Assignment 7: Time Series Analysis

Kelly Davidson

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A07_TimeSeries.Rmd") prior to submission.

The completed exercise is due on Tuesday, March 16 at 11:59 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot 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 data frame named `GaringerOzone` of 3589 observation and 20 variables.

```
# 1. Setting up my session
getwd()  #checking working directory
```

```
## [1] "/home/guest/EDA-Fall2022/EDA-Fall2022"
```

```
library(tidyverse)  #loading tidyverse
library(lubridate)  #loading lubridate
library(trend)  #loading trend
library(zoo)  #loading zoo
library(Kendall)  #loading Kendall
library(tseries)  #loading tseries
library(plyr)  #loading plyr
library(dplyr)  #loading dplyr
library(readr)  #loading readr
library(ggplot2)

# Building a ggplot theme & setting it as my default theme
A07_theme <- theme_light(base_size = 14) + theme(axis.text = element_text(color = "dark gray"),
    legend.position = "right")
```

```
theme_set(A07_theme)

# 2. Importing the 10 datasets from the Ozone_TimeSeries folder in bulk
Ozonefiles = list.files(path = "./Data/Raw/Ozone_TimeSeries", pattern = ".csv", full.names = TRUE)
Ozonefiles  #checking to be sure all 10 datasets have been read
```

```
##  [1] "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv"
##  [2] "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv"
##  [3] "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv"
##  [4] "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv"
##  [5] "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv"
##  [6] "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv"
##  [7] "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv"
##  [8] "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv"
##  [9] "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv"
## [10] "./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv"
```

```
GaringerOzone = ldply(Ozonefiles, read_csv)  #combining the 10 datasets into 1 data frame
dim(GaringerOzone)  #checking the dimensions of this new data frame
```

```
## [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. Setting the Date column of the data frame as a date class
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

# 4. Wrangling the dataset to only include the 3 columns specified above
colnames(GaringerOzone)[colnames(GaringerOzone)
                        == "Daily Max 8-hour Ozone Concentration"] <- "Daily.Max.8.hour.Ozone.Concentra
                  #renaming the Daily Max 8-hour Ozone Concentration column name

GaringerOzone_wrangled <-
  GaringerOzone %>%  #wrangling the data via a pipe
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE) #selecting only the
                                                                        #columns specified

#5. Generating a new daily data frame from the wrangled dataset that contains a sequence
#of dates from 2010-01-01 to 2019-12-31
Days <- as.data.frame(seq.Date(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"), by = "day"))
colnames(Days) <- c("Date") #renaming the column to Date
```

```
#6. Using 'left_join' to combine the Days & GaringerOzone_wrangled data frames
GaringerOzone.Daily <- left_join(Days, GaringerOzone_wrangled)
```

```
## Joining, by = "Date"
```

```
dim(GaringerOzone.Daily) #checking the dimensions of this new data frame
```

