# Assignment 8: Time Series Analysis

Nadia Barbo

Spring 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

```
#1
#checking working directory
getwd()
```

```
## [1] "C:/Users/nadia/Documents/Duke_/EDA-Spring2023"
```

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

#setting ggplot theme
mytheme <- theme_classic(base_size = 14) + #building my theme and making base font size
↪  14
  theme(axis.text = element_text(color = "black", #making axis text black
                                 family = 'sans'), #making axis text 'sans' font
        plot.title = element_text(color = "black", #making plot title black
                              family = "sans", #making plot title 'sans' font
                              hjust = 0.5), #making plot title centered
        panel.background = element_rect(fill = 'whitesmoke'), #making panel background
        ↪  off white
        plot.background = element_rect(fill = "whitesmoke"), #making plot background off
        ↪  white
        legend.background = element_rect(fill = 'whitesmoke'), #making legend background
        ↪  off white
        legend.position = "top") #print legend on top of plot
theme_set(mytheme) #setting my theme to be default
```

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.

```
# 2
import_all = list.files("./Data/Raw/Ozone_TimeSeries", pattern = "*.csv")  #importing all
↪  csv files from designated folder
GaringerOzone <- do.call(rbind, lapply(paste0("./Data/Raw/Ozone_TimeSeries/",
↪  import_all),
    sep = ",", stringsAsFactors = TRUE, header = TRUE, read.table))  #binding imported
    ↪  csv files into one dataframe
```

## Wrangle

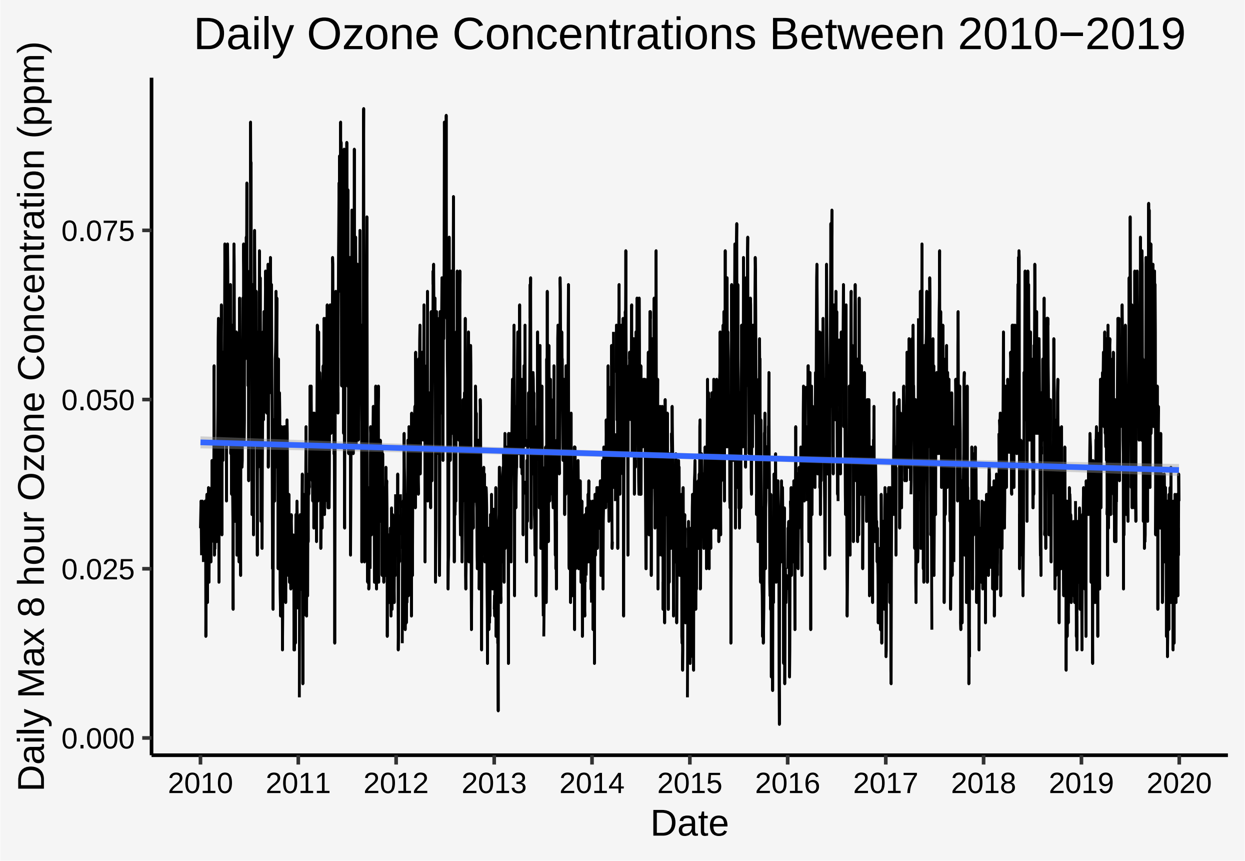
3. Set your date column as a date class.
4. Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.
5. Notice there are a few days in each year that are missing ozone concentrations. We want to generate a daily dataset, so we will need 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$Date <- mdy(GaringerOzone$Date)  #setting date column to date class

# 4
GaringerOzone <- GaringerOzone %>%
    select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)  #selecting for
    ↪  only 3 requested columns in dataframe

# 5
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), "days"))  #create
↪  dataframe with requested sequence of dates
names(Days) <- "Date"  #changing column name in this dataframe

# 6
GaringerOzone <- left_join(Days, GaringerOzone)  #joining date dataframe to ozone
↪  dataframe
```

