Assignment 7: Time Series Analysis

Yanxi Peng

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

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

getwd()
```

```
## [1] "C:/Users/16920/Documents/R/EDA-Fall2022"
```

```
setwd("C:/Users/16920/Documents/R/EDA-Fall2022")

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

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.

```
# import dataset
NC2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
    stringsAsFactors = TRUE)
NC2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
    stringsAsFactors = TRUE)
NC2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
    stringsAsFactors = TRUE)
NC2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
    stringsAsFactors = TRUE)
NC2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
    stringsAsFactors = TRUE)
NC2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
    stringsAsFactors = TRUE)
NC2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
    stringsAsFactors = TRUE)
NC2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
    stringsAsFactors = TRUE)
NC2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
    stringsAsFactors = TRUE)
NC2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
    stringsAsFactors = TRUE)
# combine dataframe
GaringerOzone <- rbind(NC2010, NC2011, NC2012, NC2013, NC2014, NC2015, NC2016, NC2017,
    NC2018, NC2019)
```

Wrangle

3. Set your date column as a date class.

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

```
## [1] "factor"
```

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

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

4. Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.

```
GaringerOzone <- GaringerOzone %>%
    select(Date, Daily.Max.8.hour.Ozone.Concentration, 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".

```
Days <- seq(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"), by = "day") %>%
    data.frame(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.

