Assignment 8: Time Series Analysis

Zhenghao Lin

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
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.

```r
#1
setwd("/Users/lzh/Desktop/EDE_Fall2023/Data/Raw/Ozone_TimeSeries")
file_list <- list.files(pattern = ".csv")
GaringerOzone <- data.frame()

for (file in file_list) {
  dataset <- read.csv(file)
  GaringerOzone <- rbind(GaringerOzone, dataset)
}

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.

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

# 4
GaringerOzone <- 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"), by = "days"))
colnames(Days) <- "Date"


# 6
GaringerOzone <- left_join(Days, GaringerOzone)
```

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

```
dim(GaringerOzone)
```

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

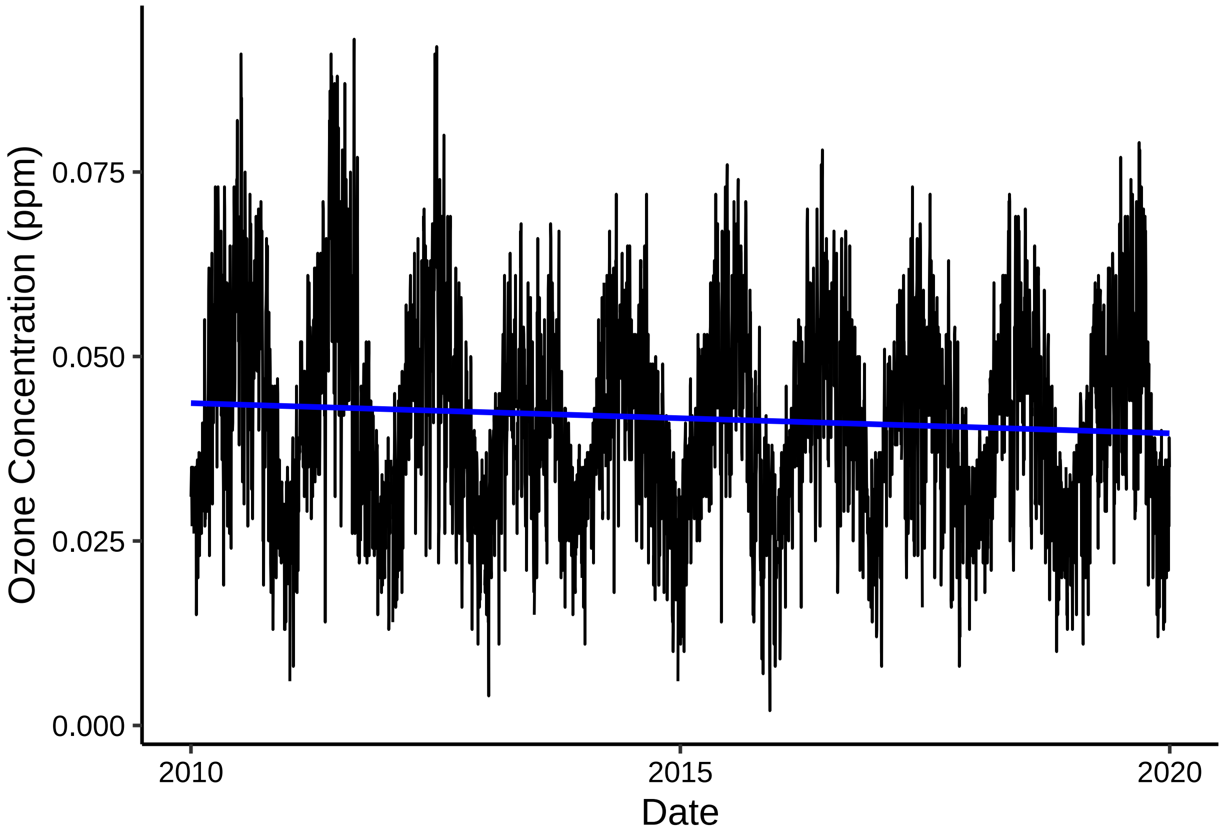
Visualize

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
 ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +  # Add a line plot
  geom_smooth(method = "lm", se = FALSE, color = "blue") +  # Add a linear trend line
  labs(x = "Date", y = "Ozone Concentration (ppm)")
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

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



Answer: There is a trend in ozone concentration over time as the smoothed line is going downward.

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 %>%
  mutate(Concentration_Clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: Based on the plot we generated in #7, there is a trend. Piecewise Constant assumes that the missing values are equal to the nearest known data point. It's suitable for data that you believe to be constant between observations. Spline interpolation uses a piecewise polynomial function to approximate the data between known points. While both may provide a more ideal number, as there is a simple linear trend, no need to us piecewise and spline to make the question and process to complex.

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
GaringerOzone.monthly <- GaringerOzone_clean %>%
  mutate(year = year(Date),
         month = month(Date))

GaringerOzone.monthly <- GaringerOzone.monthly %>%
  mutate(Date = make_date(year, month, day = 1))

# Group by year and month, then calculate the mean ozone concentration
GaringerOzone.monthly <- GaringerOzone.monthly %>%
  group_by(Date) %>%
  summarize(MeanOzoneConcentration = mean(Daily.Max.8.hour.Ozone.Concentration, na.rm = TRUE))
```

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 <- ts(GaringerOzone_clean$Concentration_Clean,
                             start = c(2010, 1),
                             end = c(2019, 365),
                             frequency = 365)

# Create a time series object for monthly data
GaringerOzone_monthly_ts <- ts(GaringerOzone.monthly$MeanOzoneConcentration,
                               start = c(2010, 1),
                               end = c(2019, 12),
                               frequency = 12)
```

