Assignment 8: Time Series Analysis

Sarah Sussman

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

```
# Load packages
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 error
```

```
library(lubridate)
library(trend)
library(zoo)
```

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

```
library(here)
```

```
## here() starts at /home/guest/EDA_Spring2025
```

```
# Check working directory
here()
```

```
## [1] "/home/guest/EDA_Spring2025"
```

```
# Set ggplot theme
mytheme <- theme_bw(base_size = 12) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top") #alternative: legend.position + legend.justification
```

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 in csv's
O3.2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"), stringsAsFactors
O3.2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"), stringsAsFactors
O3.2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"), stringsAsFactors
O3.2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"), stringsAsFactors
O3.2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"), stringsAsFactors
O3.2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"), stringsAsFactors
O3.2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"), stringsAsFactors
O3.2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"), stringsAsFactors
O3.2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"), stringsAsFactors
O3.2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"), stringsAsFactors

# combine
GaringerOzone <- rbind(O3.2010, O3.2011, O3.2012, O3.2013, O3.2014, O3.2015, O3.2016,
                       O3.2017, O3.2018, O3.2019)
```

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 <- 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"))
colnames(Days) <- "Date"

# print
#Days

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

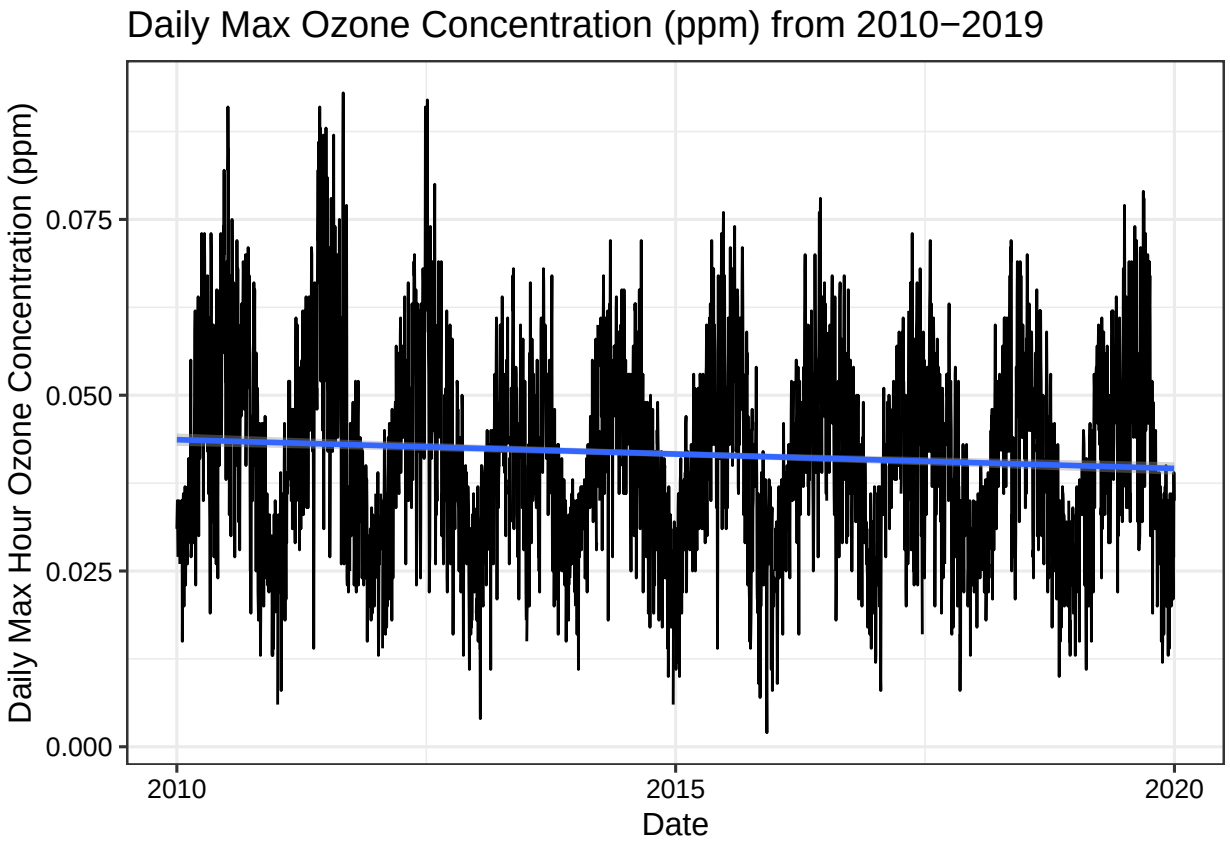
Visualize

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?

