Assignment 8: Time Series Analysis

Karina Leung

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
getwd()
```

```
## [1] "C:/Users/ktlro/OneDrive/Documents/EDA-Spring2023"
```

```
library(tidyverse)
library(lubridate)
library(trend)
library(zoo)
library(Kendall)
library(tseries)
library(formatR)
library(here)
library(cowplot)
library(viridis)
```

```
mytheme <- theme_classic(base_size = 12) +
  theme(axis.text = element_text(color = "black", size = 12),
        plot.title.position = "plot",
        plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
        axis.ticks = element_line(color = "black", linewidth = 0.5),
        legend.background = element_rect(color='grey', fill = 'white'),
        legend.title = element_text(color='black', hjust = 0.5),
        legend.position = "right")
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
GO2010 <-  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"),
↪  stringsAsFactors = TRUE)

GO2011 <-  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"),
↪  stringsAsFactors = TRUE)

GO2012 <-  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"),
↪  stringsAsFactors = TRUE)

GO2013 <-  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"),
↪  stringsAsFactors = TRUE)

GO2014 <-  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"),
↪  stringsAsFactors = TRUE)

GO2015 <-  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"),
↪  stringsAsFactors = TRUE)

GO2016 <-  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"),
↪  stringsAsFactors = TRUE)

GO2017 <-  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"),
↪  stringsAsFactors = TRUE)

GO2018 <-  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"),
↪  stringsAsFactors = TRUE)

GO2019 <-  read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"),
↪  stringsAsFactors = TRUE)

GaringerOzone <- rbind(GO2010, GO2011, GO2012, GO2012, GO2013, GO2014, GO2015, GO2016,
↪  GO2017, GO2018, GO2019)
```

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".

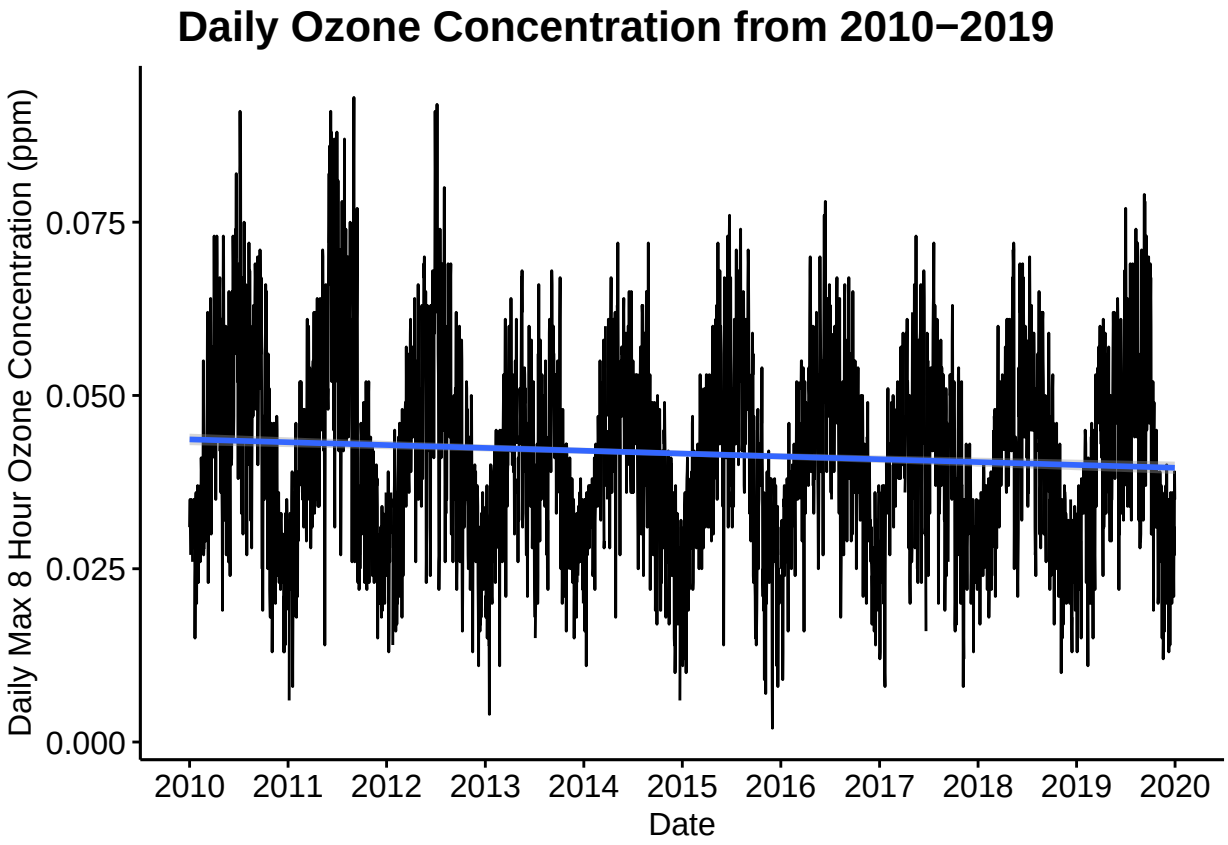
6. Use a `left_join` to combine the data frames. Specify the correct order of data frames within this function so that the final dimensions are 3652 rows and 3 columns. Call your combined data frame GaringerOzone.

