Assignment 8: Time Series Analysis

Mara Michel

Fall 2023

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on generalized linear models.

Directions

1. Rename this file `<FirstLast>_A08_TimeSeries.Rmd` (replacing `<FirstLast>` with your first and last name).
2. Change "Student Name" on line 3 (above) with your name.
3. Work through the steps, **creating code and output** that fulfill each instruction.
4. Be sure to **answer the questions** in this assignment document.
5. When you have completed the assignment, **Knit** the text and code into a single PDF file.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

```
#Check working directory
getwd()
```

```
## [1] "C:/Users/marga/OneDrive/Documents/EDE_Fall2023"
```

```
#Install packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(lubridate)
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(trend)


# Set ggplot theme
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.

```r
#2

GaringerOzone_10 <- read.csv(file="./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                             header=TRUE,
                             stringsAsFactors = TRUE)
GaringerOzone_11<- read.csv(file="./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                            header=TRUE,
                            stringsAsFactors = TRUE)
GaringerOzone_12 <- read.csv(file="./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                             header=TRUE,
                             stringsAsFactors = TRUE)
GaringerOzone_13 <- read.csv(file="./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                             header=TRUE,
                             stringsAsFactors = TRUE)
GaringerOzone_14 <- read.csv(file="./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                             header=TRUE,
                             stringsAsFactors = TRUE)
GaringerOzone_15 <- read.csv(file="./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                             header=TRUE,
                             stringsAsFactors = TRUE)
GaringerOzone_16 <- read.csv(file="./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                             header=TRUE,
                             stringsAsFactors = TRUE)
GaringerOzone_17 <- read.csv(file="./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                             header=TRUE,
                             stringsAsFactors = TRUE)
GaringerOzone_18 <- read.csv(file="./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                             header=TRUE,
                             stringsAsFactors = TRUE)
GaringerOzone_19 <- read.csv(file="./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                             header=TRUE,
                             stringsAsFactors = TRUE)

GaringerOzone <- rbind(GaringerOzone_10,
                       GaringerOzone_11,
                       GaringerOzone_12,
                       GaringerOzone_13,
                       GaringerOzone_14,
                       GaringerOzone_15,
                       GaringerOzone_16,
                       GaringerOzone_17,
                       GaringerOzone_18,
                       GaringerOzone_19)

dim(GaringerOzone)
```

```
## [1] 3589   20
```

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

# 5
Days <-as.data.frame(x= seq(as.Date("2010/1/1"),
                            as.Date("2019/12/31"),
                            by = "day"))
colnames(Days)[1] ="Date"
```

```
# 6
GaringerOzone<-left_join(x=Days,
                         y=GaringerOzone_filter,
                         by='Date')
dim(GaringerOzone)
```

```
## [1] 3652    3
```

