Assignment 8: Time Series Analysis

Griffin Bird

Spring 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

```
#1
library(knitr)
library(tidyverse)
library(lubridate)
library(here)
library(cowplot)
library(ggplot2)
library(ggthemes)
library(zoo)
library(trend)
library(Kendall)

here
getwd()

mytheme <- theme_wsj() + theme(plot.title = element_text(hjust = 0.5),
   panel.grid.minor = element_line(color = 2,
   size = 0.25,
```

```
   linetype = 1),
  legend.position = "bottom",
  axis.text = element_text(face="plain", size = 12),
  axis.title=element_text(size=14),
  title = element_text(size=14))

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.

```
#2

Garinger <- list.files(path="~/EDASpring2023/Data/Raw/Ozone_TimeSeries", full.names = TRUE) %>%
  lapply(read_csv) %>%
  bind_rows
```

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.

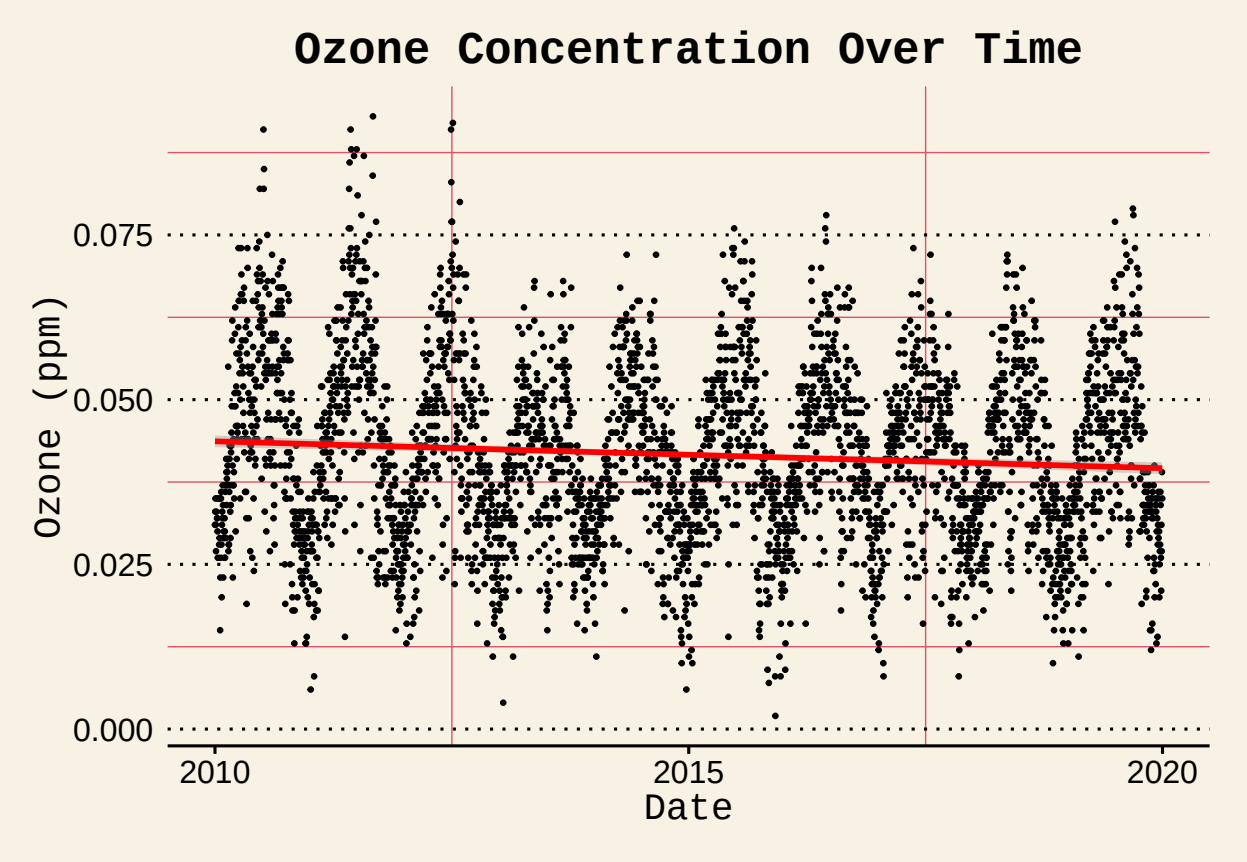
```
#3
Garinger$Date <- as.Date(Garinger$Date, format = "%m/%d/%Y")