```
#3
GaringerOzone$Date <- mdy(GaringerOzone$Date)

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

#5
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), "days"))

names(Days) <- "Date"

#6

GaringerOzone_clean <- left_join(Days, GaringerOzone_clean)
```

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_clean, aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = "lm") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  labs(y = "Daily Max 8 Hour Ozone Concentration (ppm)", title = "Daily Ozone
  ↪  Concentration from 2010-2019")
```



Answer: The plot shows a very slight downward slope on the trend line from 2010-2019, indicating that there's slight decrease in ozone 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_clean <- GaringerOzone_clean %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration_clean =
   ↪  zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: Using piecewise constant would fill in the missing data with the "nearest neighbor" approach, which would give the missing data the same value and we cannot assume that these values are the same. We also cannot use a spline interpolation because it uses a quadratic formula to interpolate and our data looks more like a descending linear line.

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)

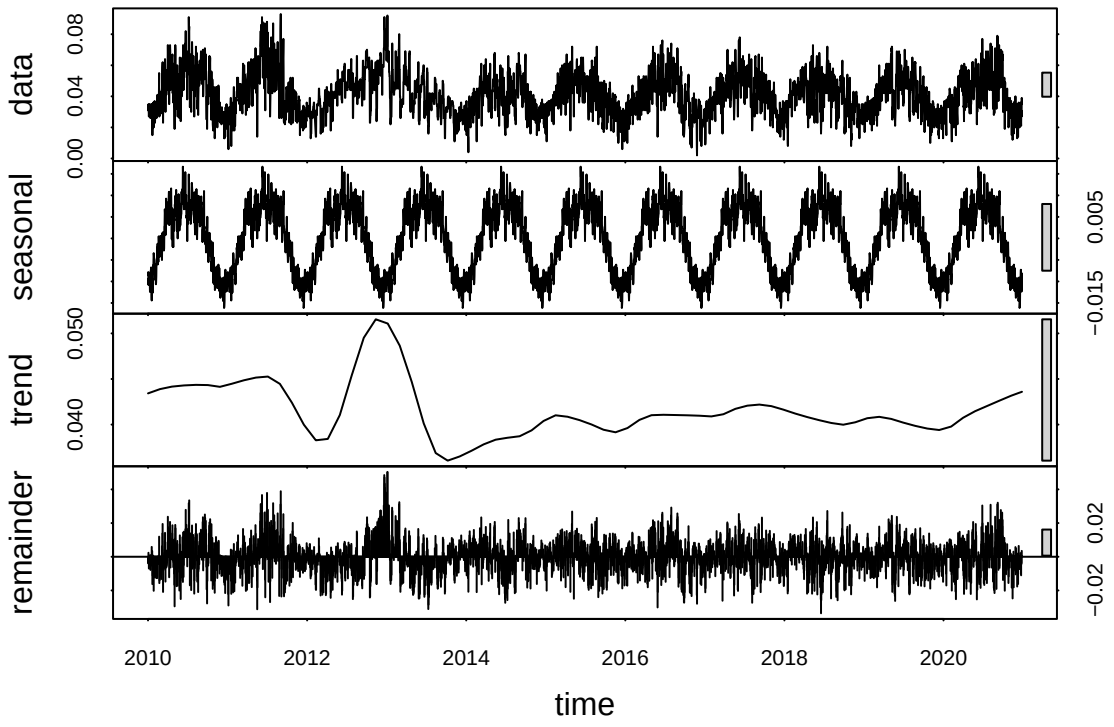
```r
#9
GaringerOzone.monthly <- GaringerOzone_clean %>%
  mutate(Year= year(Date), Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarise(Mean.Monthly.Ozone.Concentrations =
  ↪  mean(Daily.Max.8.hour.Ozone.Concentration_clean))

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.

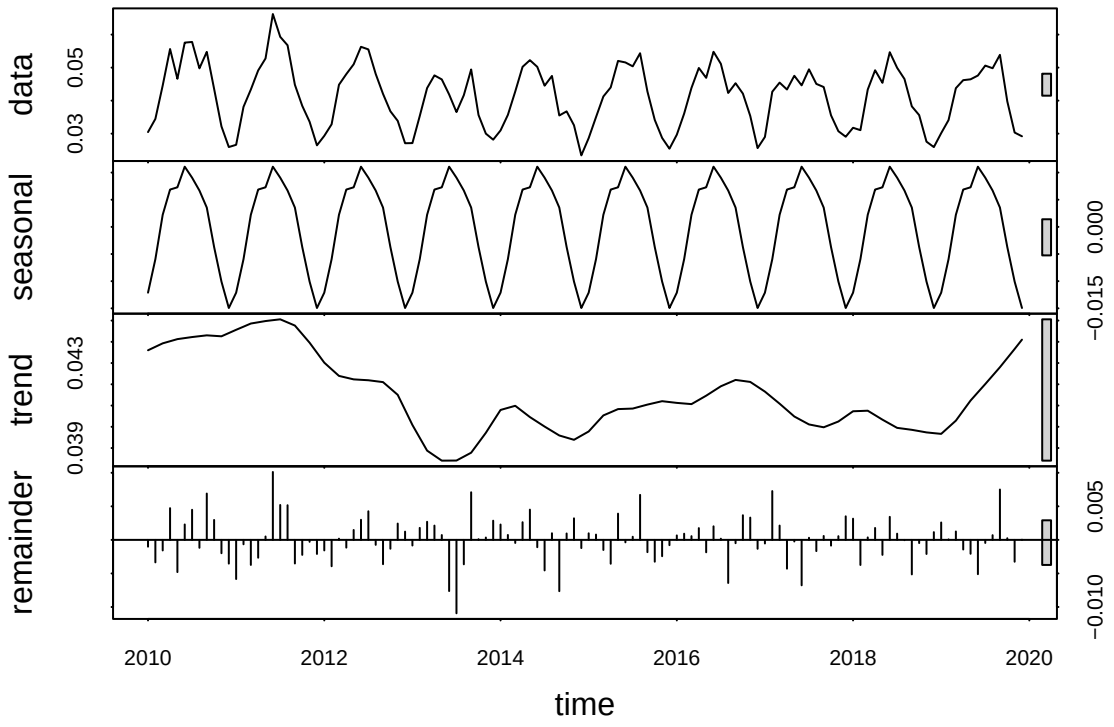
```r
#10
f_day <- day(first(GaringerOzone_clean$Date))
f_month <- month(first(GaringerOzone_clean$Date))
f_year <- year(first(GaringerOzone_clean$Date))


GaringerOzone.daily.ts <-
↪  ts(GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration_clean,
                  start=c(f_year,f_month, f_day),
                  frequency=365)

f2_year <- year(first(GaringerOzone.monthly$Date))
f2_month <- month(first(GaringerOzone.monthly$Date))

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean.Monthly.Ozone.Concentrations,
                  start=c(f2_year,f2_month),
                  frequency=12)
```

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

```r
#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?

