Assignment 7: Time Series Analysis

Cal Oakley

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] "K:/GradSchool/Spring2022/EnvironmentalDataAnalytics/Environmental_Data_Analytics_2022/Assignmen
```

```
library(tidyverse)
library(lubridate)
library(zoo)
library(trend)
library(ggplot2)

CalsTheme <- theme_classic(base_size = 16) +
  theme(axis.text = element_text(color = "gray"), legend.position = "left",
        legend.justification = 2)
theme_set(CalsTheme)
```

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
GraingerFiles <- list.files(path = "../Data/Raw/Ozone_TimeSeries",
                            pattern = "*.csv",
                           full.names = TRUE) #creates list of file names in Arc)
```

```
library(plyr)

GraingerOzone <- GraingerFiles %>%
  ldply(read.csv) %>%
  data.frame(stringsAsFactors = TRUE)

#str(GraingerOzone)
```

Wrangle

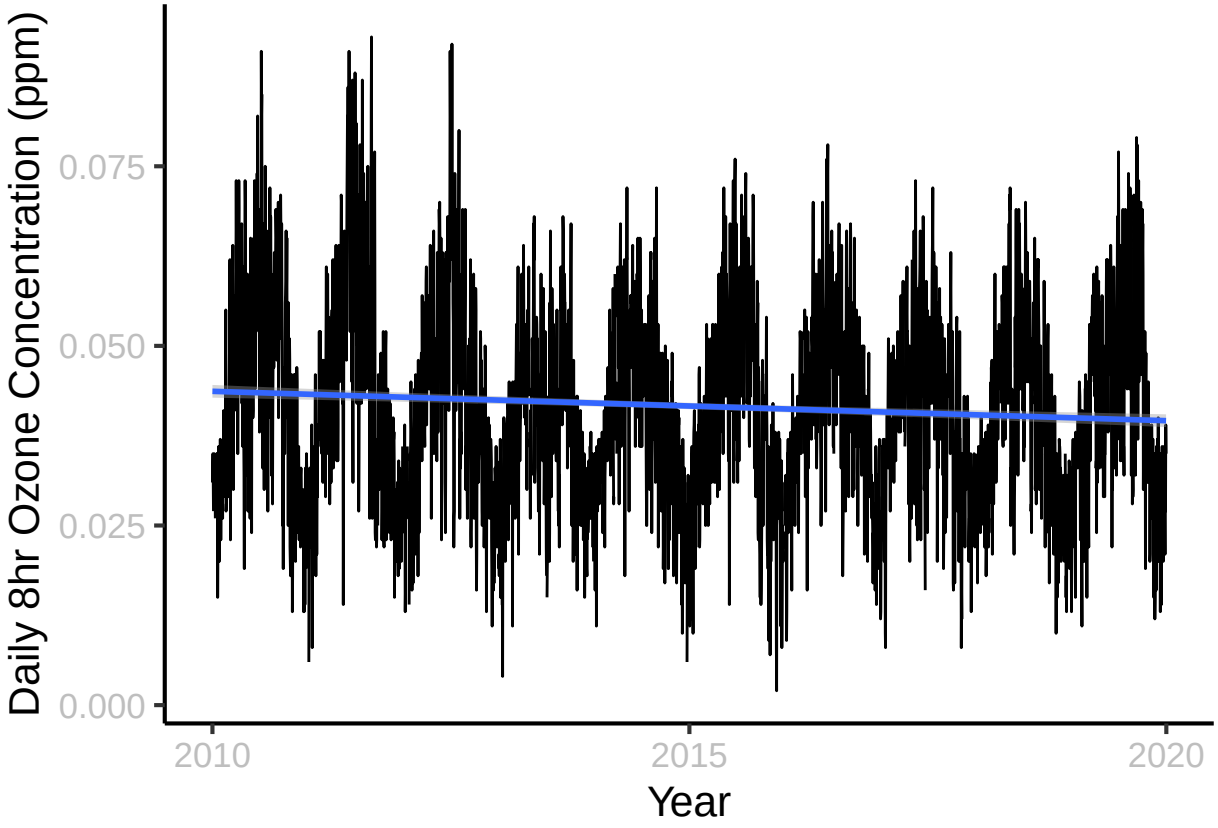
3. Set your date column as a date class.
4. Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.
5. Notice there are a few days in each year that are missing ozone concentrations. We want to generate a daily dataset, so we will need 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
GraingerOzone$Date <- mdy(GraingerOzone$Date)