```
#7
ggplot(GaringerOzone, aes(x = Date,
                          y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm") +
  labs(x= "Date",
       y = "Daily Max Hour Ozone Concentration (ppm)",
       title = "Daily Max Ozone Concentration (ppm) from 2010-2019") +
  mytheme
```

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

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



Answer: Yes, the plot suggests a slight decrease (looking at the blue line) in the daily max ozone concentration from 2010-2019.

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
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration) # 63 NA's
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```
GaringerOzone <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration, Date))
```

Answer: We did not use piecewise constant interpolation because that uses a nearest neighbor approach, where the value that is filled in is equal to the next closest value. The likelihood of two measurements being exactly the same on a given day is small, so this would create some

inaccuracies (the dataset had about 63 NA's). We did not use the spline interpolation because that uses a quadractic function to interpolate instead of a straight line, and a straight line (like in linear) is more appropriate.

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 %>%
  mutate(Month = month(Date), Year = year(Date)) %>%
  group_by(Year, Month) %>%
  summarise(mean_ozone = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date2 = ymd(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.

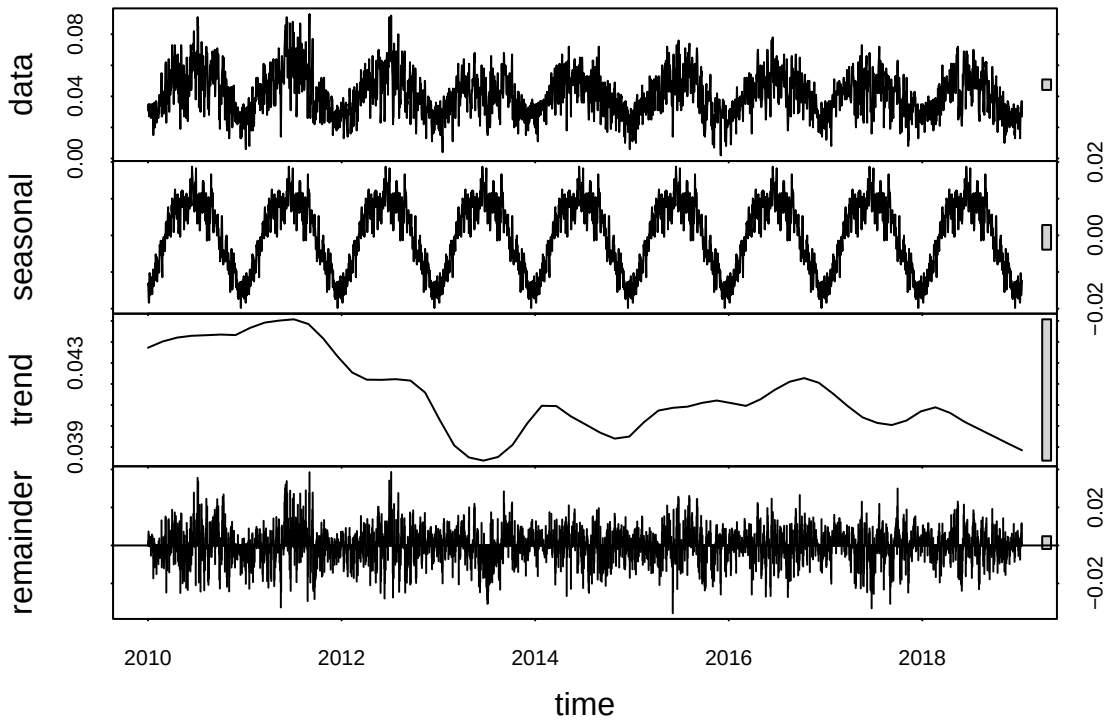
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010,1), end = c(2019,12), frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
                               start = c(2010,1), end = c(2019,12), frequency = 12)
```

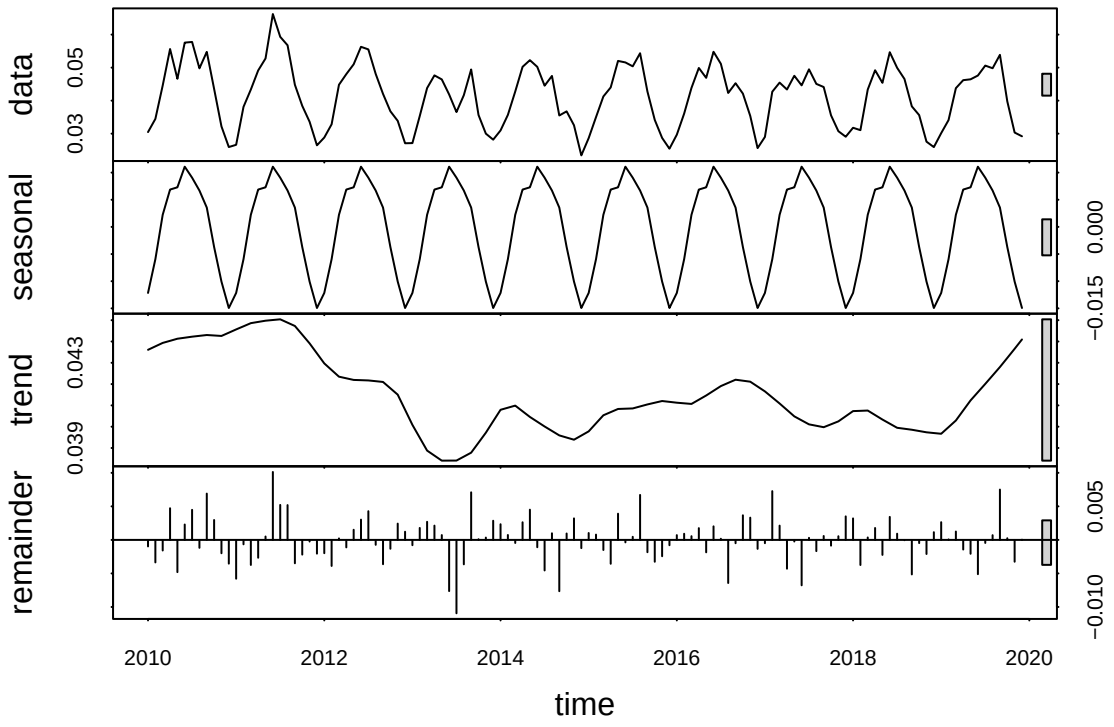
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

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

plot(GaringerOzone.daily.decomposed)
```




```
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
GaringerOzone.monthly.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# View test results
GaringerOzone.monthly.trend
```

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

Answer: The seasonal Mann-Kendall is the most appropriate out of the other tests we have learned about because it can handle seasonality. Other tests that we have learned about but thaat cannot handle seasonality are linear regression, (Non-seasonal) Mann Kendall, and Spearman Rho.

