Assignment 8: Time Series Analysis

Kendall Barton

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(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(ggthemes)

getwd()
```

```
## [1] "/Users/kendallbarton/Downloads/EDE_Fall2023"
```

```
theme_set(theme_bw())
```

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
getwd()
```

```
## [1] "/Users/kendallbarton/Downloads/EDE_Fall2023"
```

```
GaringerOzone <- data.frame() #starts as blank df

for (file in list.files("./Data/Raw/Ozone_TimeSeries")) { #iterate through each file in folder
  current_file <- read.csv(paste("./Data/Raw/Ozone_TimeSeries/", file, sep = ""),
                           stringsAsFactors = TRUE) #read file
  GaringerOzone <- rbind(GaringerOzone, current_file) #combine dfs
}
```

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) #set Date as date class

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

# 5
Days <- as.data.frame(seq(as.Date("2010/1/1"), as.Date("2019/12/31"), "day"))
#get complete sequence of days
colnames(Days) <- "Date" #change column name

# 6
GaringerOzone <- left_join(Days, GaringerOzone) #combine dfs
```

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

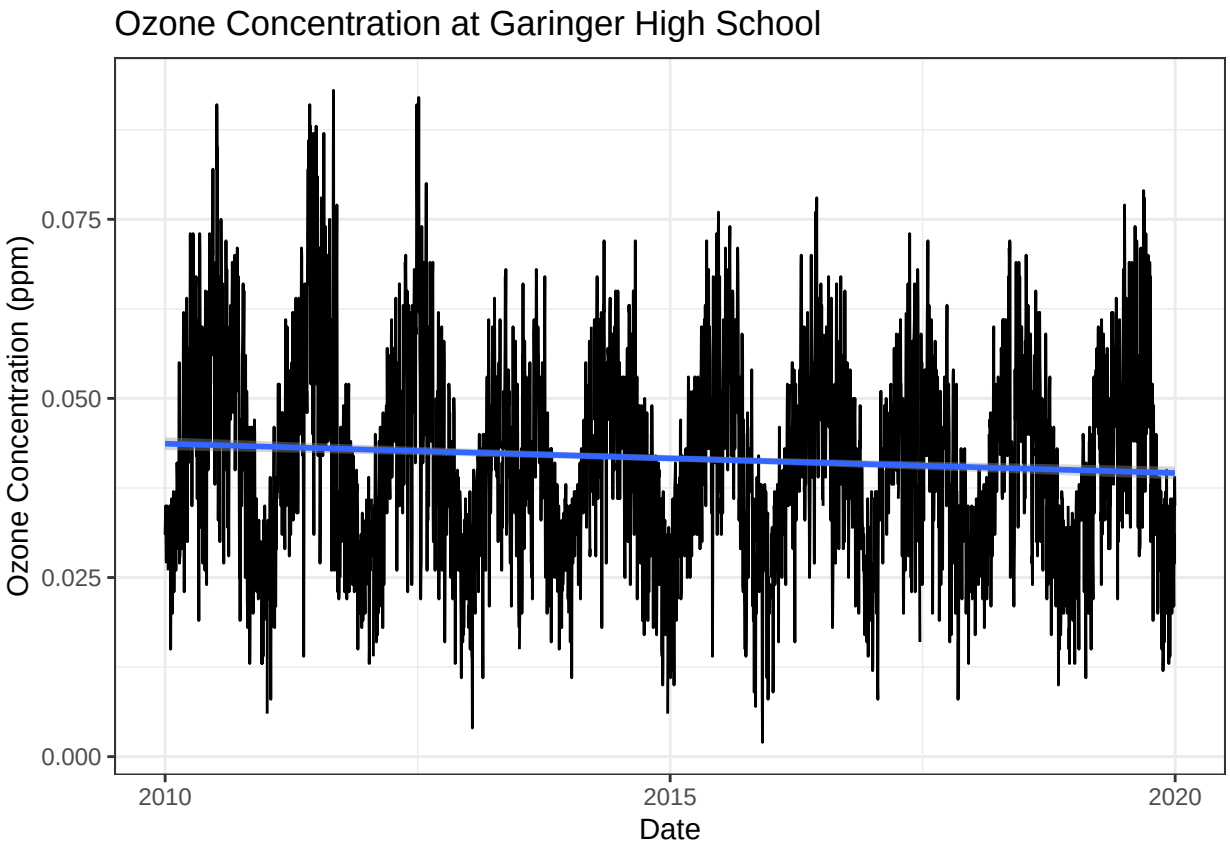
Visualize

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?

