Assignment 8: Time Series Analysis

Jake Whisler

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

```
# Getting working directory, loading packages, and setting ggplot theme
getwd()
```

```
## [1] "/home/guest/EDE_Fall2023"
```

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

```r
#2 Importing and combining datasets
O3_2010 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
O3_2011 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
O3_2012 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
O3_2013 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
O3_2014 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
O3_2015 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
O3_2016 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
O3_2017 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
O3_2018 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
O3_2019 <- read.csv("~/EDE_Fall2023/Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")
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.

```r
# 3 Setting data class as Date for the date column
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

# 4 Filtering dataset so it only contains desired columns
GaringerOzone <-
  GaringerOzone %>%
  select("Date","Daily.Max.8.hour.Ozone.Concentration","DAILY_AQI_VALUE")

# 5 Creating a Days dataframe
Days <- as.data.frame(seq(from = as.Date("2010/1/1"), to = as.Date("2019/12/31"), by = "days"))
colnames(Days)[1] = "Date"

# 6 Joining the two dataframes together
GaringerOzone <- left_join(Days, GaringerOzone)
```

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

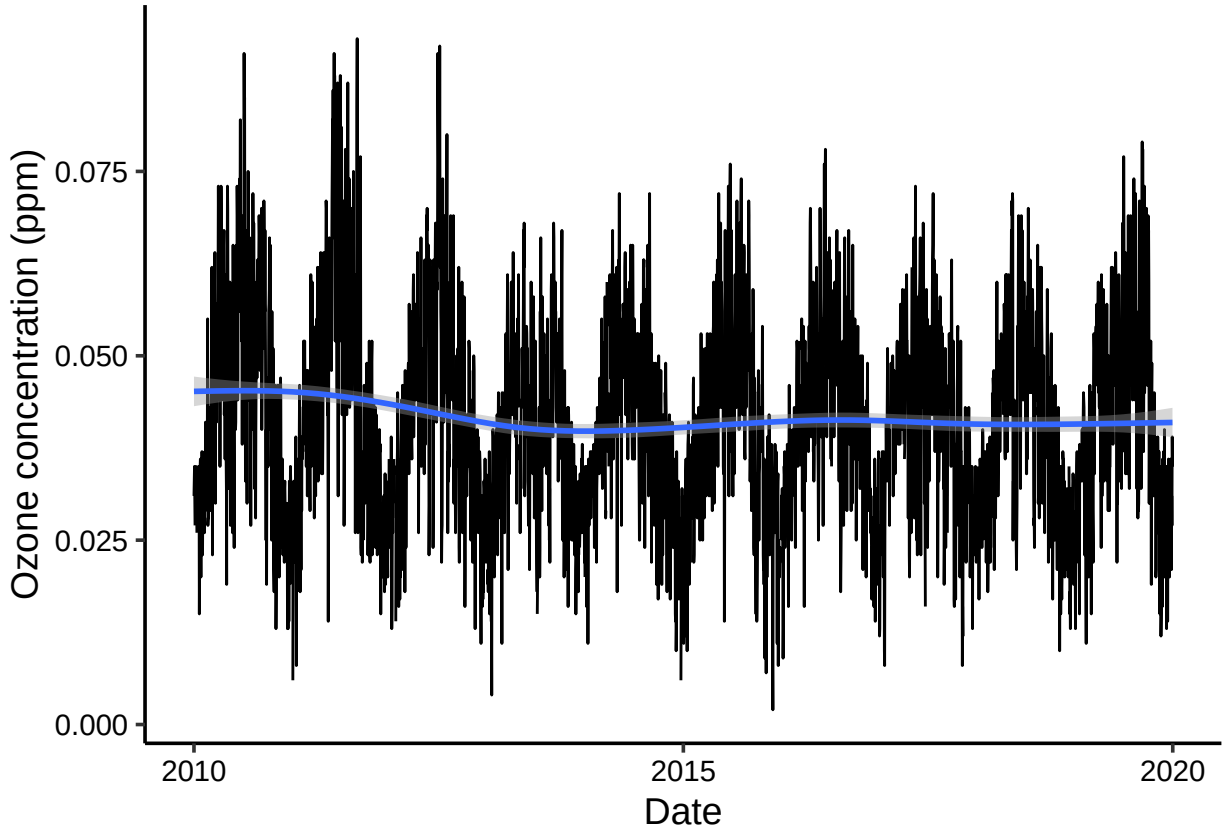
Visualize

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 Creating a line plot
Ozone_plot <- ggplot(GaringerOzone,
                     aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth() +
  xlab("Date") +
  ylab("Ozone concentration (ppm)")
print(Ozone_plot)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

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



Answer: My plot does suggest that Ozone changes over time on a yearly basis; Ozone concentration seems to minimize in the winter months and maximize during the summer months. However, I cannot detect a trend in changes from year-to-year over the decade of data collection.

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 Linear interpolation for missing data
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <-
  na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

Answer: We did not use piecewise constant or spline interpretation because at any given point, the data is following some linear trend, either negatively or positively, so lienar interpolation is the best for making appropriate assumptions about what those missing values might be.

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 Creating GaringerOzone.monthly with needed columns
GaringerOzone.monthly <-
  GaringerOzone %>%
  mutate(m = month(Date)) %>%
  mutate(Y = year(Date)) %>%
  group_by(m, Y) %>%
  summarise(meanO3 = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

