Assignment 8: Time Series Analysis

Emily Kuhlmann

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

```
getwd()
```

```
## [1] "/Users/emilykuhlmann/Documents/Documents/Fall23/EDA/EDE_Fall2023"
```

```
library(tidyverse)
library(lubridate)
library(zoo)
library(trend)
library(Kendall)

mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        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
Garinger10 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                        stringsAsFactors = T)
Garinger11 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                        stringsAsFactors = T)
Garinger12 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                        stringsAsFactors = T)
Garinger13 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                        stringsAsFactors = T)
Garinger14 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                        stringsAsFactors = T)
Garinger15 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                        stringsAsFactors = T)
Garinger16 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                        stringsAsFactors = T)
Garinger17 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                        stringsAsFactors = T)
Garinger18 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                        stringsAsFactors = T)
Garinger19 <-
  read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                        stringsAsFactors = T)

GaringerOzone <- rbind(Garinger10, Garinger11, Garinger12, Garinger13,
                        Garinger14, Garinger15, Garinger16, Garinger17,
                        Garinger18, Garinger19)
```

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$Date <- mdy(GaringerOzone$Date)

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

# 5
start_date <- as.Date("2010-01-01")
end_date <- as.Date("2019-12-31")
Days <- as.data.frame(seq(start_date, end_date, by = "day"))
Days$Date <- Days$`seq(start_date, end_date, by = "day")`
Days$`seq(start_date, end_date, by = "day")` <- NULL

# 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
O3_time <- ggplot(GaringerOzone,
                  aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(linewidth = 0.25) +
  geom_smooth(method = 'lm', se = F) +
  ylab("Daily Max 8-hour Ozone Conc, ppm") +
  xlab("Year") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
O3_time
```

Answer: The plot suggests that there is a seasonal trend in ozone concnetration as well as a small downward trend over the 10 years of data.

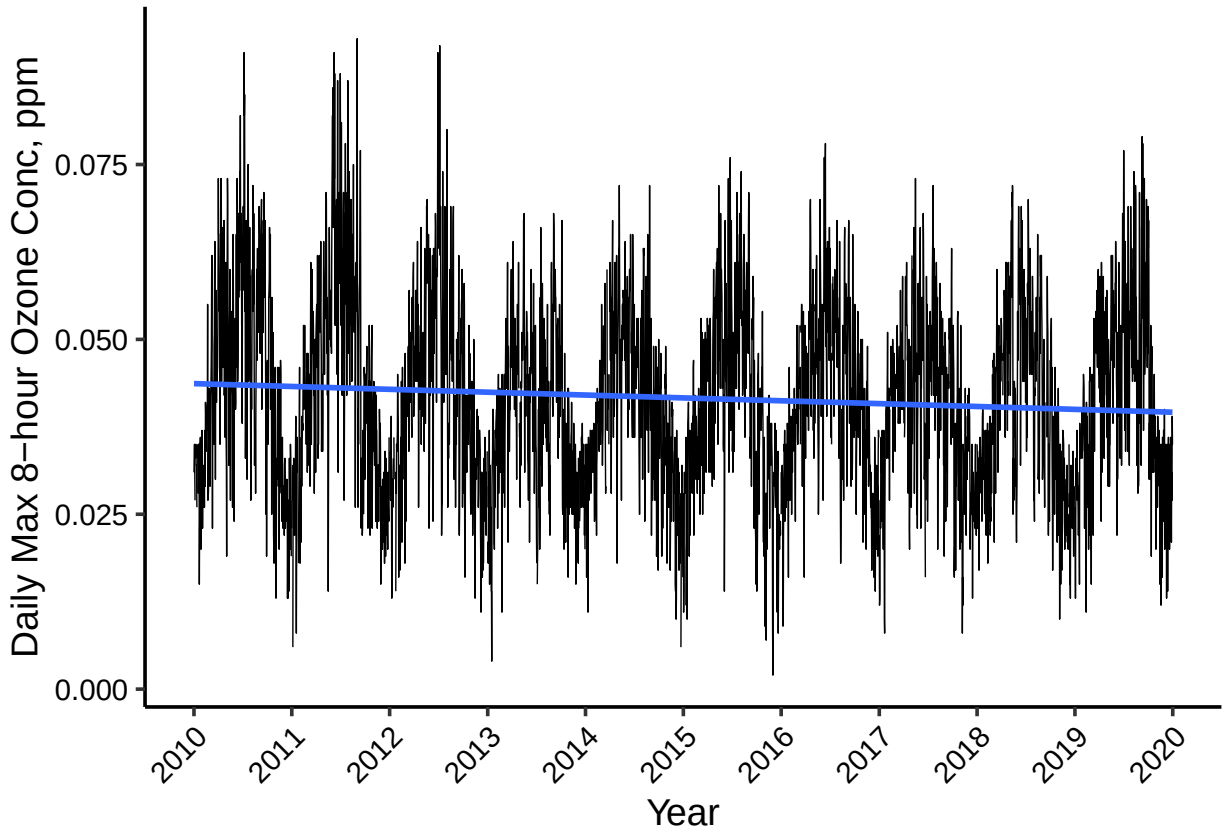

Figure 1: Daily ozone concentrations over time

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

   Answer: The piecewise constant assignes the value taken at the nearest date, but we have daily data and ozone concentration can vary daily so it's better to use a linear interpolation. The spline interpolation is not necessary because ozone concentration is not likely to change quadratically over the course of a day.

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(Year = year(Date), Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarize(Mean.Monthly.Ozone = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  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
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010, 1), frequency = 365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean.Monthly.Ozone,
                               start = c(2010, 1), frequency = 12)
```