```r
#7
library(ggplot2)
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  labs(y = "Ozone Concentration (ppm)", title = "Ozone Concentration at Garinger High School") +
  geom_smooth(method = lm) #trend line
```

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

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



Answer: The plot suggests a trend of decreasing ozone concentration 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 <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
#linear interpolation
```

Answer: We used linear interpolation to replace missing values with values in a straight line between the previous and next values. Piecewise only takes into account one other value to base a missing value on, and spline doesn't use a straight line to estimate missing values. Because of this, linear is the best option.

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 %>%
  group_by(year(Date), month(Date)) %>% #grouping and summarizing will create new columns
  summarise(Daily.Max.8.hour.Ozone.Concentration = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  rename(Year = "year(Date)", Month = "month(Date)") #rename new columns
```

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

```r
#new date column
GaringerOzone.monthly <- mutate(GaringerOzone.monthly, Date = mdy(paste(Month, 1, Year,
                                                                        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.

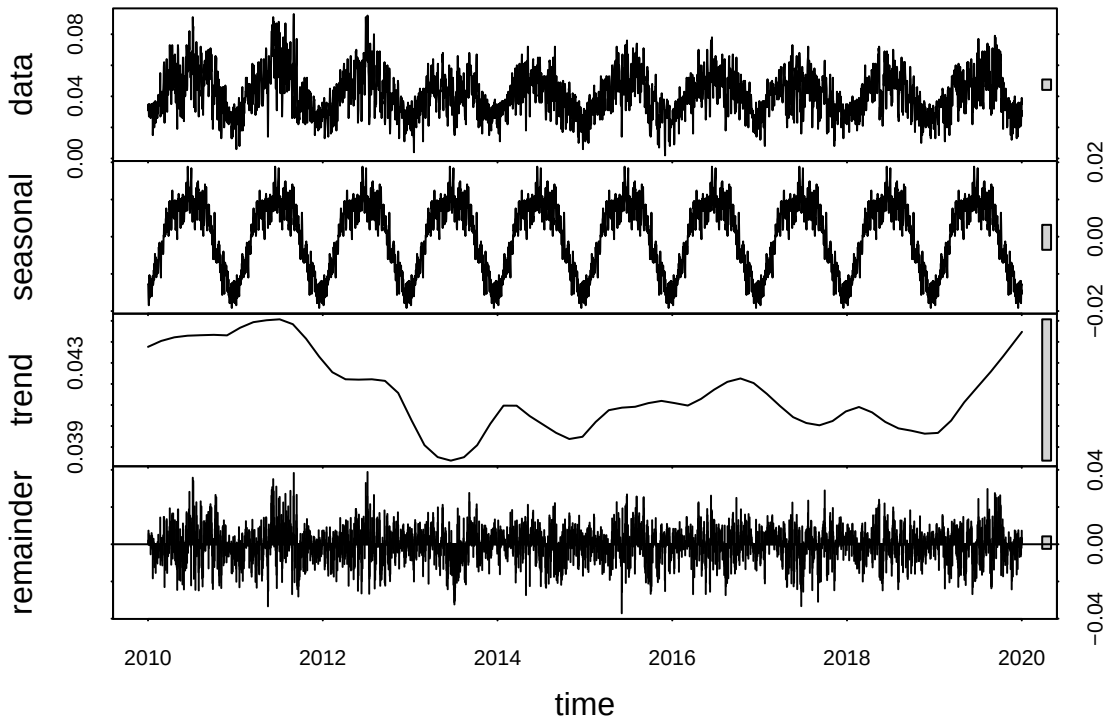
```r
#10
f_month <- month(first(GaringerOzone$Date))
f_year <- year(first(GaringerOzone$Date))

GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(f_year, f_month), frequency = 365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Daily.Max.8.hour.Ozone.Concentration,
                               start = c(f_year, f_month), frequency = 12)
```

