# Assignment 8: Time Series Analysis

Student Name

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

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

here()
```

```
## [1] "C:/Users/purec/Documents/Duke/Fall_2023/EDA/EDE_Fall2023"
```

```
mytheme <- theme_gray()+
  theme(plot.title = element_text(size = 16, hjust= 0),
        axis.title = element_text(size = 13),
        legend.position = 'right')

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.

```
#1
O3_2010 <- read.csv(file = here("Data","Raw","Ozone_TimeSeries",
                          "EPAair_O3_GaringerNC2010_raw.csv"),
                    stringsAsFactors = T)
O3_2011 <- read.csv(file = here("Data","Raw","Ozone_TimeSeries",
                          "EPAair_O3_GaringerNC2011_raw.csv"),
                    stringsAsFactors = T)
O3_2012 <- read.csv(file = here("Data","Raw","Ozone_TimeSeries",
                          "EPAair_O3_GaringerNC2012_raw.csv"),
                    stringsAsFactors = T)
O3_2013 <- read.csv(file = here("Data","Raw","Ozone_TimeSeries",
                          "EPAair_O3_GaringerNC2013_raw.csv"),
                    stringsAsFactors = T)
O3_2014 <- read.csv(file = here("Data","Raw","Ozone_TimeSeries",
                          "EPAair_O3_GaringerNC2014_raw.csv"),
                    stringsAsFactors = T)
O3_2015 <- read.csv(file = here("Data","Raw","Ozone_TimeSeries",
                          "EPAair_O3_GaringerNC2015_raw.csv"),
                    stringsAsFactors = T)
O3_2016 <- read.csv(file = here("Data","Raw","Ozone_TimeSeries",
                          "EPAair_O3_GaringerNC2016_raw.csv"),
                    stringsAsFactors = T)
O3_2017 <- read.csv(file = here("Data","Raw","Ozone_TimeSeries",
                          "EPAair_O3_GaringerNC2017_raw.csv"),
                    stringsAsFactors = T)
O3_2018 <- read.csv(file = here("Data","Raw","Ozone_TimeSeries",
                          "EPAair_O3_GaringerNC2018_raw.csv"),
                    stringsAsFactors = T)
O3_2019 <- read.csv(file = here("Data","Raw","Ozone_TimeSeries",
                          "EPAair_O3_GaringerNC2019_raw.csv"),
                    stringsAsFactors = T)

#renaming the columns that are different from the rest of the data sets
# so that they can be joined
colnames(O3_2010)[3] <- "Site.ID"
colnames(O3_2010)[5] <- "Daily.Max.8.hour.Ozone.Concentration"
colnames(O3_2010)[8] <- "Site.Name"

O3_all <- rbind(O3_2010, O3_2011, O3_2012, O3_2013, O3_2014, O3_2015, O3_2016,
                O3_2017, O3_2018, O3_2019)
```