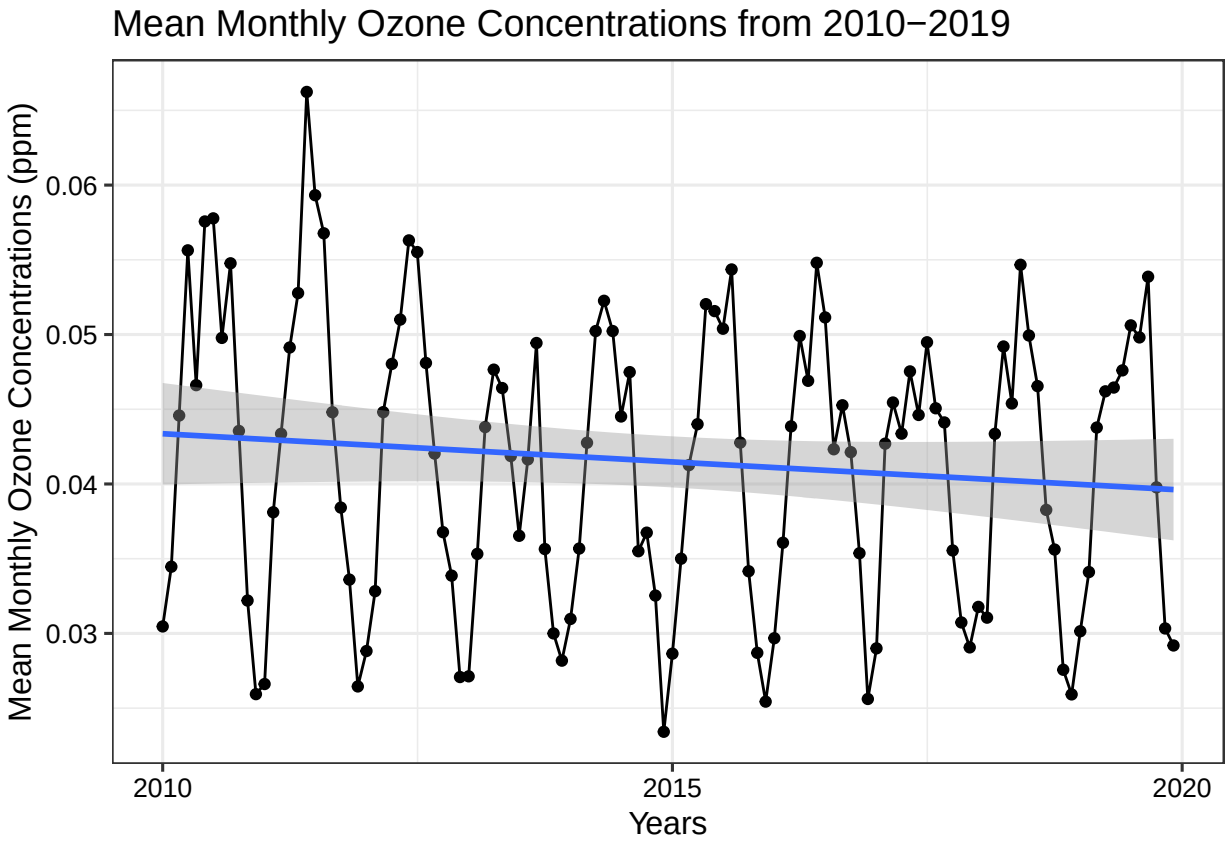
13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```
# 13
mean_monthly_ozone.plot <- ggplot(GaringerOzone.monthly,
                                  aes(x = Date2,
                                      y = mean_ozone)) +
  geom_point() +
```

```
  geom_line() +
  geom_smooth(method = "lm") +
  labs(x = "Years", y = "Mean Monthly Ozone Concentrations (ppm)",
       title = "Mean Monthly Ozone Concentrations from 2010-2019") +
  mytheme

# Visualize plot
mean_monthly_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: Ozone concentrations have changed over the 2010s at this station. The negative slope indicates that there is a decreasing trend in mean monthly ozone concentration (ppm) from 2010-2019. The results of the seasonal Mann-Kendall test also indicate that the trend is negative, or decreasing (tau = -0.143). The calculated p-value from the season Mann-Kendall test is 0.047, which is <0.05. This means that trend is statisitically significant and not a result of random chance.

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.nonseasonal <- GaringerOzone.monthly.ts - GaringerOzone.monthly.decomposed$time.s

#16

GaringerOzone.monthly.nonseasonal.trend <- Kendall::MannKendall(GaringerOzone.monthly.nonseasonal)

# View test results
GaringerOzone.monthly.nonseasonal.trend
```

```
## tau = -0.165, 2-sided pvalue =0.0075402
```

Answer: The Mannn-Kendall test run on the nonseasonal ozone monthly series shows a decreasing trend (tau = -0.165), just like the seasonal monthly series; in fact it is a slightly larger decreasing trend than the seasonal series (seasonal tau = -0.143). The p-value for the nonseasonal trend is 0.007, which is smaller than the seasonal series (seasonal p-value = 0.04). Because the p-value is well below 0.05, it means that the trend is statistically significant and not a result of random chance.