Assignment 8: Time Series Analysis

Jonathan Joyner

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] "C:/Users/jbjoy/OneDrive/Documents/Grad School/Fall 2023/Environ 872/EDE_Fall2023"
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.4.3     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0

## -- 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(trend)
library(zoo)
```

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

```
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
Ozone2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                      stringsAsFactors = TRUE)
```

```
Ozone2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                      stringsAsFactors = TRUE)
Ozone2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
                      stringsAsFactors = TRUE)
GaringerOzone <- rbind(
  Ozone2010,
  Ozone2011,
  Ozone2012,
  Ozone2013,
  Ozone2014,
  Ozone2015,
  Ozone2016,
  Ozone2017,
  Ozone2018,
  Ozone2019)
```

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 ChatGPT/AI Assistance used for this problem
Days<-as.data.frame(seq(
  as.Date("2010-01-01"),
  as.Date("2019-12-31"),
  by = "days"))
colnames(Days)<-"Date"
# 6
GaringerOzone <- left_join(Days, GaringerOzone, by = "Date")
```

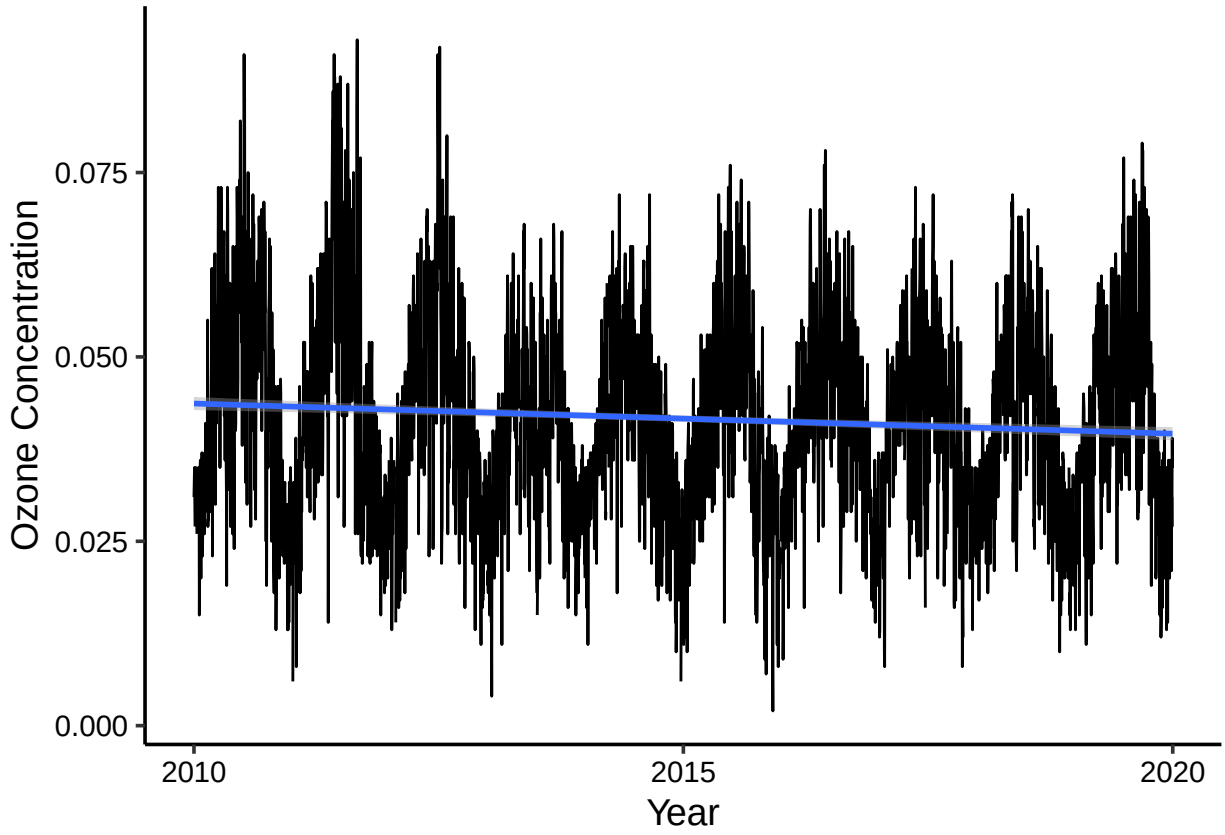
Visualize

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
ggplot(GaringerOzone,aes(x=Date,y=Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  geom_smooth(method=lm)+
  labs(x="Year",
       y="Ozone Concentration")
```

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

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



Answer: The trend line is relatively stable over the past ten years, but has a slight downward trajectory.

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 ChatGPT/AI Assistance used for this problem
GaringerOzone$Daily.Max.8.hour.Ozone.Concentration <-
  zoo::na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

Answer: Spline interpolation and Piecewise Constant are too nuanced to be of use in a large dataset with small gaps like this one. Linear interpolation will connect the adjacent data points and provide a simpler fill for the missing data.

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  ChatGPT/AI Assistance used for this problem
GaringerOzone.monthly<-
  GaringerOzone %>%
  mutate(
    Month=month(Date),
    Year=year(Date)) %>%
  group_by(Year,Month) %>%
  summarize(mean_ozone = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  ungroup() %>%
  mutate(Date = make_date(Year,Month))
```

