Assignment 8: Time Series Analysis

Simon Heinberg

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(here)
```

```
## here() starts at /Users/simonheinberg/Desktop/Duke_Coursework/Envirnmental_Data_Exploration/EDE_Fall
```

```
library(ggthemes)
library(dplyr)
library(Kendall)

getwd()
```

```
## [1] "/Users/simonheinberg/Desktop/Duke_Coursework/Envirnmental_Data_Exploration/EDE_Fall2023"
```

```
Simon_theme <- theme_base() +
  theme(
    legend.key = element_rect(
      color='purple',

    ),
    plot.background = element_rect(
      color='blue',
      fill = 'grey'
    ),
    plot.title = element_text(
      color='blue'
    ),
    panel.grid.major = element_line(color="grey44")
    ,
    legend.position="right")
theme_set(Simon_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.

```
#2

Ozone_2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
Ozone_2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
Ozone_2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
Ozone_2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
Ozone_2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
Ozone_2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
Ozone_2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
```

```
Ozone_2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
Ozone_2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
Ozone_2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")

GaringerOzone <- rbind(Ozone_2010, Ozone_2011, Ozone_2012,
Ozone_2013, Ozone_2014, Ozone_2015, Ozone_2016, Ozone_2017, Ozone_2018, Ozone_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.

```
# 3
GaringerOzone$Date <- mdy(GaringerOzone$Date)
# 4
GaringerOzone_select <- select(GaringerOzone, Date,
Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
# 5
Days <- as.data.frame(seq.Date(from= as.Date("2010-01-01"), to= as.Date("2019-12-31"), by= 'day'))

colnames(Days) <-c("Date")
# 6
Garinger_joined <- Days %>%
  left_join(GaringerOzone_select, by="Date")

#I already used the name GaringerOzone when I imported the data in question 2,
#so I am using a different name here
```

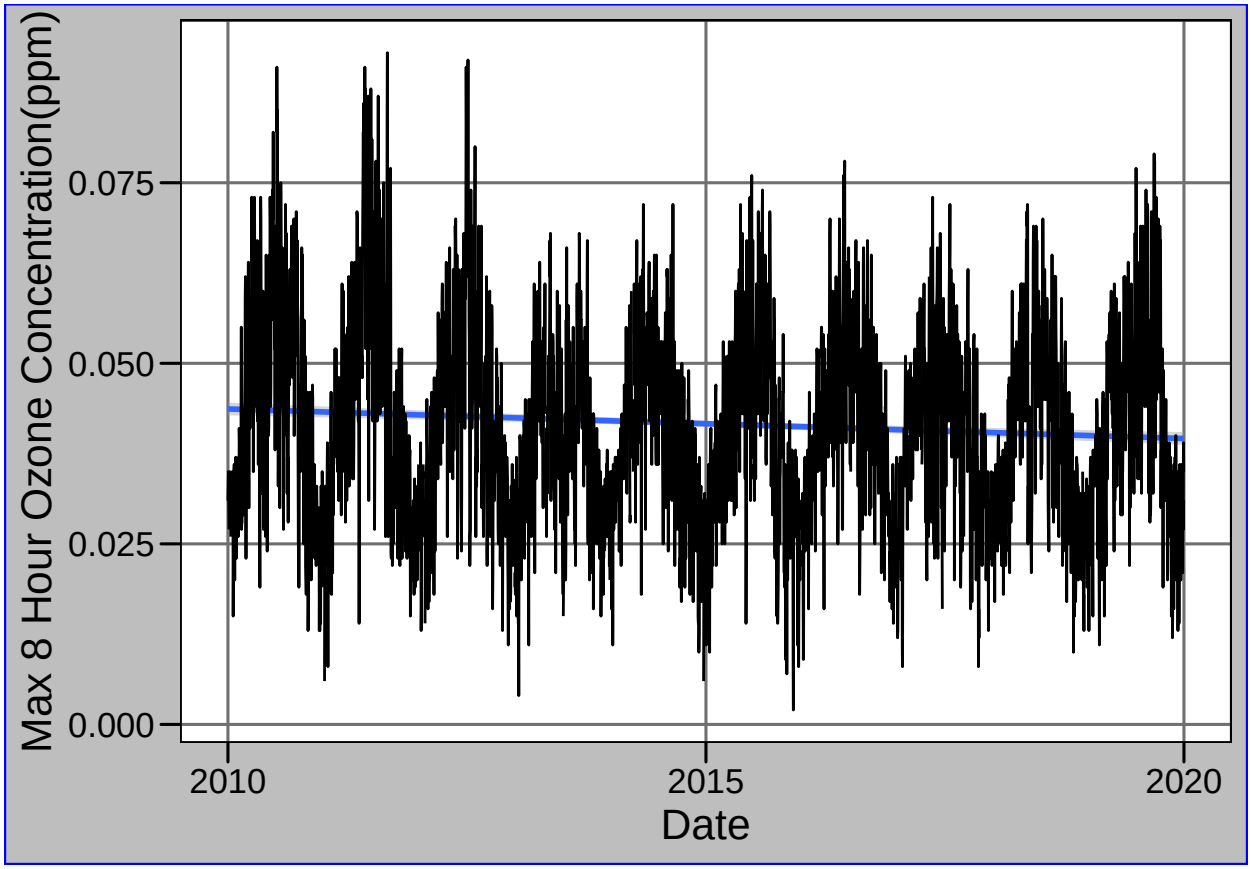
Visualize

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?

