Assignment 8: Time Series Analysis

Andreana Chou

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(here)
```

```
## here() starts at C:/Users/andre/Documents/R Studio Files/EDE_Fall2023
```

```
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 1.9.2     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(Kendall)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
#assigned custom theme to my_theme object
my_theme <- theme_classic(base_size = 12) +
  theme(panel.background = element_rect(color = "lightblue", fill = "white"),
        panel.grid.major = element_line(color = "lightblue", linewidth = 0.5),
        legend.title = element_text(color = "black"),
        legend.position = "right")

#set custom theme as default
theme_set(my_theme)
```

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

#read raw data files individually
EPA_2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
                     stringsAsFactors = TRUE)
EPA_2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
                     stringsAsFactors = TRUE)
EPA_2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
                     stringsAsFactors = TRUE)
EPA_2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
                     stringsAsFactors = TRUE)
EPA_2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
                     stringsAsFactors = TRUE)
EPA_2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
                     stringsAsFactors = TRUE)
EPA_2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
                     stringsAsFactors = TRUE)
EPA_2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
```

```
                    stringsAsFactors = TRUE)
EPA_2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
                    stringsAsFactors = TRUE)
EPA_2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
                    stringsAsFactors = TRUE)

#create a list out of all raw files
join_list <- list(EPA_2010, EPA_2011, EPA_2012, EPA_2013, EPA_2014, EPA_2015,
                  EPA_2016, EPA_2017, EPA_2018, EPA_2019)

#use reduce() function with
GaringerOzone <- reduce(join_list, full_join)
```

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

#use mdy from lubridate to set as date
GaringerOzone$Date <- mdy(GaringerOzone$Date)

#check data class
data.class(GaringerOzone$Date)
```

```
## [1] "Date"
```

```
# 4

#use select() to extract columns
GaringerOzone_4 <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration,
         DAILY_AQI_VALUE)

# 5

#create a sequence of dates and set as a dataframe
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), "days"))
#change name of singular column using colnames()
colnames(Days) <- "Date"
```

```
# 6

#left_join with Days before GaringerOzone_4
GaringerOzone <- left_join(Days, GaringerOzone_4)
```

```
## Joining with `by = join_by(Date)`
```

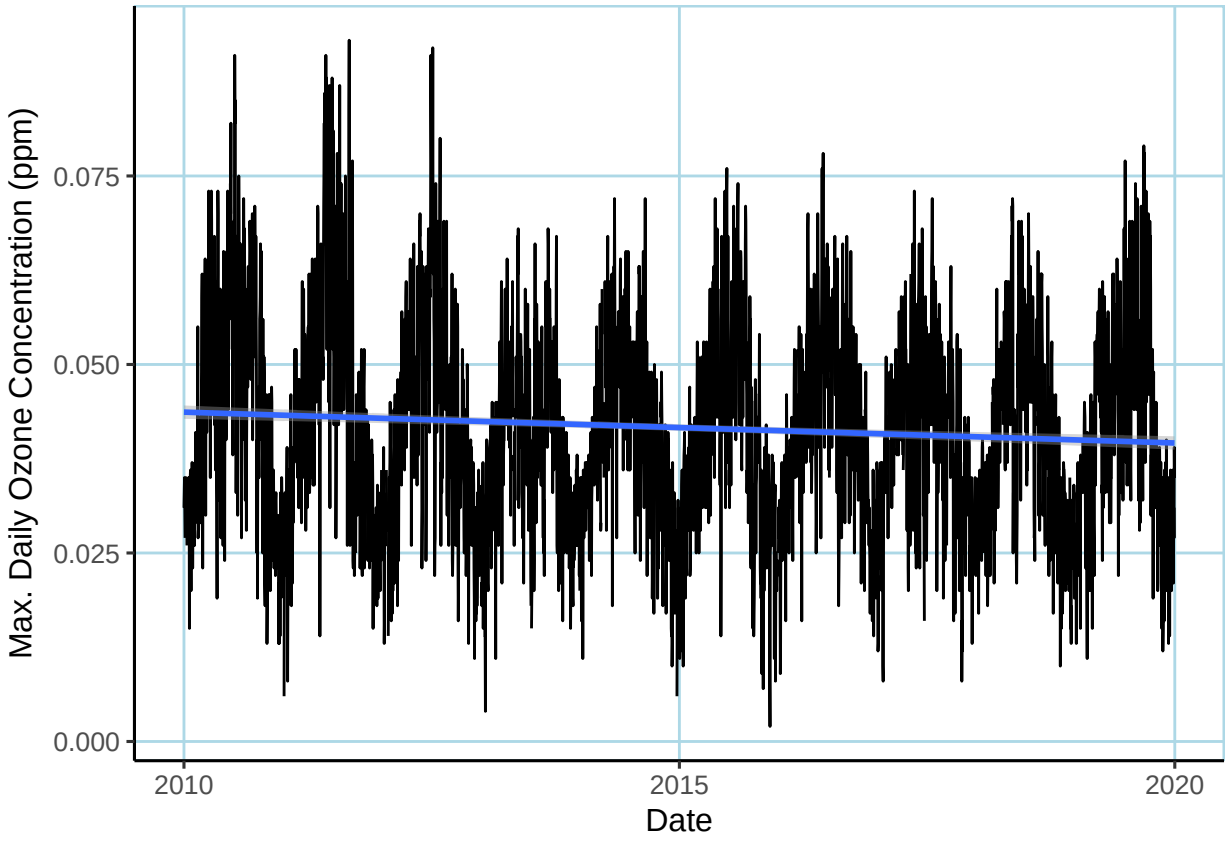
Visualize

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

#created line plot with geom_line() and inserted trend line with geom_smooth
Ozone_plot7 <- GaringerOzone %>%
  ggplot(aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method="lm") +
  ylab(expression("Max. Daily Ozone Concentration (ppm)"))
Ozone_plot7
```

