Assignment 7: Time Series Analysis

Ariel O'Callaghan

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
#install.packages('formatR')
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=50), tidy=TRUE)

library(tidyverse)
library(lubridate)
#install.packages("trend")
library(trend)
#install.packages("zoo")
library(zoo)
#install.packages("Kendall")
library(Kendall)
#install.packages("tseries")
```

```
library(tseries)

getwd()
```

```
## [1] "/home/guest/R/EDA-Fall2022"
```

```
EPA2010<- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv", stringsAsFactors = TRU
EPA2011<- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv", stringsAsFactors = TRU
EPA2012<- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv", stringsAsFactors = TRU
EPA2013<- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv", stringsAsFactors = TRU
EPA2014<- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv", stringsAsFactors = TRU
EPA2015<- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv", stringsAsFactors = TRU
EPA2016<- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv", stringsAsFactors = TRU
EPA2017<- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv", stringsAsFactors = TRU
EPA2018<- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv", stringsAsFactors = TRU
EPA2019<- read.csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv", stringsAsFactors = TRU


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

GaringerOzone<- rbind(EPA2010,EPA2011, EPA2012, EPA2013, EPA2014, EPA2015, EPA2016, EPA2017, EPA2018, EP


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

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")
class(GaringerOzone$Date)
```

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

```
# 4

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

# 5 Days<- as.data.frame(seq.Date(from =
# as.Date('2010-01-01'), to =
# as.Date('2019-12-31'), by=1), colnames('Date'))

Days <- as.data.frame(seq.Date(from = as.Date("2010-01-01"),
    to = as.Date("2019-12-31"), by = 1))
colnames(Days) <- "Date"


# 6
GaringerOzoneF <- left_join(Days, GaringerOzone_Wrangled,
    by = c("Date"))
```

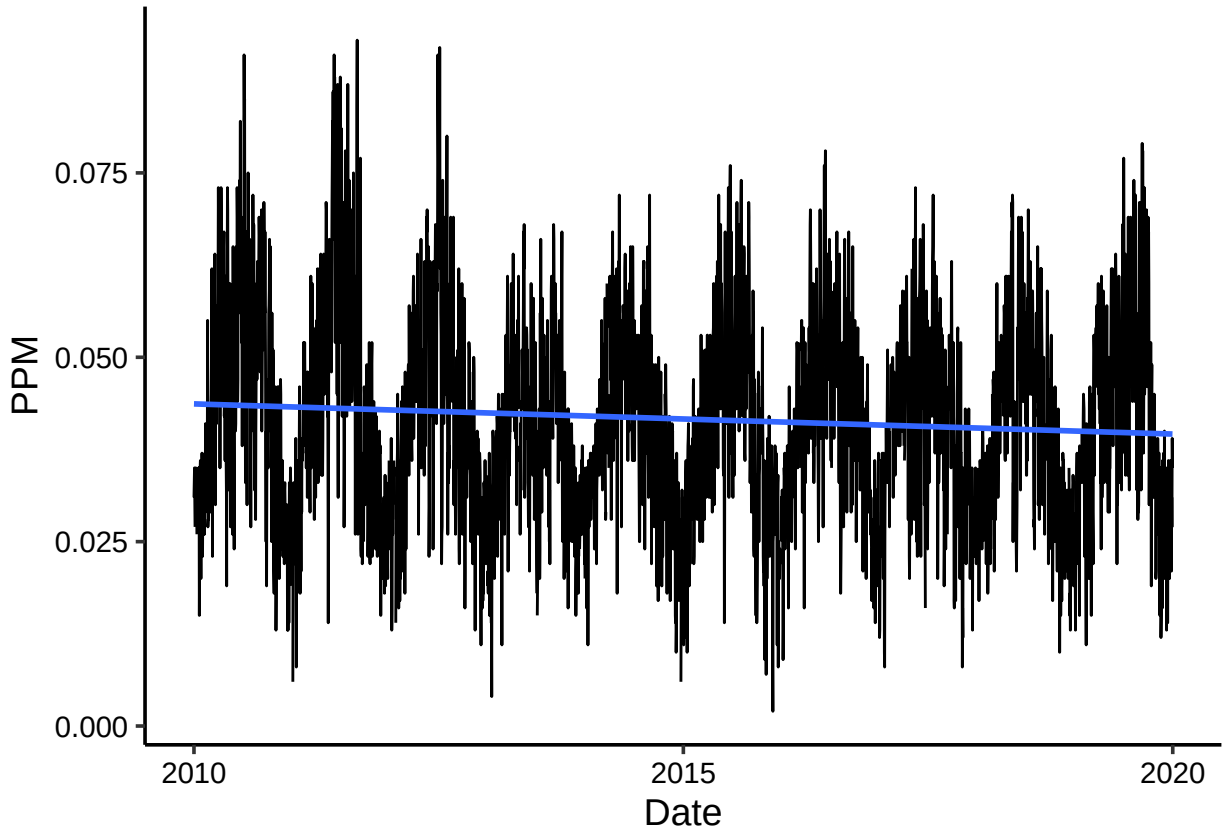
Visualize

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?