#4
Garinger_Filter <- Garinger %>%
  select(Date, `Daily Max 8-hour Ozone Concentration`, DAILY_AQI_VALUE)

#5
Days <- as.data.frame(x= seq(as.Date("2010/1/1"), as.Date("2019/12/31"), by = "day")) %>%
  rename("Date" = 1)

#6
GaringerOzone <- left_join(x=Days, y=Garinger_Filter) %>%
  rename("Daily8hrMax" = 2)
```

```
## 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?

```
#7
ozone.plot <- ggplot(GaringerOzone, aes(x = Date, y = Daily8hrMax)) +
  labs(title = "Ozone Concentration Over Time",
       x = "Date",
       y = "Ozone (ppm)") +
  geom_point(size = 0.5) +
   geom_smooth(method = lm, color = "red")

print(ozone.plot)
```



Answer: The plot suggests a negative trend in 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_Interp <- GaringerOzone %>%
  mutate(Daily8hrMax = na.approx(Daily8hrMax))
```

Answer: We used linear interpolation here because, as illustrated by the trend line in the plot above, the trend in ozone concentration looks to be roughly linear - its a good fit if we want to estimate some missing data points. We didn't use piecewise constant or spline interpolation because neither is a great fit for the data we have. Piecewise constant would assign missing values the nearest data value (unsophisticated, linear interp is easier and better than this) and spline interpolation would use higher-order polynomials to interpolate values which seems a bit sophisticated for the data we're dealing with. Linear interpolation isn't too basic or too sophisticated and seems a good fit based on the scatterplot of our data, we could always fine tune the interpolation if the linear interpolations didn't make sense.

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)

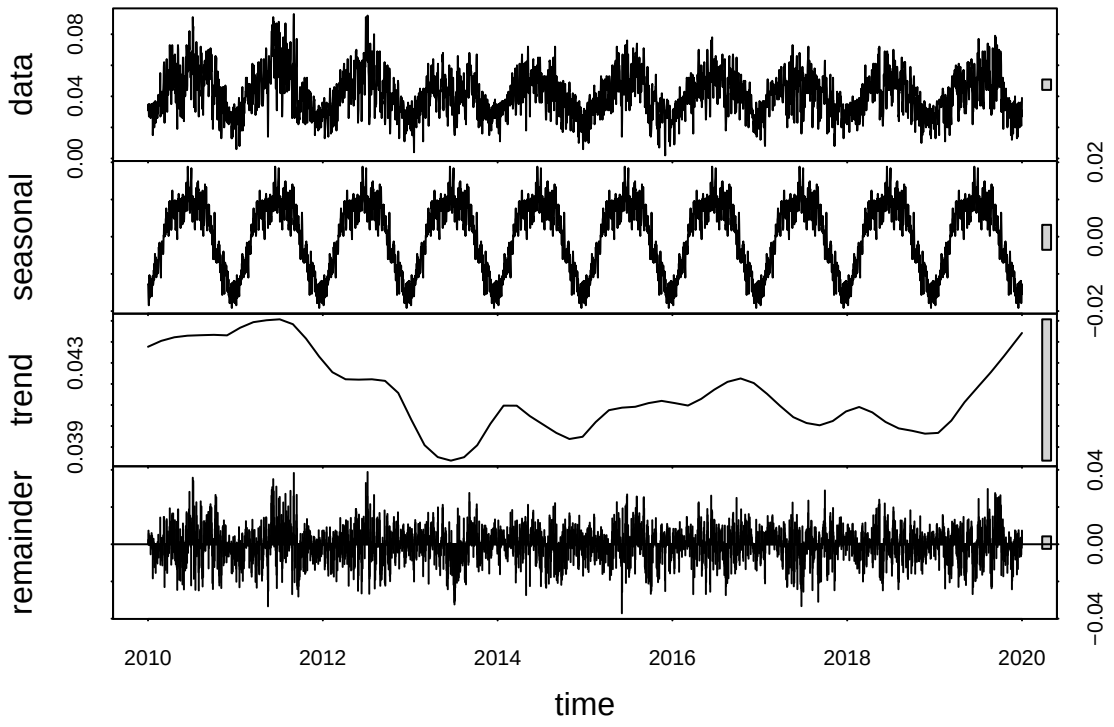
```
#9