```
#7

Garinger_plot <-ggplot(Garinger_joined, aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration))+
  geom_smooth(method=lm)+
  geom_line()+
  ylab("Max 8 Hour Ozone Concentration(ppm)")

Garinger_plot
```

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

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



Answer: The trendline suggests that ozone concentration is decreasing 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
Garinger_filled <- Garinger_joined %>%
mutate(Daily.Max.8.hour.Ozone.Concentration=zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: We didn't use the piecewise constant since the piecewise constant works best when there is no linear trend to the data, and in this case the data has a decreasing linear trend. We didn't use the spline interpolation since it works best when the trend is quadratic rather than linear, and this dataset has a linear trend.

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 <- Garinger_filled %>%
  mutate(month=month(Date))%>%
  mutate(year=year(Date))%>%
   group_by(year, month) %>%
  summarize(mean_ozone=mean(Daily.Max.8.hour.Ozone.Concentration))%>%
      mutate(Date=as.Date(paste(year, month, "01", sep="-"), "%Y-%m-%d"))
```

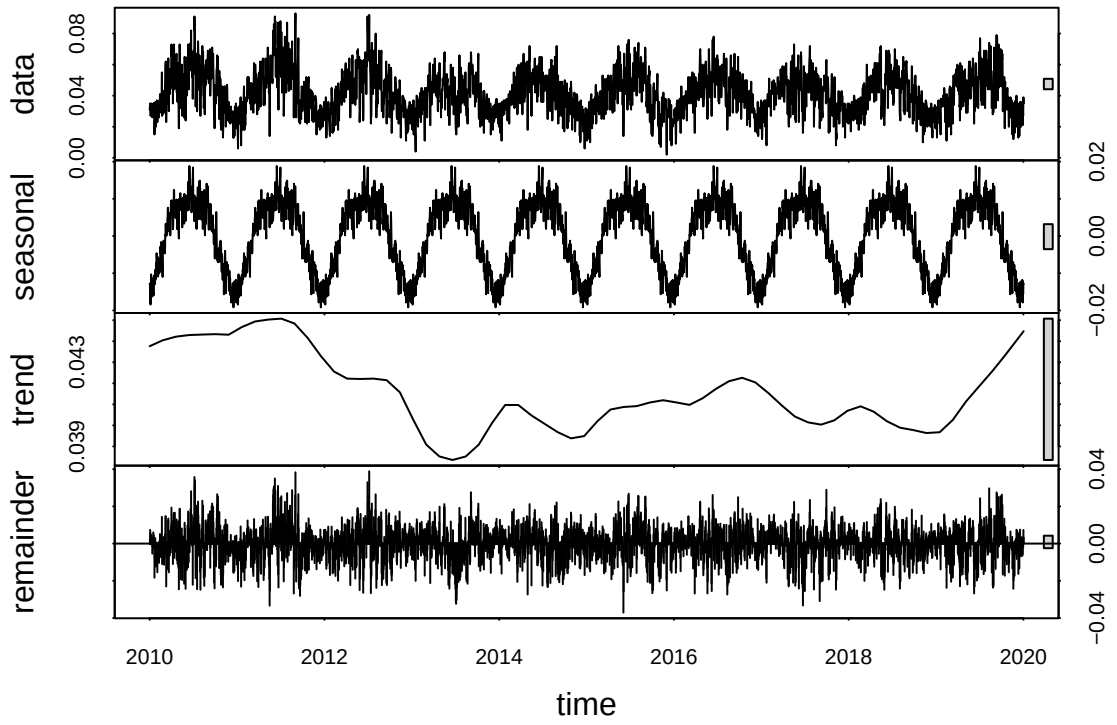
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

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.

