Assignment 8: Time Series Analysis

Sammy DiLoreto

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 Set up your session:
#Check working directory
getwd()
```

```
## [1] "/Users/sammydiloreto/Library/CloudStorage/Box-Box/ENV872-EDA/EDA-Spring2023/Assignments"
```

```
#Load packages
library(tidyverse);library(lubridate);library(trend);library(zoo)
library(Kendall);library(tseries);library(ggthemes);library(here)
here
```

```
## function (...)
## {
##     .root_env$root$f(...)
## }
## <bytecode: 0x7fd179764a80>
## <environment: namespace:here>
```

```
# Set theme
my_theme <- theme_base() +
  theme(
    plot.background = element_rect(
      linewidth = 1
      ),
    plot.title = element_text(
      size = rel(1),
      face = "bold"
      ),
    axis.title = element_text(
      size = rel(0.8),
      face = "bold"
      ),
    axis.text = element_text(
        size = rel(0.6),
      face = "bold"
      ),
    legend.title = element_text(
      size = rel(0.7),
      face = "bold"
      ),
    legend.text = element_text(
      size = rel(0.6)
      ),
    legend.position = 'top',
    complete = TRUE
  )
theme_set(my_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
GaringerOzone2010 <- read.csv(
  here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
  stringsAsFactors = TRUE)
GaringerOzone2011 <- read.csv(
  here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
  stringsAsFactors = TRUE)
GaringerOzone2012 <- read.csv(
  here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
  stringsAsFactors = TRUE)
GaringerOzone2013 <- read.csv(
  here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
  stringsAsFactors = TRUE)
GaringerOzone2014 <- read.csv(
  here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
  stringsAsFactors = TRUE)
GaringerOzone2015 <- read.csv(
  here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
```

```
  stringsAsFactors = TRUE)
GaringerOzone2016 <- read.csv(
  here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
  stringsAsFactors = TRUE)
GaringerOzone2017 <- read.csv(
  here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
  stringsAsFactors = TRUE)
GaringerOzone2018 <- read.csv(
  here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
  stringsAsFactors = TRUE)
GaringerOzone2019 <- read.csv(
  here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
  stringsAsFactors = TRUE)

GaringerOzone <- rbind(GaringerOzone2010, GaringerOzone2011,
                       GaringerOzone2012, GaringerOzone2013,
                       GaringerOzone2014, GaringerOzone2015,
                       GaringerOzone2016, GaringerOzone2017,
                       GaringerOzone2018, GaringerOzone2019 )
```

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.

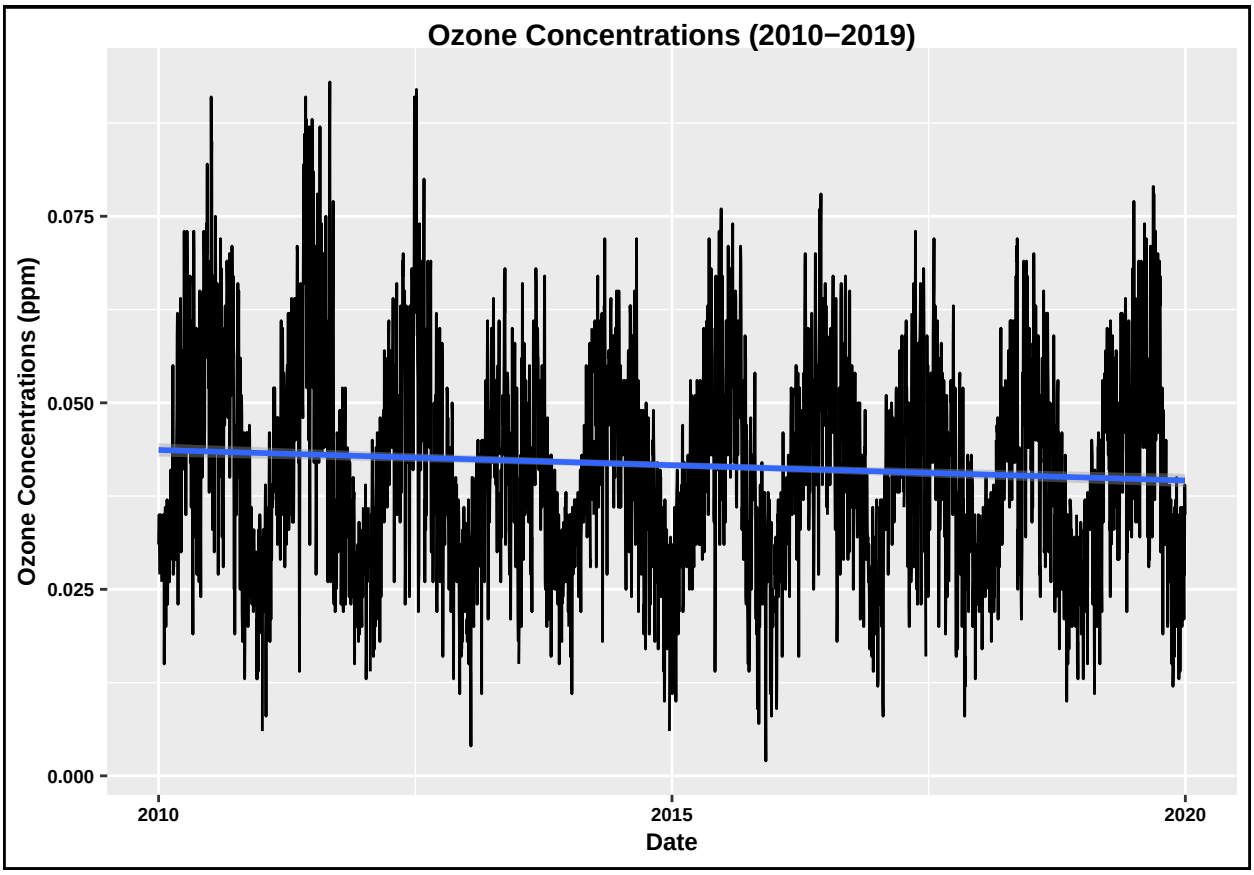
```
#3 Set your date column as a date class.
GaringerOzone$Date <- mdy(GaringerOzone$Date)

