Assignment 8: Time Series Analysis

Cammie Moore

Spring 2025

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     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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
```

```r
library(trend)
```

```
## Warning: package 'trend' was built under R version 4.4.3
```

```r
library(here)
```

```
## Warning: package 'here' was built under R version 4.4.2

## here() starts at C:/Users/cammi/OneDrive/Documents/EDE_Spring2025
```

```r
getwd()
```

```
## [1] "C:/Users/cammi/OneDrive/Documents/EDE_Spring2025"
```

```r
here()
```

```
## [1] "C:/Users/cammi/OneDrive/Documents/EDE_Spring2025"
```

```r
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "bottom")
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
EPAair_O3_GaringerNC2010_raw <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")

EPAair_O3_GaringerNC2011_raw <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")

EPAair_O3_GaringerNC2012_raw <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")

EPAair_O3_GaringerNC2013_raw <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")

EPAair_O3_GaringerNC2014_raw <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")

EPAair_O3_GaringerNC2015_raw <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")

EPAair_O3_GaringerNC2016_raw <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
```

```r
EPAair_O3_GaringerNC2017_raw <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")

EPAair_O3_GaringerNC2018_raw <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")

EPAair_O3_GaringerNC2019_raw <- read_csv("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")

GaringerOzone <- bind_rows(EPAair_O3_GaringerNC2010_raw, EPAair_O3_GaringerNC2011_raw, EPAair_O3_Garing
```

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.

```r
# 3
GaringerOzone <- GaringerOzone |>
  mutate(Date = mdy(Date))

# 4
GaringerOzone <- GaringerOzone |>
  select(Date, `Daily Max 8-hour Ozone Concentration`, DAILY_AQI_VALUE)

# 5
Days <- as.data.frame(seq.Date(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "day"))
Days <- Days |>
  rename(Date = `seq.Date(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "day")`)

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

Visualize

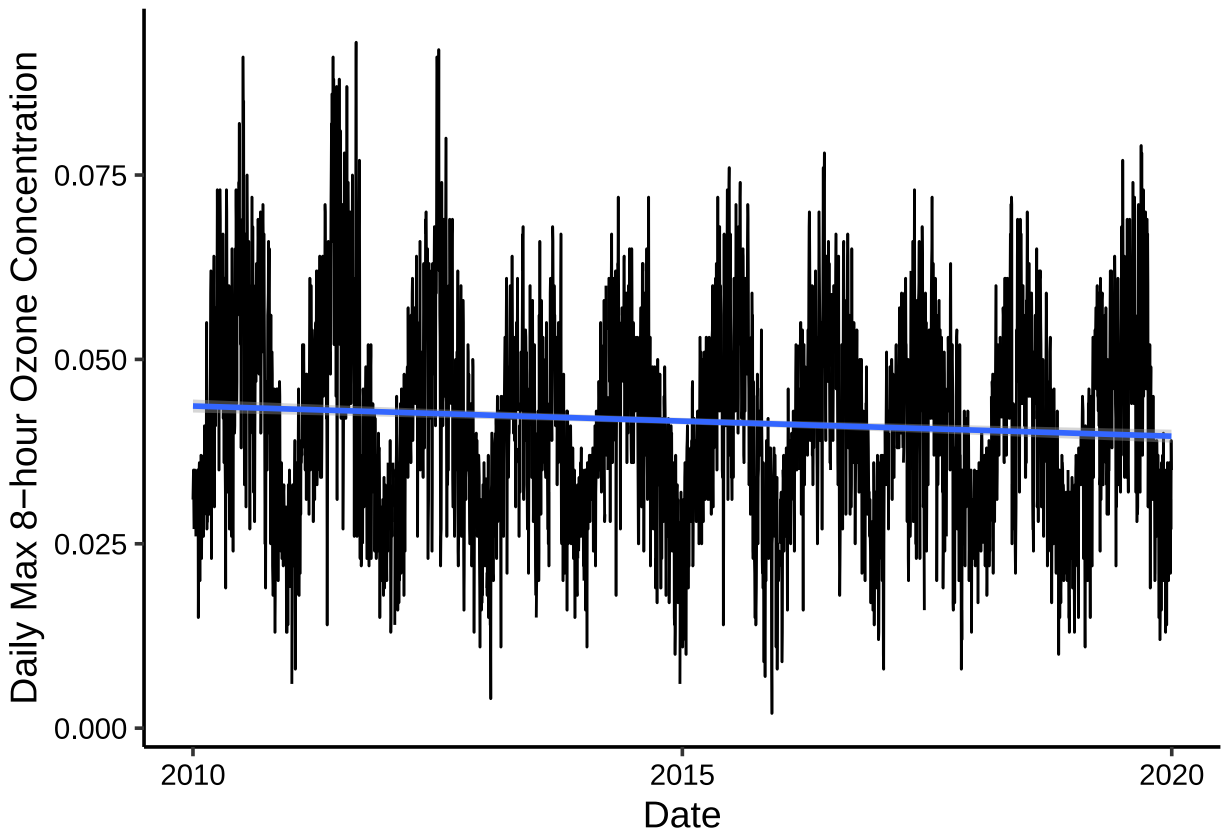
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```r
#7

