Assignment 8: Time Series Analysis

Aye Nyein Thu

Fall 2024

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

```
#1a Check the working directory
getwd()
```

```
## [1] "/home/guest/EDE_Fall2024"
```

```
#1b Load packages
library(zoo)
library(trend)
library(tidyverse)
library(lubridate)
library(here)

#1c Set the ggplot theme
mytheme <- theme_classic(base_size=12) +
  theme(
    axis.title = element_text(color="black"),
    legend.position = "top",
```

```r
    plot.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.

```r
#2 Import datasets and combine them into a single dataframe
Garinger.Ozone <- list.files(path="./Data/Raw/Ozone_TimeSeries/",
                             pattern = "*.csv", full.names = TRUE) %>%
  lapply(read.csv) %>%
  bind_rows() %>%
  data.frame(stringsAsFactors=TRUE)

dim(Garinger.Ozone)
```

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

```r
# Note: The dataset is named 'Garinger.Ozone'.
# The "." is added so as not to confuse with the file name in Question 6.
```

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.

```r
#3 Set the date column as date
Garinger.Ozone$Date <- mdy(Garinger.Ozone$Date)

#4 Wrangle the dataset to include only 3 columns
Garinger.Ozone_Processed <- Garinger.Ozone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

#5a Create a new data frame using a sequence of dates
Days <- as.data.frame(seq(as.Date("2010-01-01"),
                          as.Date("2019-12-31"),
                          "day"))
```

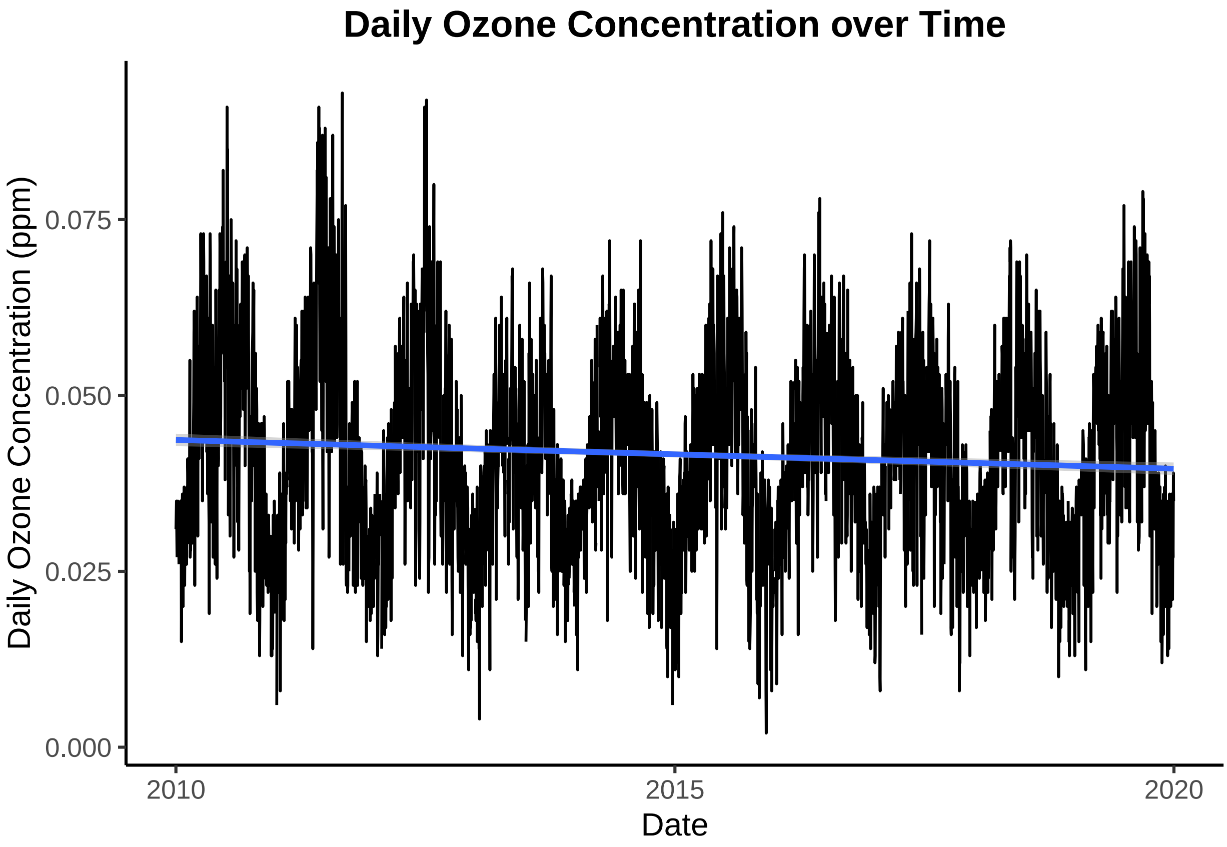
```r
#5b Change the column name of a new data frame
colnames(Days) <- "Date"