## 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.

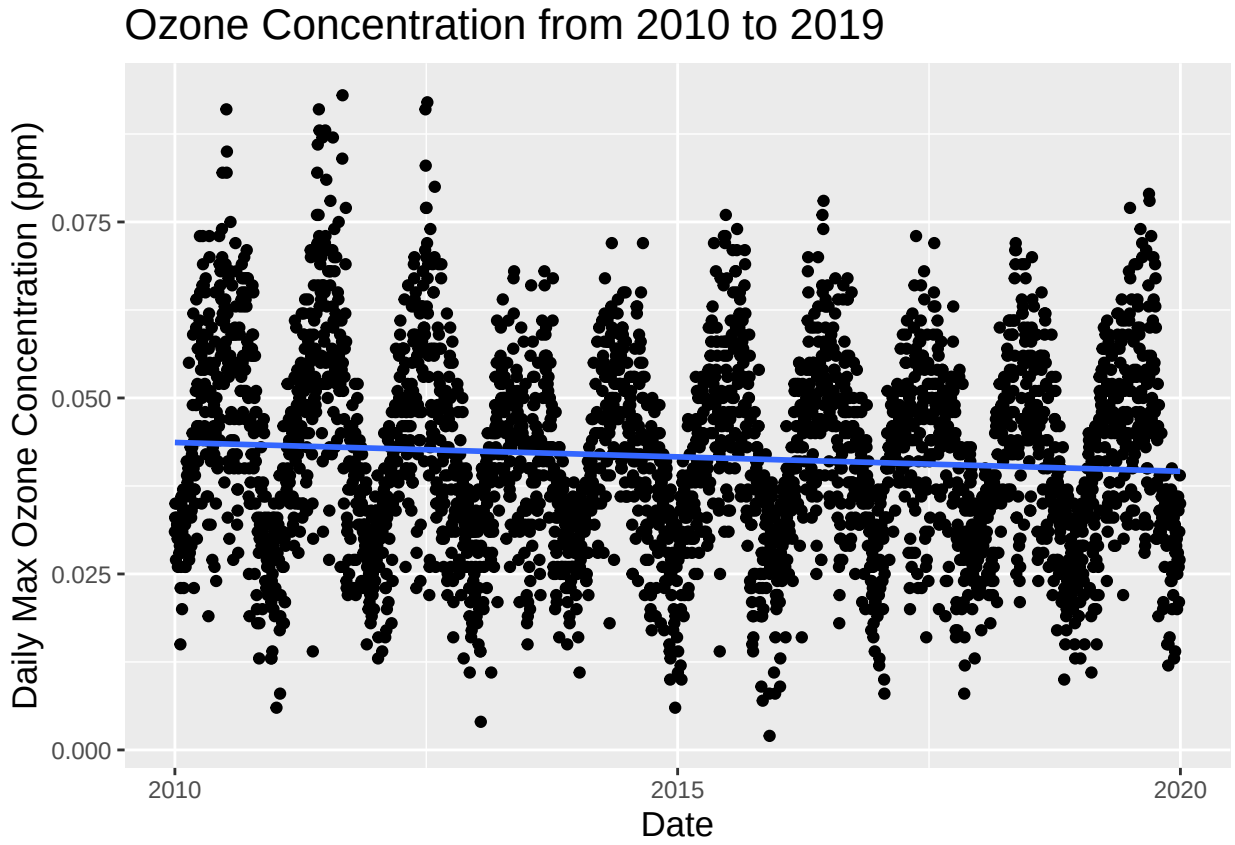
```
# 3
O3_all$Date <- mdy(O3_all$Date)

# 4
O3_summary <- O3_all %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
startdate <- ymd('2010-01-01')
enddate <- ymd('2019-12-31')

Days <- as.data.frame(seq.Date(from = startdate, to = enddate, by = 1))
colnames(Days) <- 'Date'

# 6
GaringerOzone <- left_join(Days, O3_summary)
```

## 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

O3_time <- ggplot(GaringerOzone, aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration))+
  geom_point()+
  geom_smooth(method='lm', se=F)+
  labs(y = 'Daily Max Ozone Concentration (ppm)',
       title = 'Ozone Concentration from 2010 to 2019')

O3_time
```

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




Answer: This plot does not suggest an overall trend of ozone increasing or decreasing overtime, but it does evidence that there is a seasonal or other natural fluctuation of ozone concentration.

## 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(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 <- GaringerOzone %>%
  mutate( MaxOzoneConc.clean = zoo::na.approx(
    Daily.Max.8.hour.Ozone.Concentration) )

summary(GaringerOzone$MaxOzoneConc.clean)
```

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

Answer: Piecewise constant and spline interpolation are not good choices for this data frame. Spline follows a quadratic trend, which we don't want here, and a piece wise constant would not be appropriate either. The linear interpolation keeps the same statistics for the data set and essentially connects the observations while following the trend here, which increases and decreases in a linear mode seasonally.