## 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
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) + #making
↪  plot and setting aes
  geom_line() + #line plot
  geom_smooth(method = "lm") + #add linear trend line (keeping se)
    scale_x_date(date_breaks = "1 year", date_labels = "%Y") + #Showing all years on plot
  labs(y = "Daily Max 8 hour Ozone Concentration (ppm)", title="Daily Ozone
  ↪  Concentrations Between 2010-2019") #plot axis labels and title
```



Answer: Our plot suggests a slight downward trend in ozone concentration between 2010 and 2019.

## 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_clean =
    ↪  zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))  #linear interpolation for
    ↪  missing values in ozone concentration

summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration_clean)  #making sure NAs were
↪  removed
```

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

Answer: We didn't use a piecewise constant to fill in missing daily data for ozone concentration because this method just takes the nearest neighbor of the missing data and assigns the missing data point that same value. This would not be correct because it is only taking into account a single neighbor of the missing value and our data is equidistant to the value above and below it. Also since we are seeing a slight downward trend in ozone concentration, assuming that values are the same as those next to them would be incorrect. We did not use the spline interpolation method because this method uses a quadratic function to interpolate the value for the missing data; this is not as accurate as drawing a straight line in this case since our trend is more linear than it is quadratic.

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 %>% #making new df
  mutate(Month = month(Date), Year = year(Date)) %>% #adding month and year columns to
  ↪  dataframe
  group_by(Year, Month) %>% #grouping data by month and year columns
  summarise(Mean.Monthly.Ozone.Concentration =
  ↪  mean(Daily.Max.8.hour.Ozone.Concentration_clean))