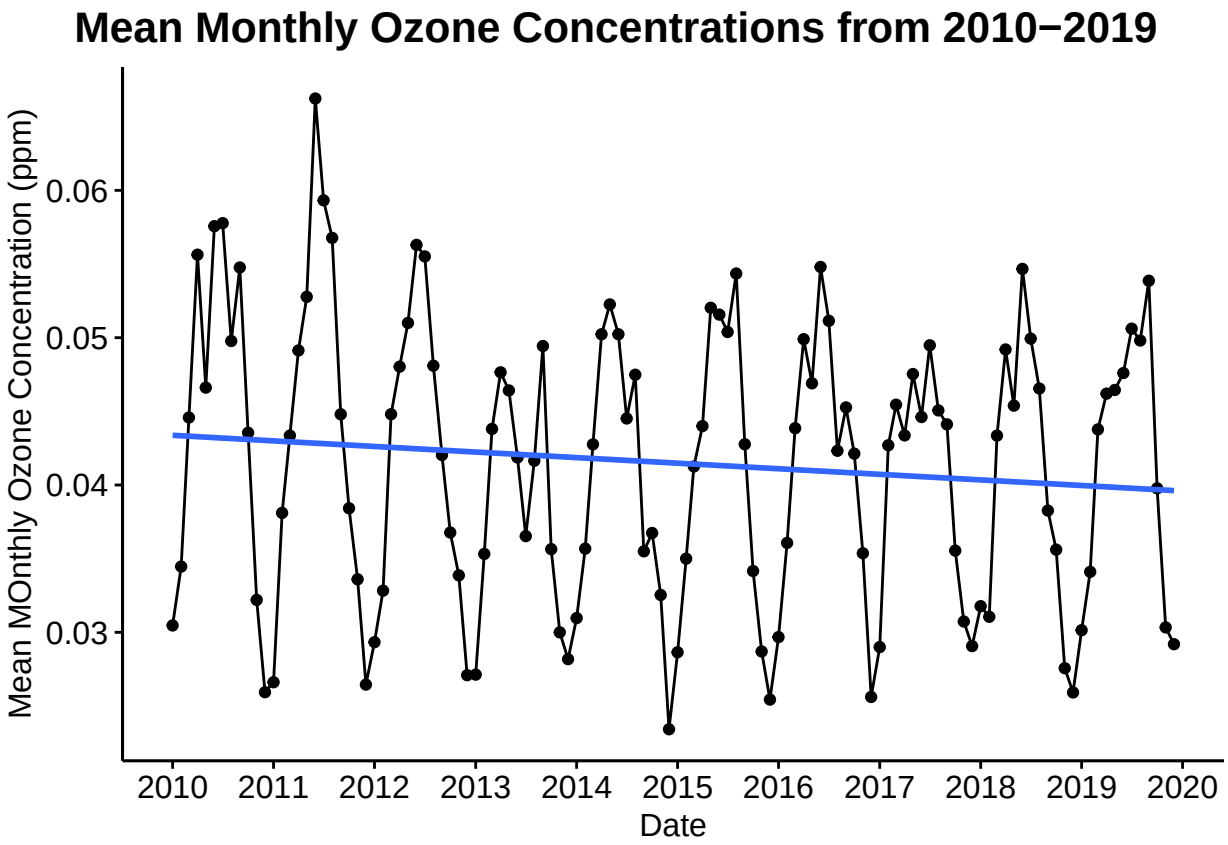
```
#12
GaringerOzone_monthly_series <- trend::smk.test(GaringerOzone.monthly.ts)

GaringerOzone_monthly_series
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -2.0146, p-value = 0.04394
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -79 1499
```

Answer: Based on the plot visualizations, it appears that ozone has some seasonality to the data since there are up and down fluctuations from year to year. The seasonal Mann-Kendall test is the only one that takes into account seasonality, whereas the other ones we learned about do not.

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
GaringerOzone_plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean.Monthly.Ozone.Concentrations)) +
  geom_point() +
  geom_line() +
  labs(y = "Mean MOnthly Ozone Concentration (ppm)", title = "Mean Monthly Ozone
  ↪  Concentrations from 2010-2019") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  geom_smooth(method = lm, se = FALSE)

print(GaringerOzone_plot)
```



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 study question is: Have ozone concentrations changed over the 2010s at this station? According to the visual plot, yes ozone concentrations do change over the 2010s. The graph shows a decreasing trend from 2010 to 2019 (Seasonal Mann-Kendall trend test, z = -2.0146, p-value = 0.04394). Therefore, we accept the alternative hypothesis that ozone concentrations have changed over the 2010s, and they have decreased.

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


#16
GaringerOzone.monthly.noseason.trend <- trend::mk.test(GaringerOzone.monthly.noseason.ts)

GaringerOzone.monthly.noseason.trend
```

```
##
##  Mann-Kendall trend test
##
## data:  GaringerOzone.monthly.noseason.ts
## z = -2.6856, n = 120, p-value = 0.00724
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
##  -1185.000000 194365.666667    -0.165978
```

Answer: The Mann-Kendall test with no seasonality returns the same result as the Mann-Kendall test with seasonality in that we can reject the null hypothesis and accept the alternative hypothesis that the ozone concentrations between 2010 and 2019 are not the same value, but the particular values returned from the test differ (z = -2.6856 and p-value = 0.00724 for the non-seasonality Mann-Kendall test vs. z = -2.0146, p-value = 0.04394 for the seasonality Mann-Kendall test).