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 <- GaringerOzone %>%
  mutate(DateMonth = month(Date),
         DateYear = year(Date)) %>%
  group_by(DateMonth, DateYear) %>%
  summarise(mean_OzoneConc = mean(MaxOzoneConc.clean)) %>%
  arrange(DateYear)
```

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

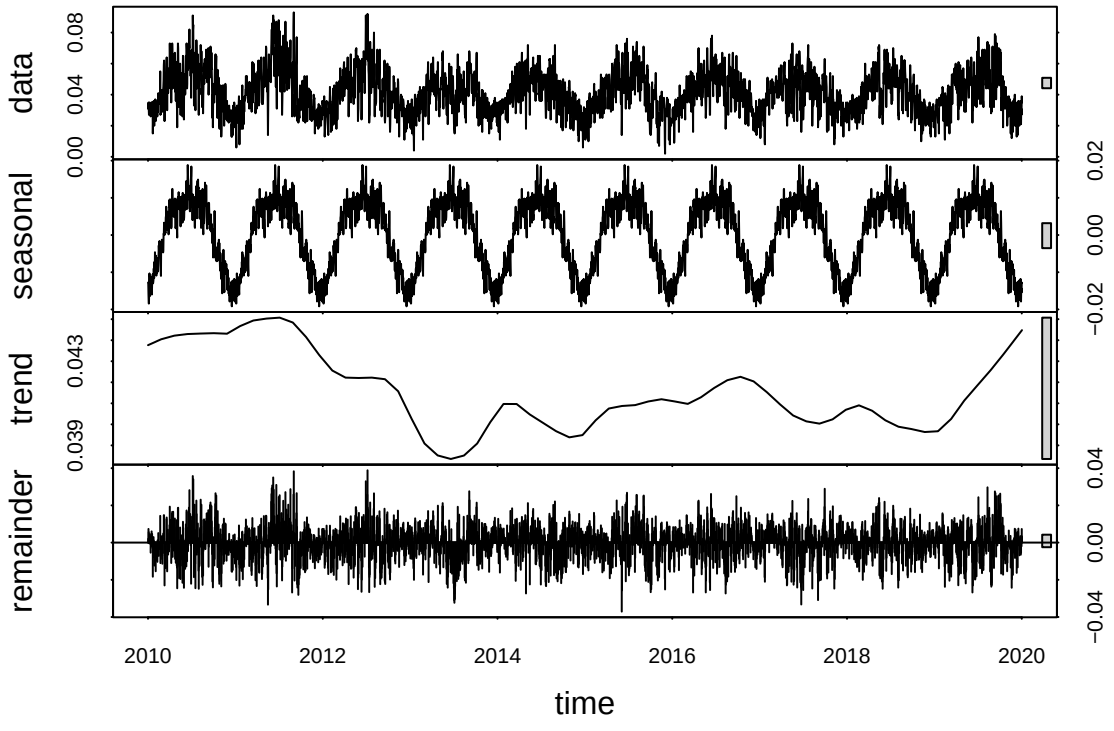
```
MonthlyDate <- seq.Date(from = startdate, to = enddate, by = 'month')
GaringerOzone.monthly$Date <- MonthlyDate
```

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.

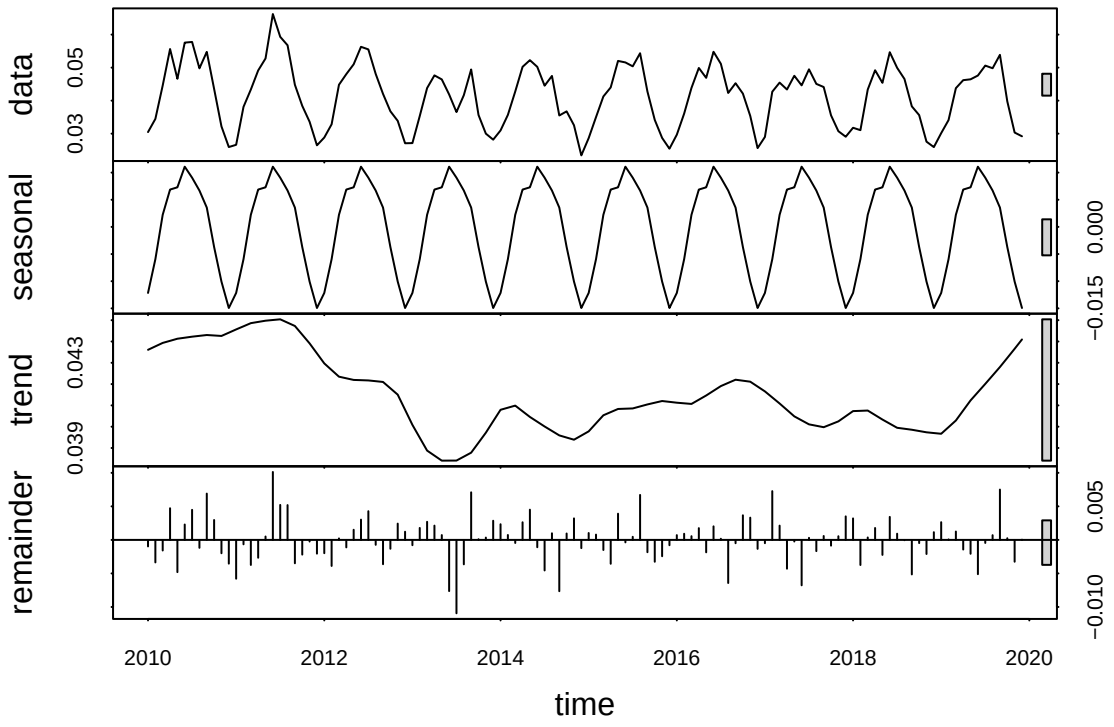
```r
#10
f_month <- month(first(GaringerOzone$Date))
f_year <- year(first(GaringerOzone$Date))
f_day <- day(first(GaringerOzone$Date))

GaringerOzone.daily.ts <- ts(GaringerOzone$MaxOzoneConc.clean,
                   start=c(f_year,f_month, f_day),
                   frequency=365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_OzoneConc,
                               start=c(f_year,f_month), frequency=12)
```

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

```r
#11
Garinger.daily_Decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(Garinger.daily_Decomposed)
```




```r
Garinger.monthly_Decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(Garinger.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
SMK_GaringerOzone <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