```
# 7
ppm.aqi.plot <- ggplot(GaringerOzoneF, aes(y = Daily.Max.8.hour.Ozone.Concentration,
    x = Date), size = 0.25, color = "#c13d75ff") +
    geom_line() + geom_smooth(method = lm, se = FALSE) +
    labs(x = "Date", y = "PPM")
print(ppm.aqi.plot)
```

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

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



Answer:Ozone concentration is decreasing over time. There is also strong seasonal variability but the linear regression decreases.

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 summary(GaringerOzoneF)

ozone.fill <- GaringerOzoneF %>%
    mutate(Daily.ozone.interp. = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
# print(ozone.fill)
```

Answer: The piecewise constant or nearest neighbor approach assmumes equal measurement to the nearest date. The spline is a linear interploation where the quaratic function is sued rather then interolating from a straight line. The linear interploation we used is a connect the dots apprach with interplates between the nearest datapoints which fills in the gaps in our time series data.

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 <- ozone.fill %>%
    mutate(month = month(Date), year = year(Date)) %>%
    mutate(Month_year = my(paste(month, "-", year))) %>%
    select(Month_year, Daily.ozone.interp.) %>%
    group_by(Month_year) %>%
    summarise(Mean_PPM = mean(Daily.ozone.interp.))
```

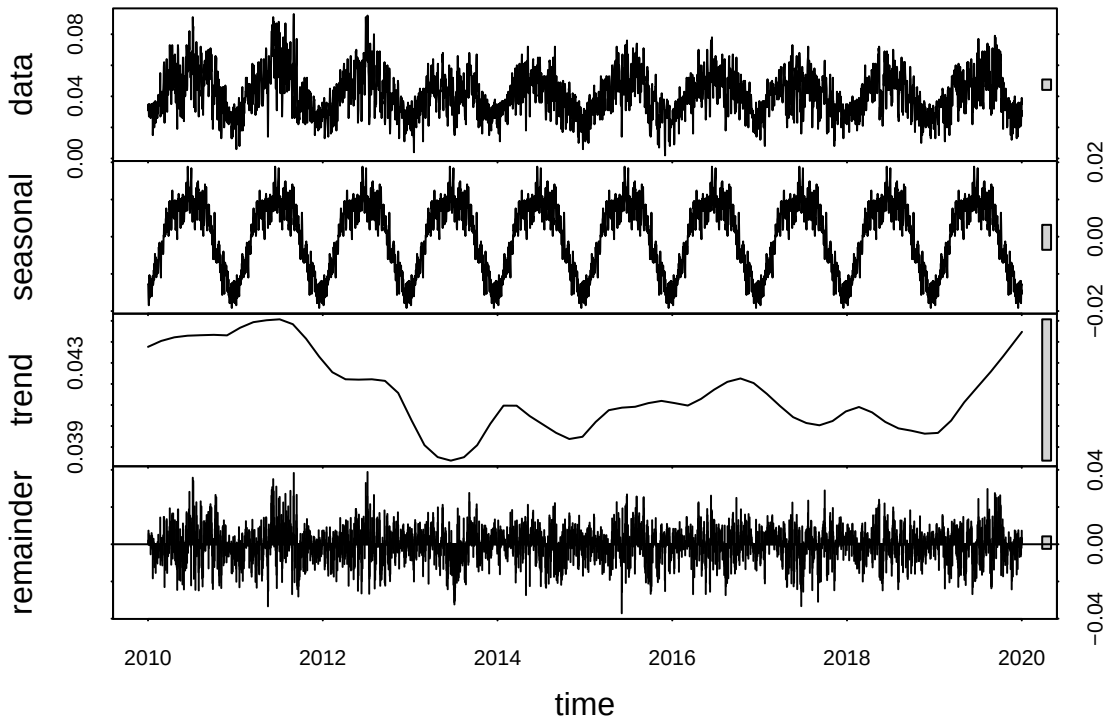
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