ozone_plot <- ggplot(GaringerOzone, aes(Date, `Daily Max 8-hour Ozone Concentration`))+
  geom_line()+
  geom_smooth(method = lm)
print(ozone_plot)
```

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

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



Answer: The plot strongly suggests annual seasonality trends in ozone levels. Additionally, Daily Max 8-hour Ozone Concentration levels appear to be decreasing slightly overall from 2010 to 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$`Daily Max 8-hour Ozone Concentration` <- na.approx(GaringerOzone$`Daily Max 8-hour Ozone
```

Answer: We used linear interpolation to fill in the missing daily data for ozone concentration because the data is moving, within a day, in a relatively simple and linear path. A piecewise constant would not be the most accurate estimation of missing days, as it would produce a step-wise pattern rather than the gradual increase or decrease we expect from ozone concentration. A spline interpolation may be too complicated a model for the straight-forward, gradual, day-to-day movement of ozone levels.

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)

```r
#9
GaringerOzone.monthly <- GaringerOzone |>
  mutate(Month = month(Date),
         Year = year(Date))|>
  group_by(Month, Year) |>
  summarise(
    mean_ozone = mean(`Daily Max 8-hour Ozone Concentration`),
    .groups = "drop"
  )

GaringerOzone.monthly <- GaringerOzone.monthly |>
  mutate(MonthYear = as.Date(paste(Year, Month, "01", sep = "-"))
  )
```

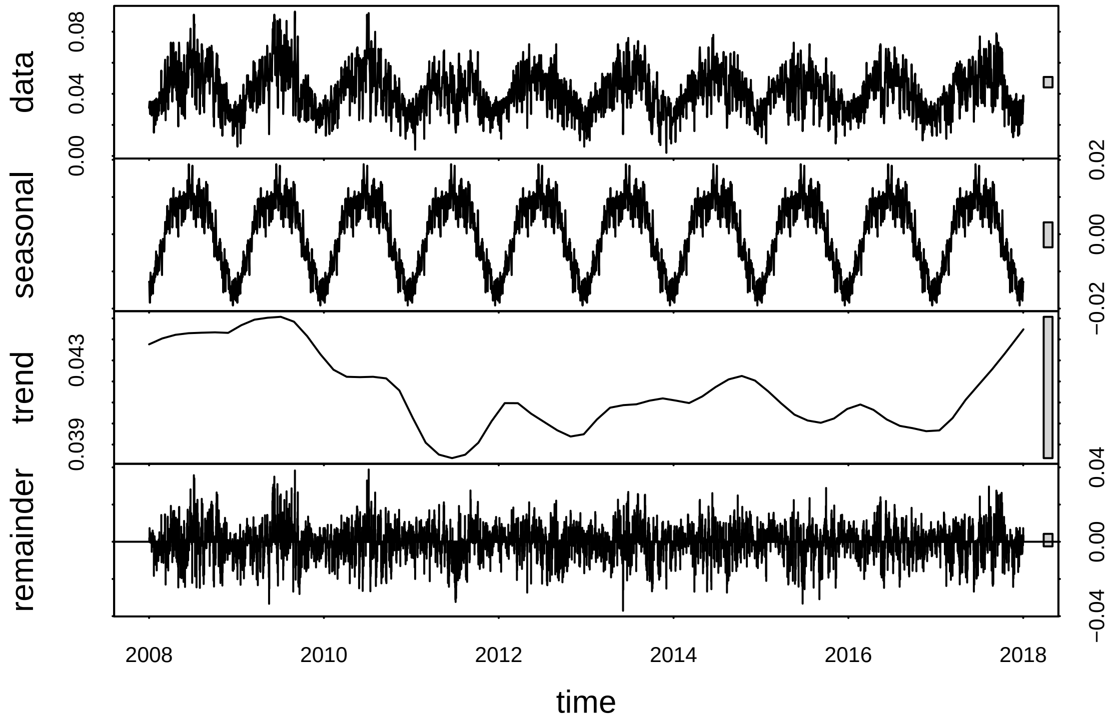
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of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

```r
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$`Daily Max 8-hour Ozone Concentration`,
                   start=c(2010-01-01),
                   frequency=365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
                               start = c(2010-01-01),
                               frequency = 12)
```