#6 Combine the data frames using 'left_join'
GaringerOzone <- left_join(
  Days, Garinger.Ozone_Processed,
  by = c ("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?

```r
#7 Create a line plot for ozone concentration over time
plot <- GaringerOzone %>%
  ggplot(aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method="lm") +
  labs(
    title="Daily Ozone Concentration over Time",
    y="Daily Ozone Concentration (ppm)") +
  theme(plot.title = element_text(hjust = 0.5, face="bold"))
print(plot)
```



Answer: The plot showcases a slight decreasing trend of the daily ozone concentrations over time. However, the downward sloping fitted line is in a gradual decrease nature. Further, a high level of fluctuations in ozone concentration is pronounced in the plot. These facts suggest to further analyze the trend of ozone concentrations over time using a time series analysis and identify the decreasing trend is statistically significant.

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?

```
#8a Check the missing values in the dataset
summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```
summary(GaringerOzone$DAILY_AQI_VALUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    2.00   30.00   38.00   41.57   47.00  169.00      63
```

```
#8b Use a linear interpolation to fill in missing values
GaringerOzone_Clean <- GaringerOzone %>%
  mutate(
    Daily.Ozone.Clean =
      zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration),
    Daily.AQI.Clean =
      zoo::na.approx(DAILY_AQI_VALUE))

dim(GaringerOzone_Clean)
```

```
## [1] 3652    5
```

Answer: The linear interpolation is the appropriate method since the dataset exhibits a slight linear decreasing trend in the plot depicted in question 7. Therefore, using the piecewise constant method that simply estimates missing values with the use of nearest data values or the spline interpolation that is better fitted for quadratic function might not be appropriate compared to the linear interpolation method.

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)

