Assignment 8: Time Series Analysis

Sophie Valkenberg

Fall 2024

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

```r
library(tidyverse)
library(lubridate)
library(trend)
library(zoo)
library(Kendall)
library(tseries)
library(here)
here
getwd()

mytheme <- theme(
  axis.text = element_text(color = "black"),
        legend.position = "top",
        plot.background = element_rect("#9073ab"))

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
#1
Ozone_2010 <- read.csv(
  file=here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
         stringsAsFactors = TRUE)
Ozone_2011 <- read.csv(
  file=here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
         stringsAsFactors = TRUE)
Ozone_2012 <- read.csv(
  file=here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
         stringsAsFactors = TRUE)
Ozone_2013 <- read.csv(
  file=here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
         stringsAsFactors = TRUE)
Ozone_2014 <- read.csv(
  file=here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
         stringsAsFactors = TRUE)
Ozone_2015 <- read.csv(
  file=here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
         stringsAsFactors = TRUE)
Ozone_2016 <- read.csv(
  file=here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
         stringsAsFactors = TRUE)
Ozone_2017 <- read.csv(
  file=here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
         stringsAsFactors = TRUE)
Ozone_2018 <- read.csv(
  file=here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
         stringsAsFactors = TRUE)
Ozone_2019 <- read.csv(
  file=here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
         stringsAsFactors = TRUE)

GaringerOzone <- rbind(Ozone_2010, Ozone_2011, Ozone_2012,
                                          Ozone_2013, Ozone_2014, Ozone_2015,
                                          Ozone_2016, Ozone_2017, Ozone_2018,
                                          Ozone_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
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

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

# 5
Days <- as.data.frame(seq(from = as.Date("2010-01-01"),
                          to = as.Date("2019-12-31"), by = "day"))
names(Days) <- ("Date")

# 6
GaringerOzone <- left_join(Days, GaringerOzone_Wrang, by = "Date")
```