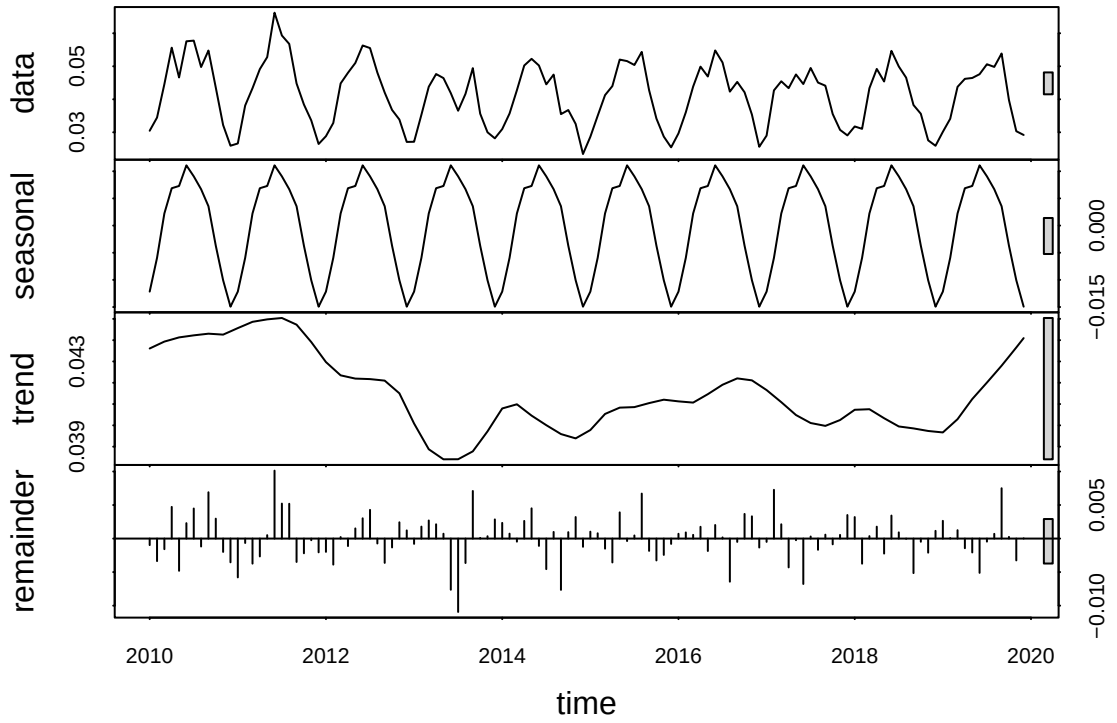
```
#10

f_date <- date(first(Garinger_filled$Date))


GaringerOzone.daily.ts <-
ts(Garinger_filled$Daily.Max.8.hour.Ozone.Concentration,
   start=2010, frequency=365)

f_month <-month(first(GaringerOzone.monthly$Date))
f_year <-year(first(GaringerOzone.monthly$Date))

GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$mean_ozone,
     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.

```
#11

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

GaringerOzone.monthly.decomposed <-stl(GaringerOzone.monthly.ts, s.window="periodic")

plot(GaringerOzone.daily.decomposed)
```




```
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

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

GaringerOzone.monthly.trend
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

```
summary(GaringerOzone.monthly.trend)
```