GaringerOzone.monthly <- GaringerOzone_Interp %>%
  mutate(Year = format(Date, "%Y"), Month = format(Date, "%m")) %>%
           group_by(Year, Month) %>%
  summarise(Daily8hrMax.Mean = mean(Daily8hrMax))

GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = as.Date(as.yearmon(paste(Year, Month), "%Y %m")))
```

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.

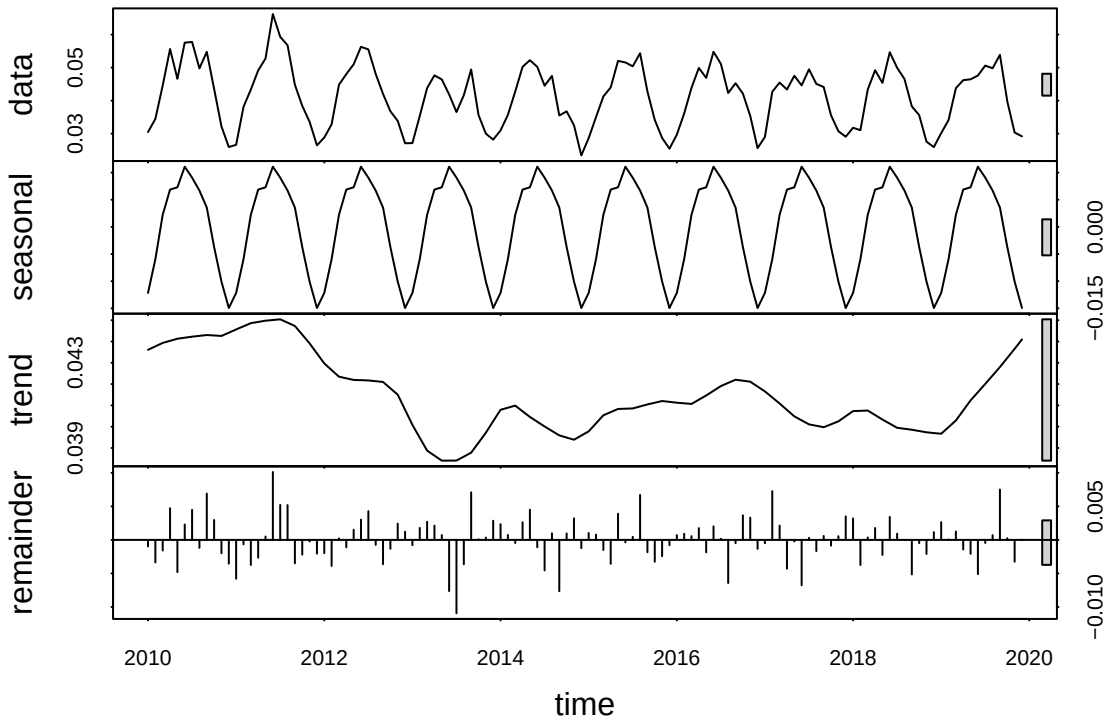
```
#10

GaringerOzone.daily.ts <-
  ts(GaringerOzone_Interp$Daily8hrMax, start = c(2010,1), end = c(2020), frequency=365)

GaringerOzone.monthly.ts <-
  ts(GaringerOzone.monthly$Daily8hrMax.Mean, 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
GaringerOzone.daily.ts_decomposed <- stl(GaringerOzone.daily.ts, s.window="periodic")
plot(GaringerOzone.daily.ts_decomposed)
```




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

smk.test(GaringerOzone.monthly.ts)
```