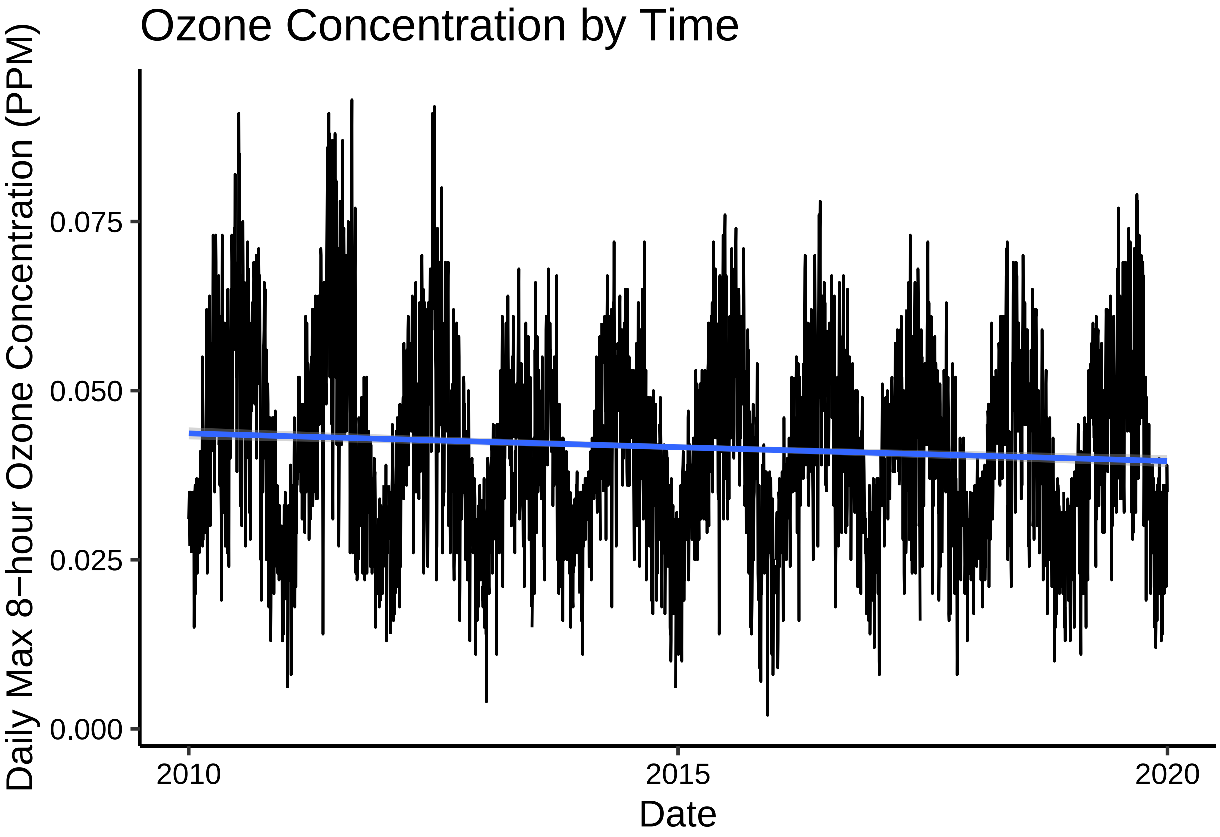
Visualize

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
GaringerOzone_Plot <- ggplot(GaringerOzone,aes(x=Date,
                y=Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  geom_smooth(method=lm)+
  labs(title="Ozone Concentration by Time",y="Daily Max 8-hour Ozone Concentration (PPM)")

GaringerOzone_Plot
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

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



Answer:The trend line of the plot suggests a slightly negative linear trend 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

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

Answer: We use a linear interpolation in this dataset to fit values between the previous and following values along the same slope to maintain the trend. Piecewise constant would have selected the "nearest neighbor" and filled in the Ozone concentration value with the same value as the closest date observation. The Spline interpolation would potentially produce results that are outside of the range of the values before and after it.

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
#Group by year and month and then find average of Ozone
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Month = month(Date),Year = year(Date)) %>%
  group_by(Year,Month)%>%
  summarize(mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

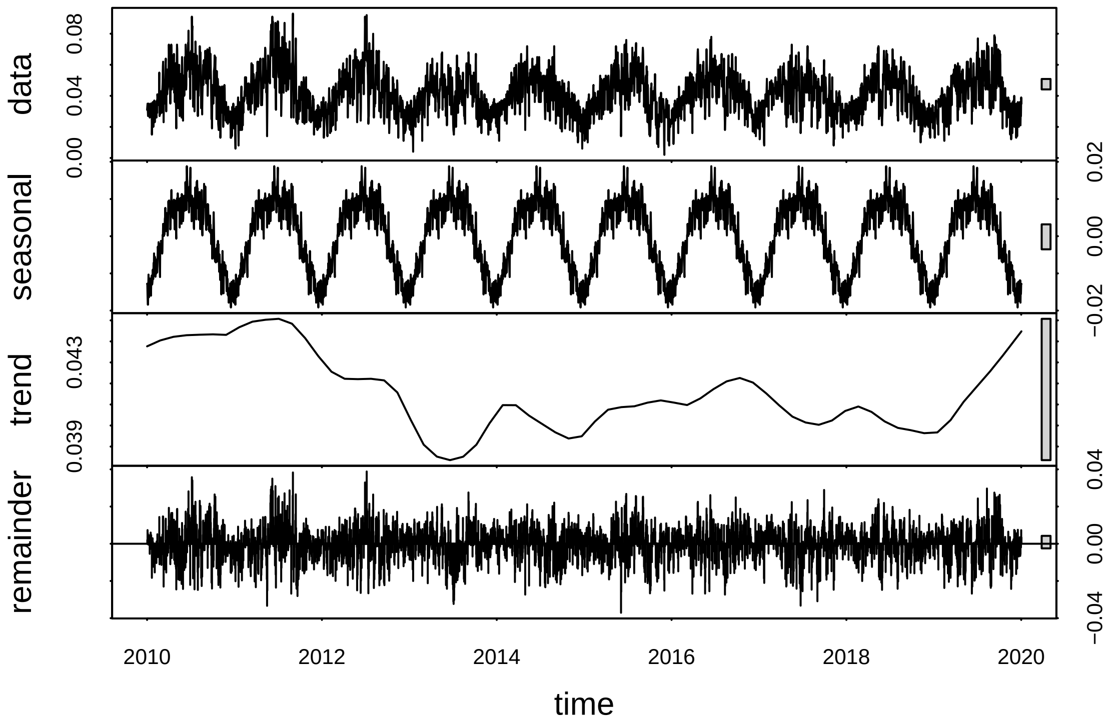
```
#Rename columns for ease of use in future formulas

colnames(GaringerOzone.monthly)[3] ="Mean_Ozone"

#Create new Date column
GaringerOzone.monthly$Date <- lubridate::ymd(paste(GaringerOzone.monthly$Year,
                                                   GaringerOzone.monthly$Month,
                                                   "1"))
```

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.