```
GaringerOzone.monthly$Date <- as.yearmon(paste(GaringerOzone.monthly$m,
                                GaringerOzone.monthly$Y), "%m %Y")

GaringerOzone.monthly <- transform(GaringerOzone.monthly, Date = as.Date(Date, frac = 0))
class(GaringerOzone.monthly$Date)
```

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

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.

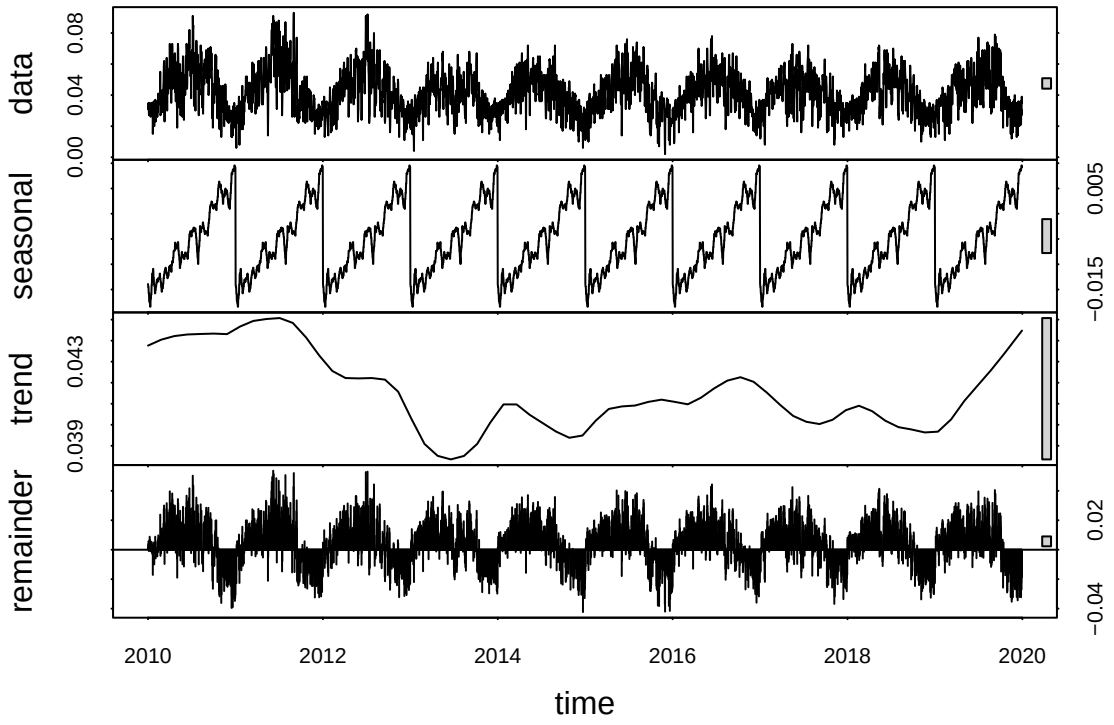
```
#10 Generating the two requested time series
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start = c(2010,1),
                             frequency = 365.25)
view(GaringerOzone.daily.ts)

