Assignment 8: Time Series Analysis

Xianhang Xie

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
# Verify the home directory
getwd()
```

```
## [1] "C:/Users/my/Documents/GitHub/EDA-Spring2023/Assignments"
```

```
# Load the necessary packages
library(tidyverse)
library(lubridate)
library(zoo)
library(trend)
# Set your ggplot theme
custom_theme <- theme(
  plot.background = element_rect(fill = "White"),
  plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
  panel.grid.major = element_line(colour = "gray", linetype = "dashed"),
  panel.grid.minor = element_line(colour = "gray", linetype = "dotted"),
  legend.title = element_text(face = "bold")
)

# Set the custom theme as the default theme
theme_set(custom_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
# Set the directory containing the CSV files
dir_path <- "../Data/Raw/Ozone_TimeSeries/"

# Get a list of CSV files in the directory
csv_files <- list.files(path = dir_path, pattern = "*.csv")
# Paste dir_path into the begining of the file names
csv_files <- paste(dir_path, csv_files, sep = "")

# Import all CSV files into a list of data frames
dfs <- lapply(csv_files, read.csv)

# Combine all data frames into a single data frame
GaringerOzone <- do.call(rbind, dfs)

# Verify the resulting data frame
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
# Convert the date column to a date class
GaringerOzone$Date <- mdy(GaringerOzone$Date)

#4
```

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

#5
# Create a sequence of dates from 2010-01-01 to 2019-12-31
Days <- as.data.frame(seq(as.Date("2010-01-01"), as.Date("2019-12-31"), by = "day"))
colnames(Days) <- "Date"



#6
# Use left_join to combine the data frames
GaringerOzone <- Days %>%
  left_join(GaringerOzone, by = "Date")

# Verify the resulting data frame
dim(GaringerOzone)
```

