Assignment 8: Time Series Analysis

Abby Liu

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

#Load packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.0
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(trend)
library(zoo)
```

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

```
library(here)
```

```
## here() starts at /home/guest/EDA-Spring2023
```

```
here()
```

```
## [1] "/home/guest/EDA-Spring2023"
```

```
#Set theme
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.

```
#2

#Import datasets
Ozone2010 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"), stringsAsFacto
Ozone2011 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"), stringsAsFacto
Ozone2012 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"), stringsAsFacto
Ozone2013 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"), stringsAsFacto
Ozone2014 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"), stringsAsFacto
Ozone2015 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"), stringsAsFacto
Ozone2016 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"), stringsAsFacto
Ozone2017 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"), stringsAsFacto
Ozone2018 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"), stringsAsFacto
Ozone2019 <- read.csv(here("Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"), stringsAsFacto

#Combine datasets
Ozone <- rbind(Ozone2010, Ozone2011, Ozone2012, Ozone2013, Ozone2014, Ozone2015, Ozone2016, Ozone2017, O
```

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 Transform date column
Ozone$Date <- as.Date(Ozone$Date, "%m/%d/%Y")

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

#5 Generate daily dataset
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "day"))
colnames(Days) <- "Date"

#6 Combine datasets
GaringerOzone <- left_join(Days, Ozone)
```

```
## Joining with `by = join_by(Date)`
```

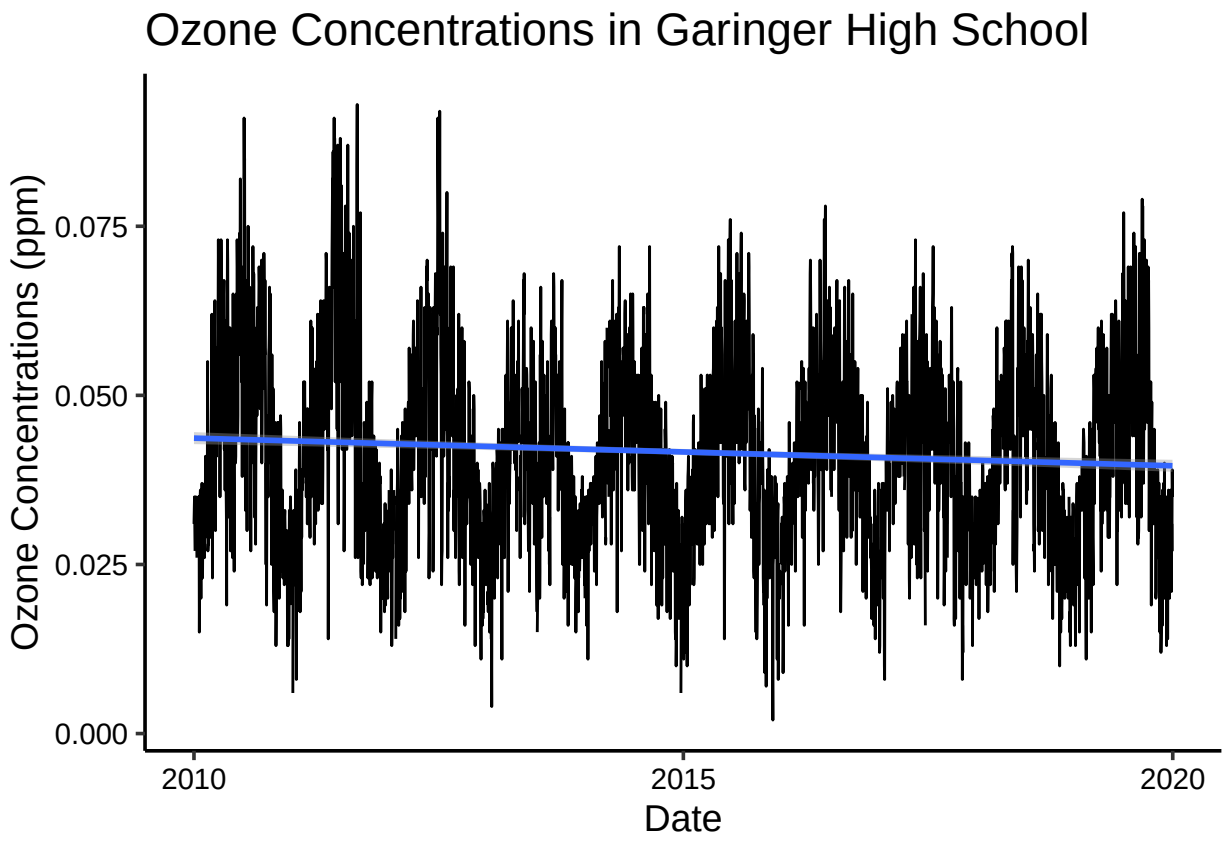
Visualize

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 Visualization
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line(linewidth = 0.5) +
  geom_smooth(method = lm) +
  labs(y = "Ozone Concentrations (ppm)", title = "Ozone Concentrations in Garinger High School")
```