f_month <- month(first(GaringerOzone.monthly$Date))
f_year <- year(first(GaringerOzone.monthly$Date))
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$meanO3,
                   start=c(f_year,f_month),
                   frequency=12)
```

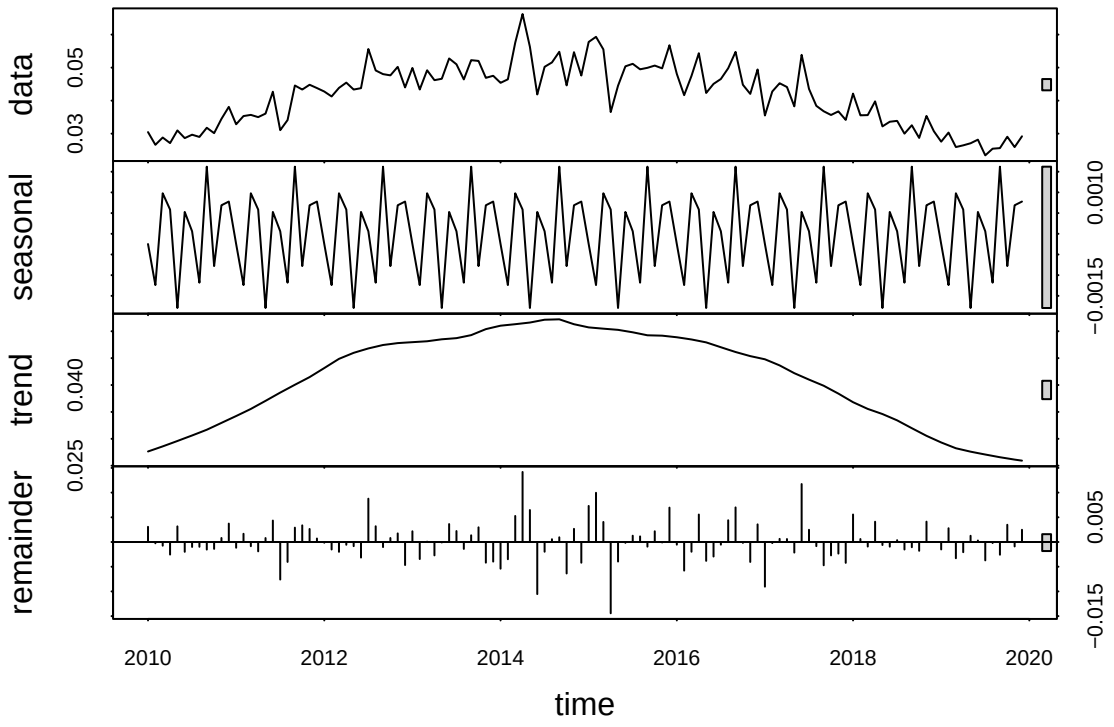
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```r
#11 Decomposing the time series objects and plotting them
GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
GaringerOzone.daily.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")

plot(GaringerOzone.daily.decomposed)
```




```r
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 Running the seasonal Mann-Kendall trend analysis
GaringerOzone.monthly.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(GaringerOzone.monthly.trend)
```

```
## Score =  -54 , Var(Score) = 1500
## denominator =  540
## tau = -0.1, 2-sided pvalue =0.16323
```

```
GaringerOzone.monthly.trend.2 <- smk.test(GaringerOzone.monthly.ts)
summary(GaringerOzone.monthly.trend.2)
```

```
##
##  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|)
```

