Assignment 8: Time Series Analysis

Justin Maynard

Fall 2023

```
#Clear environment
rm(list = ls())
```

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)
```

```
## -- 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   3.4.4      v tibble    3.2.1
## v lubridate 1.9.2      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
```

```
library(lubridate)
library(zoo)


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

library(trend)
library(here)


## here() starts at /Users/justinmaynard/Fall_2023_EDE

here()


## [1] "/Users/justinmaynard/Fall_2023_EDE"

mytheme <-
  theme(
      plot.background = element_rect(fill = "#f5f0d9", color = NA),
      panel.background = element_rect(fill = "#f5f0d9", color = NA),
      legend.background = element_rect(fill = "#f5f0d9", color = NA),
      panel.grid.major =  element_line(color = "#d0d0d0", linewidth = .2),
      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.

```
#1
Ozone2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"))
Ozone2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"))
Ozone2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"))
Ozone2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"))
Ozone2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"))
Ozone2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"))
Ozone2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"))
Ozone2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"))
Ozone2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"))
Ozone2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"))

GaringerOzone  <- rbind(Ozone2010, Ozone2011, Ozone2012, Ozone2013, Ozone2014, Ozone2015,
                        Ozone2016, Ozone2017, Ozone2018, Ozone2019)
```

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 )
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(as.Date("2010-01-01"), as.Date("2019-12-31"), "day"))

colnames(Days) <- "Date"

# 6
GaringerOzone <- left_join(Days, GaringerOzone.wrangled)
```

```
## Joining with 'by = join_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
q7 <- ggplot(GaringerOzone, aes(y = Daily.Max.8.hour.Ozone.Concentration, x = Date)) +
  geom_line() +
  geom_smooth()
print(q7)
```

```
## 'geom_smooth()' using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

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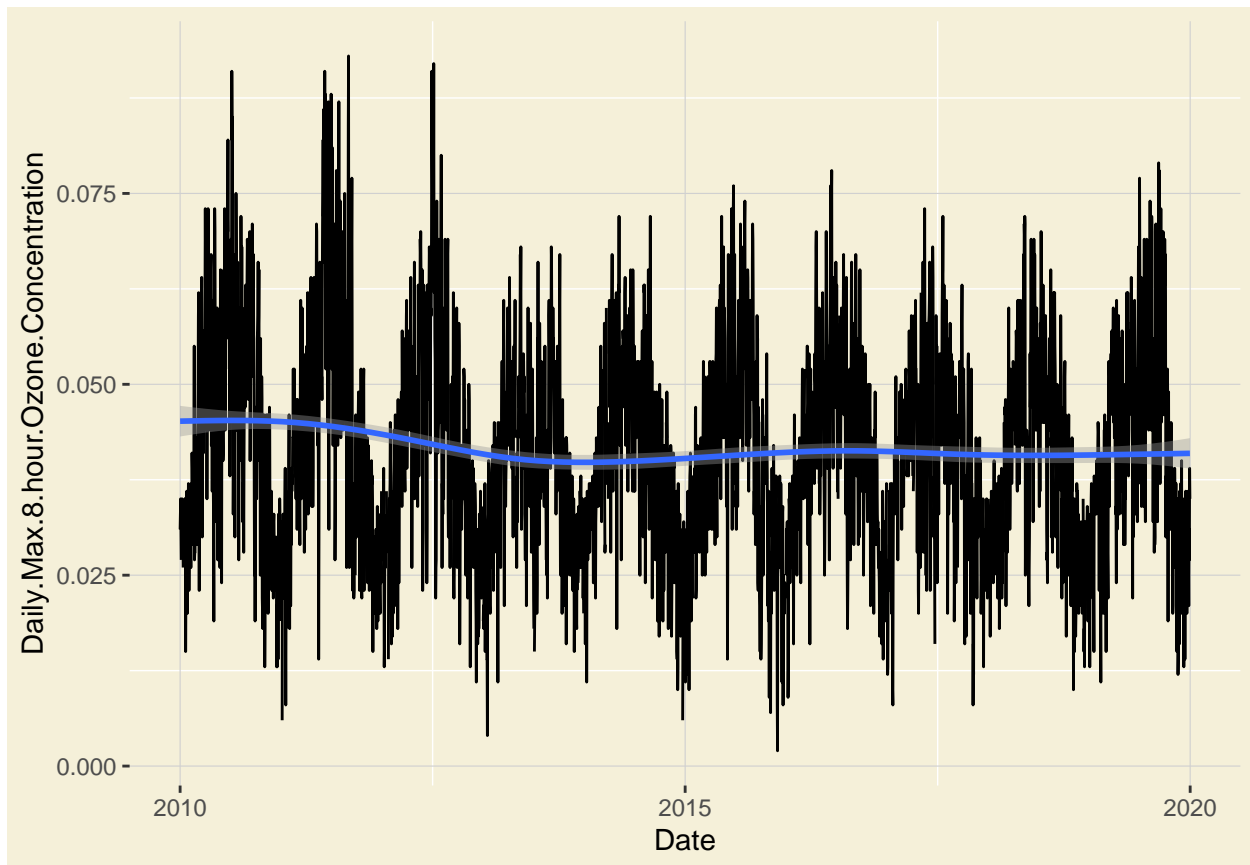