11

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

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

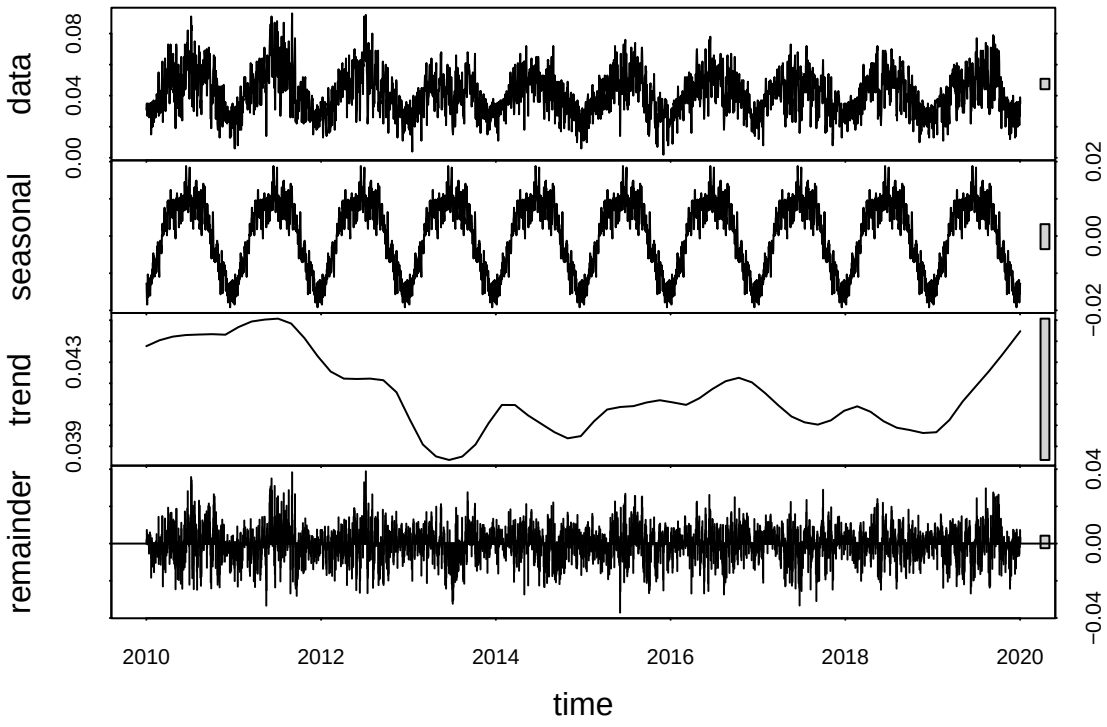

Figure 2: Decomposition plots of daily ozone time series

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

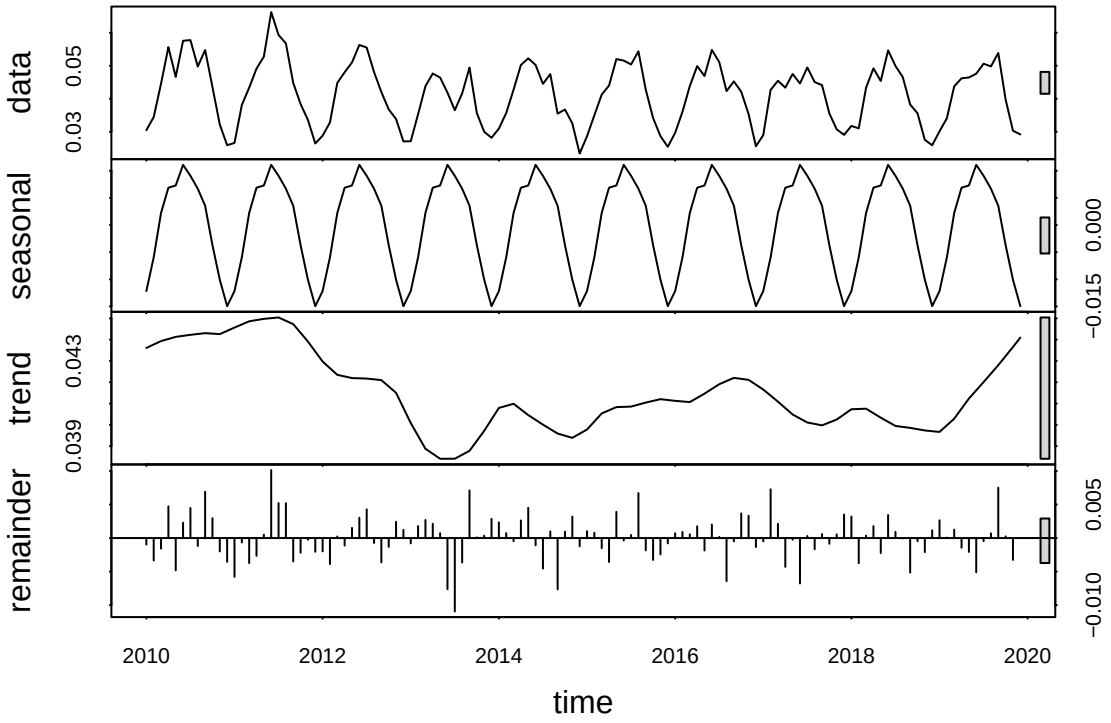

Figure 3: Decomposition plots of montly ozone time series

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_ozone_trend <- SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(monthly_ozone_trend)
```

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

Answer: The seasonal Mann-Kendall is appropriate becasue there is clearly a seasonal trend in the ozone concentration.

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
MeanMonthlyOzone <- ggplot(GaringerOzone.monthly,
                           aes(x = Date, y = Mean.Monthly.Ozone)) +
  geom_point(color = "darkgreen") +
  geom_line() +
  ylab("Mean Monthly Ozone Concentration, ppm") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
MeanMonthlyOzone
```