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

# 5
Days <- as.data.frame(seq(as.Date(first(GraingerOzone$Date)),
                          as.Date(last(GraingerOzone$Date)),
                          by = "day"))
colnames(Days)[1] <- "Date"

# 6
GraingerOzone <- left_join(Days, GraingerOzone, 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
A07_plot1 <- ggplot(GraingerOzone,
                    aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  labs(x = "Year", y = "Daily 8hr Ozone Concentration (ppm)") +
  geom_smooth(method = lm)

print(A07_plot1)
```



Answer: Looking at this plot, there does appear to be a trend over time. It appears as though the Ozone concentrations at this site decrease 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
GraingerOzone <- GraingerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration.fill =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration)) %>%
  mutate(DAILY_AQI_VALUE.fill =
           zoo::na.approx(DAILY_AQI_VALUE))
```

Answer: We are using linear interpolation because we are trying to connect the dots. We want these dots connected in ways that make sense. Since piecewise uses the neighboring values we risk interpolating values that are more or less than they should be on a given date. Spline uses more than just the local data and we dont want that either because our data has seasonality (visible in the line plot) and interpolating at a larger time scale could create variables that don't make sense either.

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)

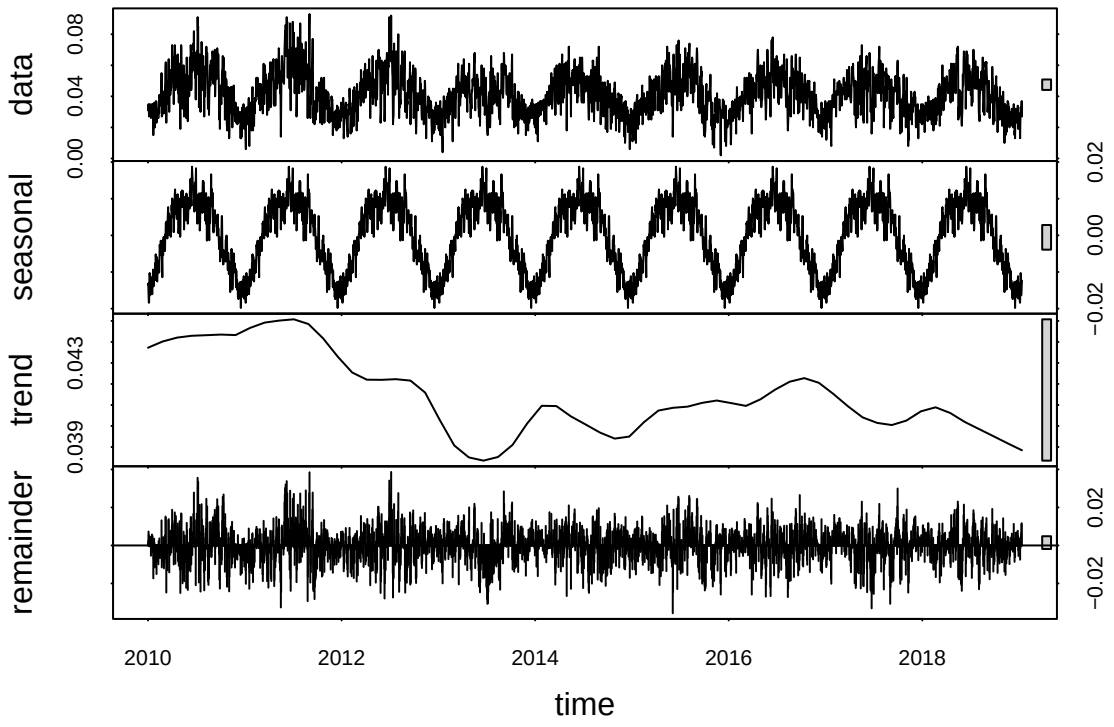
```r
#9
GraingerOzone_monthly <- GraingerOzone %>%
  separate(Date, c("Year", "Month", "Day"), sep = "-") %>%
  mutate(Date = paste(Year, Month, "01", sep = "-")) %>%
  group_by(Date) %>%
  dplyr::summarise(Mean_Ozone_Conc = mean(Daily.Max.8.hour.Ozone.Concentration.fill))

GraingerOzone_monthly$Date <- as.Date(GraingerOzone_monthly$Date, "%Y-%m-%d")

#str(GraingerOzone_monthly)
```