```
# 10
GaringerOzone.daily.ts <- ts(ozone.fill$Daily.ozone.interp.,
    start = c(2010, 1), frequency = 365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_PPM,
    start = c(2010, 1), frequency = 12)
```

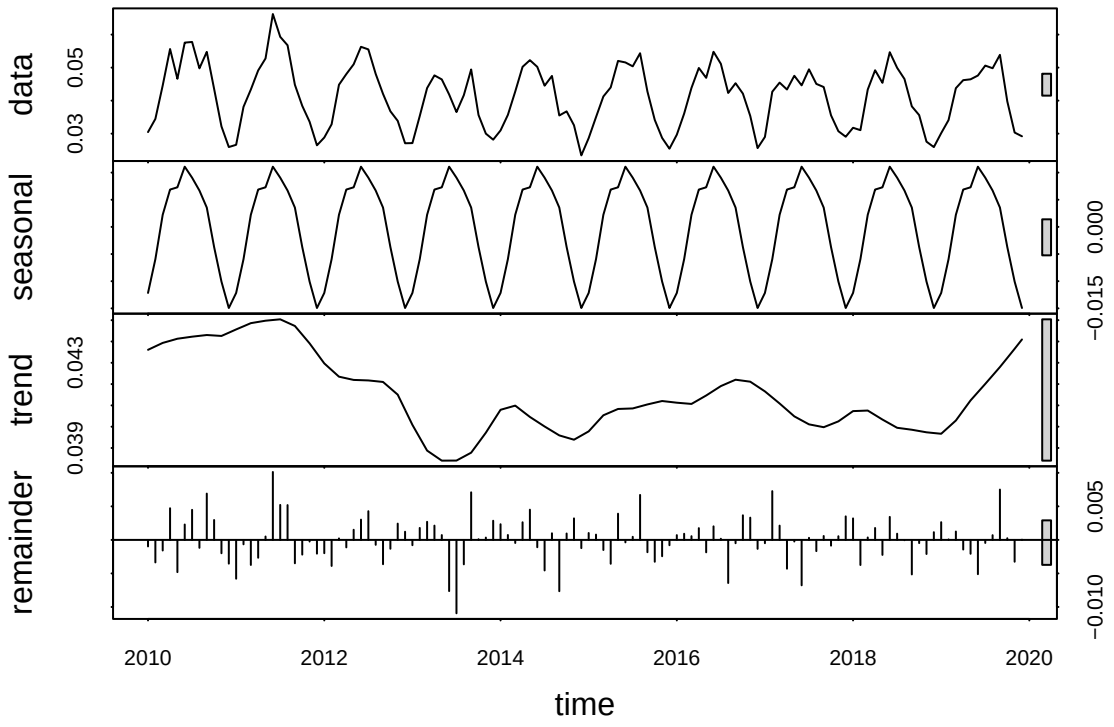
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

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

plot(GaringerOzone.daily.ts.decomposed)
```




```
plot(GaringerOzone.monthly.ts.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
monthly.ozone.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# summary(monthly.ozone.trend)
```