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

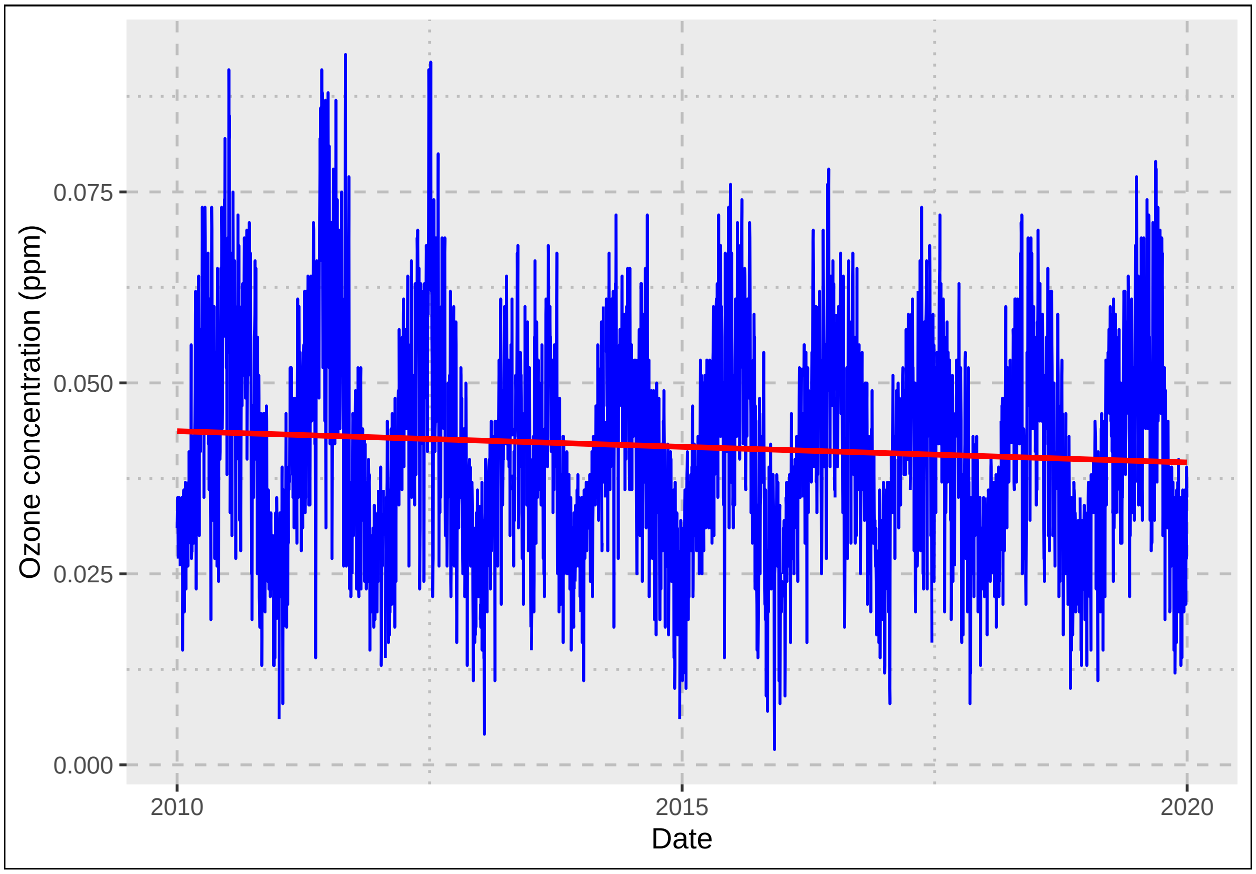
Visualize

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
ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration, group = 1)) +
  geom_line(color = "blue") +
  geom_smooth(aes(y = Daily.Max.8.hour.Ozone.Concentration), method = "lm", se = FALSE, color = "red")
  labs(x = "Date", y = "Ozone concentration (ppm)")
```

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

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



Answer: The resulting plot show a line plot of ozone concentrations over time with a red smoothed line showing any linear trend of the data. It appears that there is a slight decreasing average trend from 2010 to 2019. But the ozone concentrations also seems to be periodic.

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.Co
```

Answer: As for why we did not use a piecewise constant or spline interpolation, linear interpolation is frequently the most straightforward and fast way for filling in missing data, particularly when the data is pretty smooth and the gaps are very short. Piecewise constant interpolation may provide results that are jagged or unnatural, whereas spline interpolation may create false fluctuations or overfit the data. Consequently, linear interpolation is frequently the method of choice in such situations.

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
# Add columns for year and month to form the groupings
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 with each month-year combination being set as the first day of the month
GaringerOzone.monthly$Date <- as.Date(paste(GaringerOzone.monthly$year, GaringerOzone.monthly$month, "0
```

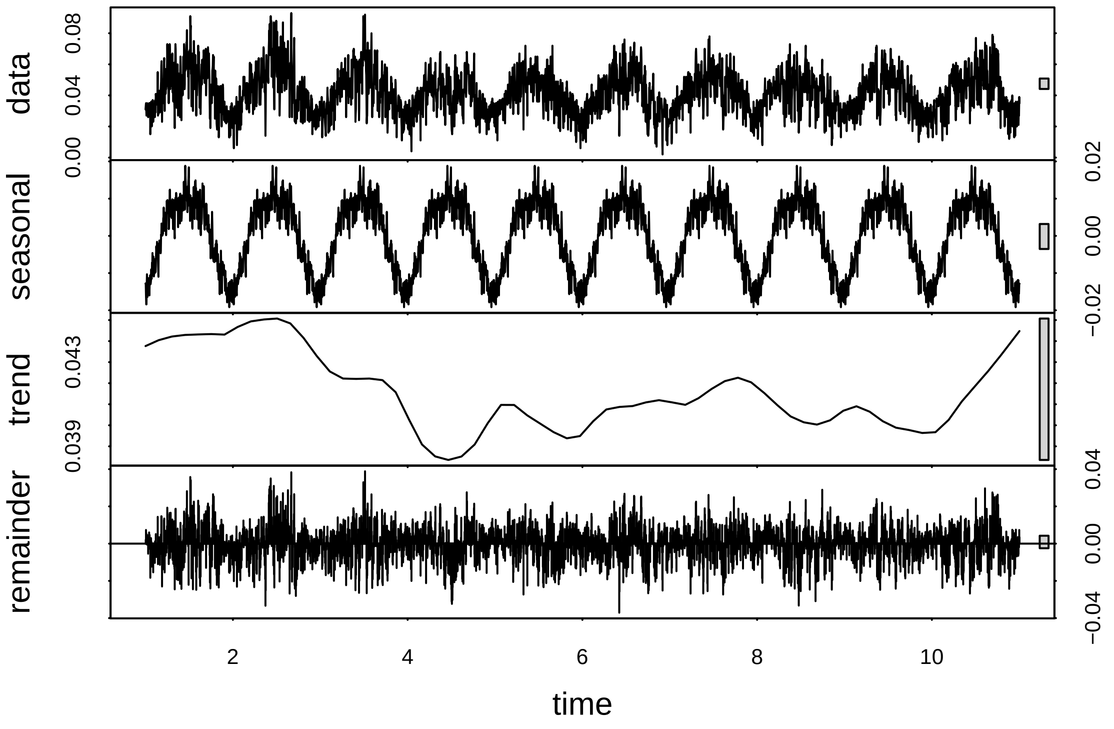
10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

```
#10
GaringerOzone.daily.ts <- zoo(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration, GaringerOzone$Date)
GaringerOzone.daily.ts <-
  window(GaringerOzone.daily.ts,
         start = as.Date("2010-01-01"),
         end = as.Date("2019-12-31"))
GaringerOzone.daily.ts = ts(GaringerOzone.daily.ts, frequency = 365)