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.

```r
#10
GraingerOzone_daily_ts <-
  ts(GraingerOzone$Daily.Max.8.hour.Ozone.Concentration.fill,
     start = c(2010, 1),
     end = c(2019, 12),
     frequency = 365)
     # frequency = length(GraingerOzone$Daily.Max.8.hour.Ozone.Concentration.fill))

GraingerOzone_monthly_ts <-
  ts(GraingerOzone_monthly$Mean_Ozone_Conc,
     start = c(2010,1),
     end = c(2019, 12),
     frequency = 12)
     #frequency = length(GraingerOzone_monthly$Mean_Ozone_Conc))
```

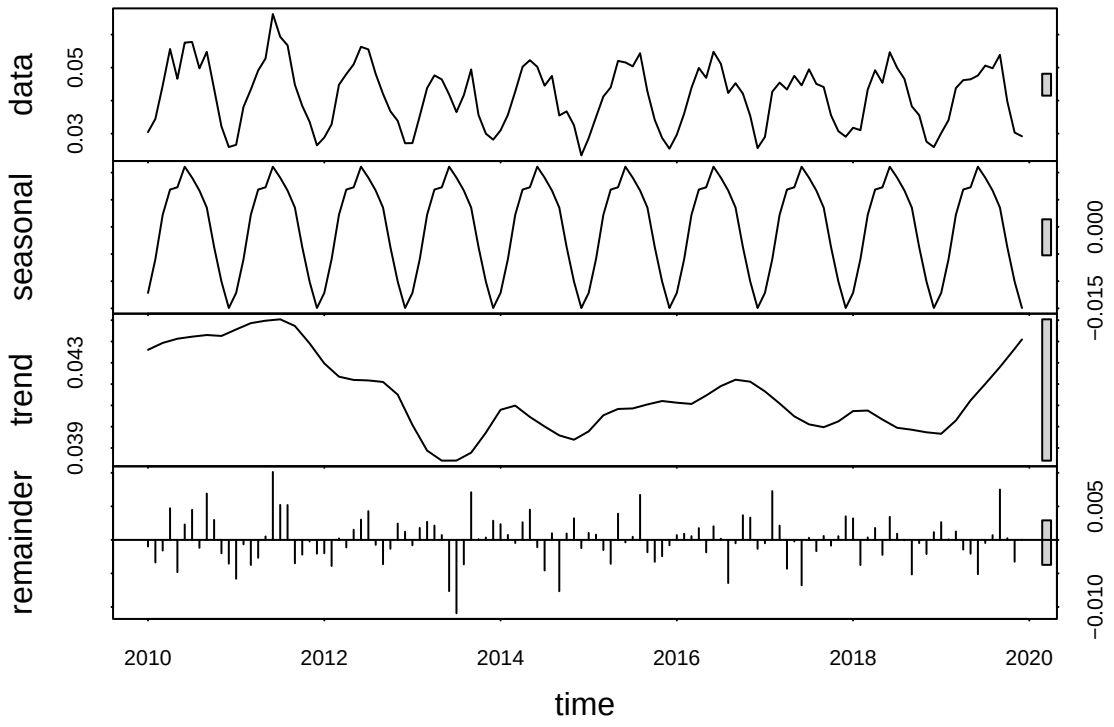
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```r
#11
Daily_decomp <- stl(GraingerOzone_daily_ts, s.window = "periodic")
print(plot(Daily_decomp))
```




```
## NULL
```

```
Monthly_decomp <- stl(GraingerOzone_monthly_ts, s.window = "periodic")
print(plot(Monthly_decomp))
```