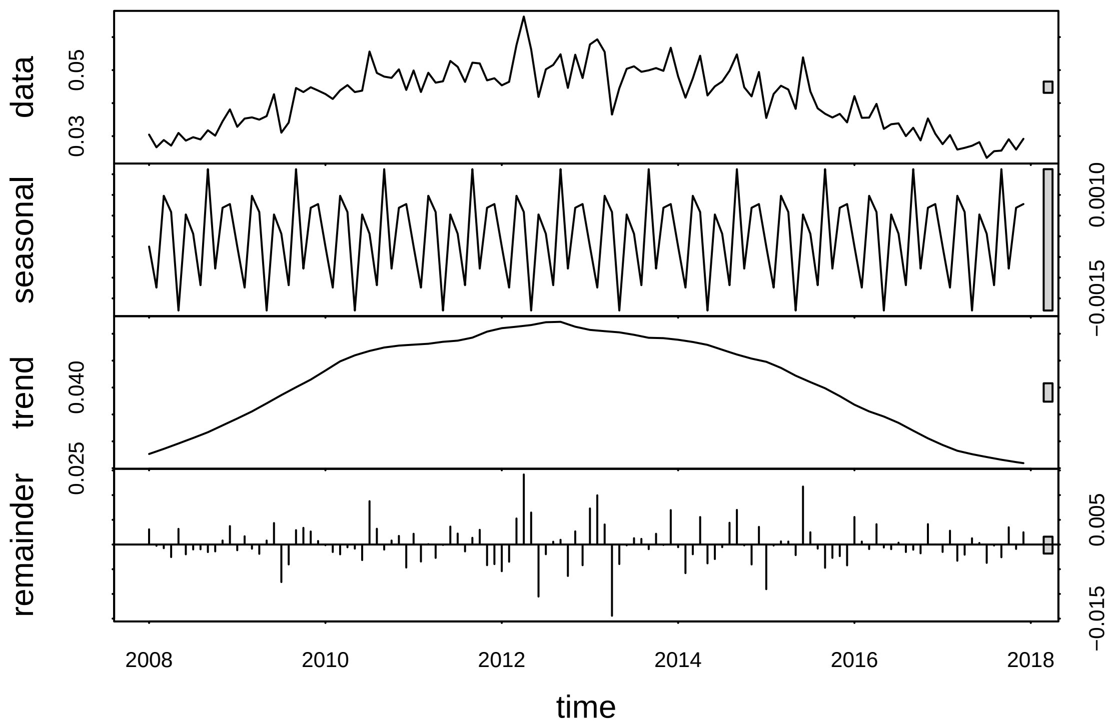
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```r
#11

