Assignment 8: Time Series Analysis

Shiqi Zheng

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

```r
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
```

```r
library(lubridate)
library(trend)
library(zoo)
```

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

```r
library(Kendall)
library(tseries)
```

```
## Registered S3 method overwritten by ’quantmod’:
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(here)
```

```
## here() starts at /Users/shiqizheng/Desktop/ENV872/EDE_Fall2023
```

```r
here()
```

```
## [1] "/Users/shiqizheng/Desktop/ENV872/EDE_Fall2023"
```

```r
mytheme <- theme_gray() +
  theme(axis.text = element_text(color = "black"),
        axis.ticks = element_line(color = "gray", linewidth = 0.5),
        panel.grid = element_line(color = "white", linewidth = 0.2),
        legend.position = "right",
        legend.background = element_rect(fill = "lightyellow"),
        plot.title = element_text(color = "black", size = 16, face = "bold", hjust = 0.5)
        )
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.

```r
#1 load data
folder_path <- "Data/Raw/Ozone_TimeSeries"
files <- list.files(path = folder_path, pattern = ".csv", full.names = TRUE)
dataframes <- list()
for (file in files[1:10]) {
  df <- read.csv(file)
  dataframes <- append(dataframes, list(df))
}
GaringerOzone <- bind_rows(dataframes)
dim(GaringerOzone)
```

```
## [1] 3589   20
```

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 date
GaringerOzone$Date <- mdy(GaringerOzone$Date)

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

# 5 date frame
start_date <- as.Date("2010-01-01")
end_date <- as.Date("2019-12-31")
Days <- as.data.frame(seq(start_date, end_date, by = "days"))
colnames(Days) <- "Date"

# 6 create daily frame
GaringerOzone <- left_join(Days, GaringerOzone, by = "Date")
dim(GaringerOzone)
```

```
## [1] 3652    3
```

