Assignment 7: Time Series Analysis

Zoe Wong

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/Zoe/OneDrive/DukeMEM_Yr1/Spring/Environmental_Data_Analytics_2021/Assignments"
```

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

mytheme <- theme_light(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
```

```
        legend.position = "top")
theme_set(mytheme)

#2
set2010 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
set2011 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
set2012 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
set2013 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
set2014 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
set2015 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
set2016 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
set2017 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
set2018 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
set2019 <- read.csv("../Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")

GaringerOzone <- rbind(set2010, set2011, set2012, set2013, set2014, set2015, set2016, set2017, set2018,
```

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 <- 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"), by = "day"))
Days <- rename(Days, Date = `seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "day")`)

# 6
GaringerOzone <- left_join(Days, GaringerOzone)
```

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

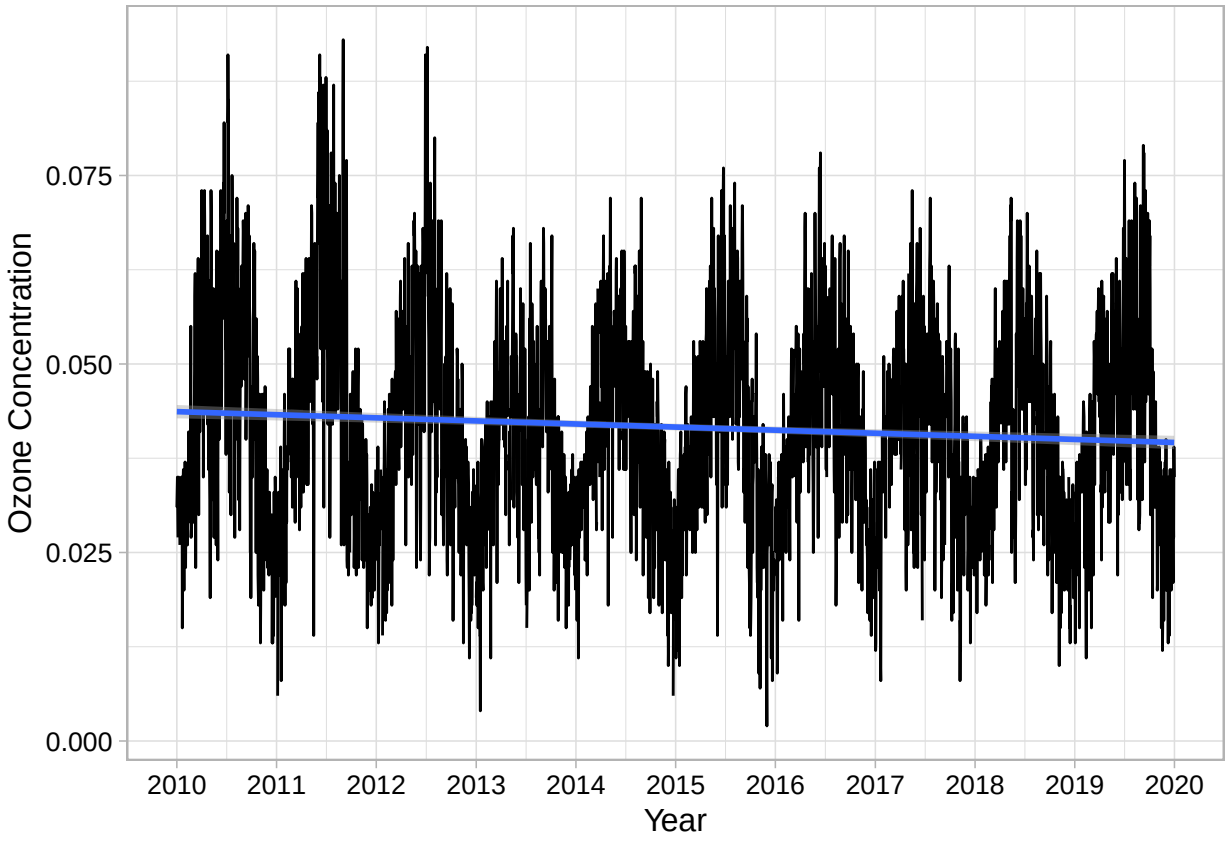
Visualize

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?