#4  Wrangle your dataset
GaringerOzone <- GaringerOzone %>%
  select(Date,Daily.Max.8.hour.Ozone.Concentration,DAILY_AQI_VALUE)

#5 Generate a daily dataset
start <- first(GaringerOzone$Date)
end <- last(GaringerOzone$Date)

Days <- as.data.frame(seq(start, end, by = "day"))
Days <- rename(Days, "Date" = 'seq(start, end, by = "day")')

#6 Use a `left_join` to combine the data frames.
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?

```
#7
ggplot(GaringerOzone, aes(x= Date, y= Daily.Max.8.hour.Ozone.Concentration))+
  geom_line()+
  geom_smooth(method = "lm")+
  labs(y = "Ozone Concentrations (ppm)", title = "Ozone Concentrations (2010-2019)")
```



Answer: The plot suggests that over the years 2010 to 2019, there has been a slight decline in ozone concentrations.

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_AQI_VALUE.clean = zoo::na.approx(DAILY_AQI_VALUE),
         Ozone.clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: The piecewise function assumes that missing data are equal to the nearest neighbor which is not suitable for this data because we see a slight decrease in the data over time so assuming that the missing data is equal to its neighbor could be very incorrect. The spline approach is also not suited for this interpolation because the trend is more consistent with a straight line rather than a quadratic function.

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)

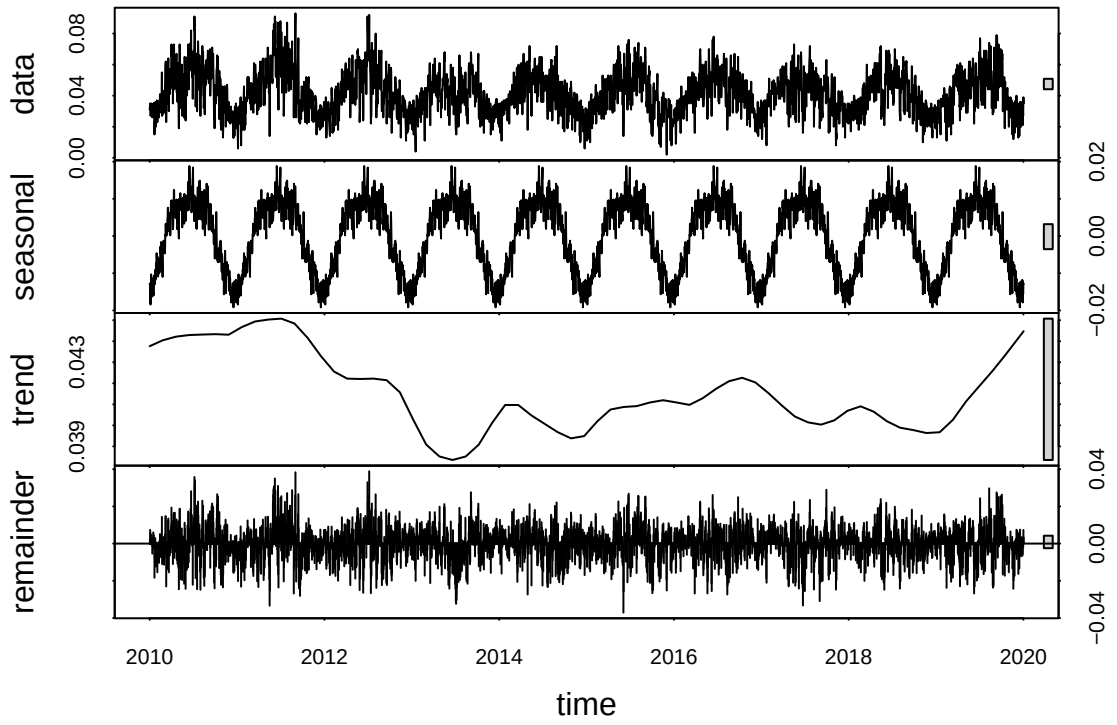
```
#9
GaringerOzone.monthly <- GaringerOzone  %>%
  mutate(Year = year(Date), Month = month(Date))  %>%
  group_by(Year, Month)%>%
  summarise(Mean.Ozone = mean(Ozone.clean),
            Mean.AQI = mean(DAILY_AQI_VALUE.clean))

GaringerOzone.monthly$Date <-ymd(paste0(GaringerOzone.monthly$Year, "-",
         GaringerOzone.monthly$Month, "-01"))
```