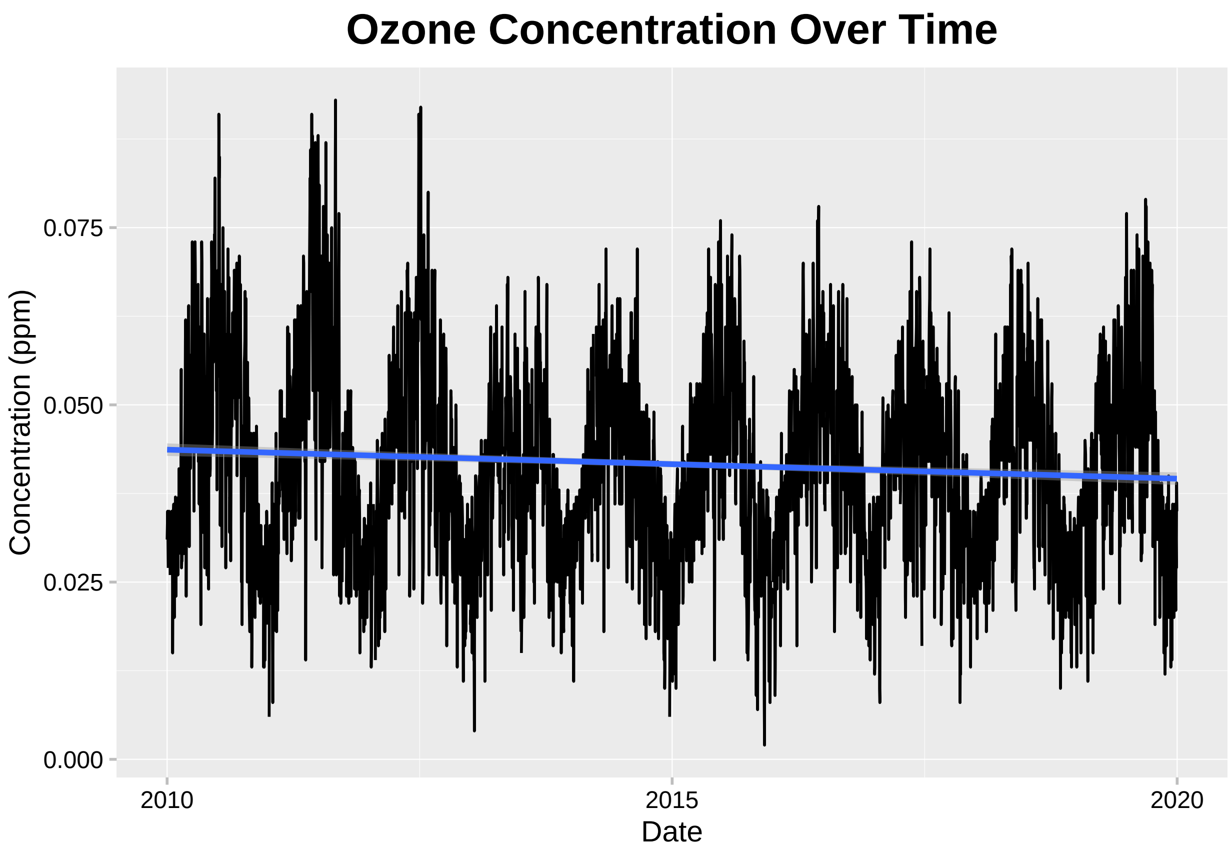
Visualize

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 plot
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  labs(x = "Date", y = "Concentration (ppm)", title = ("Ozone Concentration Over Time"))+
  geom_smooth( method = lm )
```

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

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



Answer: The plot suggest that there is a general decreasing trend in ozone concentration over time. There is a clear seasonal trend for concentration each year with a peak in each summer.

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$Daily.Max.8.hour.Ozone.Concentration <-
  na.approx(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

Answer: Because there is no rapid and irregular changes and shows general linear changes between ozone concentration data, linear interpolation is then a good choice as it's simple and effective. Piecewise constant interpolation suits for the value remains constant between observed data