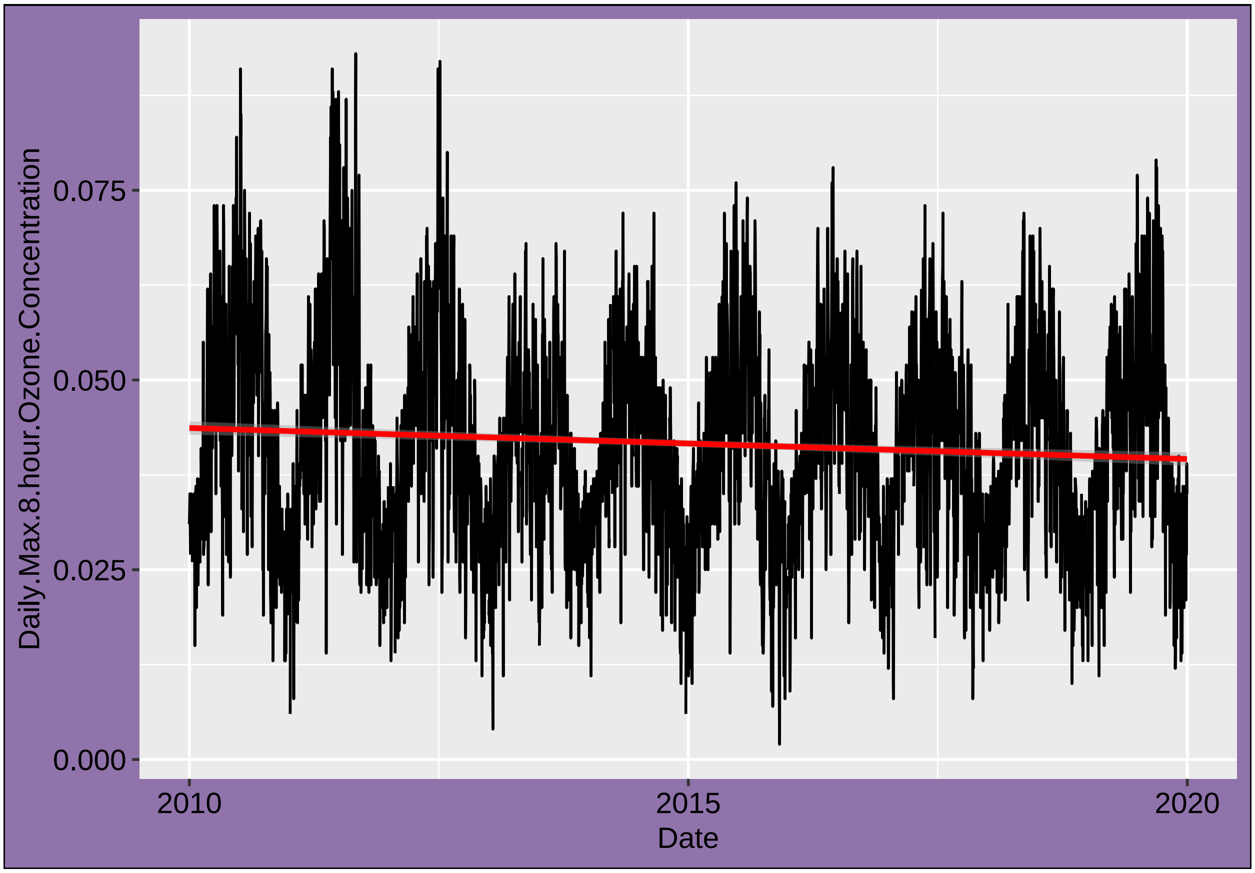
Visualize

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?

```
#7
GaringerPlotLine <- ggplot(GaringerOzone, aes(
  x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm", col="red")

print(GaringerPlotLine)
```

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

```
## Warning: Removed 63 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```



Answer: There is definitely a seasonal trend, as shown by the up-and-down pattern of the line plot. Additionally, there seems to be a very slight downward trend over time, shown by the linear trend line.

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

Answer: We used a linear interpolation instead of a piecewise constant since this data is seasonal, meaning that it changes as compared to the nearest neighbor. Additinoally, we did not use a spline interpolation because that method uses a quadratic function, but our data appeared to change linearly over time, so we wanted to use a linear function.

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)

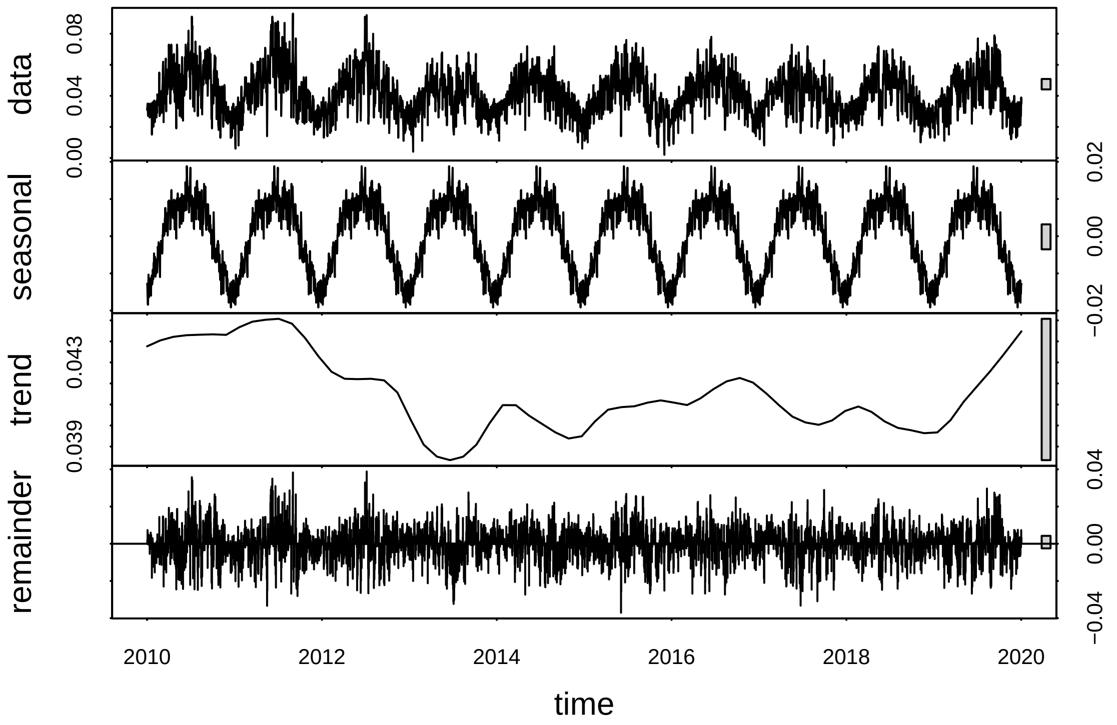
```
#9
GaringerOzone.monthly_pipe <-
  GaringerOzone_Interpolation %>%
  mutate(Month = month(Date)) %>%
  mutate(Year = year(Date)) %>%
  group_by(Year, Month) %>%
  summarise(meanOzone = mean(Conc.Clean, na.rm=TRUE),
            .groups = 'drop')