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

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



Answer: There exists very significant seasonality in ozone concentration. It goes up in the summer and goes down during winter. The plot suggested a very slight downward trend in ozone concentration over time but a decomposed trend is needed to draw a better conclusion.

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 Fill in missing values
GaringerOzone <-
  GaringerOzone %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: In this case, the linear interpolation method fits the data better. Linear interpolation assumes the missing data to fall between the previous and next measurement, with a straight line drawn between the known points determining the values of the interpolated data on any given date. Based on the plot, ozone concentrations seem continuous and pretty linear. A piecewise constant will lead to very abrupt steps in the data and make the data less continuous as it assumes missing data to be equal to the measurement made nearest to that date. A spline interpolation uses quadratic function which adds unnecessary complexity to the dataset.

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 Aggregate monthly data
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(Year = year(Date), Month = month(Date)) %>%
  group_by(Year, Month)%>%
  summarise(AverageOzone = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%
  mutate(Date = mdy(paste0(Month,"-",01,"-",Year)))
```

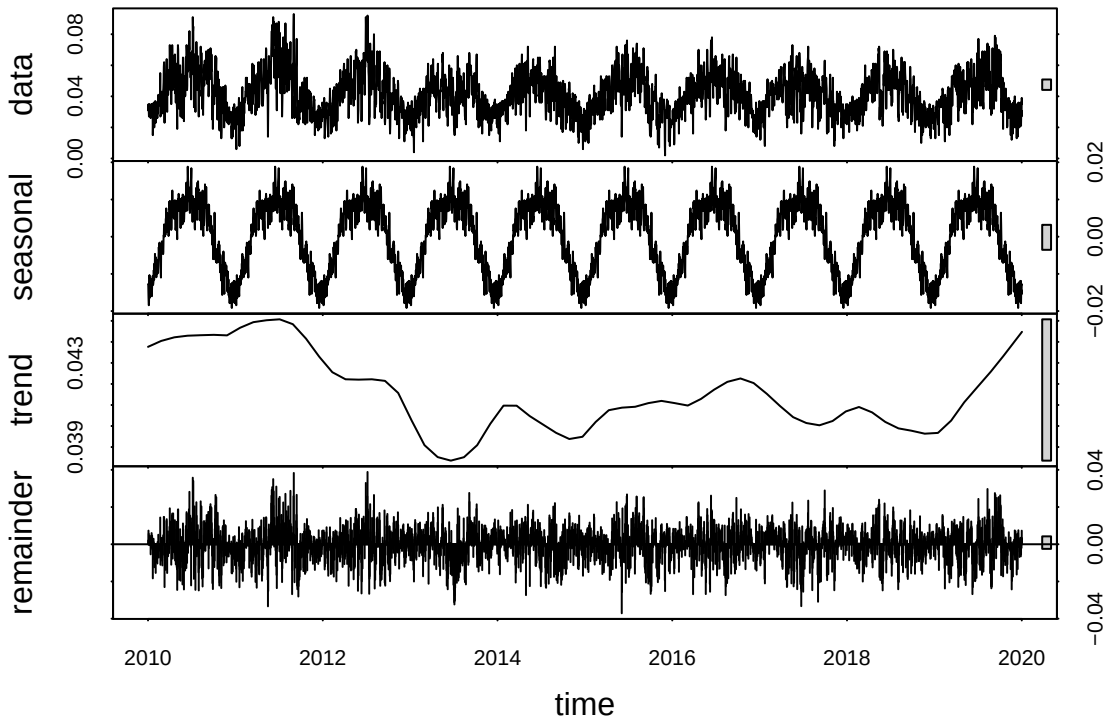
```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

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.