points and jumps to the next observed value. This approach may be less suitable for this continuous, gradual change in the dataset. Spline interpolation is more suitable for more complex, non-linear patterns, it's computationally more intensive and can be sensitive to the choice of spline order and parameters.

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 group
GaringerOzone.monthly <- GaringerOzone %>%
  mutate(year = year(Date), month = month(Date)) %>%
  group_by(year, month) %>%
  summarise(mean_ozone = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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

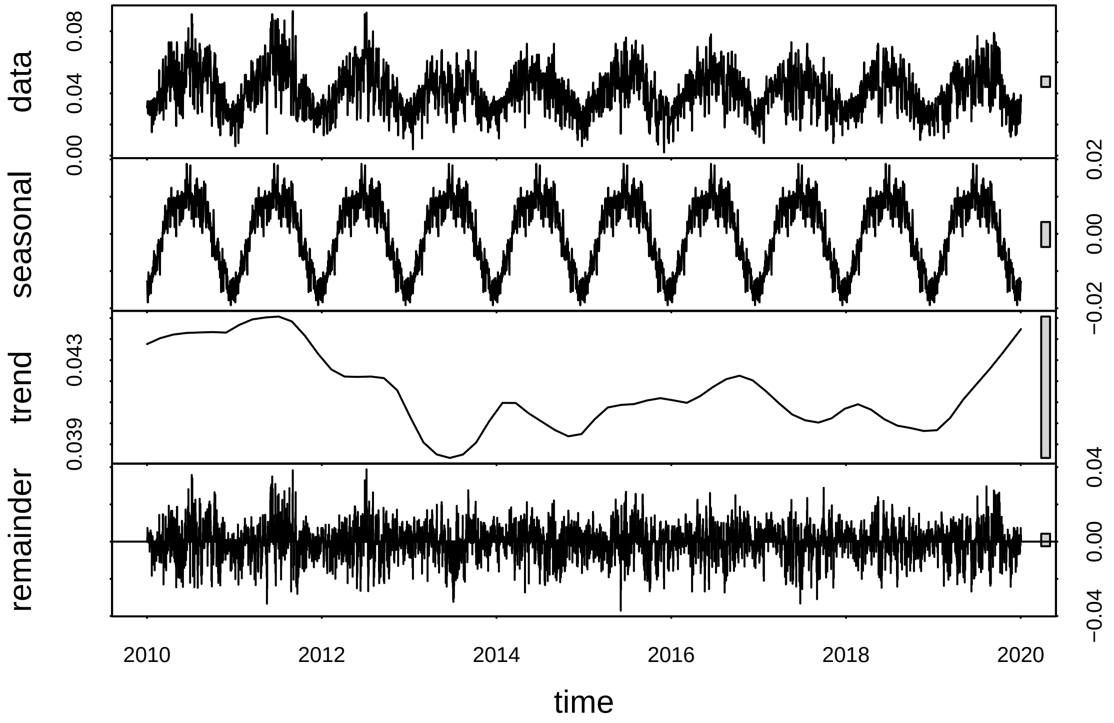
```
# Create a new Date column
GaringerOzone.monthly <- mutate(GaringerOzone.monthly,
                                Date = as.Date(paste(year, month, "01", sep = "-")))
```

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.

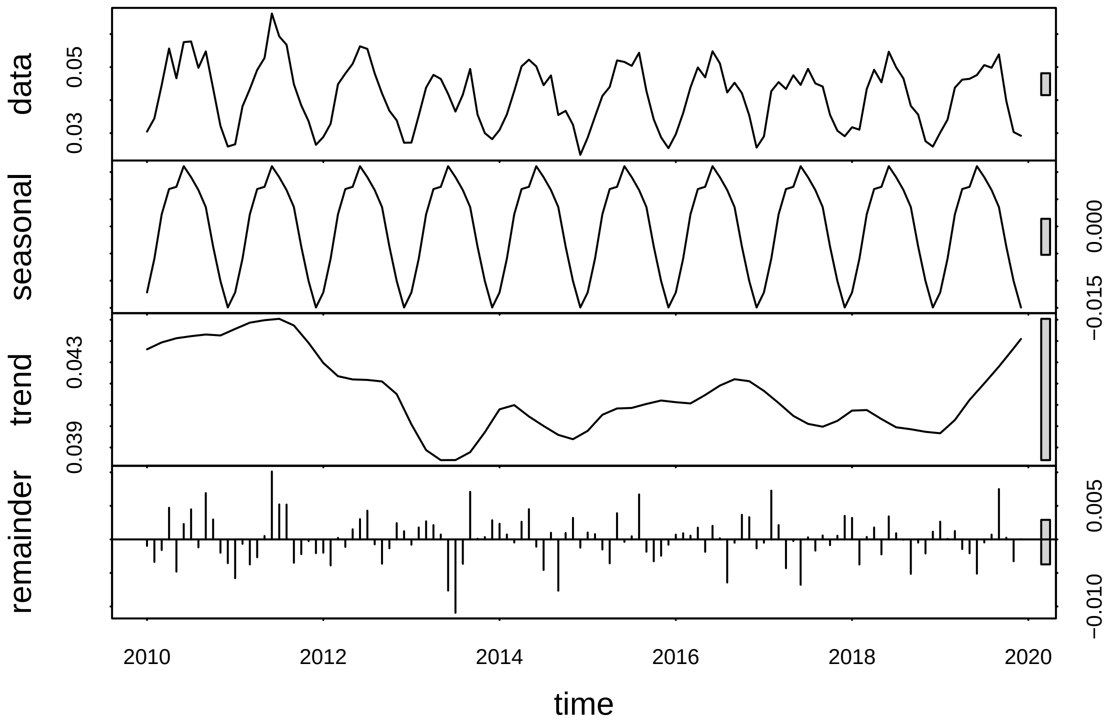
```
#10 ts
f_month <- month(first(GaringerOzone$Date))
f_year <- year(first(GaringerOzone$Date))


GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
                             start=c(f_year,f_month),frequency = 365)
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$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 decomp and plot
GaringerOzone.daily.decomp <- stl(GaringerOzone.daily.ts,s.window = "periodic")
plot(GaringerOzone.daily.decomp)
```



```r
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?

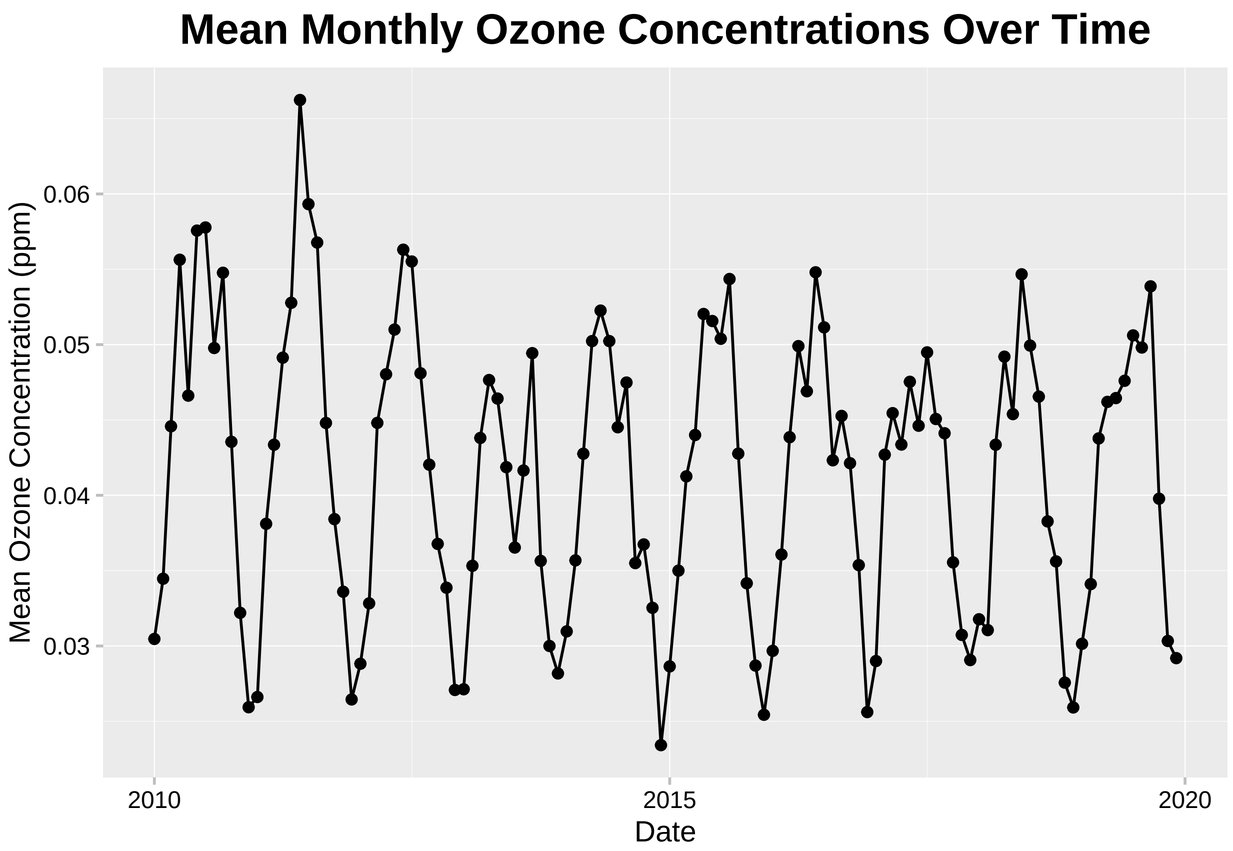
```
#12 seasonal Mann-Kendall
GaringerOzone.monthly.ts.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(GaringerOzone.monthly.ts.trend)
```

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

Answer: Because the concentration data shows cyclic (seasonal) trend, and seasonal Mann-Kendall is the only monotonic trend analysis counts for seasonality.

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
ggplot(GaringerOzone.monthly, aes(x = Date, y = mean_ozone)) +
  geom_point() +
  geom_line() +
  labs(x = "Date", y = "Mean Ozone Concentration (ppm)",
       title = ("Mean Monthly Ozone Concentrations Over Time"))
```



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 plot and the test show that ozone concentrations show a negative trend over the 2010s at this station (tau = 0.143).This suggests that ozone concentrations have indeed changed over the 2010s at this station, with a statistically significant downward trend according to the seasonal Mann-Kendall test ($p = 0.046724 < 0.05$).

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 extract the components
GaringerOzone.monthly.comp <-
  as.data.frame(GaringerOzone.monthly.decomp$time.series[,1:3])

GaringerOzone.monthly.nonseason <-
  GaringerOzone.monthly.ts - GaringerOzone.monthly.comp$seasonal


#16 test
GaringerOzone.monthly.nonseason.trend <-
```

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

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

> Answer: The results suggest a stronger and more statistically significant decreasing trend with greater variability and stabiloty in the non-seasonal Ozone monthly series compared to the original series, as evidenced by the more negative score, larger variance, larger denominator, and smaller p-value. This implies that the seasonal component had an impact on the observed trend, and when removed, the trend becomes more pronounced.