Assignment 8: Time Series Analysis

Elizabeth Good

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(here)
```

```
## here() starts at C:/Users/goode/OneDrive/Documents/Duke/ENV872_EDE/EDE_Fall2023
```

```
here()
```

```
## [1] "C:/Users/goode/OneDrive/Documents/Duke/ENV872_EDE/EDE_Fall2023"
```

```
new_theme <- theme_bw() +
  theme(axis.text = element_text(color = "navy",
                                 size = 12),
        axis.title = element_text(color = "gray40",
                                  size = 12),
        plot.title = element_text(color = "gray40",
                                  face = "bold",
                                  hjust = 0.5),
        legend.position = "bottom")

theme_set(new_theme)
```

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

Garinger2010 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
           stringsAsFactors = TRUE)
Garinger2011 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
           stringsAsFactors = TRUE)
Garinger2012 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
           stringsAsFactors = TRUE)
Garinger2013 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
           stringsAsFactors = TRUE)
Garinger2014 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
           stringsAsFactors = TRUE)
Garinger2015 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
```

```
           stringsAsFactors = TRUE)
Garinger2016 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
           stringsAsFactors = TRUE)
Garinger2017 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
           stringsAsFactors = TRUE)
Garinger2018 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
           stringsAsFactors = TRUE)
Garinger2019 <-
  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
           stringsAsFactors = TRUE)

GaringerOzone <-
  do.call("rbind", list(Garinger2010,
                        Garinger2011,
                        Garinger2012,
                        Garinger2013,
                        Garinger2014,
                        Garinger2015,
                        Garinger2016,
                        Garinger2017,
                        Garinger2018,
                        Garinger2019))
```

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$Date <-
  mdy(GaringerOzone$Date)

# 4

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

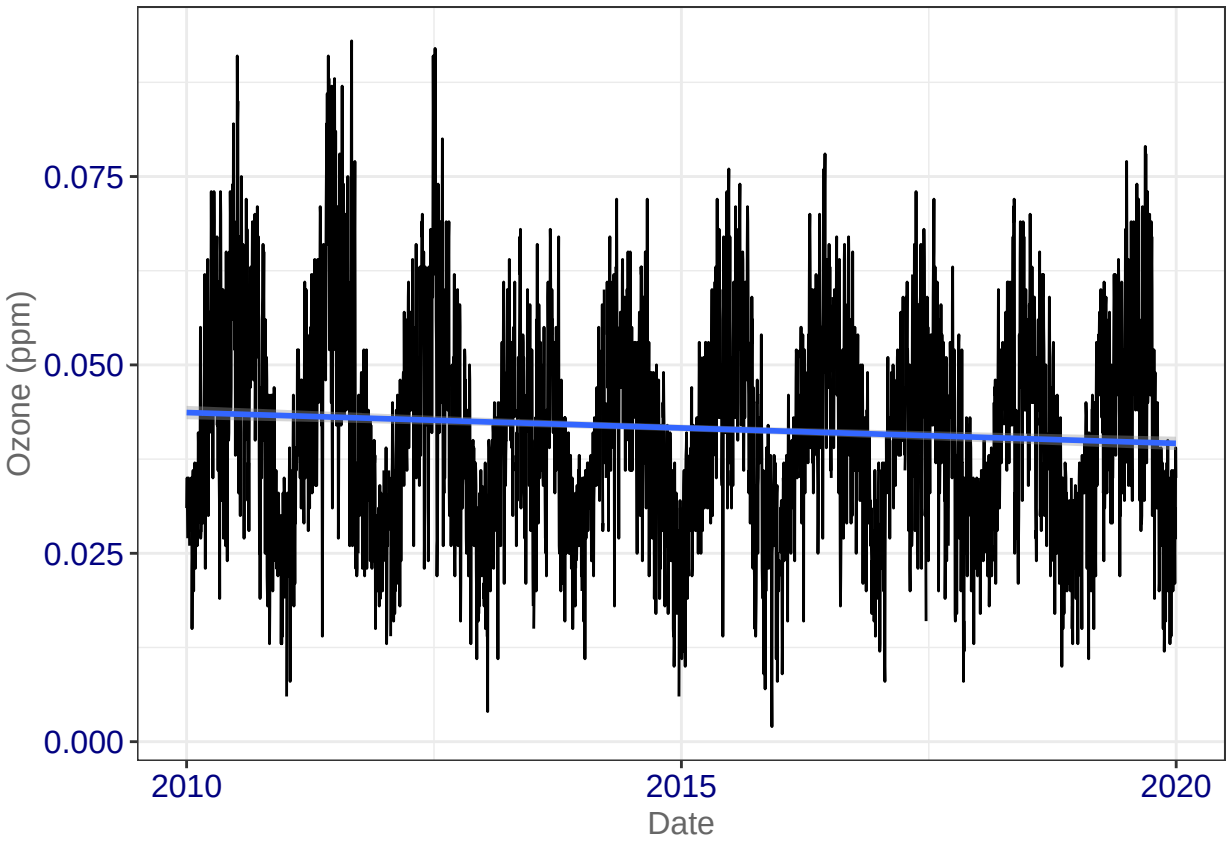
```r
# 5

Days <- as.data.frame(seq(ymd("2010-1-1"),
                          ymd("2019-12-31"),
                          by = "days"))

colnames(Days)[1] ="Date"

# 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_plot <-
ggplot(GaringerOzone,
       aes(x = Date,
           y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  ylab("Ozone (ppm)") +
  geom_smooth( method = lm)
print(O3_plot)
```