# Create a time series object for monthly average ozone values
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone, start = c(2010, 1), end = c(2019, 12),
```

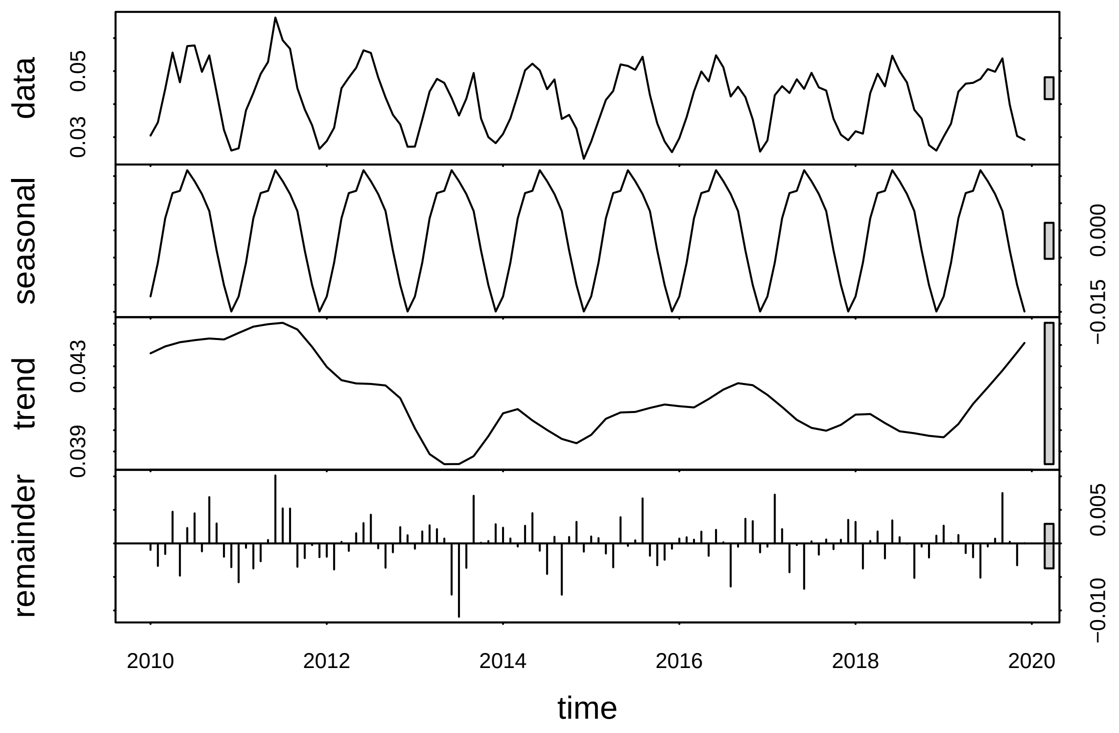
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11
# Decompose the daily time series object
decomp_daily <- stl(GaringerOzone.daily.ts, s.window = "periodic")
# Plot the daily decomposition
plot(decomp_daily)
```



```
# Decompose the monthly time series object
decomp_monthly <- stl(GaringerOzone.monthly.ts, s.window = "periodic")

# Plot the monthly decomposition
plot(decomp_monthly)
```



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 SMK test
wind_data_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results
wind_data_trend1
```

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