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

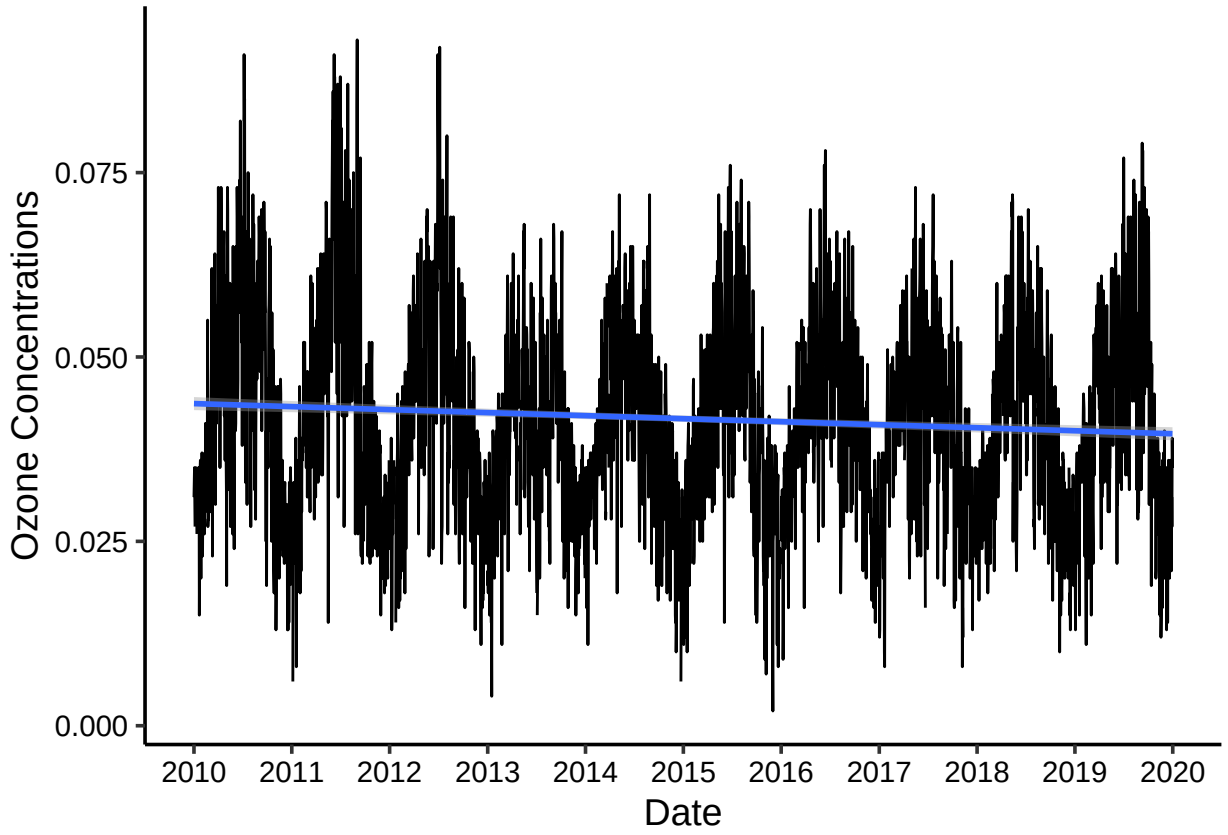
Visualize

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?

```
plot7 <- ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
    geom_line() + geom_smooth(method = "lm") + ylab("Ozone Concentrations") + scale_x_date(date_breaks =
    date_labels = "%Y")
plot7
```

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

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



Answer: The ozone concentration decrease slightly 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?

```
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

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

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

Answer: From the lecture Rmd example, the Piecewise constant will convert missing data to the measurement made nearest to that date (could be earlier or later), which will make our dataset unaccurate because the ozone concentration different by the time. Since from the plot I made in question 7, the relationship between date and concentration are linear not quadratic, so the linear interpolation is better than spline interpolation 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)

```
GaringerOzone.monthly <- GaringerOzone_clean %>%
    mutate(Month = month(Date)) %>%
    mutate(Year = year(Date)) %>%
    group_by(Year, Month) %>%
    summarise(meanOzone.Concentration = mean(Ozone.Concentration)) %>%
    mutate(newDate = my(paste0(Month, "-", Year)))
```

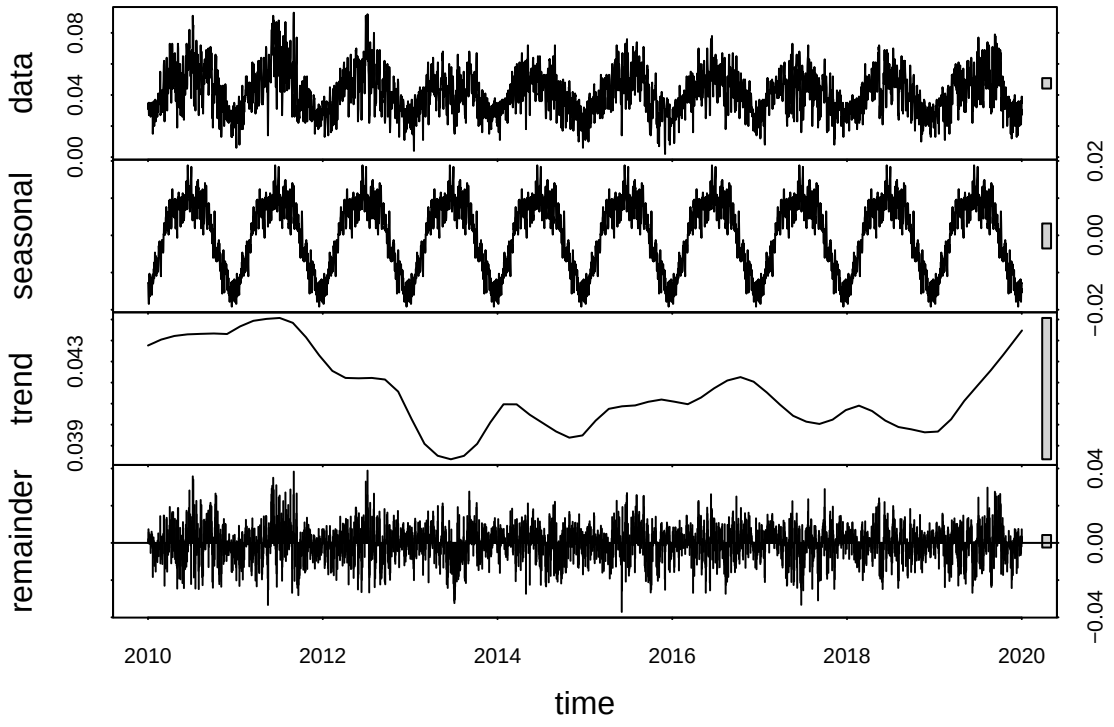
```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

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.