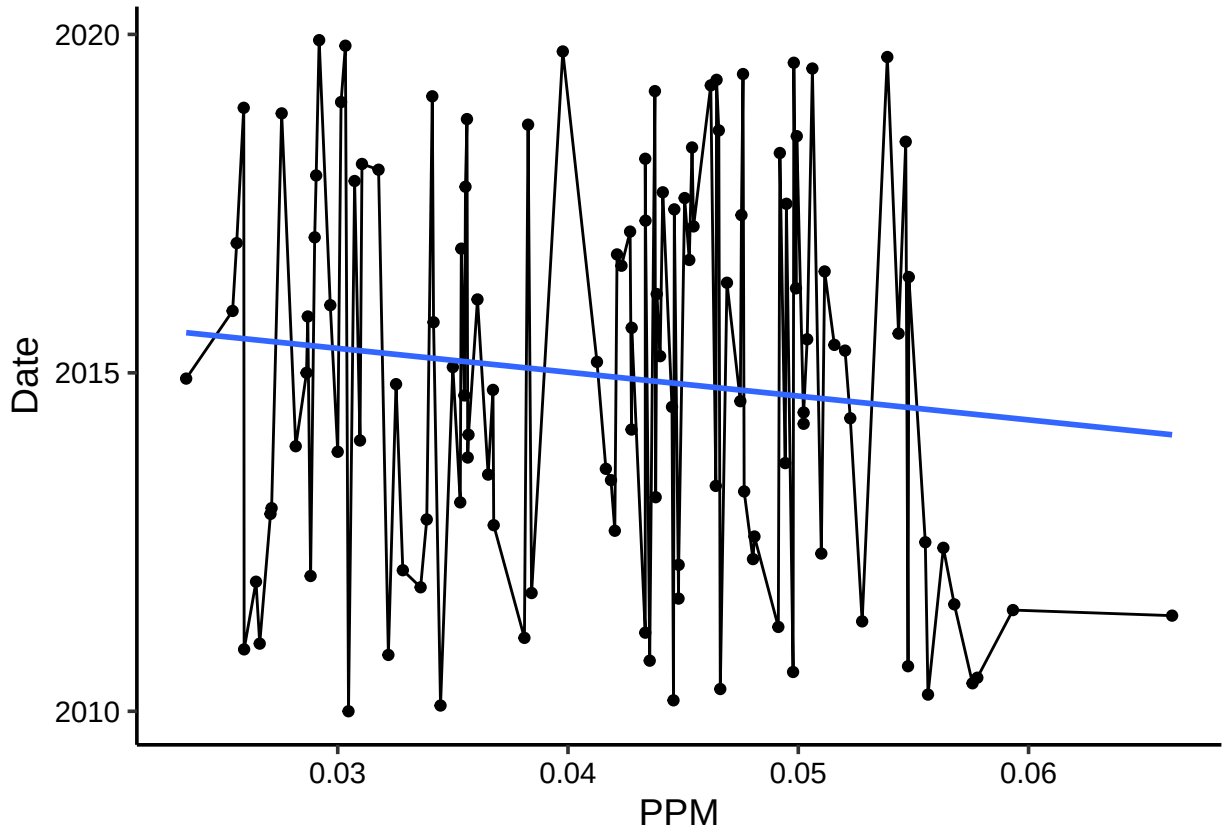
Answer: You can see from the graphs above that there is seasonability in the ozone data.

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.

```
# 13

ozone.month.plot <- ggplot(GaringerOzone.monthly, aes(x = Mean_PPM,
    y = Month_year)) + geom_point() + geom_line() +
    geom_smooth(method = lm, se = FALSE) + labs(x = "PPM",
    y = "Date")
print(ozone.month.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: There is a negative trend as ozone concentrations decreased.. The tau from the Mann-Kendall test shows -0.14. The p value is less then 0.5 so the correlation between ozone and time is significant. The score is -77 which also shows a negative trend.

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

seasonal.removed <- (GaringerOzone.monthly.ts - GaringerOzone.monthly.ts.decomposed$time.series[,
    1])

# 16

monthly.ozone.trend.noseasonal <- Kendall::MannKendall(seasonal.removed)
summary(monthly.ozone.trend.noseasonal)
```

```
## Score =  -1179 , Var(Score) = 194365.7
```

```
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer: The score of the MannKenall with the seasonal removed shows a score of -1179 whie the score of the seasonal kendall has a score of -77. The two sided p value of the seasonal removed data is 0.007 much lower then p value of the seasonal data of 0.046. The tau for seasonal removed is -0.165 and for seaosonal -0.143. Both tests show that ozone is statistically significant with time even though the statstics vary between the two tests.