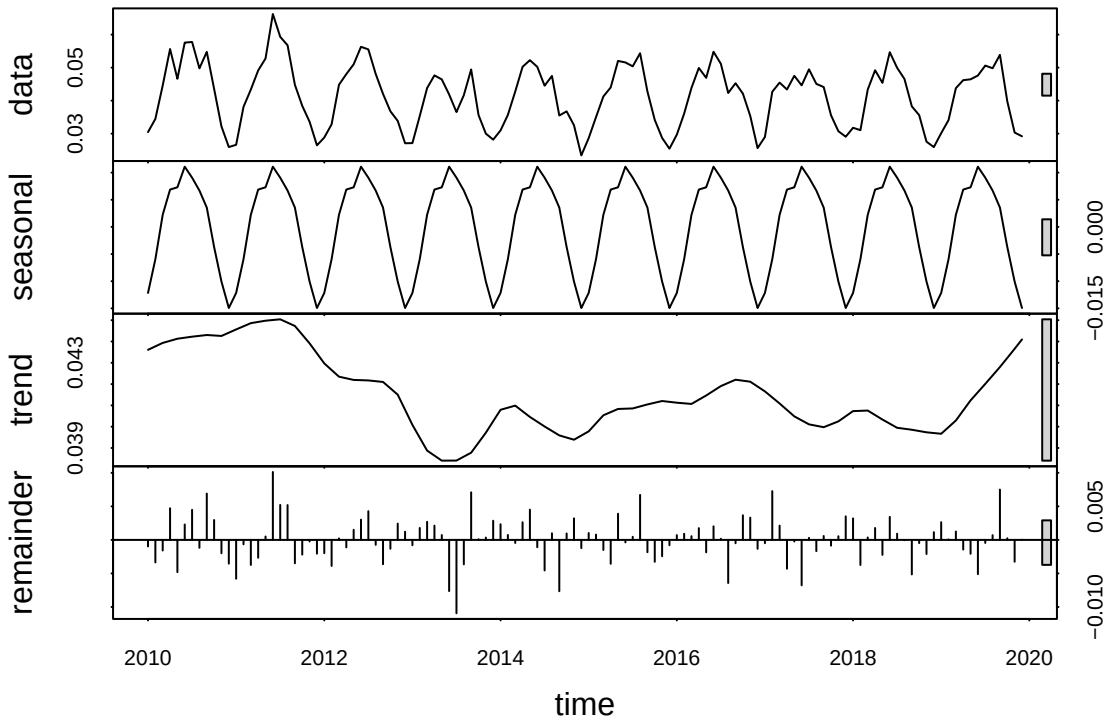
```
#10 Generate time series objects
year1 <- year(first(GaringerOzone$Date))
month1 <- month(first(GaringerOzone$Date))
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration, start = c(year1, month

year2 <- year(first(GaringerOzone.monthly$Date))
month2 <- month(first(GaringerOzone.monthly$Date))
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$AverageOzone, start = c(year2, month2), frequency =
```

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

```
#11

#Decompose the time series objects
GaringerOzone.daily.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
GaringerOzone.monthly.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

#Plot the components
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 Trend analysis
Ozone.monthly.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(Ozone.monthly.trend)
```

