Assignment 7: Time Series Analysis

Danlei Zou

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

#checking working directory
getwd()
```

```
## [1] "/Users/danleizou/EDA-Fall2022"
```

```
#loading necessary packages
library(tidyverse)
library(lubridate)
library(zoo)
library(trend)
```

```
#set theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)

#2

#importing raw datasets from Ozone_TimeSeries folder
Ozone2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFactors =
Ozone2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFactors =
Ozone2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFactors =
Ozone2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFactors =
Ozone2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFactors =
Ozone2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFactors =
Ozone2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFactors =
Ozone2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFactors =
Ozone2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFactors =
Ozone2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFactors =

#combining into one dataset
GaringerOzone <- rbind(Ozone2010, Ozone2011, Ozone2012, Ozone2013, Ozone2014, Ozone2015, Ozone2016, Ozon
```

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

#setting sampledate column to date objects
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

#checking sampledate column
class(GaringerOzone$Date)
```

```
## [1] "Date"
```

```
# 4

#wrangling datasets according to parameters
```

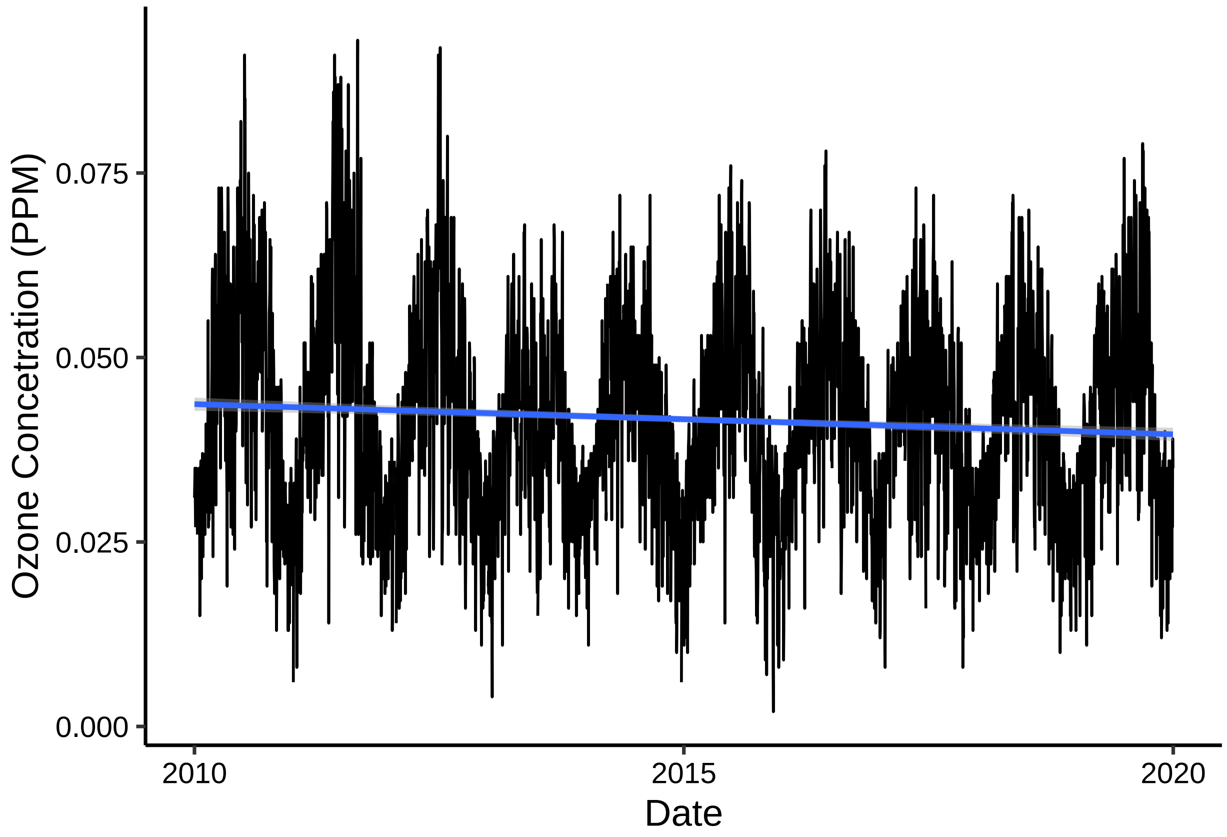
```r
GaringerOzone.processed <-
  GaringerOzone %>%
  select(Date,Daily.Max.8.hour.Ozone.Concentration,DAILY_AQI_VALUE)

# 5

#generating daily data frame and renaming to date
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"),by = "day"))

colnames(Days) <- 'Date'

# 6

#combining data frames
GaringerOzone <- left_join(Days, GaringerOzone.processed, by = c("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

GaringerOzone.lineplot <- ggplot(GaringerOzone,
                                 aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm", formula= y~x) + ylab("Ozone Concetration (PPM)") +
  xlab("Date")

print(GaringerOzone.lineplot)
```

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



Answer: The line plot's trend line shows a small decrease in ozone concentration over time from 2010 to 2020.