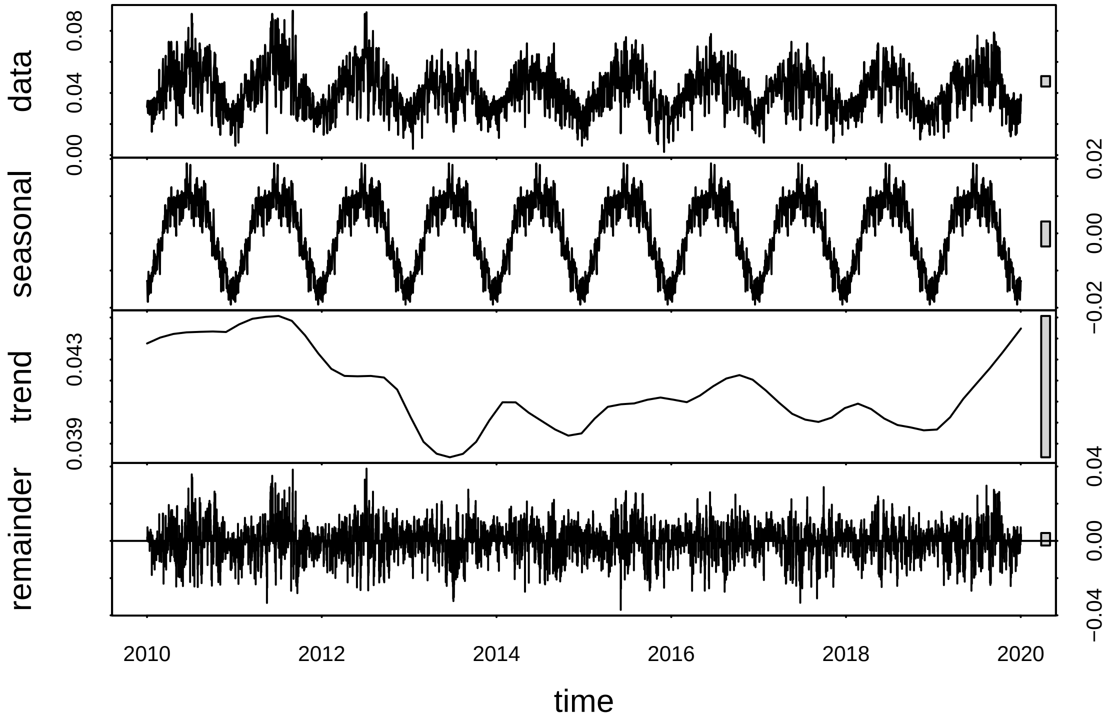
```
#9a Create a new data frame by adding Year and Month columns
GaringerOzone_Clean$Date <- as.Date(GaringerOzone_Clean$Date,"%Y-%m-%d")

GaringerOzone.monthly <- GaringerOzone_Clean %>%
  mutate(Year = year(Date), Month = month(Date)) %>%
  group_by(Year, Month) %>%
  summarize(Monthly.Mean.Ozone = mean(Daily.Ozone.Clean)) %>%
  mutate(Dates = ymd(paste0(Year, "-", 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.

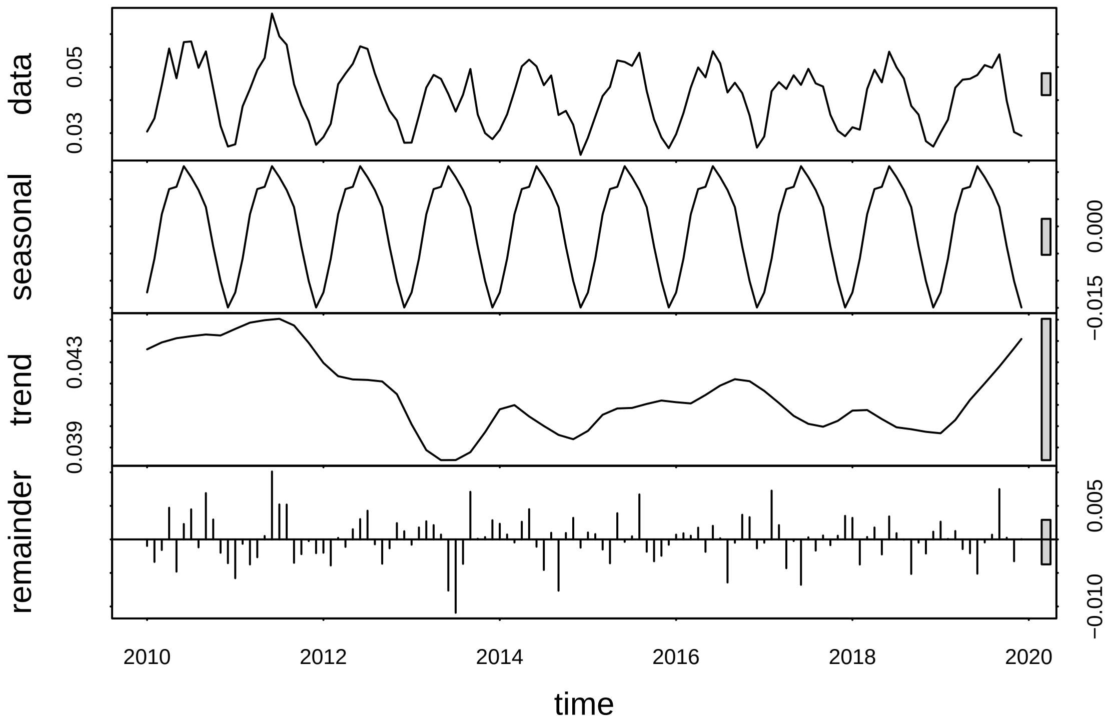
```
#10a Generate the daily time series objects
GaringerOzone.daily.ts <- ts(GaringerOzone_Clean$Daily.Ozone.Clean,
                             start=c(2010,1),
                             frequency = 365)