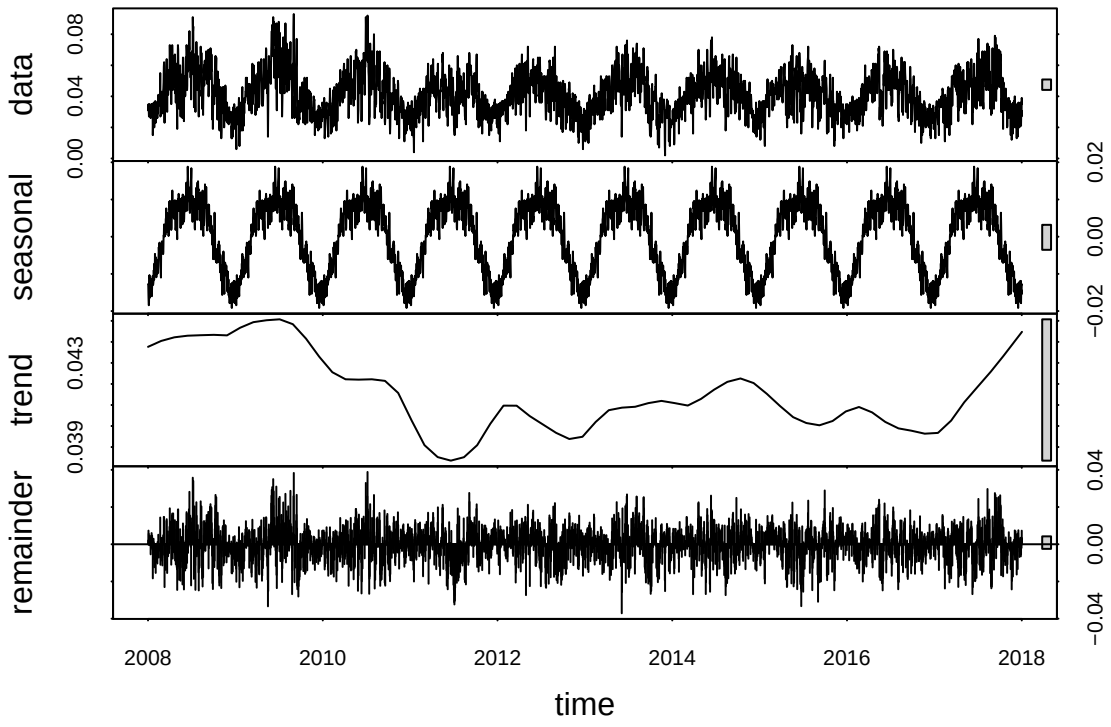
```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

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.