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

#filling in missing daily data for ozone concentration
GaringerOzone.complete <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: We wouldn't use a piecewise constant because it assumes that any missing data is equal to the measurement made nearest to that date, which would cause the trend to flatten instead of following the downward trend. We also wouldn't use a splint interpolation because it would use a quadratic function to interpolate, rather than drawing it linearly in a straight line. Linear interpolation is the best fit for this because it will show the connection from day to day in a straight line.

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

#creating new data frame containing aggregated data
GaringerOzone.monthly <-
  GaringerOzone.complete %>%
  mutate(Month = month(Date), Year = year(Date)) %>%
  mutate(Month_Year = my(paste0(Month, "-", Year))) %>%
  dplyr::group_by(Month, Year, Month_Year) %>%
  dplyr::summarise(MeanOzone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

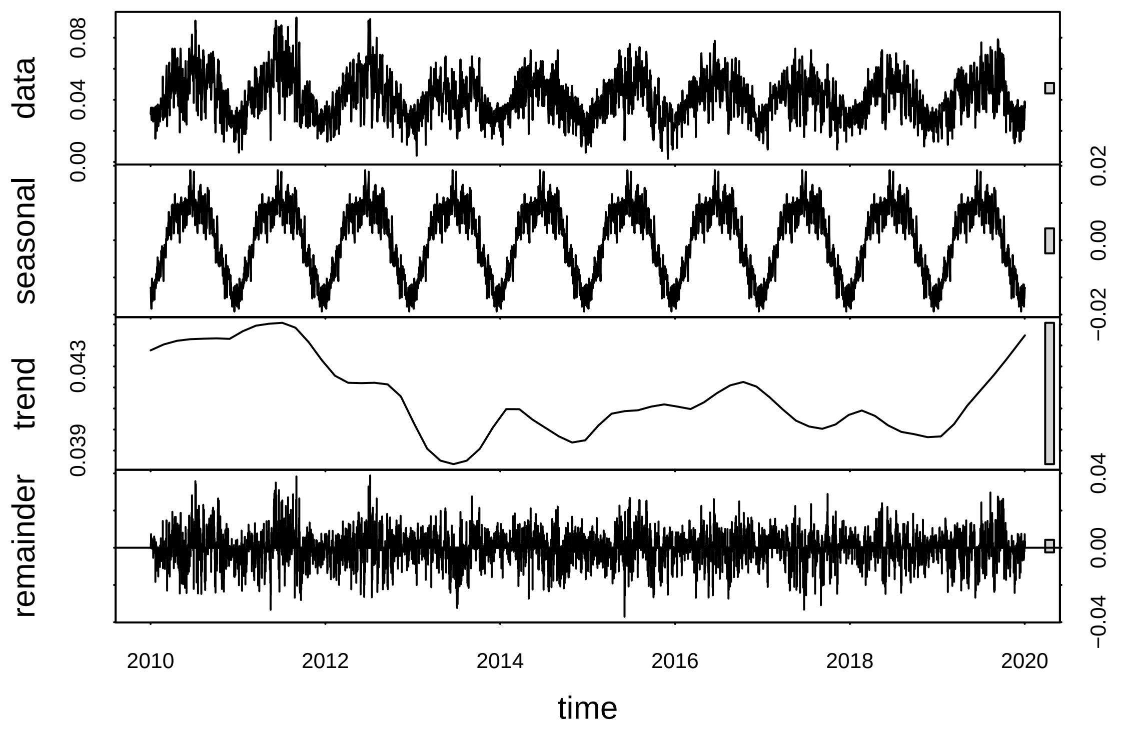
```
## `summarise()` has grouped output by 'Month', 'Year'. You can override using the
## `.groups` argument.
```

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.