```r
#7
ozone.line <- 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") +
  scale_x_date(breaks = "year", date_labels = "%Y") +
  labs(y = "Ozone Concentration", x = "Year")
print(ozone.line)
```

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

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



Answer: There does seem to be a negative trend, but it's very slight.

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 <- mutate(GaringerOzone, Daily.Max.8.hour.Ozone.Concentration =
                          na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: Linear interpolation allows us to connect the dots between two points on either side of the missing data. This makes sense in this case because the data exhibit pretty steady variation and the gaps are small - individual days rather than large spans like weeks or months. A piecewise constant interplation isn't suitable because the rest of the data are changing in a smooth curve with only small gaps between days, and a spline interpolation would be too complicated for the data and the small missing gaps.

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), month = month(Date)) %>%
  group_by(month, year) %>%
  summarise(mean.ozone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

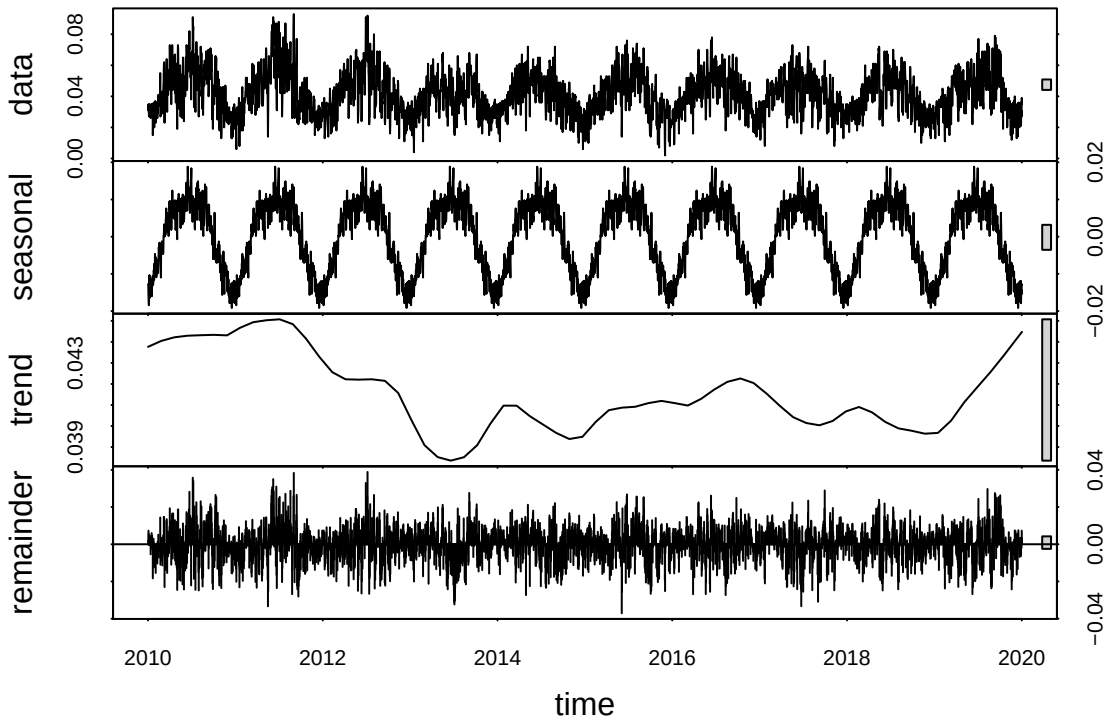
```
GaringerOzone.monthly$Date <- paste(GaringerOzone.monthly$year, GaringerOzone.monthly$month, "01", sep =
GaringerOzone.monthly$Date <- as.Date(GaringerOzone.monthly$Date, format = "%Y-%m-%d")
```

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.

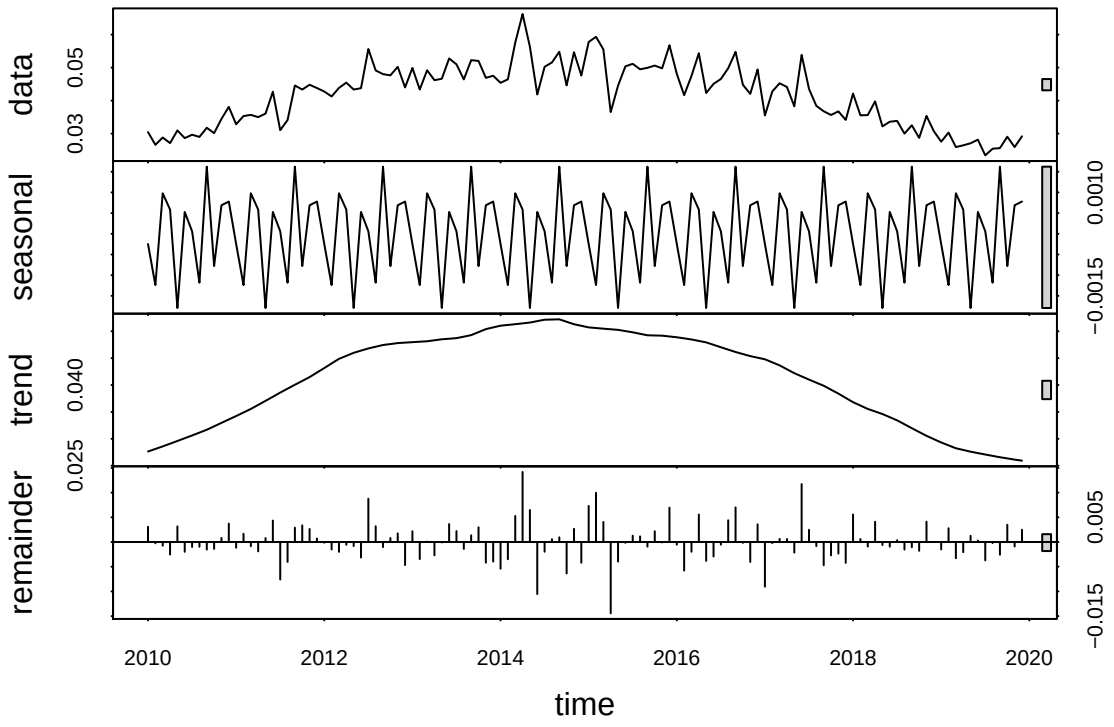
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration, start = c(2010, 01), fr
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean.ozone, start = c(2010, 01), frequency = 12)
```