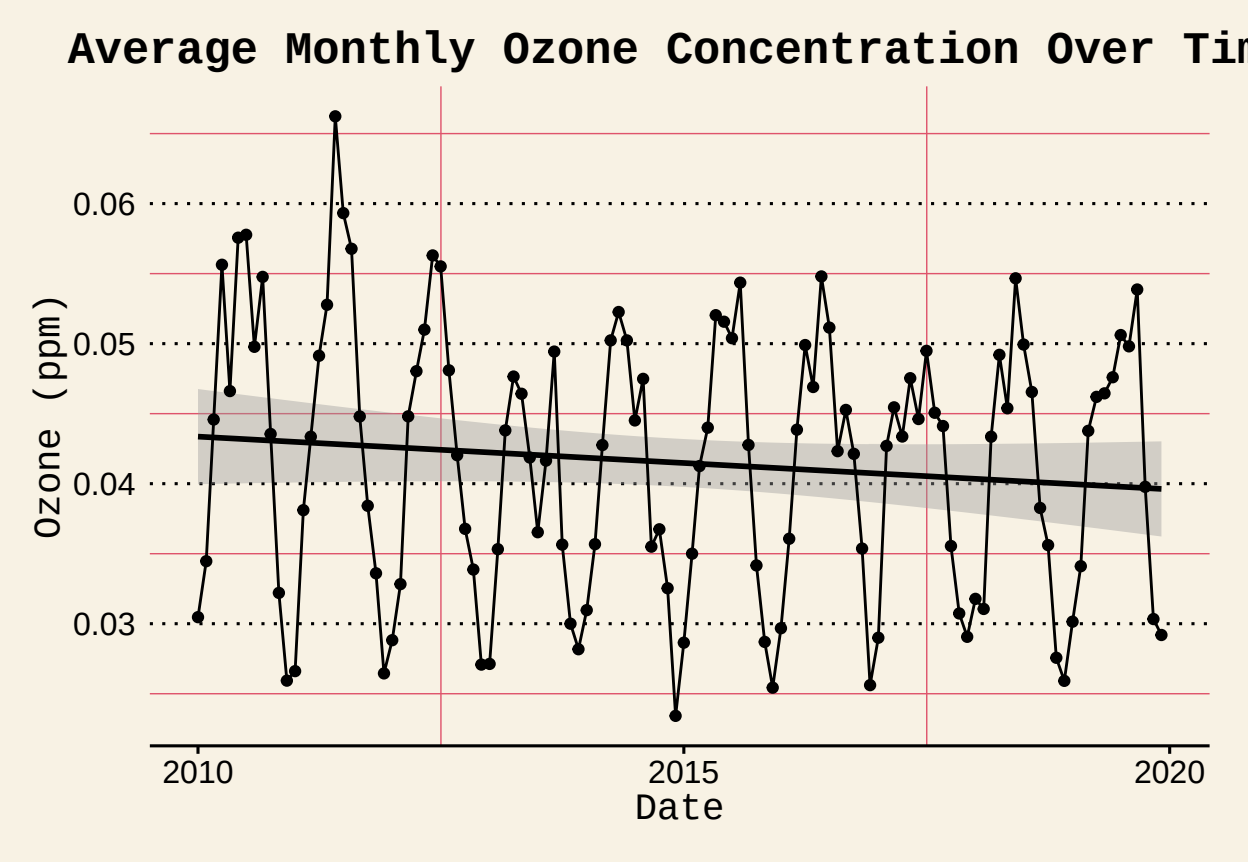
```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

Answer: The Seasonal Mann-Kendall test is appropriate here because our time series has seasonality that we haven't removed, and the Mann-Kendall test cannot be applied to seasonal data.

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 = Daily8hrMax.Mean)) +
  labs(title = "Average Monthly Ozone Concentration Over Time",
       x = "Date",
       y = "Ozone (ppm)") +
  geom_smooth(method = lm, color = "black") +
  geom_point() +
  geom_line()

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: The results of the Seasonal Mann Kendall test allow us to reject the null hypothesis, indicating that there is a trend in the monthly ozone time series data (z = -1.963, p-value = 0.04965). This test just indicates the presence of a trend in the data, it doesn't tell us whether that trend is positive or negative. However, the plot above does provide some insight as the trend line is negative. We have good reason to believe there is a trend in monthly ozone averages since 2010 and the trend line indicates a negative trend, but there's more statistical investigating to be done.

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.ts.Components <- as.data.frame(GaringerOzone.monthly.ts_decomposed$time.series)

GaringerOzone.monthly.ts.nonseasonal <- (GaringerOzone.monthly.ts - GaringerOzone.monthly.ts.Components$

#16

MannKendall(GaringerOzone.monthly.ts.nonseasonal)
```

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

Answer: Both tests, the Seasonal Mann Kendall and Man Kendall, yielded p-values sufficient for rejecting the null hypothesis, indicating that a trend is present in the Ozone monthly average time series. It's reaffirming that both tests yielded the same conclusion.