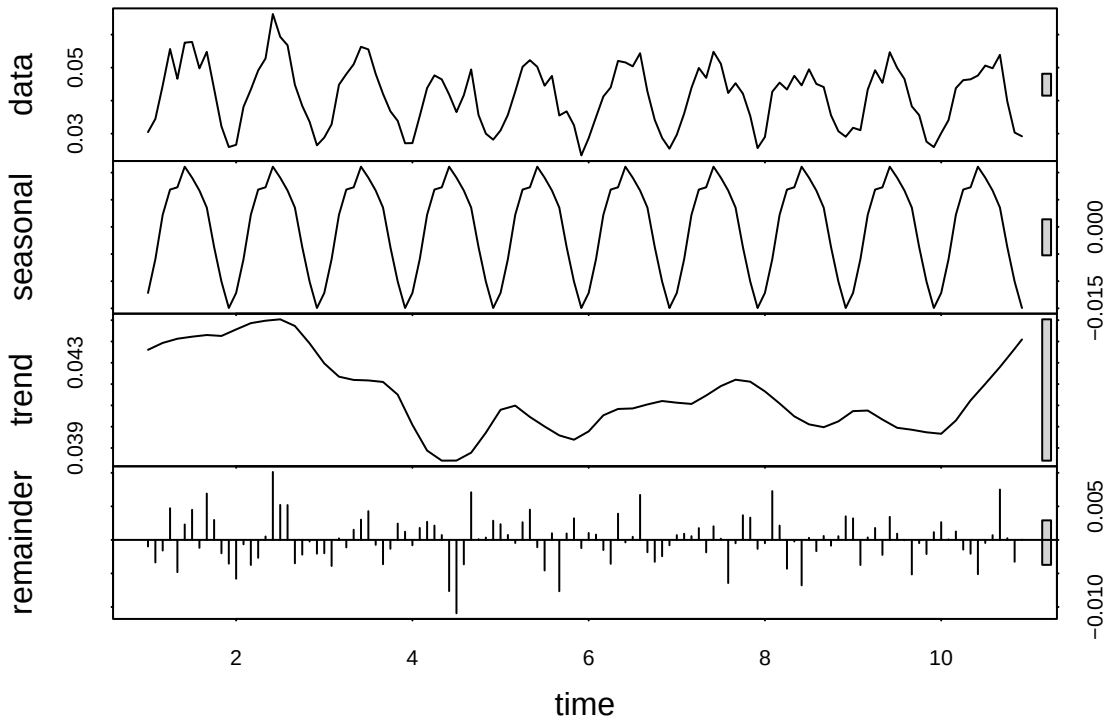
```
GaringerOzone.daily.ts <- ts(GaringerOzone_clean$Ozone.Concentration, start = c(2010,
    1, 1), frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$meanOzone.Concentration, frequency = 12)
```

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

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




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



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

```
library(Kendall)
```

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

```
monthly_data_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

summary(monthly_data_trend1)
```