```
## NULL
```

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

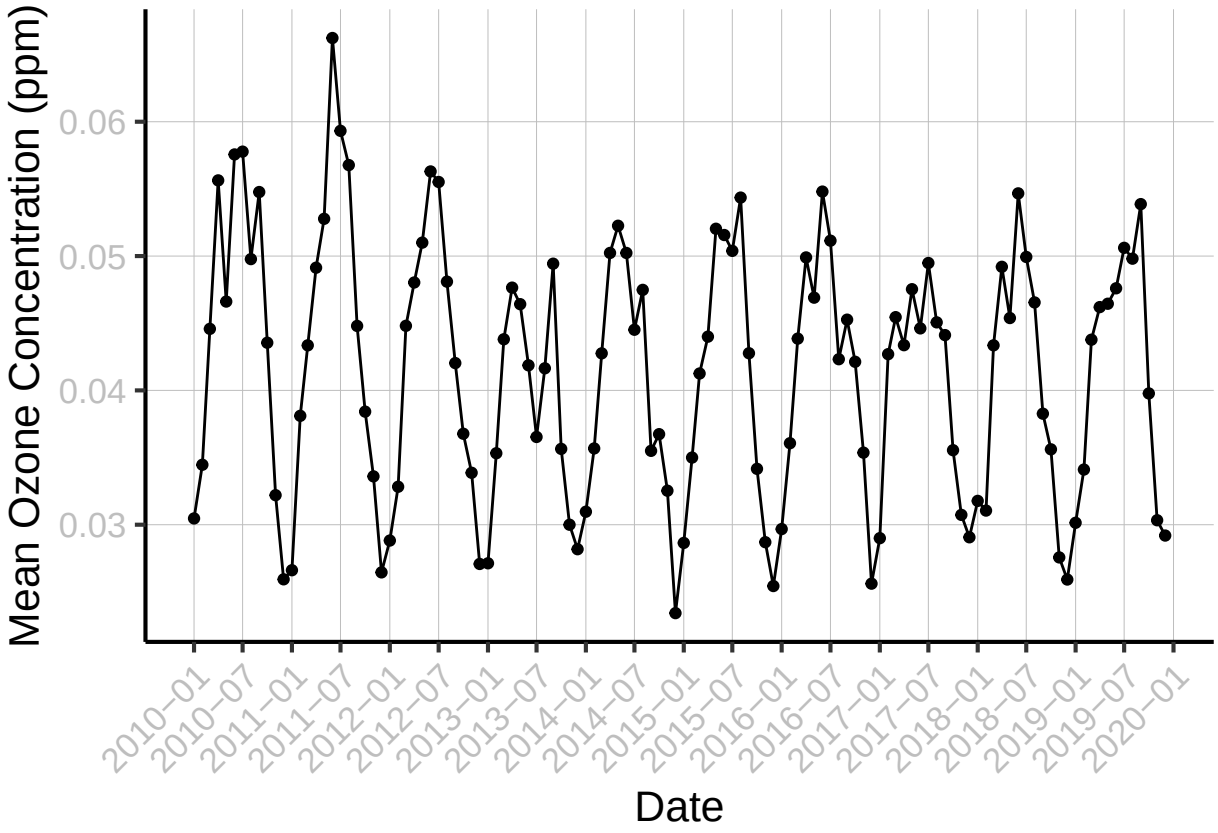
```
#12
MonthlyTrend1 <- Kendall::SeasonalMannKendall(GraingerOzone_monthly_ts)
print(MonthlyTrend1)
```

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

Answer: Seasonal Mann-Kendall test works here because we have a seasonal time series (evidenced by the cyclic variation in the plots above) with non-parametric variables.

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
A07_plot2 <- ggplot(GraingerOzone_monthly,
                    aes(x = Date, y = Mean_Ozone_Conc)) +
  geom_point() +
  geom_line() +
  labs(y = "Mean Ozone Concentration (ppm)") +
  scale_x_date(date_breaks = "6 months", date_labels = "%Y-%m") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        panel.grid.major.x = element_line(size = 0.1, color = "gray"),
        panel.grid.major.y = element_line(size = 0.1, color = "gray"))

print(A07_plot2)
```



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 the Seasonal Mann-Kendall test there, is a significant negative trend in our data (tau = -0.143, p-value = 0.046724). What this means for our original question is that Ozone concentrations did decrease between 2010 and 2020.

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
Monthly_components <- as.data.frame(Monthly_decomp$time.series[,1:3]) %>%
  mutate(Observed = GraingerOzone_monthly_ts)

Monthly_components <- Monthly_components %>%
  mutate(Obs_min_Seasn = Observed - seasonal)

#16
MonthlyTrend2 <- Kendall::MannKendall(Monthly_components$Obs_min_Seasn)
print(MonthlyTrend2)
```

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

Answer: Based on a (non-seasonal) Mann-Kendall test, there is still a significant negative trend in our data (tau = -0.165, p-value = 0.0075402). These values indicated a stronger and more

significant trend than the results of the Seasonal Mann-Kendall test performed earlier. In terms of our orignal question, this means that after removing seasonal variation in atmospheric Ozone concentrations, the decrease in concetrations between 2010 and 2020 becomes stronger.