or spline interpolation.

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_Filled %>%
  mutate(Year = year(Date)) %>%
  mutate(Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarize(meanConc = mean(OzoneConc), .groups ='drop') %>%
  mutate (Date = paste(Month, Year, sep = "/")) %>%
  select(Date,meanConc)

#GaringerOzone.monthly$Date <-as.Date(GaringerOzone.monthly$Date,format = "%m/%YY")
```

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.

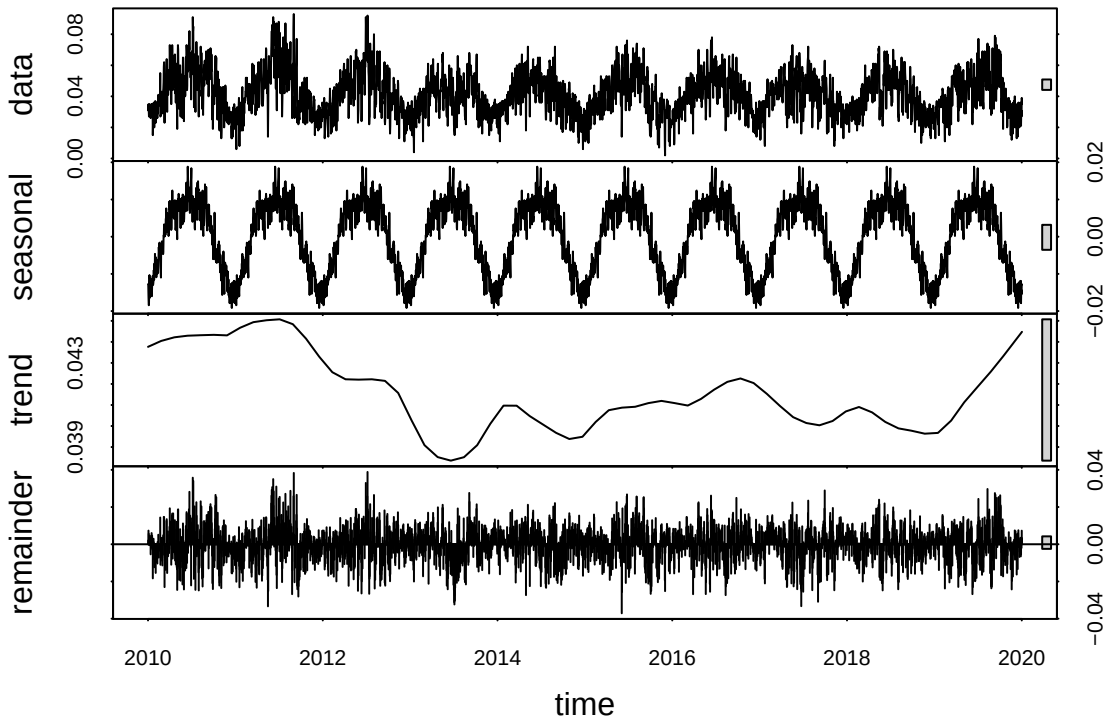
```
#10
f_day <-day(first(GaringerOzone$Date))
f_month <- month(first(GaringerOzone$Date))
f_year <- year(first(GaringerOzone$Date))

GaringerOzone.daily_ts <- ts(GaringerOzone_Filled$OzoneConc, start = c(f_year,f_month,f_day),frequency =
GaringerOzone.monthly_ts <- ts(GaringerOzone.monthly$meanConc, start = c(f_year,f_month), frequency = 1
```

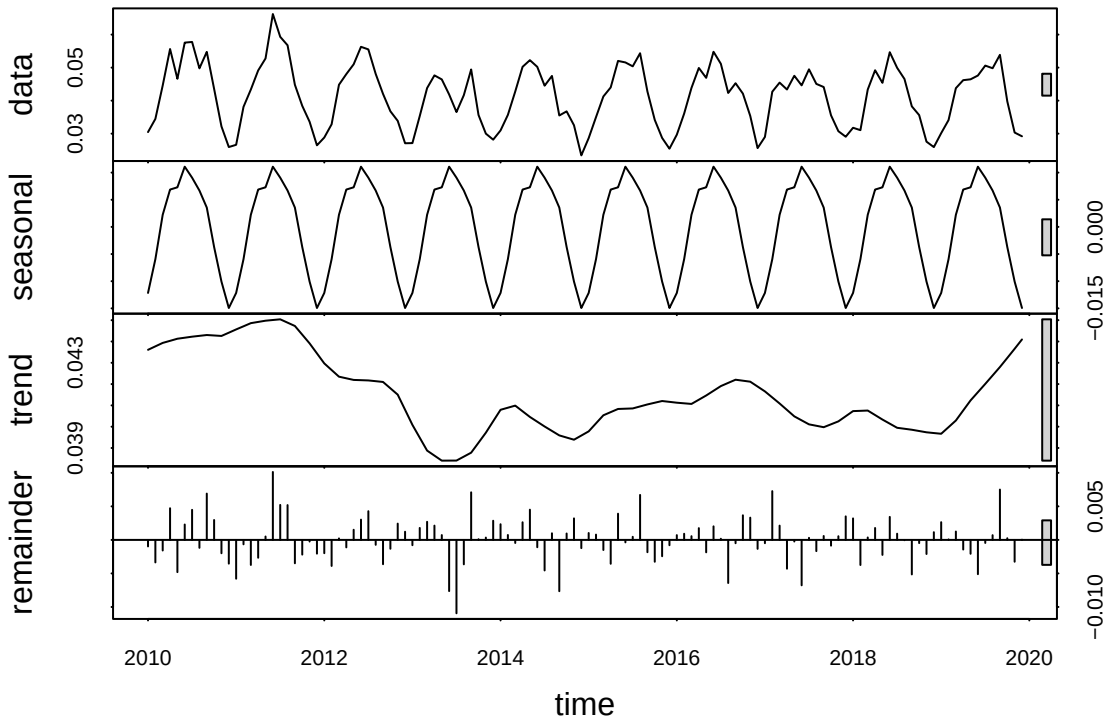
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#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
trend::smk.test(GaringerOzone.monthly_ts)
```