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

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



Answer: The line plot suggests that there might be a slight decrease in 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 <- GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration) )
```

Answer: A piecewise constant would have made the missing value equal to the next nearest measurement, while the linear interpolation draws a line between the value around the missing value to find the missing value. This would account for any trends in the data when assigning the missing value. The spline interpolation uses a quadratic function, which doesn't seem neccessary in our data set which doesn't seem to have a quadratic relationship.

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 %>%
  select(c(Date, Daily.Max.8.hour.Ozone.Concentration)) %>%
  mutate(month = month(Date)) %>%
  mutate(year = year(Date))

GaringerOzone.monthly <- GaringerOzone.monthly %>%
  group_by(month, year) %>%
  summarise(avg.ozone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

```
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(day = "01")

GaringerOzone.monthly$Date <-
  as.Date(paste(GaringerOzone.monthly$year,
                GaringerOzone.monthly$month,
                GaringerOzone.monthly$day,
                sep="-"), "%Y-%m-%d")
```

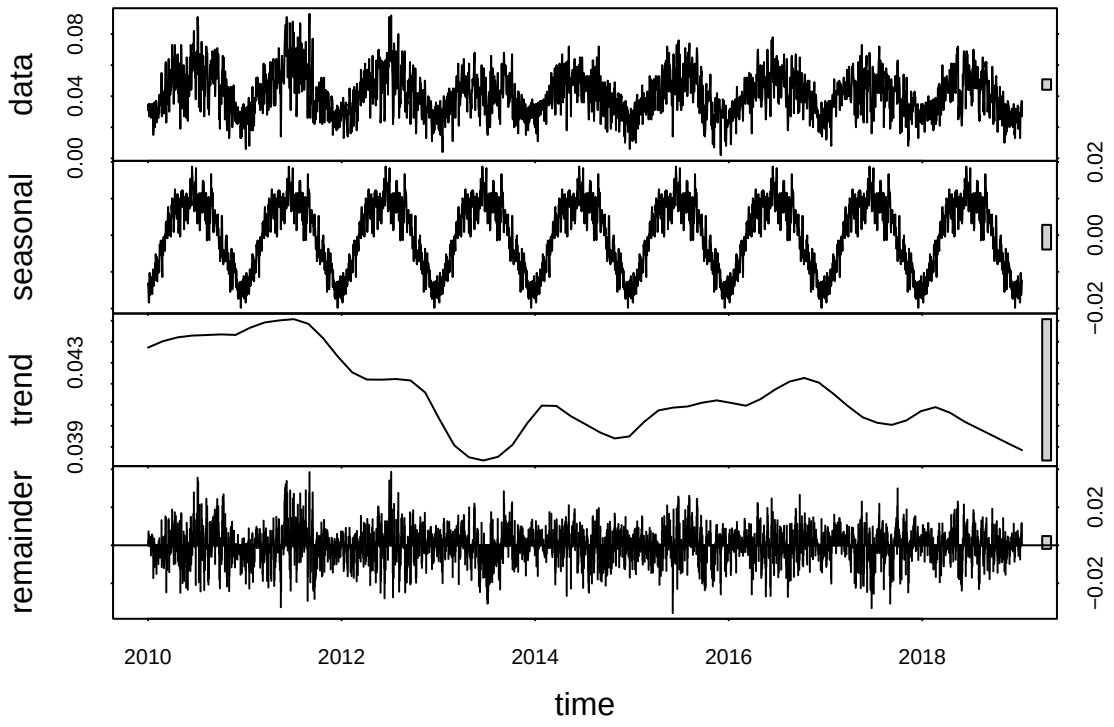
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of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