SMK_GaringerOzone
```

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

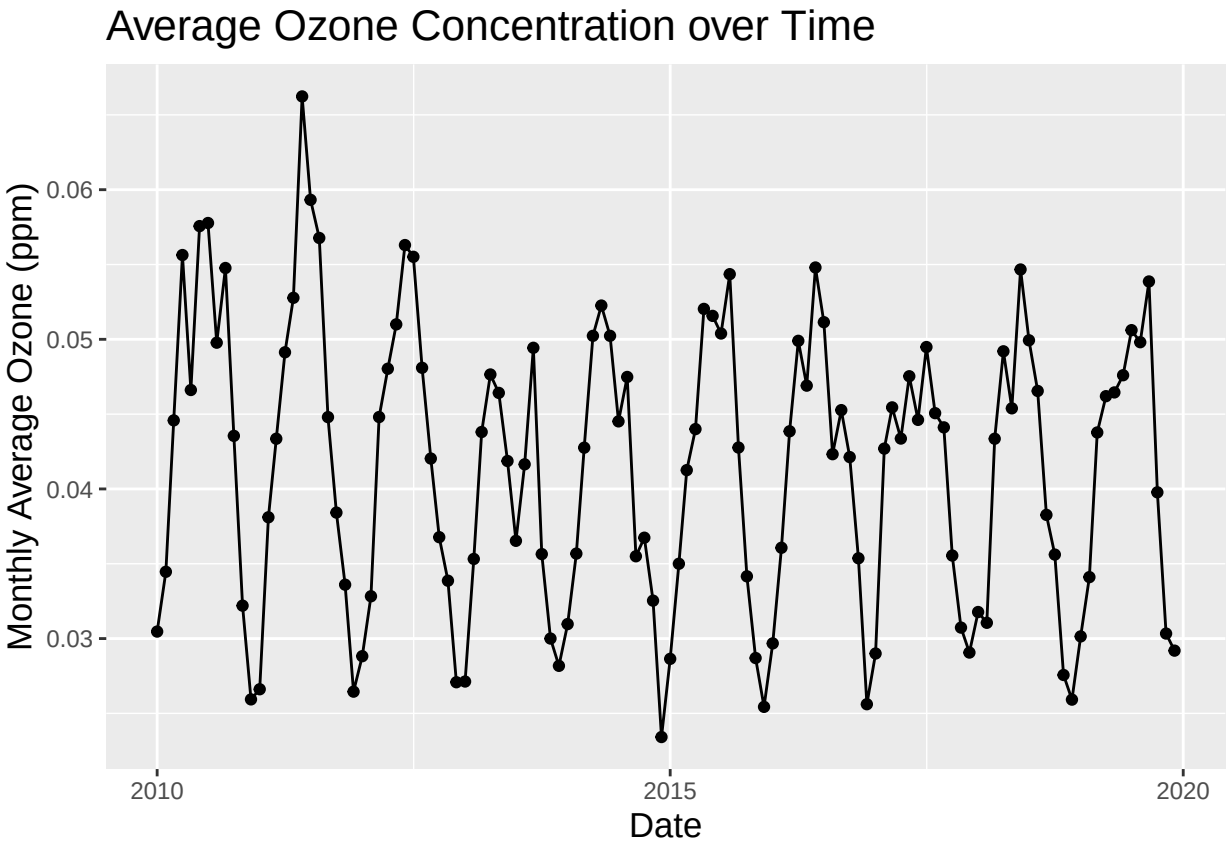
```
summary(SMK_GaringerOzone)
```

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

Answer: This data shows clear seasonality in the repeated increase and decrease in concentration throughout the year, and the Seasonal Mann-Kendall test is the only test that accounts for 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.

```r
# 13

Monthly_Ozone <- ggplot(GaringerOzone.monthly, aes(x=Date, y=mean_OzoneConc))+
  geom_point()+
  geom_line()+
  labs(y = "Monthly Average Ozone (ppm)",
       title = "Average Ozone Concentration over Time")


Monthly_Ozone
```




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 Seasonal Mann-Kendall test performed indicates that the ozone concentration has changed since 2010 (p-value = 0.0467). We reject the null hypothesis that the mean concentration remains stationary.

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.

```r
#15
Garinger.monthly_noSeason <- Garinger.monthly_Decomposed$time.series[, c(2,3)]


#16

SMK_Garinger_noSeason <- Kendall::SeasonalMannKendall(Garinger.monthly_noSeason)

SMK_Garinger_noSeason
```

```
## tau = -0.582, 2-sided pvalue =< 2.22e-16
```

```r
summary(SMK_Garinger_noSeason)
```

```
## Score =  -1326 , Var(Score) = 11400
## denominator =  2280
## tau = -0.582, 2-sided pvalue =< 2.22e-16
```

Answer: The result of the SMK after removing the seasonal component shows a much stronger conclusion that the ozone concentration has changed since 2010 (p-value =< 2.22e-16).