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.

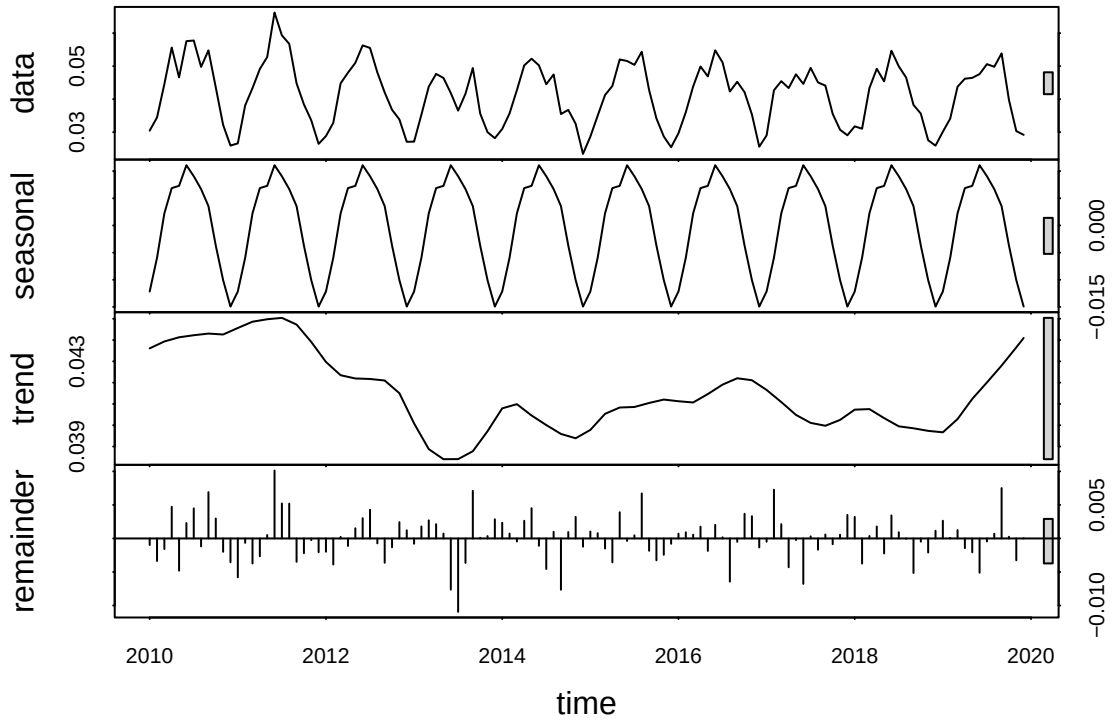
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone$Ozone.clean,
                             start = c(2010,1),
                             frequency = 365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean.Ozone,
                               start = c(2010,1),
                               frequency = 12)
```

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

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




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



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

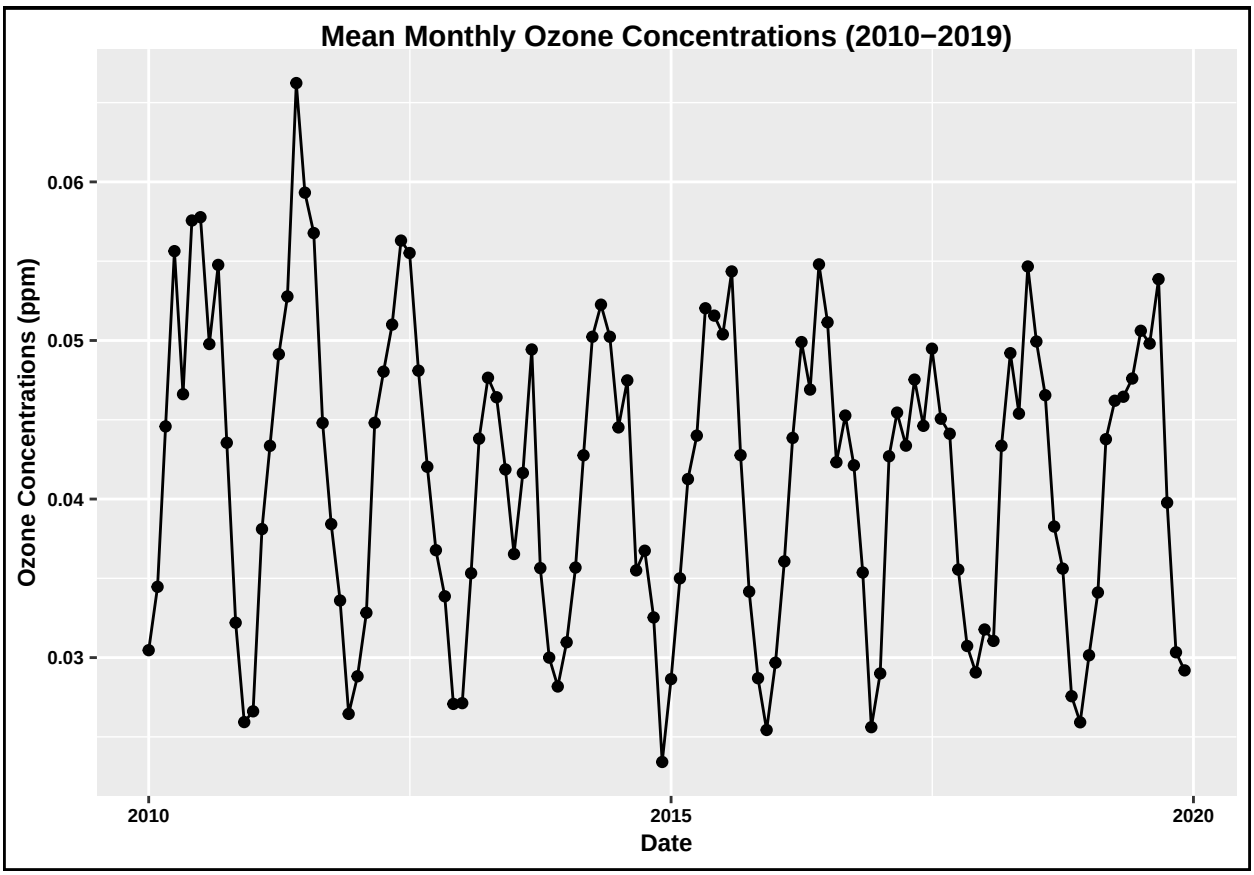
```r
#12
GaringerOzone.monthly.trend <- trend::smk.test(GaringerOzone.monthly.ts)
GaringerOzone.monthly.trend
```