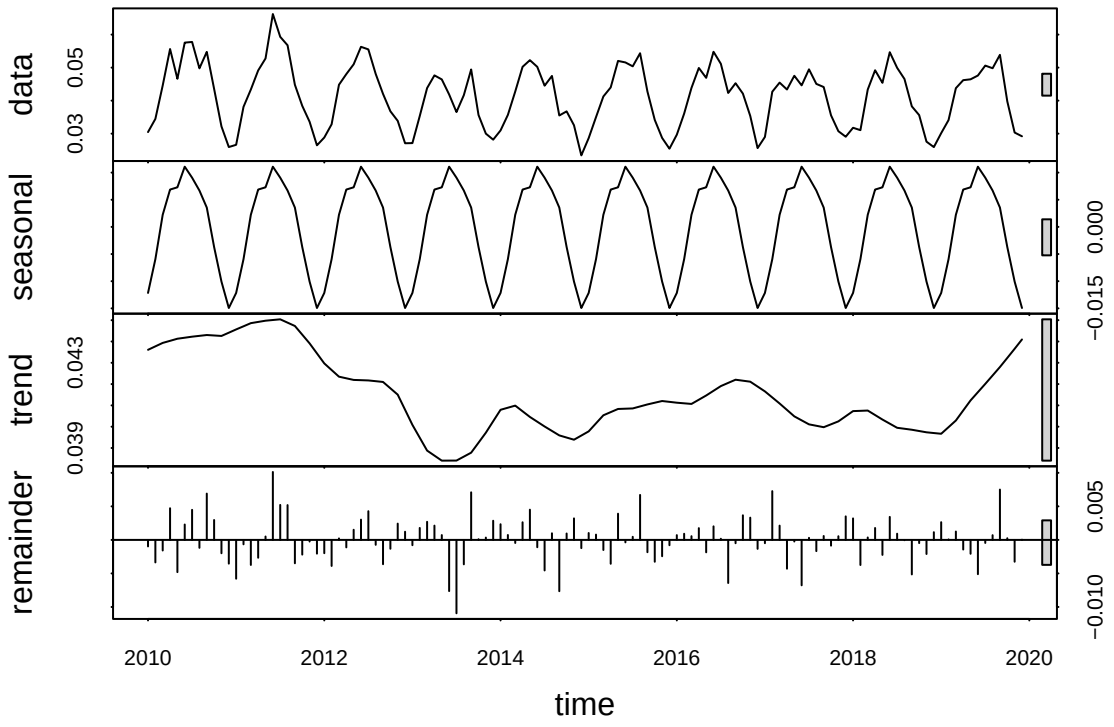
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

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

plot(GO_daily_decomp)
```




```
plot(GO_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
GO_monthly_mk <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(GO_monthly_mk)
```

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