```
#10

GaringerOzone.daily.ts <-
  ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                              start = c(2010,1,1),
                              end = c(2019, 12,31),
                              frequency = 365)

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

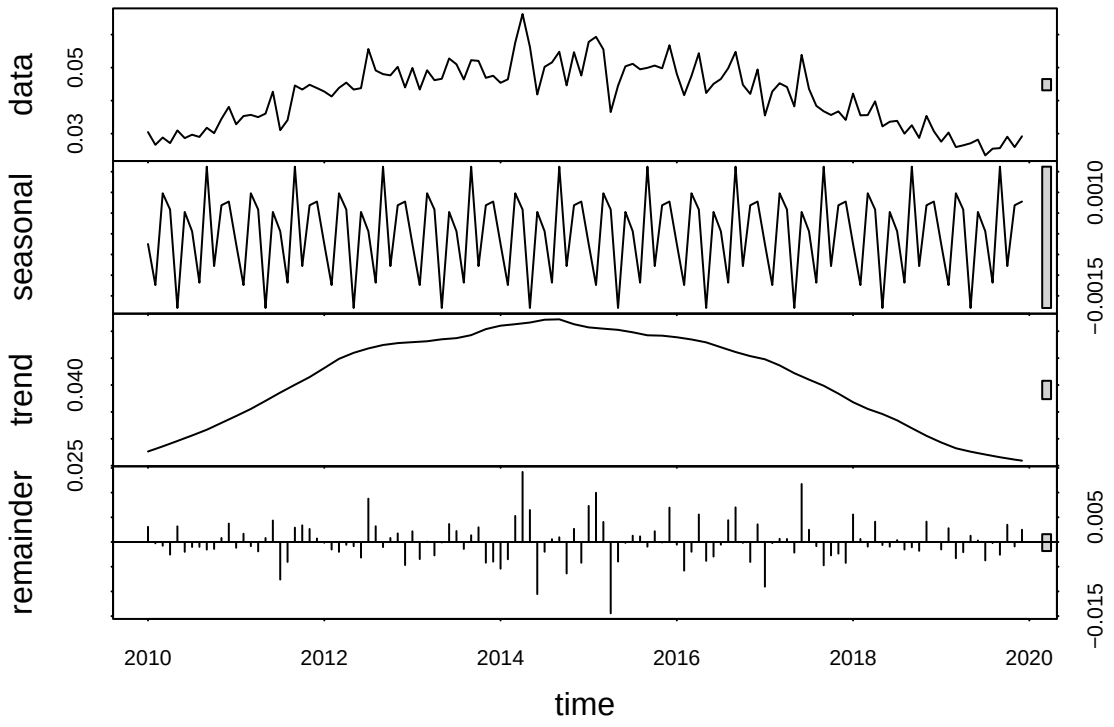
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11

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

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

plot(daily_decomposed)
```




```r
plot(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

# Run SMK test
monthly.ozone.smk <-
  Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results
monthly.ozone.smk
```

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