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

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




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



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

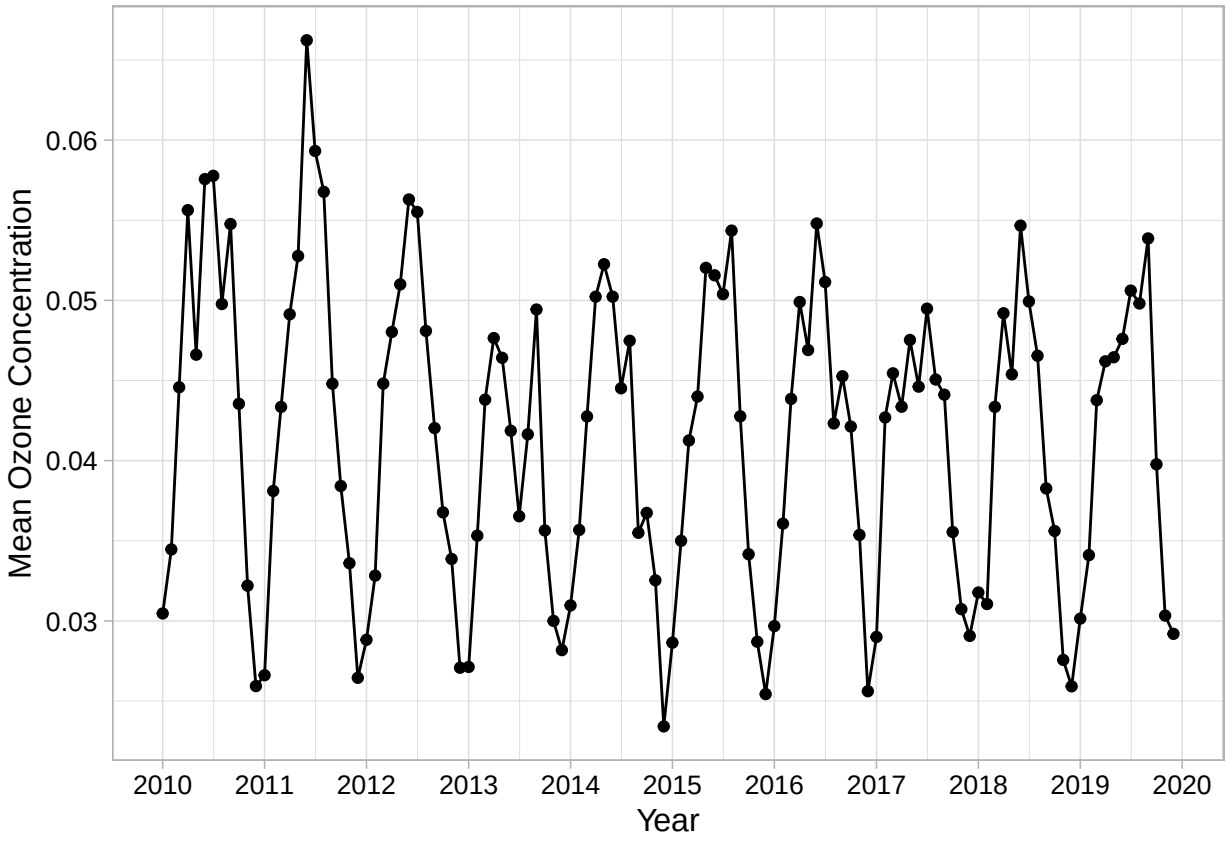
```
#12
GaringerOzone.monthly.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(GaringerOzone.monthly.trend)
```

```
## Score =  -54 , Var(Score) = 1500
## denominator =  540
## tau = -0.1, 2-sided pvalue =0.16323
```

Answer: The seasonal Mann-Kendall is appropriate here because the data exhibit a seasonal variation in addition to the larger trend.

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.monthly.plot <- ggplot(GaringerOzone.monthly) +
  geom_point(aes(x = Date, y = mean.ozone)) +
  geom_line(aes(x = Date, y = mean.ozone)) +
  scale_x_date(breaks = "year", date_labels = "%Y") +
  labs(y = "Mean Ozone Concentration", x = "Year")
print(GaringerOzone.monthly.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: Ozone concentrations have not changed significantly at this station during the 2010s. After analyzing the data, the seasonal Mann-Kendall test showed that there is not enough evidence to conclude that ozone concentrations have changed significantly over time ($p = 0.16$).

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.decomp$time.series[,1:3])
GaringerOzone.monthly.components <- mutate(GaringerOzone.monthly.components,
                                     mean.ozone = GaringerOzone.monthly$mean.ozone)
GaringerOzone.monthly.noseason <- ts(data = GaringerOzone.monthly.components$mean.ozone - GaringerOzone
                                   start = c(2010, 01), frequency = 12)

#16
GaringerOzone.monthly.mk <- mk.test(GaringerOzone.monthly.noseason)
print(GaringerOzone.monthly.mk)
```

```
##
##  Mann-Kendall trend test
##
## data:  GaringerOzone.monthly.noseason
## z = -1.6263, n = 120, p-value = 0.1039
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -7.180000e+02  1.943667e+05 -1.005602e-01
```

Answer: After removing the seasonal component from the time series, the Mann-Kendall test still shows that there is not a significant trend in the data ($p = 0.10$). The Mann-Kendall test returns a smaller p-value than the seasonal Mann-Kendall test, but the result is still not statistically significant.