```
summary(wind_data_trend1)
```

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

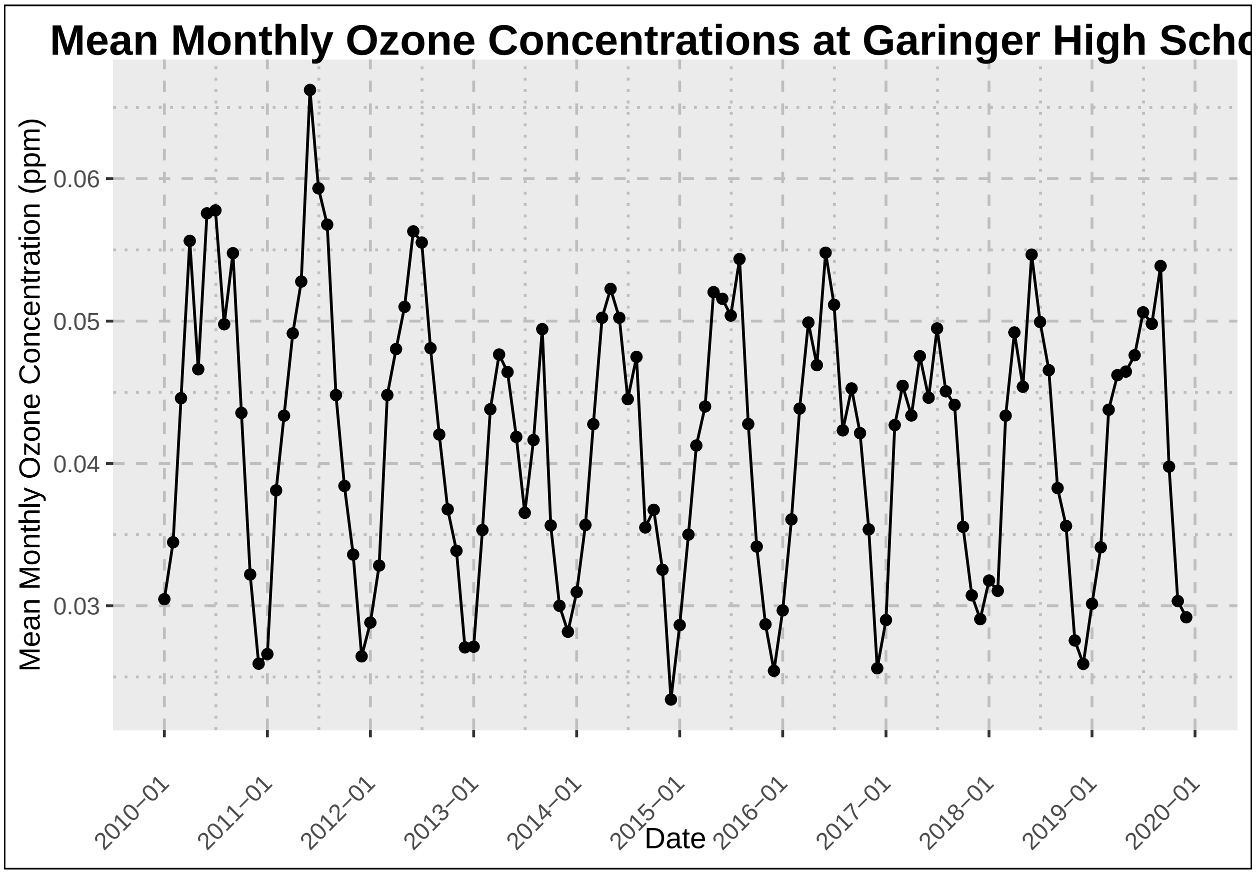
Answer: The seasonal Mann-Kendall test goes beyond the Mann-Kendall test by taking into account how the data changes with the seasons. It looks at the seasonal parts of the time series instead of the whole series to see if there is a steady trend. By taking seasonal patterns into account, the test is more sensitive to changes in trend that happen at certain times of the year.

In the monthly ozone series, the data are likely to show seasonal patterns because of things like changes in weather and atmospheric conditions. So, the seasonal Mann-Kendall test is a good way to look for trends in these numbers.

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 new data frame with mean monthly ozone concentrations
GaringerOzone.monthly.mean <- aggregate(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
             by = list(year = format(GaringerOzone$Date, "%Y"),
             month = format(GaringerOzone$Date, "%m")), FUN = mean)
GaringerOzone.monthly.mean$Date <- as.Date(paste(GaringerOzone.monthly.mean$year,
        GaringerOzone.monthly.mean$month,"01", sep = "-"))

# Create a plot with both a point and line layer
ggplot(GaringerOzone.monthly.mean, aes(x = Date, y = x)) +
  geom_point() +
  geom_line() +
  labs(x = "Date", y = "Mean Monthly Ozone Concentration (ppm)",
       title = "Mean Monthly Ozone Concentrations at Garinger High School") +
  scale_x_date(date_labels = "%Y-%m", date_breaks = "1 year")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



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: From 2010 to 2019, the average amount of ozone at Garinger High School in North Carolina was looked at. The time series plot shows that the amount of ozone changes with the seasons, with peaks in the summer and lower levels in the winter. In 2010, the average monthly ozone levels were around 0.044 ppm. In 2019, they are around 0.038 ppm.

We did a seasonal Mann-Kendall test to see if the amount of ozone in the air changed in the same way each month. The test showed that the monthly ozone concentrations at the Garinger High School station are getting higher (tau = - 0.143, p-value = 0.046724<0.05), which means that the ozone concentrations have gone down at this station in the 2010s.

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
# Subtract the seasonal component from the monthly time series
non_seasonal <- GaringerOzone.monthly.ts - as.data.frame(decomp_monthly$time.series)$seasonal
#16
mk_test_non_seasonal <- Kendall::MannKendall(non_seasonal)
summary(mk_test_non_seasonal)
```

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

Answer: Comparing the results of the two tests, after removing the seasonal effect, the p value of the Mann Kendall decreased from 0.048 to 0.0075. So when we remove the seasonal effect, the result trend becomes quite clear.