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

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



Answer: The plot suggests there is a slight downward trend in ozone concentration over time, but more analysis is required to determine the strength of the relationship.

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

#applied mutate() function and nested interpolation function approx() within
GaringerOzone <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           approx(Date, Daily.Max.8.hour.Ozone.Concentration, method = "linear",
                  xout = Date)$y)
```

Answer: Piecewise constant is not used because the "nearest neighbor" approach would not work well if the dates are equidistantly apart. Spline is not used because we are plotting Ozone concentrations using a linear, not quadratic model.

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)

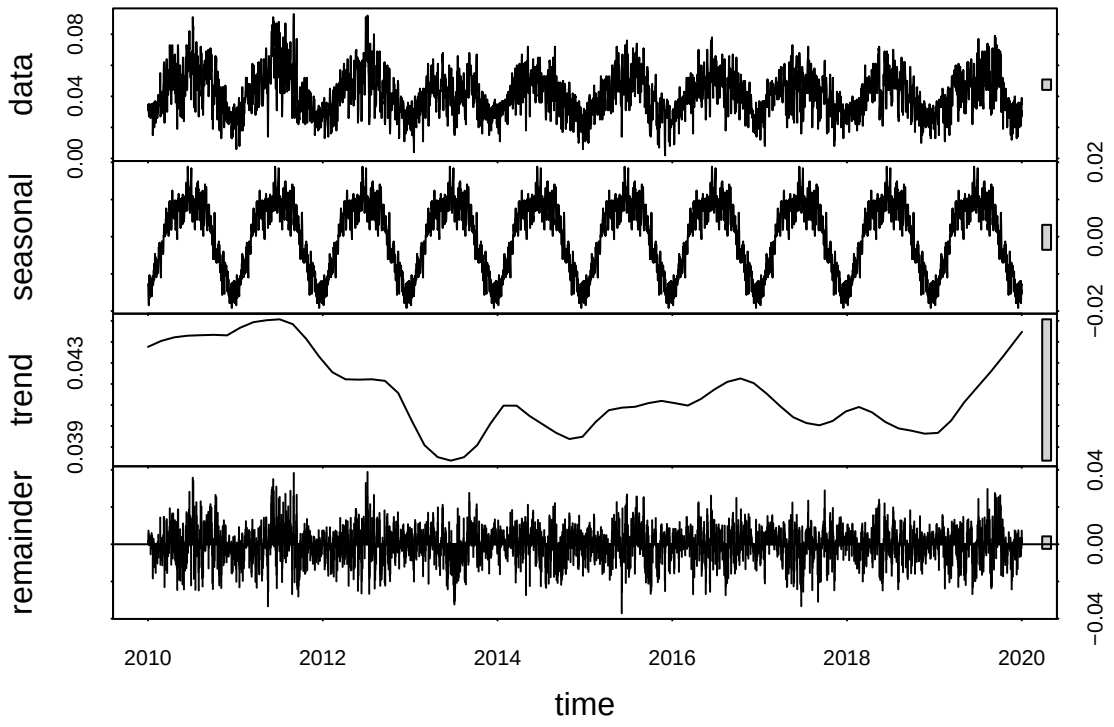
```
#9

#added new Month and Year columns with mutate() and lubridate functions
#obtained mean ozone by grouping month, then year, and applying mean()
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Month = month(Date), Year = year(Date)) %>%
  group_by(Year, Month) %>%
  summarize(Mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration), .groups="drop")

#ungroup()
#created new Date column with mutate() and paste() function, reassigned to dataframe
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = make_date(year=Year, month=Month, day=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.