```
summary(monthly.ozone.smk)
```

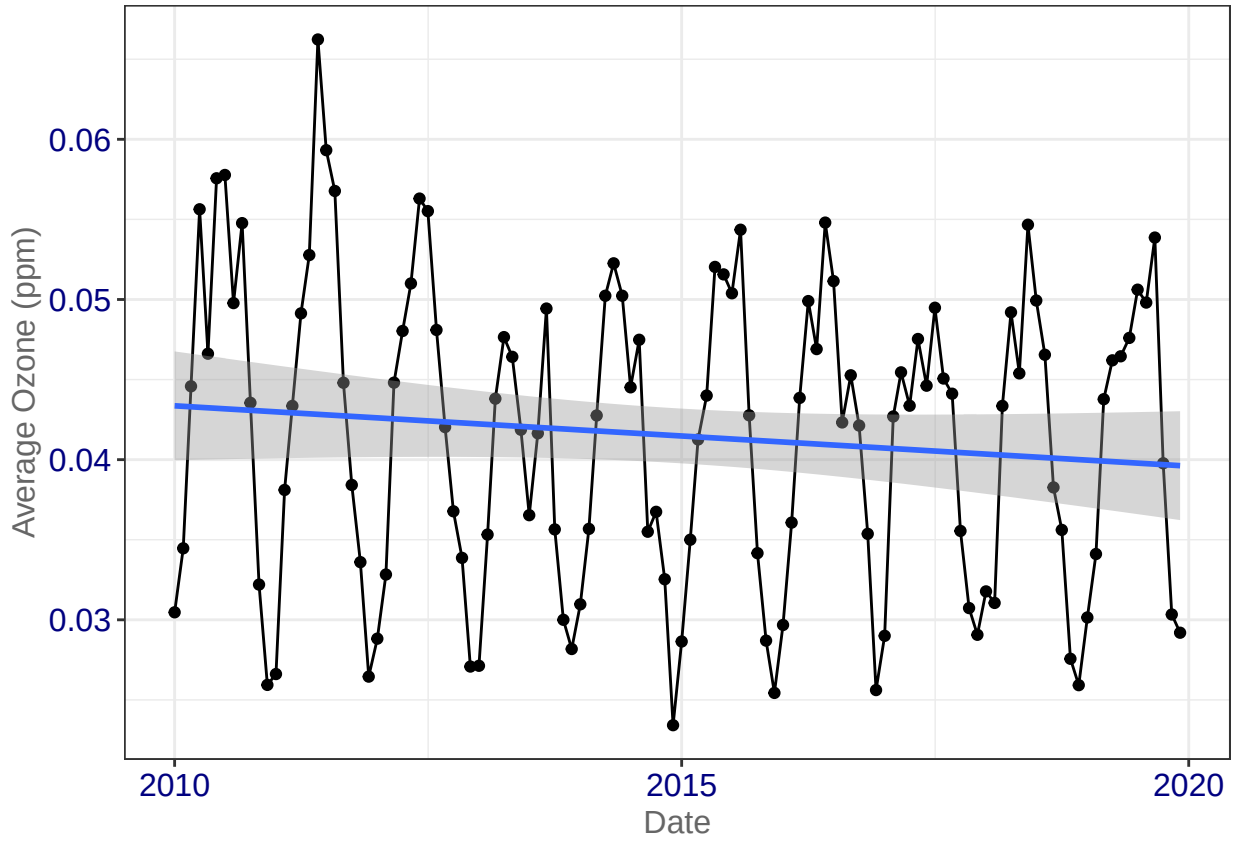
```
## Score =  -54 , Var(Score) = 1500
## denominator =  540
## tau = -0.1, 2-sided pvalue =0.16323
```

Answer: Ozone has a seasonal cycle, so we need a test that accounts for that, making the seasonal Mann-Kendall most appropriate.

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

monthly_ozone_plot <-
  ggplot(GaringerOzone.monthly,
       aes(x = Date, y = avg.ozone)) +
  geom_point() +
  geom_line() +
  ylab("Average Ozone (ppm)") +
  geom_smooth( method = lm )
print(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: Based on the results of the seasonal Mann-Kendall test, ozone concentrations have not changed significanly over the 2010s at this station (pvalue = 0.16323).

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.

```r
#15

ozone_monthly_components <-
  as.data.frame(monthly_decomposed$time.series[,1:3])

ozone_monthly_components <-
  mutate(ozone_monthly_components,
         Observed = GaringerOzone.monthly$avg.ozone,
         Date = GaringerOzone.monthly$Date)

ozone_monthly_components <-
  mutate(ozone_monthly_components,
          non.seasonal =
            ozone_monthly_components$Observed -
            ozone_monthly_components$seasonal)

GaringerOzone.monthly.nonseasonal.ts <-
  ts(ozone_monthly_components$non.seasonal,
                               start = c(2010,1),
                               end = c(2019,12),
                               frequency = 12)

#16

# Run MK test
monthly.ozone.mk <-
  Kendall::MannKendall(GaringerOzone.monthly.nonseasonal.ts)

# Inspect results
monthly.ozone.mk
```

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

```r
summary(monthly.ozone.mk)
```

```
## Score =  -718 , Var(Score) = 194366.7
## denominator =  7140
## tau = -0.101, 2-sided pvalue =0.10388
```

Answer: With the seasonal component removed from the monthly ozone data, the pvalue decreases from 0.16323 to 0.10388. While this pvalue is still too high to say that there is a significant trend in the data, it is lower than the seasonal Mann-Kendall pvalue and indicates that removing the seasonal component shows more of a trend in ozone concentrations in the 2010s.