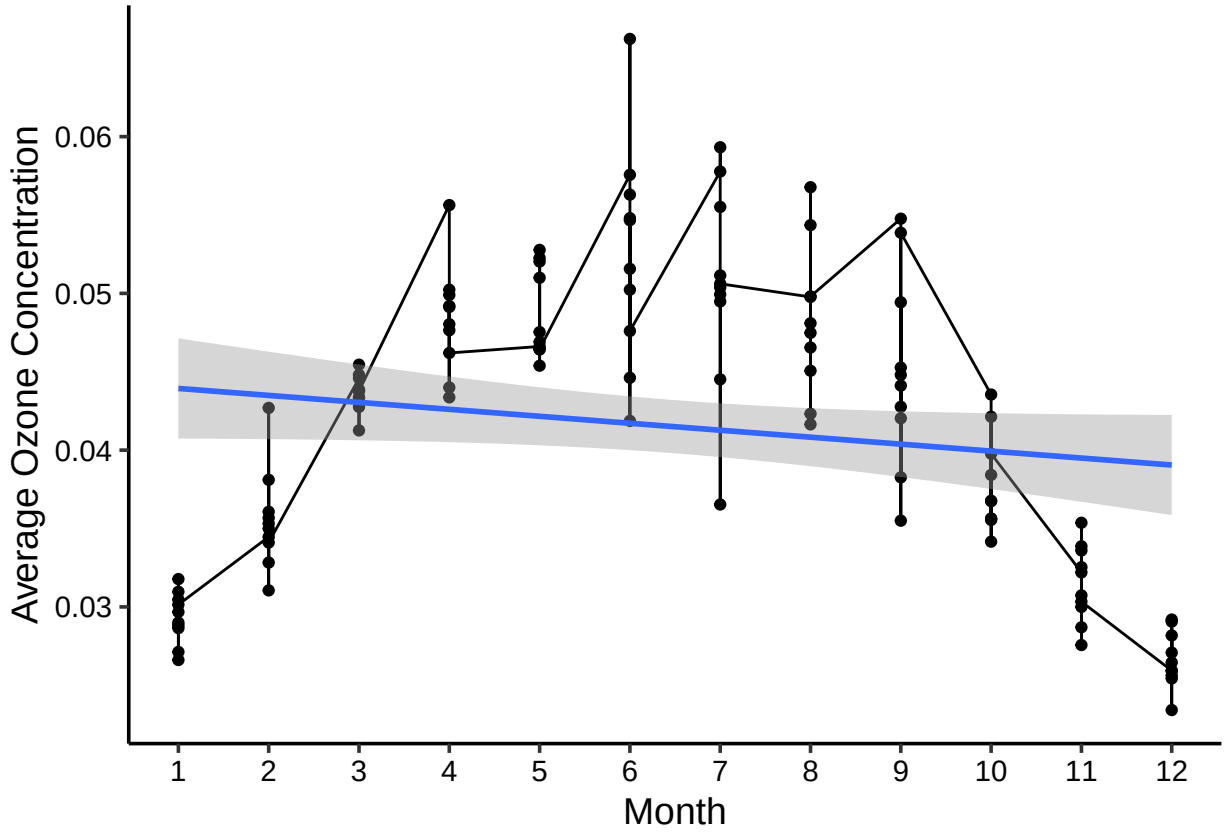
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The model is not linear so in this case linear regression can not be use. And our ozone concentration are different for each season, which is said to be seasonal. In this case, only seasonal Mann-Kendall can deal with seasonality.

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.

```
plot_13 <- ggplot(GaringerOzone.monthly, aes(x = Month, y = meanOzone.Concentration)) +
    geom_point() + geom_line() + ylab("Average Ozone Concentration") + scale_x_continuous(breaks = seq(
    12, 1)) + geom_smooth(method = lm)
print(plot_13)
```

```
## `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: From the graph, the ozone concentrations changed over the 2010s, overall, from the linear line, the average ozone concentration decrease with the month increase. The highest ozon concentration occure in June and July. Moreover, the P value from the monotonic trend analysis reject the null hypothesis. Which means that we have a trend between the average ozone concentration and the month.

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.

```
monthly_Components <- as.data.frame(GaringerOzone_monthlydecop$time.series[, 1:3])

monthly_Components <- mutate(monthly_Components, Observed = GaringerOzone.monthly$meanOzone.Concentrati
    Date = GaringerOzone.monthly$newDate)
```

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.

```
GaringerOzone.non_seasonal_monthly.ts <- ts(monthly_Components$Observed, frequency = 12)
monthly_data_trend2 <- Kendall::MannKendall(GaringerOzone.non_seasonal_monthly.ts)

summary(monthly_data_trend2)
```

```
## Score =  -424 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.0594, 2-sided pvalue =0.33732
```

Answer: The P value we got here do not reject the null hypothesis, the ozone concentration and the month do not have relationship, which have the opposite conclusion compare to the one obtained with the seansonal Mann Kendall.