GaringerOzone.monthly <-
  GaringerOzone.monthly_pipe %>%
  mutate(Date = my(paste0(Month,"-",Year)))
```

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.

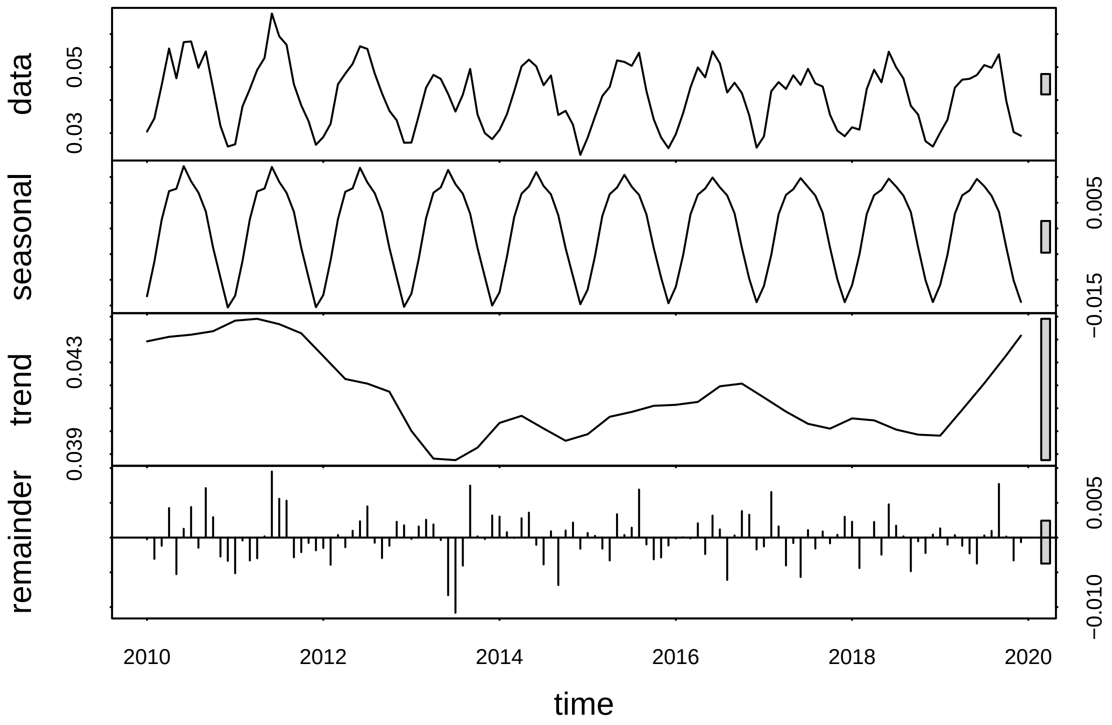
```
#10
#daily
f_month <- month(first(GaringerOzone_Interpolation$Date))
f_year <- year(first(GaringerOzone_Interpolation$Date))
f_day <- day(first(GaringerOzone_Interpolation$Date))

GaringerOzone.daily.ts <- ts(
  GaringerOzone_Interpolation$Conc.Clean,
  start=c(f_year, f_month, f_day),
  frequency = 365)

#monthly
GaringerOzone.monthly.ts <- ts(
  GaringerOzone.monthly$meanOzone,
  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.

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



```r
GaringerMonthly_Decomp <- stl(GaringerOzone.monthly.ts, s.window = 12)
plot(GaringerMonthly_Decomp)
```