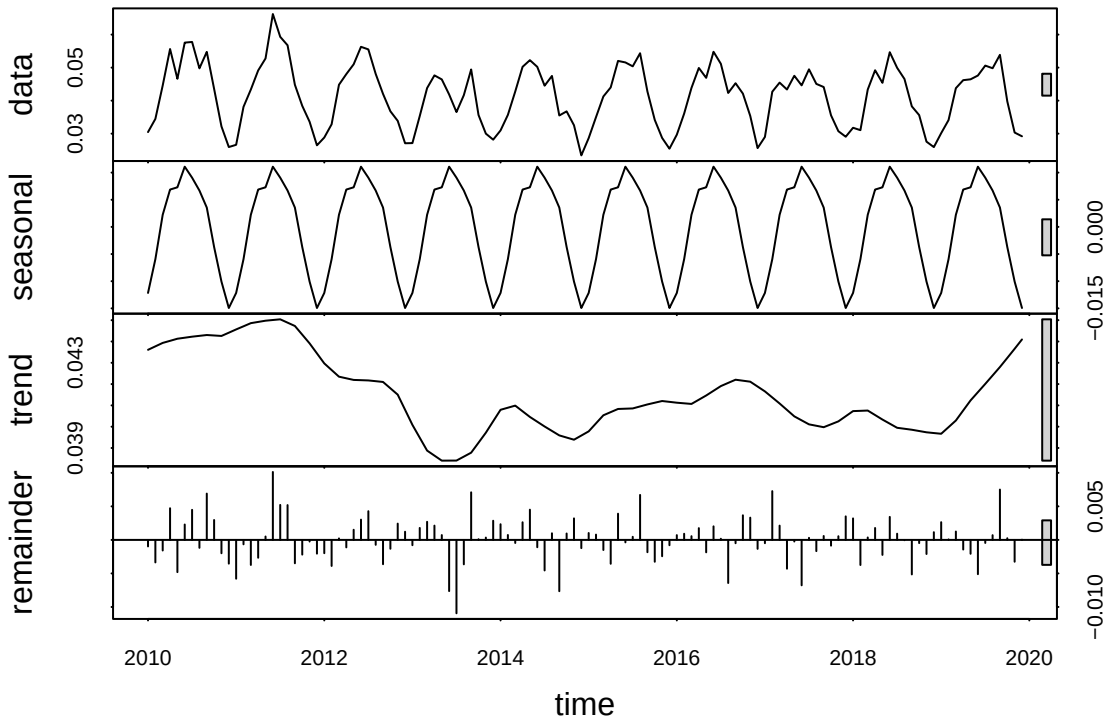
```
#10

#created a time-series object with frequency=365 due to daily values
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             frequency=365, start=c(2010, 1))

#created a time-series object with frequency=12 due to monthly values
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Ozone,
                               frequency=12, start=c(2010, 1))
```

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

```
#11

#applied time-series objects as arguments within stl()
Garinger_daily_decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")
#plot() stl object to obtain decomposed trends
plot(Garinger_daily_decomp)
```