#10b Generate the monthly time series objects
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$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.

```
#11a Decompose the daily time series objects
GaringerOzone.daily.decom <- stl(GaringerOzone.daily.ts,
                                 s.window="periodic")

#11b Plot the daily time series objects
plot(GaringerOzone.daily.decom)
```



```
#11c Decompose the monthly time series objects
GaringerOzone.monthly.decom <- stl(GaringerOzone.monthly.ts,
                                   s.window="periodic")

#11d Plot the monthly time series objects
plot(GaringerOzone.monthly.decom)
```



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 Run the Mann-Kendall test for the monthly ozone time series
GaringerOzone.monthly.MKtest <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
GaringerOzone.monthly.MKtest
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

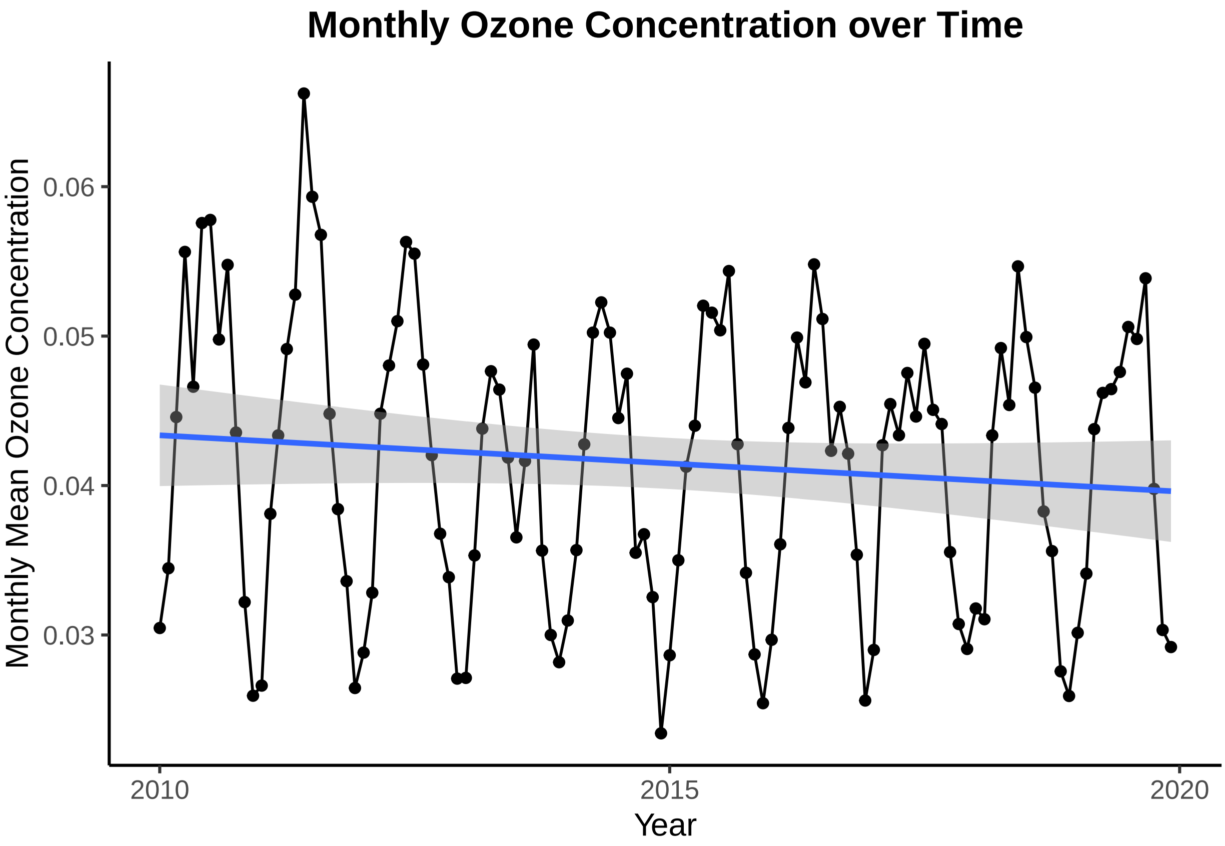
```
summary(GaringerOzone.monthly.MKtest)
```