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
SeasonalMannKendall(GaringerOzone.monthly.ts)
```

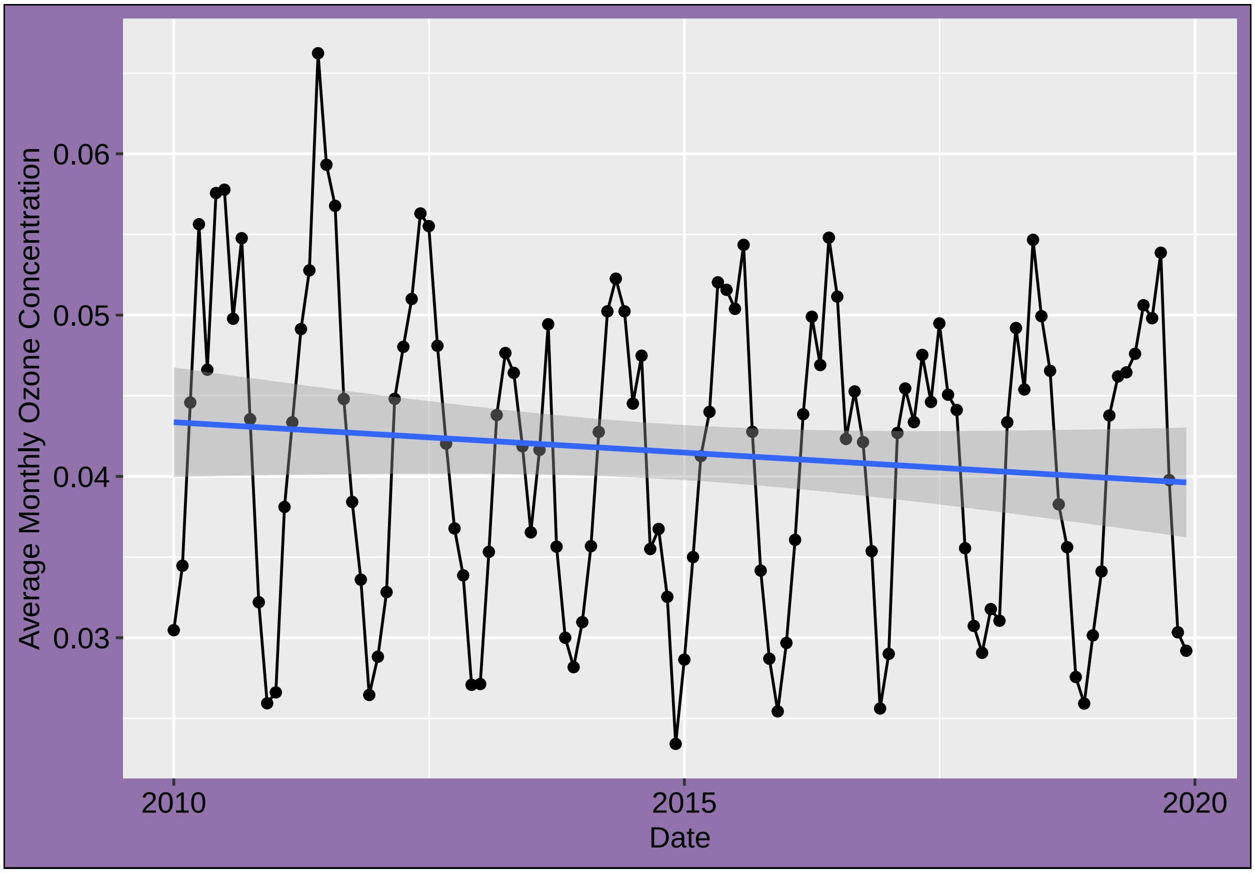
```
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The seasonal Mann-Kendall is the most appropriate because it takes seasonality into account, which our data set clearly presented with. The other monotonic trend analysis methods do not take seasonality into account,so would not give an accurate analysis for this time series.

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 = meanOzone)) +
  geom_point() +
  geom_line() +
  ylab("Average Monthly Ozone Concentration") +
  xlab("Date")+
  geom_smooth( method = lm )

print(GaringerOzone.monthly_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: Our original research question investigated whether or not ozone concentrations have changed over the 2010s at this station. According to our findings, the ozone concentrations have decreased in a statistically signficiant way, as our p-value is 0.047, which is less than 0.05 with a decay rate of -0.143% (tau = -0.143, 2-sided pvalue = 0.046724). The graph shows this visually, with a trend line showing a slight decline.

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_Components <- as.data.frame(
  GaringerMonthly_Decomp$time.series[,1:3])

GaringerOzone.monthly_NonSeas <-
  GaringerOzone.monthly.ts - GaringerMonthly_Decomp$time.series[,1:3]
```

```r
GaringerOzone.monthly_Components <-
  mutate(GaringerOzone.monthly_Components,
         Nonseasonal = GaringerOzone.monthly_NonSeas,
         Observed = GaringerOzone.monthly$meanOzone,
         Date = GaringerOzone.monthly$Date)

#16
GaringerOzone.monthly_MKtest <- MannKendall(GaringerOzone.monthly_NonSeas)
print(GaringerOzone.monthly_MKtest)
```

```
## tau = -0.00963, 2-sided pvalue =0.78548
```

> Answer: With removing seasonality, our new p-value is 0.7855, meaning that the decrease is not statistically signifcant when seasonality is removed, so ozone concentrations have not significantly decreased over time according to this analysis. The new tau value is very small, which may expalin why the decrease is not significant. These results differ from the seasonal Mann Kendall test, which did point to signifcant decreases.