```
GO_monthly_mk2 <- trend::smk.test(GaringerOzone.monthly.ts)
summary(GO_monthly_mk2)
```

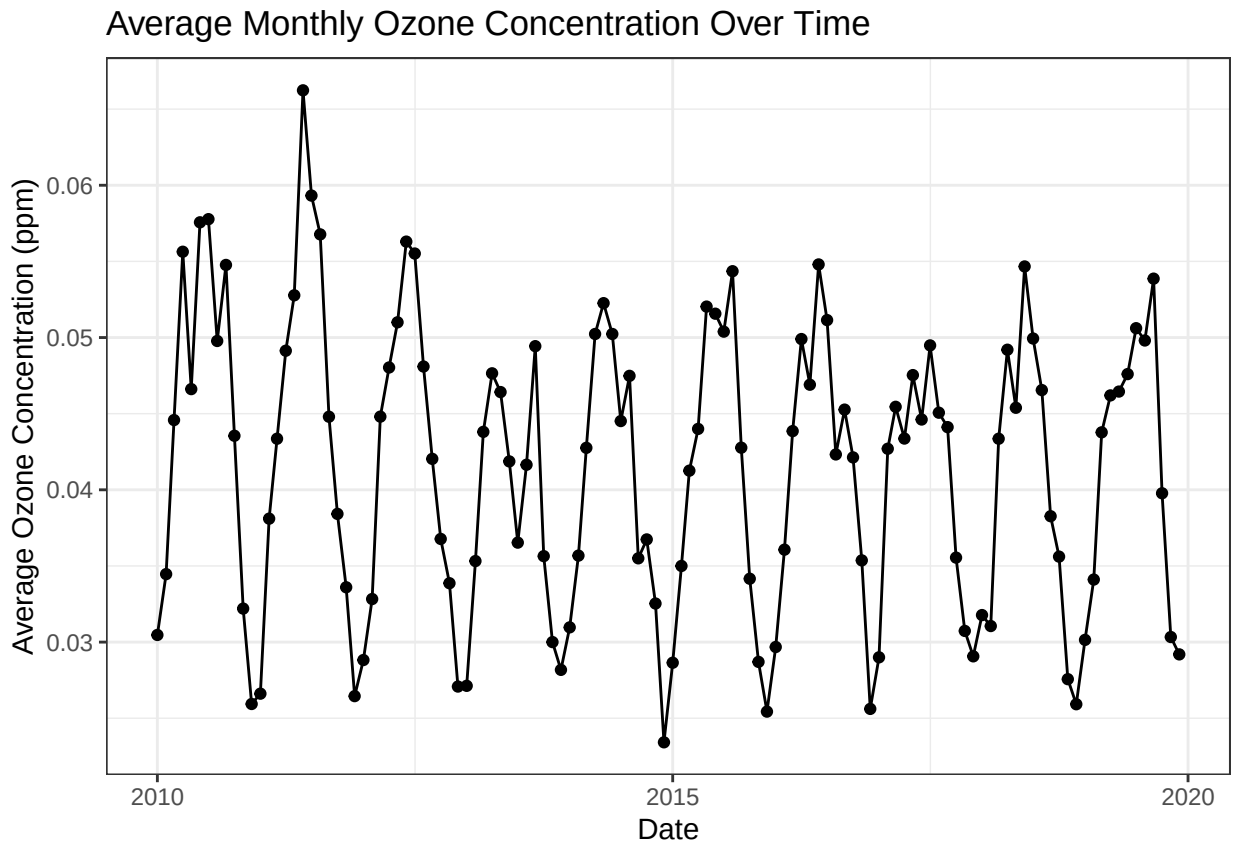
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                     S varS    tau      z Pr(>|z|)
```

```
## Season 1:   S = 0   15  125  0.333  1.252  0.21050
## Season 2:   S = 0   -1  125 -0.022  0.000  1.00000
## Season 3:   S = 0   -4  124 -0.090 -0.269  0.78762
## Season 4:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 5:   S = 0  -15  125 -0.333 -1.252  0.21050
## Season 6:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 7:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 8:   S = 0   -7  125 -0.156 -0.537  0.59151
## Season 9:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 10:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 11:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 12:   S = 0  11  125  0.244  0.894  0.37109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The seasonal Mann-Kendall is best because we expect there will be fluctuations in ozone depending on the time of year, and the Seasonal Mann-Kendall assumes seasonality.

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(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_point() +
  geom_line() + #both line and point
  labs(y = "Average Ozone Concentration (ppm)", title ="Average Monthly Ozone Concentration Over Time")
```



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: Using a Seasonal Mann-Kendall monotonic trend analysis on average monthly ozone over time, we got a tau of -0.143 and a p-value of 0.046724. The p-value of less than 0.05 means that our result is significant and the negative tau means that the trend is negative. Therefore, we can conclude that the ozone concentration is significantly decreasing.

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
GO_monthly_Components <- as.data.frame(GO_monthly_decomp$time.series[,1:3])
GO_monthly_non_seasonal_df <- GO_monthly_Components %>%
  mutate(Observed = GaringerOzone.monthly$Daily.Max.8.hour.Ozone.Concentration,
         Date = GaringerOzone.monthly$Date, Non_Seas = Observed - seasonal) #subtract out seasonal

#16
GO_monthly_non_seasonal.ts <- ts(GO_monthly_non_seasonal_df$Non_Seas,
                                 start = c(f_year, f_month), frequency = 12)

GO_monthly_ns_mk <- Kendall::MannKendall(GO_monthly_non_seasonal.ts)
#not seasonal mk because seasonality has been removed
summary(GO_monthly_ns_mk)
```

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

Answer: In the new Mann-Kendall test, the tau is -0.165 and the p-value is 0.0075402. The tau is slightly more negative than the last test, suggesting an even more decreasing trend. The p-value is also smaller than before. This means that the result is even more likely to be significant.