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

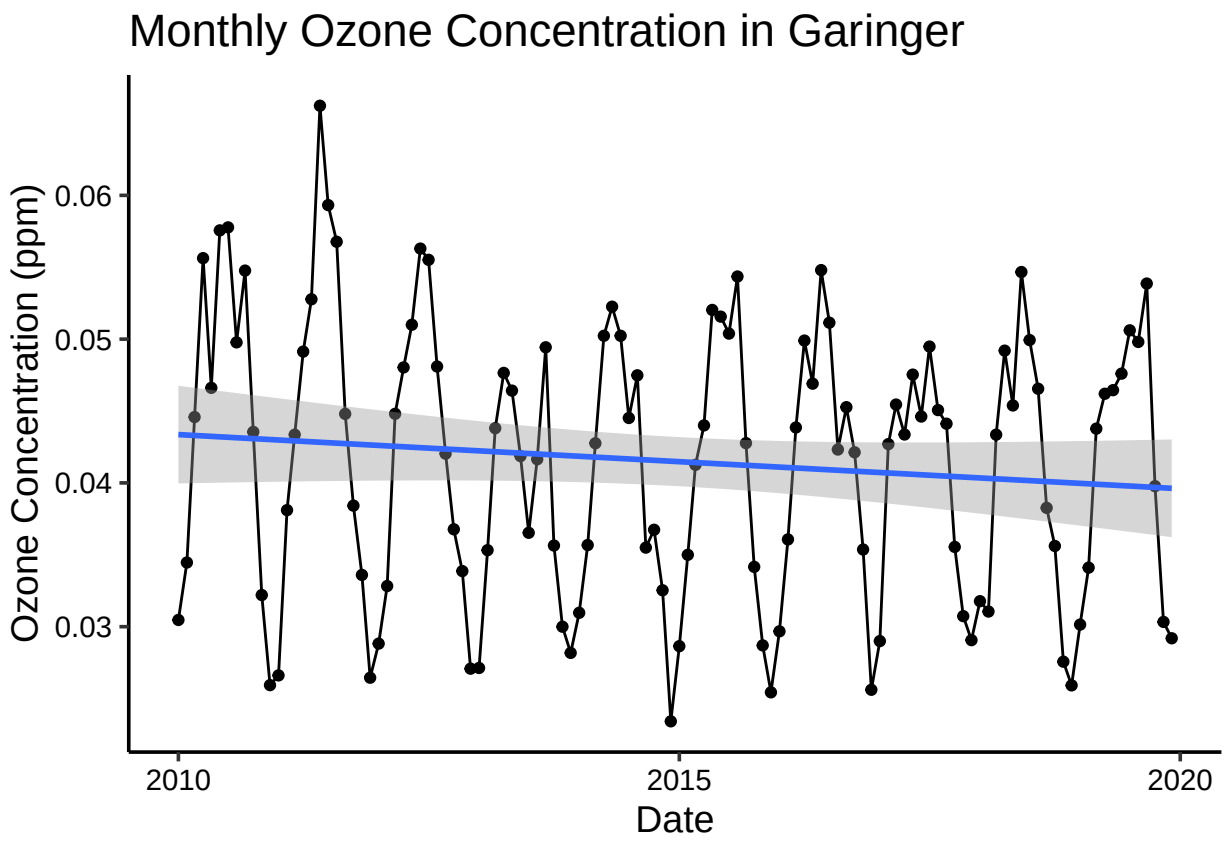
Answer: In this case the seasonal Mann-Kendall is most appropriate because it takes into account of seasonality while other trend analysis methods don't consider. Also, it's non-parametric, which means the data doesn't have to be normally distributed.

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 Visualization
Ozone.monthly.plot <-
ggplot(GaringerOzone.monthly, aes(x = Date, y = AverageOzone)) +
  geom_point() +
  geom_line() +
  labs(y = "Ozone Concentration (ppm)", title = "Monthly Ozone Concentration in Garinger") +
```

```
  geom_smooth( method = lm )
print(Ozone.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: Based on the monthly ozone concentration graph, there's clearly seasonality in the data but it's hard to tell the trend solely in the graph. That's where the seasonal Mann-Kendall test comes into play. The null hypothesis for the test is that there's no monotonic trend. The p-value of test result is 0.0467 (<0.05). We reject the null hypothesis, suggesting a monotonic trend is observed. In that case, we can conclude that ozone concentration has changed over the 2010s in a decreasing trend.

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.

```r
#15 Extract series components
Ozone.monthly.components <- as.data.frame(GaringerOzone.monthly.decomposed$time.series[,1:3])

Ozone.monthly.components <- Ozone.monthly.components %>%
  mutate(Ozone.monthly.components, Observed = GaringerOzone.monthly$AverageOzone, Date = GaringerOzone.m

Ozone.monthly.components <- Ozone.monthly.components %>%
  mutate(Nonseasonal = Ozone.monthly.components$Observed - Ozone.monthly.components$seasonal)

#16 Run the Mann Kendall test
GaringerOzone.monthly.nonseasonal.ts <- ts(Ozone.monthly.components$Nonseasonal, start = c(2010,1), freq

Ozone.nonseasonaly.monthly.trend <- Kendall::MannKendall(GaringerOzone.monthly.nonseasonal.ts)
summary(Ozone.nonseasonaly.monthly.trend)
```

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

Answer: The p-value of the Mann Kendall test is 0.0075, which is more significant than the seasonal Mann Kendall test result. It shows that when taking out the seasonality, the trend can be more evident. Both tests have negative scores, indicating a decreasing trend.