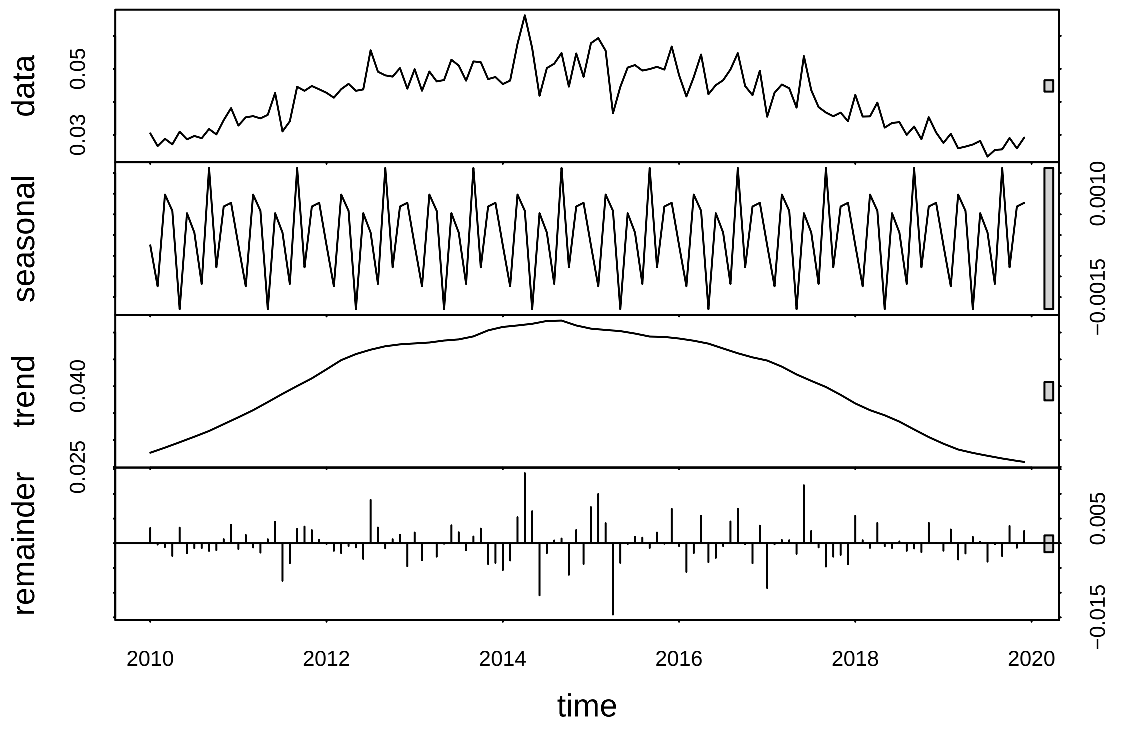
```
#10

#time series for daily ozone observations
GaringerOzone.daily.ts <- ts(GaringerOzone.complete$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010,01,01), frequency = 365)

#time series for monhtly average ozone values
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$MeanOzone,
                               start = c(2010,01,01), frequency = 12)
```

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

```
#11

#decomposing daily time series objects
daily.ts <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(daily.ts)
```



```
#decomposing monthly time series objects
monthly.ts <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(monthly.ts)
```



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

#loading Kendall package
library(Kendall)