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

Answer: To analyze the monthly ozone concentration, the seasonal Mann-Kendall test is the most appropriate since it allows for both seasonality, non-parametric and monotonic trend in the dataset. The monthly ozone concentration time series plot in 11d suggests that there is a seasonal variation and non-parametric nature in the monthly ozone concentration dataset. Further, the results of the seasonal Mann-Kendall test indicates that there is a decreasing monotonic trend at tau value -0.143 and it is stastically significant at p-value being less than 0.05. Therefore, the seasonal Mann-Kendall is the best fit test.

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 Create a plot for the mean monthly ozone concentrations over time
ggplot(GaringerOzone.monthly, aes(x=Dates, y=Monthly.Mean.Ozone)) +
  geom_point() +
  geom_line() +
  geom_smooth(method="lm") +
  labs(
    title="Monthly Ozone Concentration over Time",
    y="Monthly Mean Ozone Concentration",
    x="Year") +
  theme(plot.title = element_text(hjust = 0.5, face="bold"))
```



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 research question is to find out if the ozone concentrations changed over the 2010s at the station under study. The season Mann-Kendall test conducted in question 12 and the plot generated in question 13 exhibit the fact that ozone concentration in the studied station has a slight decreasing monotonic trend during 2010s (tau=-0.143, and p-value=0.047). The tau value at -0.143 represents a weak negative trend and it matches with the plot showcasing only a slight and not a strong pronouncing decreasing linear fitted line over time. The p-value at 0.047 rejects the null hypothesis that assumes the ozone concentration over time is stationery and suggests that a weak decreasing trend is statistically significant.

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.

```
#15a Subtract the seasonal component
GaringerOzone.monthly.seasonal <- GaringerOzone.monthly.decom$time.series[,1]

#15b Change to the nonseasonal component
GaringerOzone.monthly.nonseasonal <- GaringerOzone.monthly.ts -
  GaringerOzone.monthly.seasonal

#16 Run the Mann Kendall test on the non-seasonal Ozone monthly series
GaringerOzone.monthly.nonseasonal.MKtest <-
  Kendall::MannKendall(GaringerOzone.monthly.nonseasonal)
GaringerOzone.monthly.nonseasonal.MKtest
```

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

```
summary(GaringerOzone.monthly.nonseasonal.MKtest)
```

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

Answer: When the seasonal component is subtracted and the Mann Kendall test is conducted on the non-seasonal Ozone monthly series, the result still indicates that there is a significant decreasing trend. But this time, both tau value and p-value are more pronounced than the ones using the Seasonal Mann Kendall test on the complete series. The tau value at -0.165 leans more towards the negative value and the p-value at 0.008 is stronger.