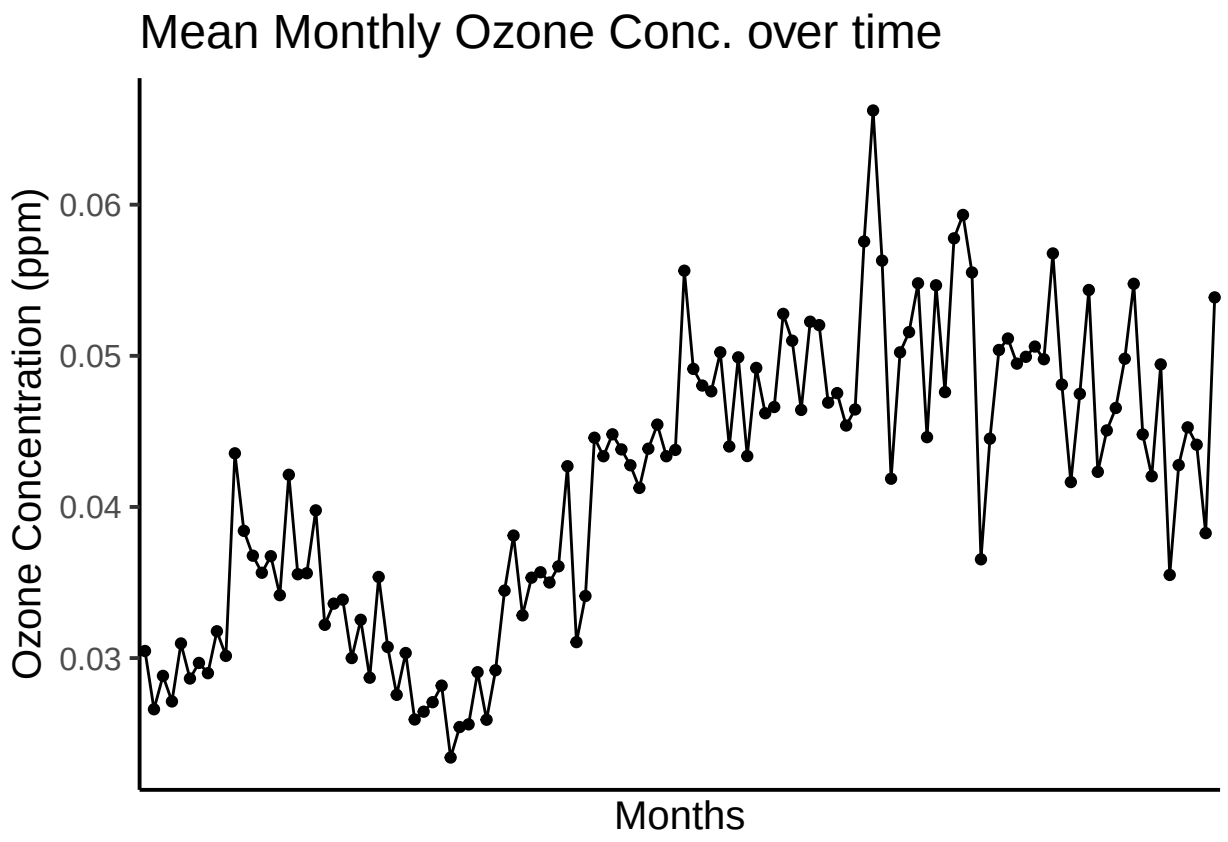
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly_ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

Answer: the seasonal Mann-Kendall is most appropriate monotoinc trend analysis because there is a clear seasonal component observed in this 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.

```r
# 13

ggplot(GaringerOzone.monthly,aes(x=Date, y= meanConc)) +
  geom_point()+
  geom_line(aes(group=1))+
  labs(title = "Mean Monthly Ozone Conc. over time", x= "Months", y="Ozone Concentration (ppm)")+
  scale_x_discrete(breaks = seq(1, nlevels(GaringerOzone.monthly$Date),length.out = 12)) +
  theme(axis.text.x = element_text(angle = 60, vjust = 0.5, hjust=1))
```



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: there is an overall increase of ozone concentration over the years, and there is an overall seasonal variations in ozone concentration. the p value for seasonal Mann-Kendall was 0.04 which is smaller than 0.05 so we reject null hypothesis that there is no seasonality in this data. However, overall trend of increase and decrease is also observed disregarding the seasonality. therefore, further analysis should be done to determine if other variation other than seasonality is causing the fluctuation.

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.nonseasonal <- GaringerOzone.monthly_ts - GaringerOzone.monthly_Decomposed$time.s
```

```
#16
library(Kendall)
```

```
## Warning: package 'Kendall' was built under R version 4.2.2
```

```
MannKendall(GaringerOzone.monthly.nonseasonal)
```

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

Answer: tau = -0.165, 2-sided pvalue =0.0075402; p vlaue is less than 0.05.therefore we reject null hypothesis that the data is stationary. there is a trend in this data. the seasonal mann kendall test had much higher p value od 0.049 than that of this test.