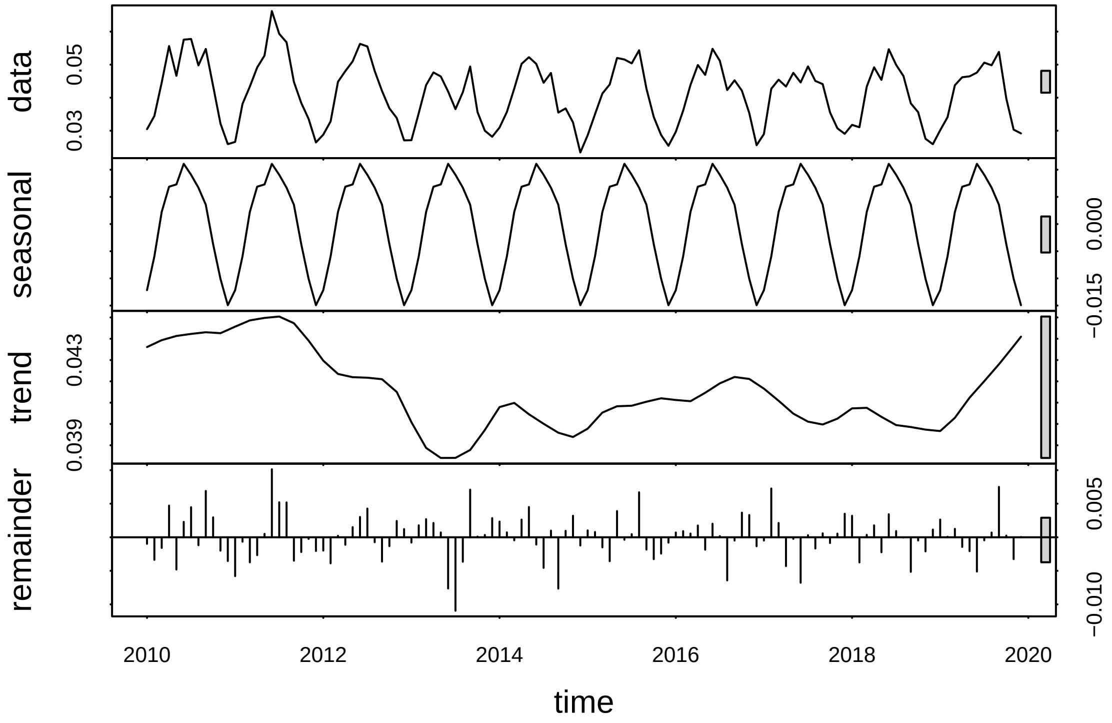
```
#10
GaringerOzone.daily.ts <-ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                            start= c(2010,1),
                            frequency=365)

GaringerOzone.monthly.ts <-ts(GaringerOzone.monthly$Mean_Ozone,
                              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.Decomposed <-stl(GaringerOzone.daily.ts,s.window = "periodic")
plot(GaringerOzone.daily.Decomposed)
```