GaringerOzone.monthly$Date <- ymd(paste0(GaringerOzone.monthly$Year, "-",
↪  GaringerOzone.monthly$Month, "-01")) #adding new column to monthly df with date
↪  (setting day to first day of month)
```

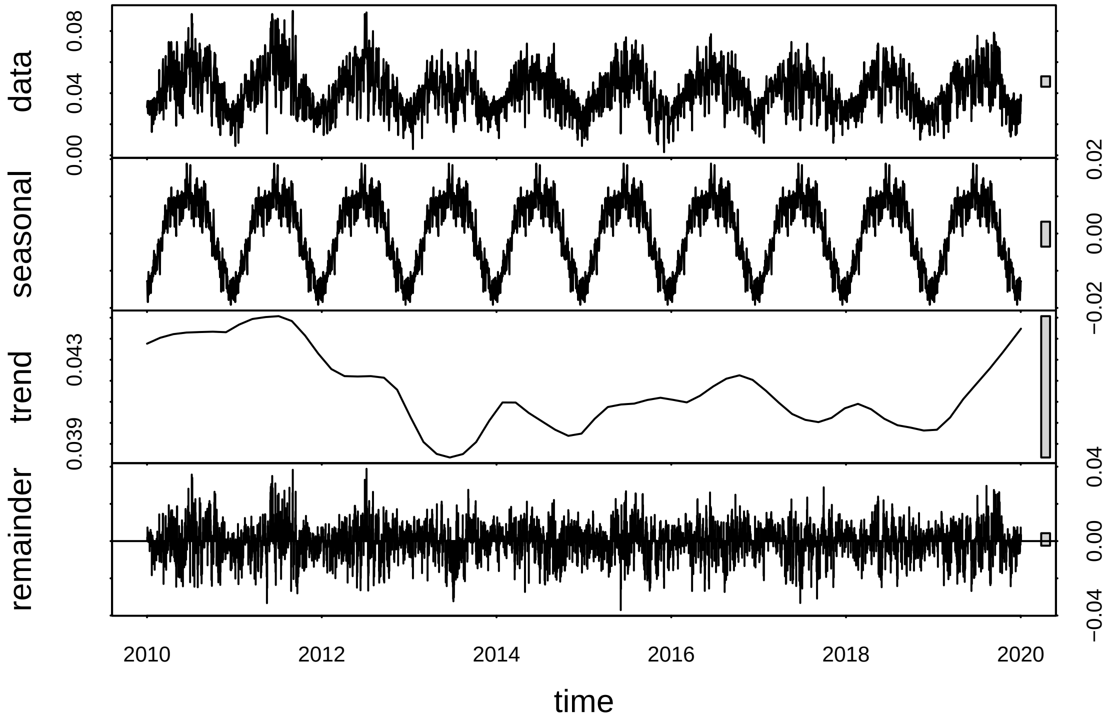
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of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

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

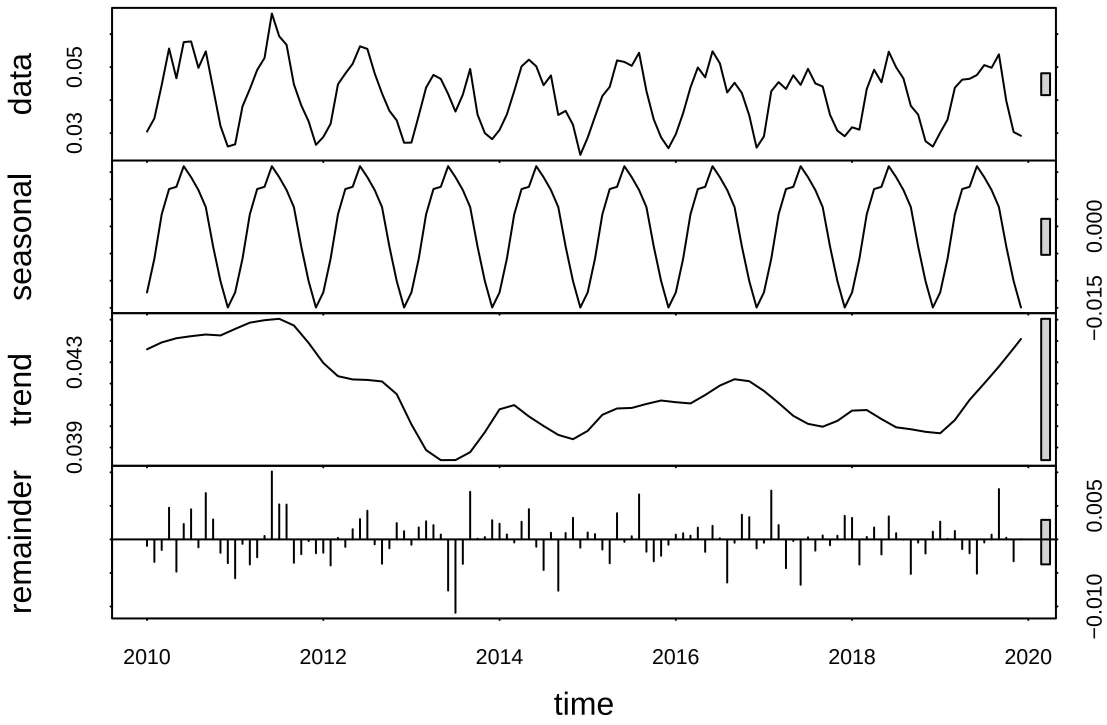
```
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration_clean,
    start = c(f_year, f_month, f_day), frequency = 365)  #creating time series for daily
    ↪  ozone concentrations