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




```r
GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GaringerOzone.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?

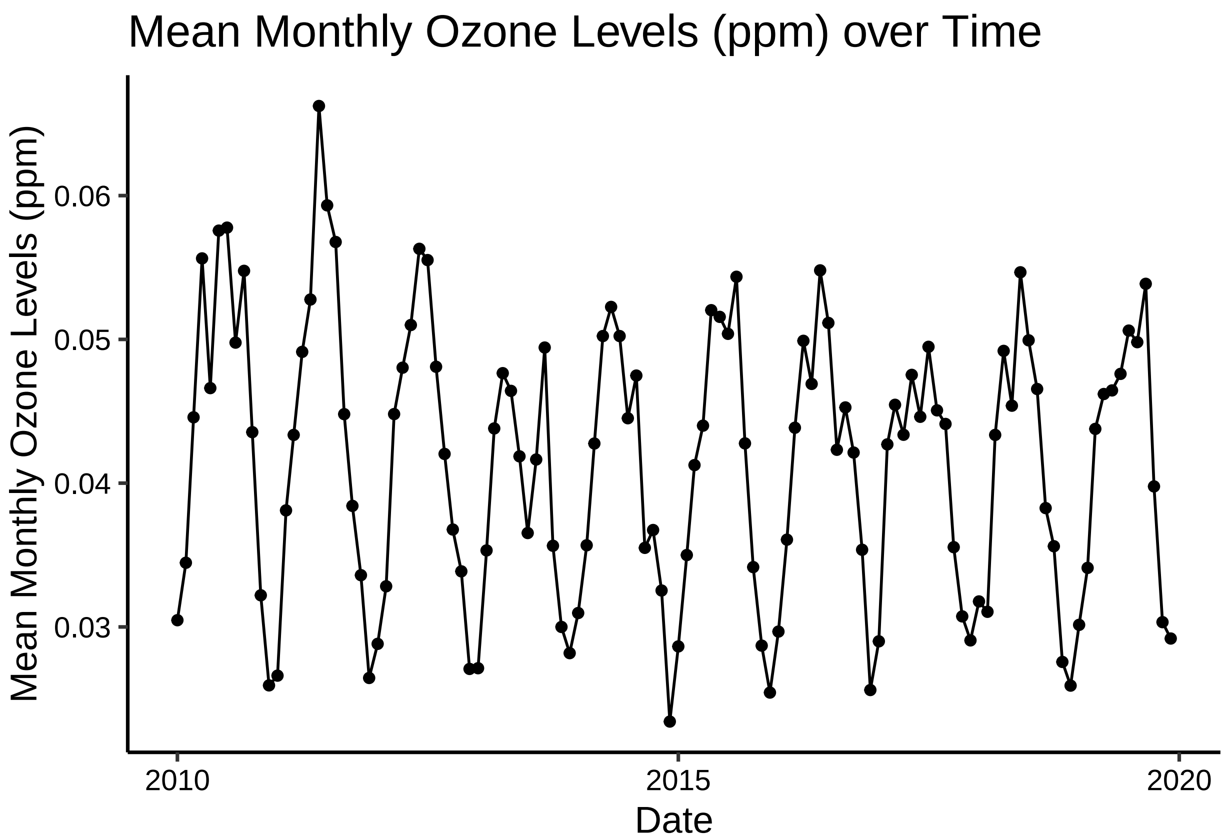
```
#12
Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
```

```
## tau = -0.1, 2-sided pvalue =0.16323
```

Answer: There is a large seasonal component to the data, therefore, the non-parametric seasonal test will allow the data to fit the test assumptions and be analyzed.

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
ggplot(GaringerOzone.monthly, aes(MonthYear, mean_ozone))+
  geom_point()+
  geom_line()+
  labs(title = "Mean Monthly Ozone Levels (ppm) over Time",
       x = "Date",
       y = "Mean Monthly Ozone Levels (ppm)")
```



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 levels are changing throughout the year due to seasonality. However, the monthly mean ozone level data did not have a statistically signifcant monotonic trend (Seasonal Kendall-Man, p = 0.163) over the years gathered for the study (2010 - 2019).

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
seasonal_component <- GaringerOzone.monthly.decomposed$time.series[, "seasonal"]
GaringerOzone.monthly.deseasonalized <- GaringerOzone.monthly.ts - seasonal_component


#16

Kendall::MannKendall(GaringerOzone.monthly.deseasonalized)
```

```
## tau = -0.101, 2-sided pvalue =0.10388
```

Answer: After seasonality was removed from the mean monthly Ozone data, there still does not exist strong evidence for a monotonic trend (p-value = 0.103). Though the deseasonalized data had a lower p-value, it appears there is not a large decrease in mean monthly ozone levels over time.