```
##
##  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: Seasonal Mann-Kendall is most appropriate here because as seen in the plots of this data, the data goes up an down throughout each year, suggesting a seasonal trend. The small gray bar next to the seasonal component in the decomposition plot suggests that the variation in the seasonal component is small compared to the variation in the 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
ggplot(GaringerOzone.monthly, aes(x=Date, y= Mean.Ozone))+
  geom_point()+
  geom_line()+
  labs(y = "Ozone Concentrations (ppm)", title = "Mean Monthly Ozone Concentrations (2010-2019)")
```



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: We can reject the null and conclude that there is a negative trend in ozone concentrations over time (z = -1.963, S = -77, p-value = 0.04965). To answer the research question "Have ozone concentrations changed over the 2010s at this station?", yes ozone concentrations have decreased over time at this station.

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.noseason <- GaringerOzone.monthly.decomp$time.series[,2] +
  GaringerOzone.monthly.decomp$time.series[,3]

#16
GaringerOzone.monthly.noseason.trend <- MannKendall(GaringerOzone.monthly.noseason)
GaringerOzone.monthly.noseason.trend
```

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

Answer: We can reject the null and conclude that there is a negative trend in non-seaonal ozone monthly concentrations (tau = -0.165, p-value = 0.0075402). Both the seasonal and non-seasonal data suggests that there is a decline in monthly ozone concentrations over time. Therefore, seasonility does not play a major role in the decline in ozone concentrations over time.