f_year2 <- year(first(GaringerOzone.monthly$Date))  #finding first year in dataframe
f_month2 <- month(first(GaringerOzone.monthly$Date))  #finding first month in dataframe

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean.Monthly.Ozone.Concentration,
    start = c(f_year2, f_month2), frequency = 12)  #creating time series for mean monthly
    ↪  ozone concentrations
```

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")
↪  #decomposing daily ozone time series
plot(GaringerOzone.daily.decomposed)  #plotting decomposed time series components
```



```
GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
↪  #decomposing monthly ozone time series
plot(GaringerOzone.monthly.decomposed)  #plotting decomposed time series components
```



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

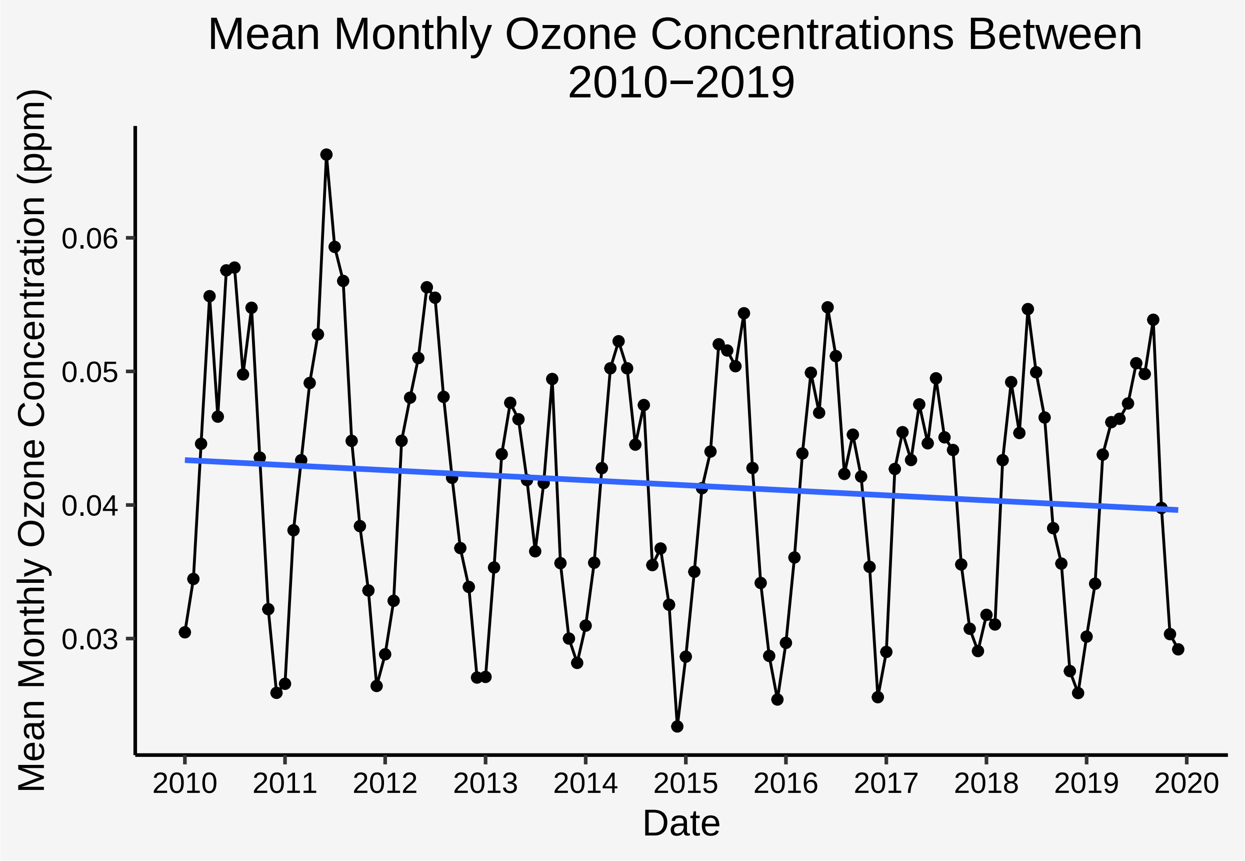
```
# 12
GaringerOzone.monthly.trend <- trend::smk.test(GaringerOzone.monthly.ts)  #running
↪  seasonal Mann-Kendall on monthly ozone time series
GaringerOzone.monthly.trend  #showing seasonal Mann-Kendall results
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

Answer: The seasonal Mann-Kendall monotonic trend analysis is most appropriate because the monthly ozone data appears to have seasonality and none of the other trend analyses that we learned are appropriate for seasonal data. THis seasonality can be seen in the decomposed plot, where the seasonal row appears to occilate - suggesting a seasonal component.

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
GaringerOzone.monthly.plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean.Monthly.Ozone.Concentration)) +
↪  #creating plot and assigning aes
  geom_point() + #adding points to plot
  geom_line() + #adding line plot
  labs(y="Mean Monthly Ozone Concentration (ppm)", title="Mean Monthly Ozone
  ↪  Concentrations Between \n2010-2019") + #labels
  geom_smooth( method = lm , se=FALSE)+ #adding linear trend line (no SE)