Answer: The plot suggests a small decrease in ozone concentration over time. It appears to be too small to be considered a trend, but more testing would need to be conducted to determine this.

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 = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration) )

print(summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration))
```

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

Answer: Linear interpolation was used as the ozone concentration vary linearly and increase or decrease seasonally. Because of this trend individual missing values are assumed to fall inbetween their neighbors, and the 'connect the dots' approach can be used. Piecewise is not used as the missing data would be either larger or smaller than their neighbor depending on the trend direction, not equal to their neighbor. Spline is not needed as the trend is not 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

GaringerOzone.monthly$FloorDate <- floor_date(GaringerOzone$Date, 'month')

GaringerOzone.monthly <- GaringerOzone.monthly %>%
  group_by(FloorDate) %>%
  summarise(Mean = mean(Daily.Max.8.hour.Ozone.Concentration))

colnames(GaringerOzone.monthly) <- c("Date", "MeanOzoneConcentration")
```

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(2010-01-01), frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$MeanOzoneConcentration,
                               start = c(2010-01-01), frequency = 12)
```
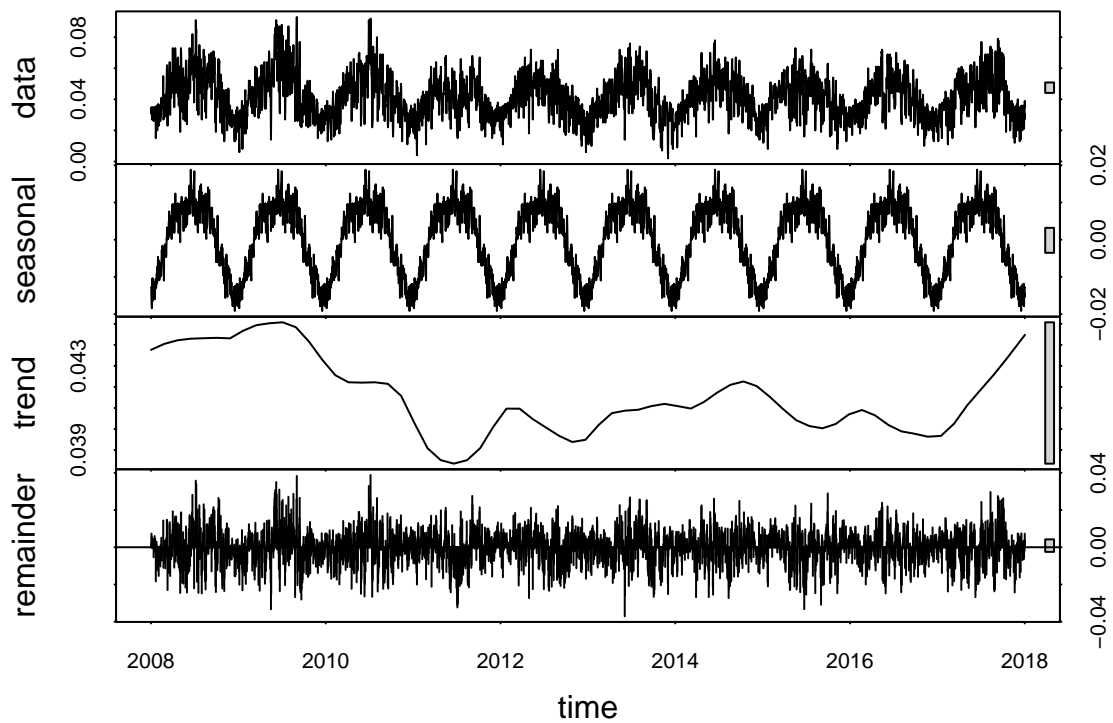
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.
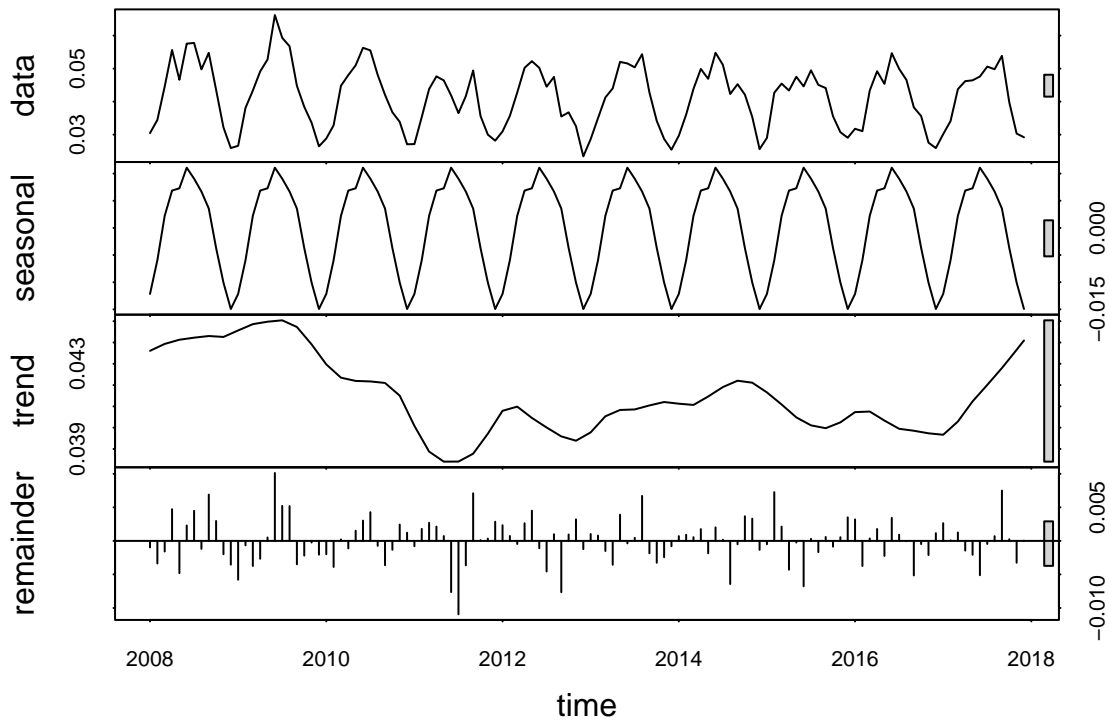
```
#11

GaringerOzone.daily.decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")
print(plot(GaringerOzone.daily.decomp))
```

```
## NULL
```

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

```
## NULL
```

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_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results

print(summary(monthly_trend))
```

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

Answer: Using a seasonal test is most appropriate because we know there is seasonal variation in our data, and the SeasonalMannKendall test will account for this.