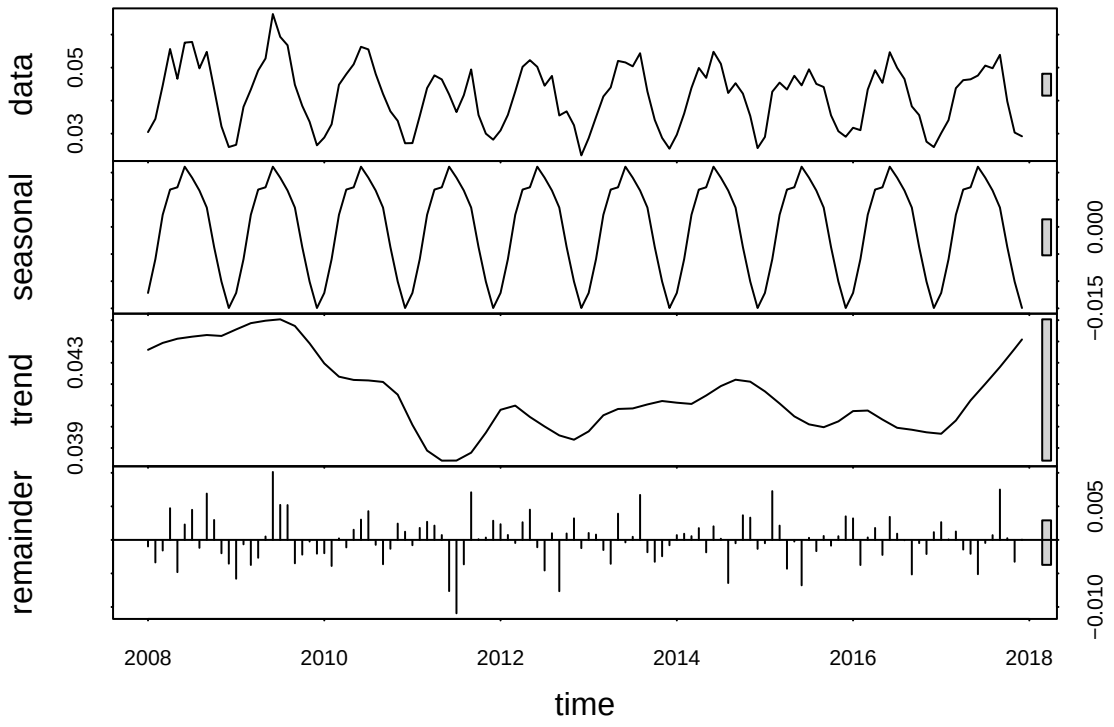
```
#10
GaringerOzone.daily.ts<-
  ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
     start=c(2010-01-01),
     frequency=365)