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

# Run SMK test
GaringerOzone.SMKTrend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

GaringerOzone.SMKTrend
```

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

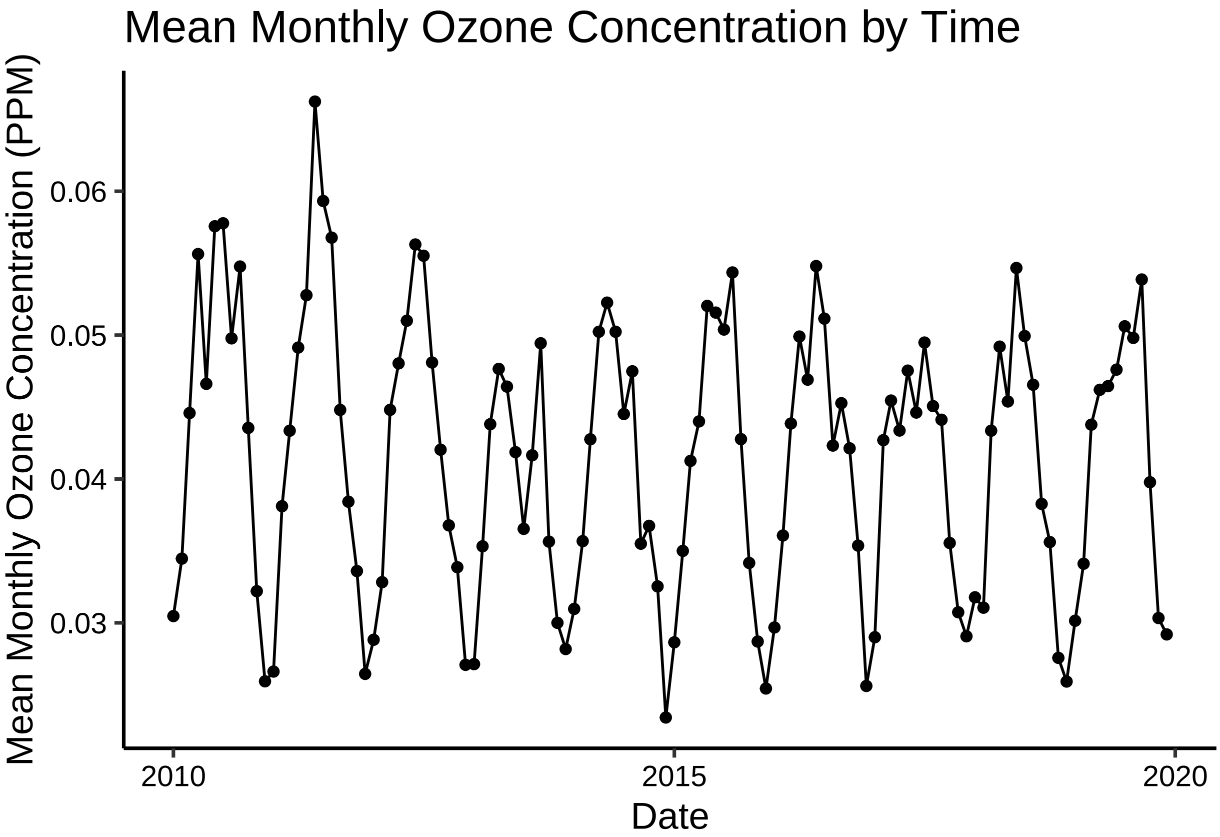
```
summary(GaringerOzone.SMKTrend)
```

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

Answer: The seasonal Mann-Kendall is most appropriate because the Ozone levels follow a consistent seasonal trend year to year. Not all the trends are able to handle this 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.

```
# 13
GaringerOzone.monthly.plot <-ggplot(GaringerOzone.monthly,aes(x=Date,y=Mean_Ozone))+
  geom_point()+
  geom_line()+
  labs(title="Mean Monthly Ozone Concentration by Time",y="Mean Monthly Ozone Concentration (PPM)")

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: The graph demonstrates the seasonality of monthly mean ozone concentrations between the years 2010-2019. The p-value, 0.046724, is less than 0.05 which leads us to reject the null-hypothesis and accept the alternative hypothesis that there is likely a relationship in the concentration changes over time. The tau value (-0.143) is negative and indicates that the Ozone concentrations decreased over time.

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
#Split GaringerOzone.monthly into components
GaringerOzone.monthly_Components <- as.data.frame(GaringerOzone.monthly.Decomposed$time.series[,1:3])

#Subtract seasonal component from observed values
GaringerOzone.monthly_deaseasoned <- GaringerOzone.monthly.ts-GaringerOzone.monthly_Components$seasonal

#16
#Run Mann Kendall test
GaringerOzone.Deaseasoned.MKTrend <- Kendall::MannKendall(GaringerOzone.monthly_deaseasoned)

#Results of deaseasoned
GaringerOzone.Deaseasoned.MKTrend
```

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

```
summary(GaringerOzone.Deaseasoned.MKTrend)
```

```
## Score =  -1179 , Var(Score) = 194365.7
## denominator =  7139.5
## tau = -0.165, 2-sided pvalue =0.0075402
```

```
#Previous results for comparison
GaringerOzone.SMKTrend
```

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

```
summary(GaringerOzone.SMKTrend)
```

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

> Answer: The p-value for the non-seasonal data is 0.0075402 which is significantly smaller than the original P-value of 0.046724. The tau value for the non-seasonal data is -0.165, which is smaller than the original tau value of -0.143. After removing the seasonality, the overall correlation and negative trend becomes more pronounced.