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

#applied seasonal mann kendall() and summary() to time series object
smk_Garinger_monthly <- SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(smk_Garinger_monthly)
```

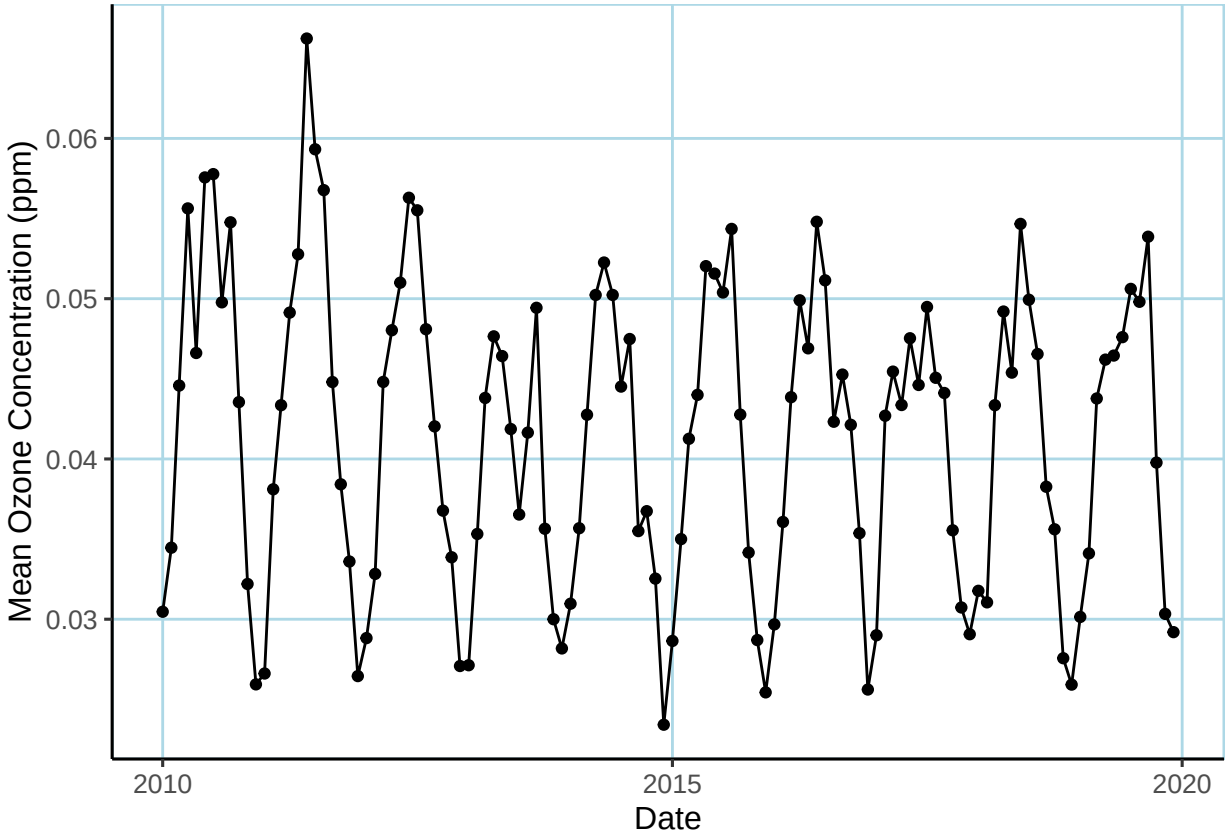
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: A seasonal Mann-Kendall is the most appropriate because the data points are temporally spaced by month, which means we will cover all the seasons of a year.

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

#generated monthly plot with geom_point(), geom_line()
monthly_plot_13 <- GaringerOzone.monthly %>%
  ggplot(aes(x=Date, y=Mean_Ozone)) +
```

```
  geom_point() +
  geom_line() +
  ylab("Mean Ozone Concentration (ppm)")

monthly_plot_13
```



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 shows a seasonal variation in mean ozone concentration by month. With a tau of -0.143, the Seasonal Mann-Kendall test reveals a negative monotonic trend. The p-value of 0.046724 rejects the H0 of no monotonic trend present, which implies mean ozone concentration has monotonically decreased over time (between 2010 and 2020).

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

#select for only the "seasonal" series in the Grainger_monthly_decomposition time series from
```

```r
#GaringerOzone.ts, resulting in a time-series object with modified ozone values
GaringerOzone.monthly_nonseasonal <- GaringerOzone.monthly.ts -
  Garinger_monthly_decomp$time.series[, "seasonal"]

#16

#applied MannKendall() and summary() to time-series object
Garinger.monthly.noseasonal.mk <- MannKendall(GaringerOzone.monthly_nonseasonal)
summary(Garinger.monthly.noseasonal.mk)
```

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

Answer: By analyzing the monthly ozone data without their seasonality component, we still find a negative monotonic trend due to tau value of -0.165, which is slightly stronger than that from the Seasonal Mann Kendall analysis. We reject the H0 that there is no monotonic trend present due to the p-value of 0.0075402. The p-value is also more significant than that from the Seasonal Mann Kendall analysis. This implied that the non-seasonal mean ozone concentration has monotonically decreased over time (between 2010 and 2020).