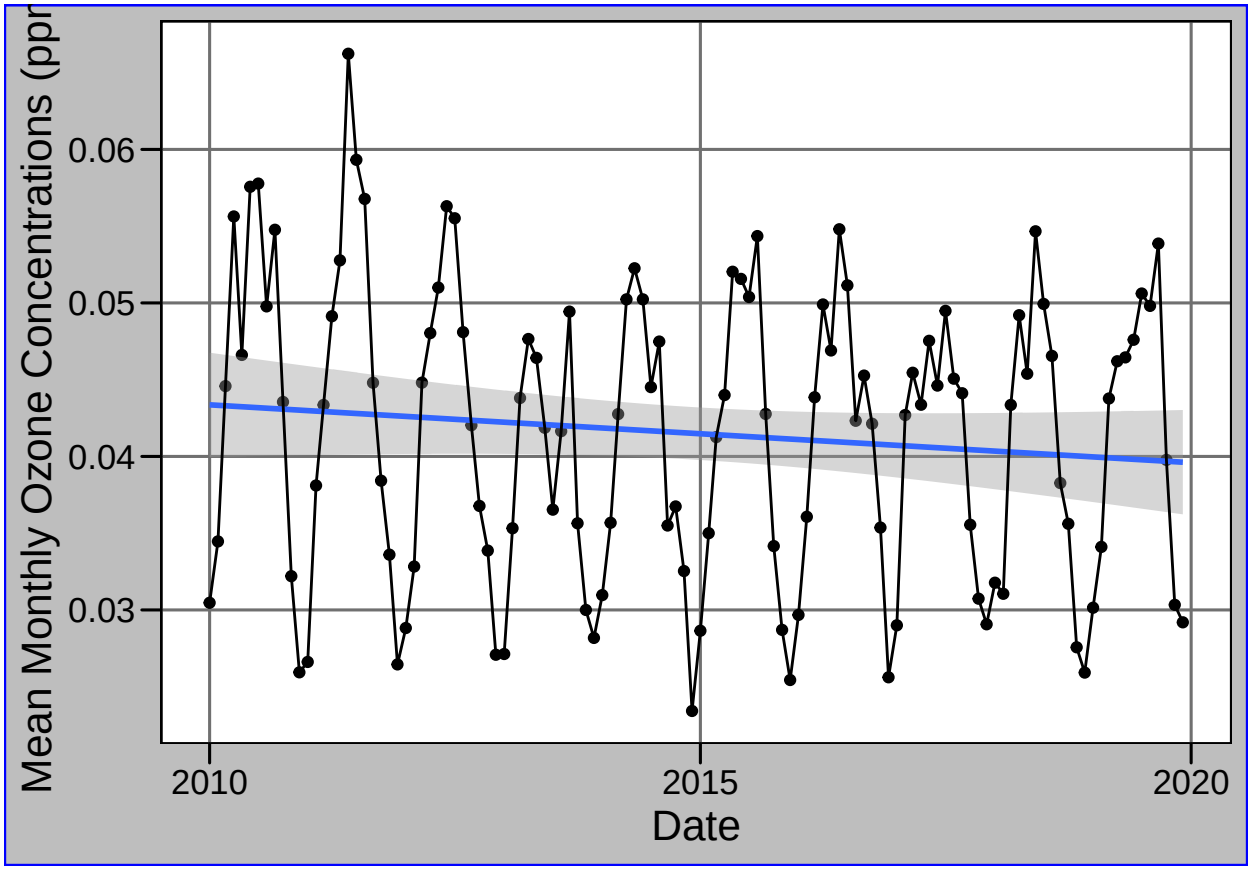
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: The seasonal Mann-Kendall test is most appropriate since our trend is not linear and because our dataset is seasonal.

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

GaringerOzone.monthly.plot <- ggplot(GaringerOzone.monthly, aes(x=Date, y=mean_ozone))+
  geom_point()+
  geom_smooth(method=lm)+
  geom_line()+
  ylab("Mean Monthly Ozone Concentrations (ppm)")

GaringerOzone.monthly.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: Our p-value is significant (p=.046724) so we can reject the null hypothesis that mean monthly ozone values have not changed during the 2010s. The negative tau value (-.143) suggests that the trend is negative, meaning that mean monthly ozone concentration have decreased during the 2010s. This is further supported by the negative slope of the trend line in our 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.

```
#15

GaringerOzone.monthly.dataframe <- as.data.frame(GaringerOzone.monthly.decomposed$time.series[,1:3])

GaringerOzone.monthly.components <- GaringerOzone.monthly.dataframe %>%
mutate(mean_ozone=GaringerOzone.monthly$mean_ozone,
        Date=GaringerOzone.monthly$Date,
       mean_nonseasonal=(mean_ozone-seasonal))

ff_date <- date(first(GaringerOzone.monthly.components$Date))
```

```
GaringerOzone.nonseasonal.ts <- ts(GaringerOzone.monthly.components$mean_nonseasonal, start=2010, frequ
```

```
#16

GaringerOzone.nonseasonal.trend <-Kendall::MannKendall(GaringerOzone.nonseasonal.ts)

GaringerOzone.nonseasonal.trend
```

```
## tau = -0.165, 2-sided pvalue =0.0075402
```

```
summary(GaringerOzone.nonseasonal.trend)
```

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

Answer: Like the Seasonal Mann Kendall, the non-seasonal Mann Kendall yielded a significant p value (.0075402) and a negative tau value (-.165). This suggests that we can reject the null hypothesis that mean monthly ozone values have not changed during the 2010s and that the trend is negative. However, the lower p value shows that after removing the seasonal component, we have stronger evidence to reject the null hypothesis. The tau value is also lower (although greater in absolute value) after removing the seasonal component. This suggests that removing the seasonal component generated a stronger negative trend.