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") #adding all years to plot
print(GaringerOzone.monthly.plot) #printing 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: Ozone concentrations at this location have decreased through the 2010s according to our plot - this can be seen by the decreasing/negative slope of the trend line (z = -1.963, p = 0.04965). This means that we can reject the null hypothesis that there is no decreasing trend in ozone concentration over this 10-year time period (S is not equal to 0, S = -77). We accept the alternative hypothesis that there is a trend in ozone concentration between 2010 and 2019, and it is 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
GaringerOzone.monthly.noseasonal.ts <- GaringerOzone.monthly.decomposed$time.series[,
    2] + GaringerOzone.monthly.decomposed$time.series[, 3]  #removing seasonal data from
    ↪  monthly time series (and creating new time series)

# 16
GaringerOzone.monthly.noseasonal.trend2 <-
↪  trend::mk.test(GaringerOzone.monthly.noseasonal.ts)  #running Mann-Kendall on new
↪  time series without seasonal data
GaringerOzone.monthly.noseasonal.trend2  #showing results of Mann-Kendall
```

```
##
##  Mann-Kendall trend test
##
## data:  GaringerOzone.monthly.noseasonal.ts
## z = -2.672, n = 120, p-value = 0.00754
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -1.179000e+03  1.943657e+05 -1.651376e-01
```

Answer: The results of the Mann Kendall show we can reject the null hypothesis that there is no trend in ozone concentrations between 2010 and 2019 and accept the alternative hypothesis that there is a trend in ozone concentrations between 2010 and 2019 (z = -2.672, p = 0.00754). This is because S is not equal to zero (S = -1179). The results of the Mann Kendall after removing the seasonal component of the data agree with the results from the Seasonal Mann Kendall that we ran with seasonal data included in the data.