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

montly_ozone_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = MeanOzoneConcentration)) +
  geom_point() +
  geom_line() +
  ylab("Mean Ozone Concentration") +
  geom_smooth( method = lm )
print(montly_ozone_plot)
```

`geom_smooth()` using formula = 'y ~ x'
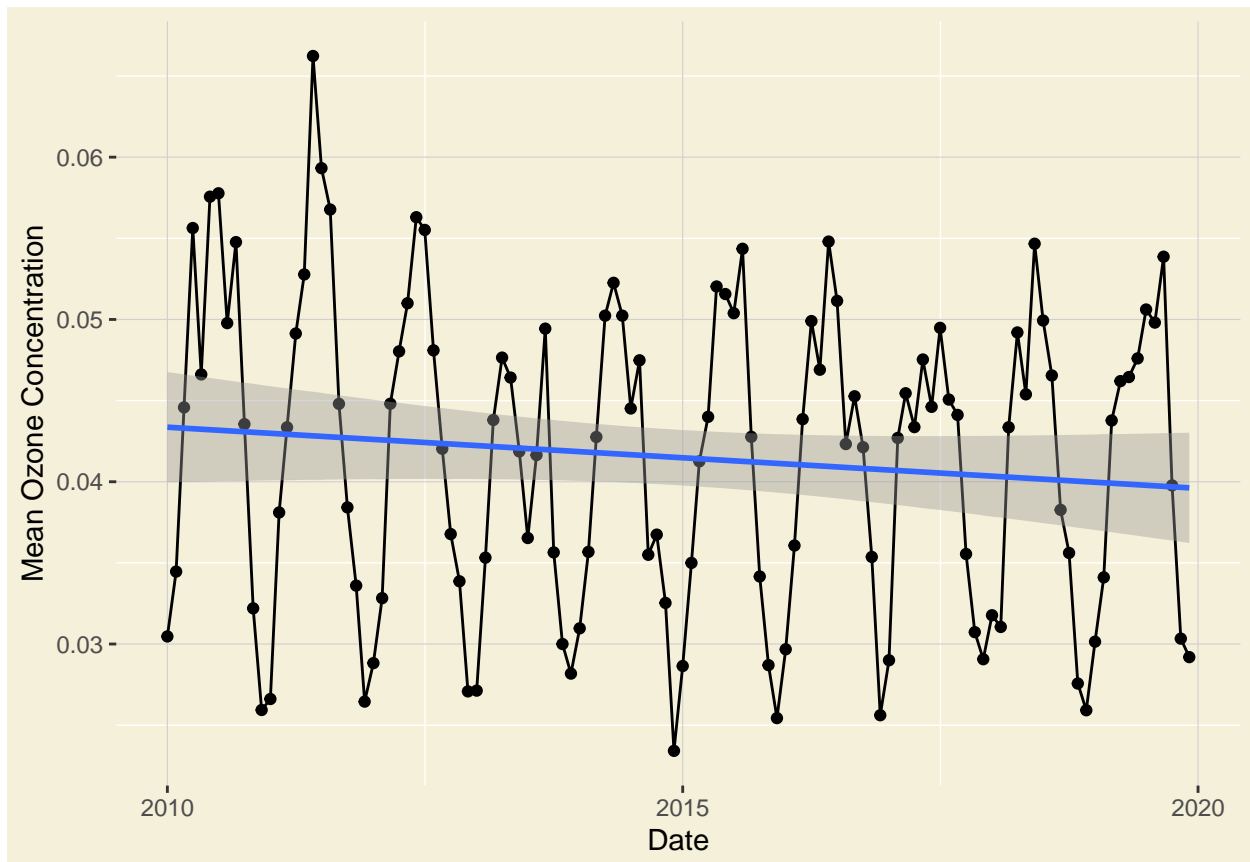
14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

Answer: Mean ozone concentration has decreased at a rate of -.143 per year, within the 95% confidence interval (tau = -.143, 2-sided p value = .0467).

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

GaringerOzone.monthly.decomp.components <- as.data.frame(GaringerOzone.monthly.decomp$time.series[,1:3])
GaringerOzone.monthly.decomp.seasonal <- GaringerOzone.monthly.decomp.components$seasonal


GaringerOzone.noseasonal.ts <- GaringerOzone.monthly.ts - GaringerOzone.monthly.decomp.seasonal



#16
GaringerOzone.noseasonal_MK <- Kendall::MannKendall(GaringerOzone.noseasonal.ts)

print(summary(GaringerOzone.noseasonal_MK))
```

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

```r
print(summary(monthly_trend))
```

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

Answer: The Seasonal Mann Kendall test on the complete series results in a tau of -.143 while the non seasonal Mann Kendall test on the non seasonal series results in a tau of -.165. The non seasonal Mann Kendall shows a stronger negative trend of mean ozone concentration over time as the absolute value of the tau is larger. Both have 2 sided p values that pass the 95% confidence threshold.