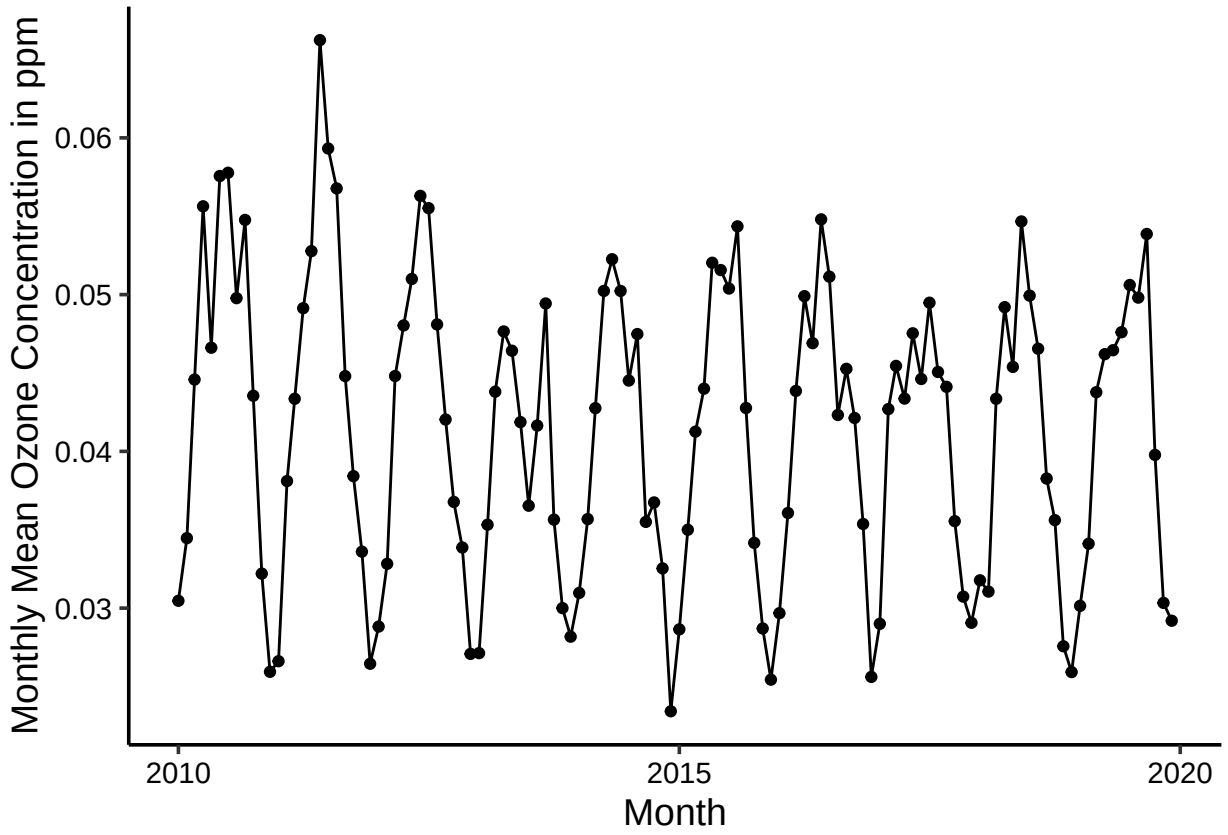
```
## Season 1:   S = 0    1  125  0.022  0.000  1.00000
## Season 2:   S = 0    5  125  0.111  0.358  0.72051
## Season 3:   S = 0   -3  125 -0.067 -0.179  0.85803
## Season 4:   S = 0    1  125  0.022  0.000  1.00000
## Season 5:   S = 0   -9  125 -0.200 -0.716  0.47427
## Season 6:   S = 0    1  125  0.022  0.000  1.00000
## Season 7:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 8:   S = 0   -3  125 -0.067 -0.179  0.85803
## Season 9:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 10:   S = 0 -11  125 -0.244 -0.894  0.37109
## Season 11:   S = 0 -15  125 -0.333 -1.252  0.21050
## Season 12:   S = 0  -5  125 -0.111 -0.358  0.72051
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The seasonal Mann-Kendall is most appropriate because the data is seasonal and non-parametric, as well as not missing any data points.

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.

```r
# 13 Creating the monthly mean plot
GaringerOzone.monthly.plot <-
  ggplot(GaringerOzone.monthly, aes(x = Date, y = meanO3)) +
  geom_point() +
  geom_line() +
  xlab("Month") +
  ylab("Monthly Mean Ozone Concentration in ppm")

print(GaringerOzone.monthly.plot)
```



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: I would say that, no, ozone levels at this location have not changed over the course of the 2010's. The tau value is a measure of the strength of the trend, but as seen with the seasonal Mann-Kendall test results, the tau value varies from positive to negative across various seasons (Season 1, tau = 0.022; Season 3: tau = -0.179.) The p-value is also returned as being 0.16, which is not a level of statistical significance that can confirm our trend.

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 Subtracting the seasonal component from GaringerOzone.monthly.ts
GaringerOzone.monthly.noSeasonal <- GaringerOzone.monthly.decomposed$time.series[,2]

#16 Running the Mann-Kendall test on the non-seasonal series
GaringerOzone.monthly.NoSeasonal.trend <- Kendall::MannKendall(GaringerOzone.monthly.noSeasonal)
summary(GaringerOzone.monthly.NoSeasonal.trend)
```

```
## Score =  -930 , Var(Score) = 194366.7
## denominator =  7140
## tau = -0.13, 2-sided pvalue =0.035101
```

Answer: In this new Mann-Kendall test, the 2-sided p-value is 0.03, which is statistically significant enough to confirm our trend, as opposed to the the seasonal test which was not. The tau value also changed from -0.1 to -0.13.