GaringerOzone.monthly.ts<-
  ts(GaringerOzone.monthly$mean_ozone,
     start=c(2010-01-01),
     frequency=12)
```

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

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




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



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

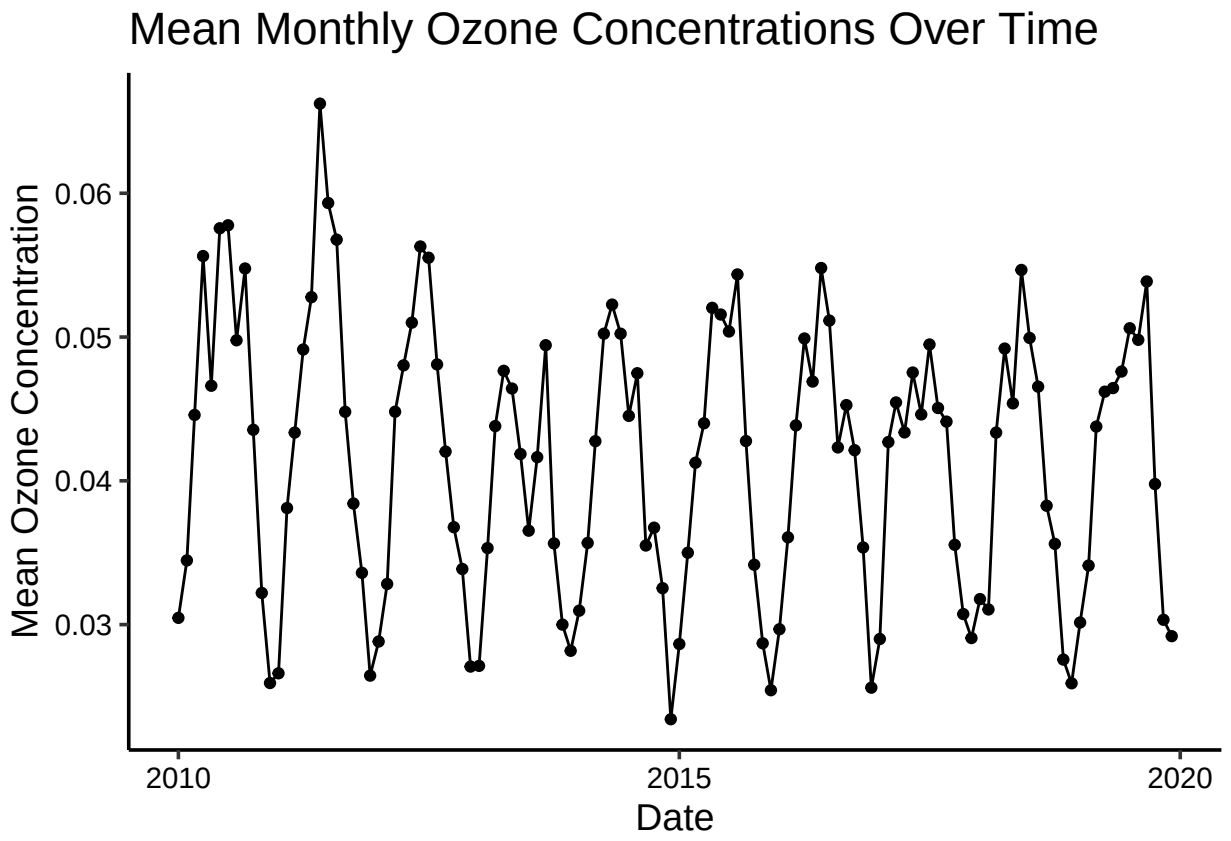
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: Because we are dealing with a monthly dataset over a decade, the seasonal Mann-Kendall test will be able to detect nuances in trend normally caused by seasonal change in the environment.

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  ChatGPT/AI Assistance used for this problem
MeanOzonePlot<-
  ggplot(GaringerOzone.monthly,aes(x=Date,y=mean_ozone)) +
  geom_point()+
  geom_line()+
  labs(x = "Date", y = "Mean Ozone Concentration") +
```

```
  ggtitle("Mean Monthly Ozone Concentrations Over Time")
plot(MeanOzonePlot)
```



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: Have ozone concentrations changed over the 2010s at this station? In examining the various plots from #s 7, 11, and 13, one can see that ozone levels are relatively stable over the decade. There is a slight downward trajectory in the initial couple years after 2010 for each, but none show a significant change with most volatility caused by seasonality. Much of the overall returns to 2010 levels by 2020 in all plots.

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  ChatGPT/AI Assistance used for this problem
seasonal_component <- Monthly.stl$time.series[, "seasonal"]
GaringerOzone_deseasonalized_ts <- GaringerOzone.monthly.ts - seasonal_component
#16
FinalMKTrend<- Kendall::MannKendall(GaringerOzone_deseasonalized_ts)
summary(Monthly.trend)
```

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

```
summary(FinalMKTrend)
```

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

Answer: The Mann Kendall test on the deseasonalized ozone time series has a significantly smaller pvalue which means the likelihood of the original research question being correct is low. The results show that most variation came from seasonality and strengthens the findings from previous graphing.