#running seasonal Mann-Kendall test on monthly Ozone series
MonthlyOzone.MK <- SeasonalMannKendall(GaringerOzone.monthly.ts)
print(MonthlyOzone.MK)
```

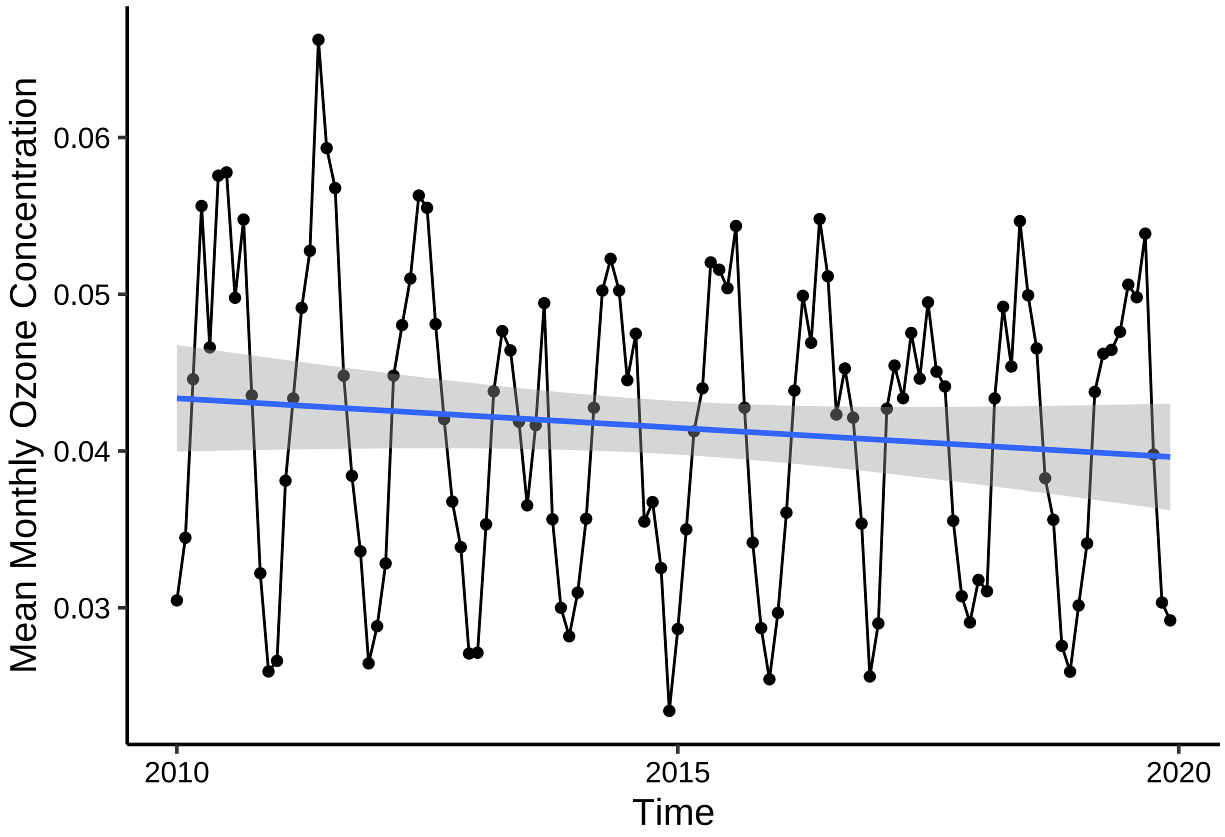
```
## tau = -0.1, 2-sided pvalue =0.16323
```

Answer: Seasonal Mann-Kendall is the most approrpiate because it is used to analyze data that follows a monotonic trend, meaning a gradual shift over time in the same direction. The monthly ozone data follows a consistent downwards trend. We also have not factored out seasonality, and this seasonal Mann-Kendall test accounts for data that is collected seasonally.

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

#creating plot showing mean monthly ozone concentrations over time
MonthlyOzone.plot <-
  ggplot(GaringerOzone.monthly, aes(x = Month_Year, y = MeanOzone)) +
  geom_point() +
  geom_line() +
  ylab("Mean Monthly Ozone Concentration") +
  xlab("Time") +
  geom_smooth(method = lm)
print(MonthlyOzone.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: The data suggests that monthly ozone concentrations have decreased from 2010 to 2020, as shown by the trend line in the graph that shows the mean monthly ozone concentrations over time. While the graph does show a negative trend, it is not a significant decrease because the Mann-Kendall test gave us a tau of -0.1 and a 2-sided pvalue of 0.16323, which indicates that the trend is statistically insignificant.

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

#subtracting seasonal component from GaringerOzone.monthly.ts
GaringerOzone.monthly.noseasonal <- as.data.frame(monthly.ts$time.series[,1:3])

GaringerOzone.monthly.noseasonal <- mutate(GaringerOzone.monthly.noseasonal,
        Observed = GaringerOzone.monthly$MeanOzone,
        Date = GaringerOzone.monthly$Month_Year)

#converting to new time series
GaringerOzone.monthly.ts2 <- ts(GaringerOzone.monthly.noseasonal$Observed,
                                start=c(2010,01,01), frequency=12)

#16

#running Mann-Kendall test on non-seasonal ozone monthly series
MonthlyOzone.MK2 <- Kendall::MannKendall(GaringerOzone.monthly.ts2)
print(MonthlyOzone.MK2)
```

```
## tau = -0.105, 2-sided pvalue =0.088483
```

```
#comparing with Seasonal Mann-Kendall results
summary(MonthlyOzone.MK2)
```

```
## Score =  -752 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.105, 2-sided pvalue =0.088483
```

```
summary(MonthlyOzone.MK)
```

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

Answer: Both the non-seasonal Mann-Kendall test and the Seasonal Mann-Kendall test showed a negative trend but were still statistically insignificant. The non-seasonal Mann-Kendall test had a tau of -0.105 and a 2-sided pvalue of 0.088483, while the Seasonal Mann-Kendall test had a tau of -0.1 and a 2-sided pvalue of 0.16323.