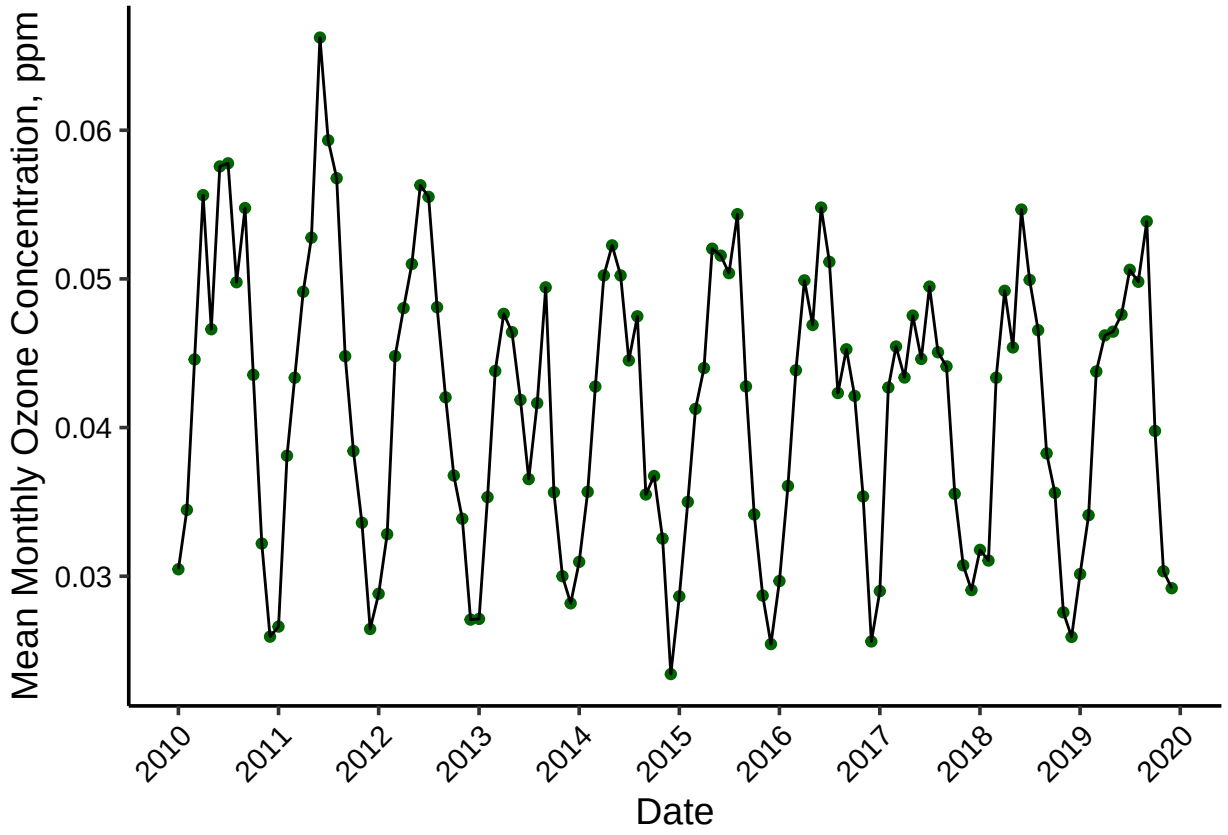

Figure 4: Mean monthly ozone concentrations

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. Have ozone concentrations changed over the 2010s at this station? > Answer: Ozone concentrations have changed significantly over the 2010s at this station. The Seasonal Mann Kendall test gave a tau of -0.143 and a 2-sided p-value of 0.0467. Since this p-value is less than 0.05, the null hypothesis is rejected and we can say that there is a trend in the data. Since tau is negative, the ozone concentrations have decreased over the time period. This is a small change but it can be visualized slightly in the plot.

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.

```
GaringerOzone.monthly.ts.Components <- as.data.frame(GaringerOzone.monthly.ts_decomp$time.series[,1:3])

GaringerOzone.monthly.nonseasonal <- GaringerOzone.monthly.ts.Components$trend +
  GaringerOzone.monthly.ts.Components$remainder

#16
GaringerOzone.monthly.nonseasonal.ts <- ts(GaringerOzone.monthly.nonseasonal)
nonseasonal_monthly_trend <- MannKendall(GaringerOzone.monthly.nonseasonal.ts)
summary(nonseasonal_monthly_trend)
```

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

Answer: The tau for the non-sesonal Mann Kendall test is -0.165 with a p-value of 0.0075. This is a stronger negative trend in the data than when seasonality was included.