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

```
#11
daily_decomp <- stl(GaringerOzone_daily_ts, s.window = "periodic")
monthly_decomp <- stl(GaringerOzone_monthly_ts, s.window = "periodic")
```

12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

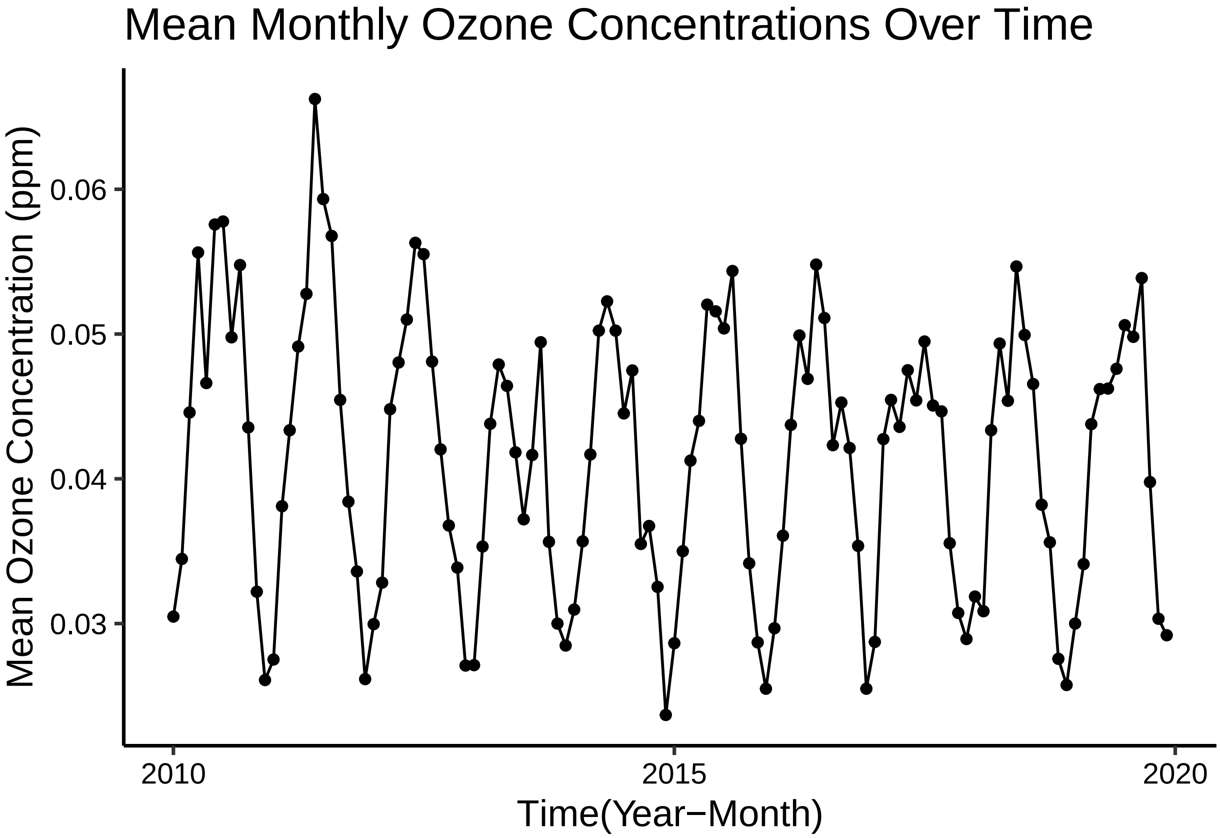
```
#12

Ozone_monthly_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone_monthly_ts)
summary(Ozone_monthly_trend1)
```

```
## Score =  -88 , Var(Score) = 1498
## denominator =  538.9944
## tau = -0.163, 2-sided pvalue =0.022986
```

> Answer: Based on the plots before, we can tell that the Ozone concentration exhibits certain levels of seasonality, and SMK specifically addresses trends in ts data that have seaonality. What's more, SMK test is a non-parametric test, which means it doesn't assume a specific distribution for the data.

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
ggplot(GaringerOzone.monthly, aes(x = Date, y = MeanOzoneConcentration)) +
  geom_point() +
  geom_line() +
  labs(x = "Time(Year-Month)", y = "Mean Ozone Concentration (ppm)") +
  ggtitle("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: There is a clear trend in the mean monthly ozone concentrations at Garinger High School in North Carolina from 2010 to 2019. The plot of mean monthly ozone concentrations over time shows a declining trend, suggesting a decrease in ozone concentrations over the years. This observation is supported by the seasonal Mann-Kendall trend analysis, which yielded a statistically significant negative tau value of -0.163 with a p-value = 0.022986. This negative tau value indicates a decreasing trend in ozone concentrations over time, further confirming the downward trend observed in the plot.

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

seasonal_component <- monthly_decomp$time.series[, "seasonal"]

# Subtract the seasonal component from the original time series
non_seasonal_ozone_monthly_ts <- GaringerOzone_monthly_ts - seasonal_component
summary(non_seasonal_ozone_monthly_ts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02810 0.03938 0.04128 0.04152 0.04338 0.05509
```

```
#16
MKtest <- Kendall::MannKendall(non_seasonal_ozone_monthly_ts)
summary(MKtest)
```

```
## Score =  -1278 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.179, 2-sided pvalue =0.0037728
```

Answer: In this case: The non-seasonal Mann-Kendall test gives a more negative score, indicating a stronger downward trend, compared to the seasonal Mann-Kendall test. The non-seasonal Mann-Kendall test also has a lower p-value = 0.0037728, compared to the seasonal Mann-Kendall test 0.022986.