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

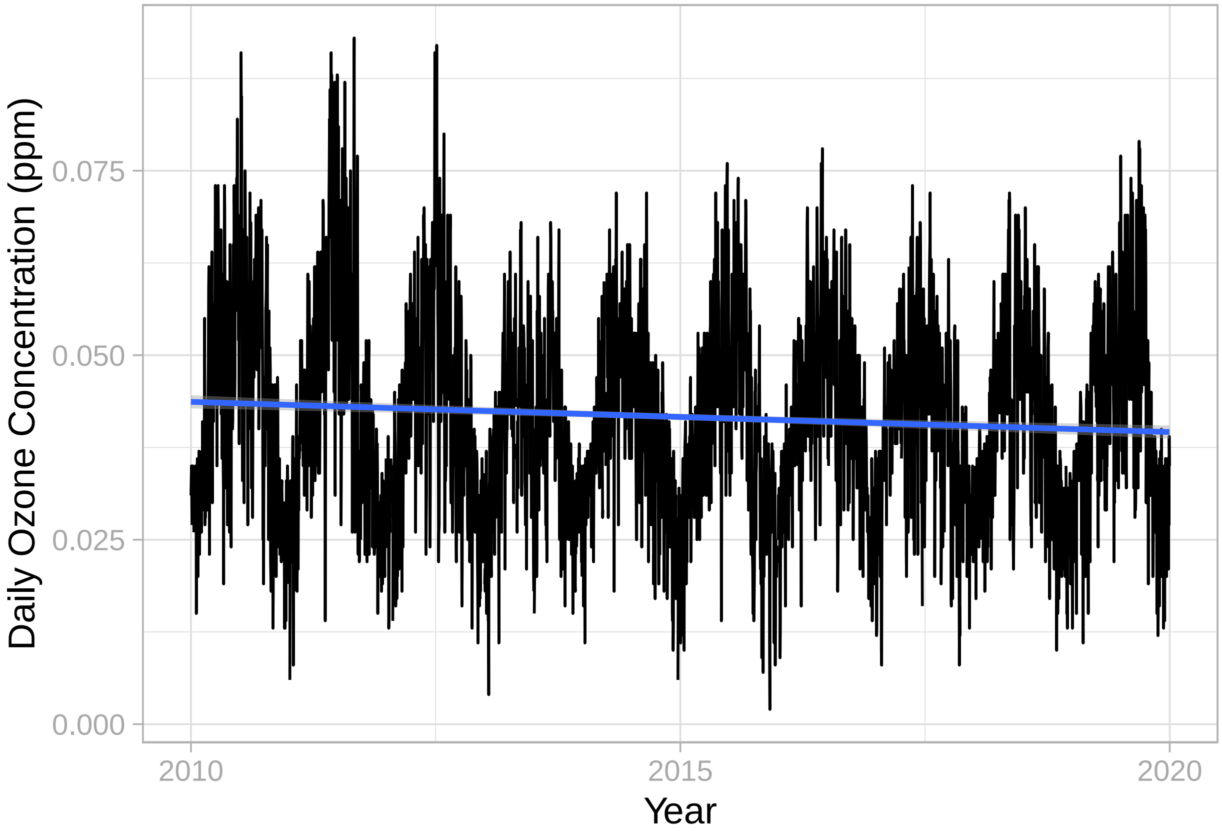
Visualize

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. Creating a line plot depicting ozone concentrations over time
DailyOzone_Concentration_plot <-
  ggplot(GaringerOzone.Daily, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
                                                   #assigning x & y values
  geom_line() + #specifying a line plot
  geom_smooth(method = lm) + #adding a smoothed line to show potential linear trend
  ylab("Daily Ozone Concentration (ppm)") + #renaming the y-axis
  xlab("Year")
print(DailyOzone_Concentration_plot)
```

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

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



Answer: In analyzing the line graph that plots daily ozone concentrations in ppm over time, there is a very slight negative trend present from 2010 to 2020. This graph also depicts the seasonality of ozone concentrations each year that seem to peak over spring and summer and decline during the winter months.

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. Using a linear interpolation to fill in missing daily data for ozone
# concentration
summary(GaringerOzone.Daily$Daily.Max.8.hour.Ozone.Concentration)  #checking to see how many
```

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

```
# na's are in the ozone concentration column (63 total)

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

summary(GaringerOzone.Daily$Daily.Max.8.hour.Ozone.Concentration)  #checking to make sure there
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

```
# are no longer any na's in the ozone concentration column - meaning the linear
# interpolation worked correctly
```

Answer: In a linear interpolation, any gaps or missing data is assumed to fall between the previous and following measurement. This is also referred to as a 'connect the dots' approach for interpolation. In this example, a linear interpolation was appropriate as opposed to a piecewise constant or spline interpolation because there were only 63 missing values (na's) across 3652 total observations. This method ensured there was not an over or under estimation of those 63 missing values.

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)

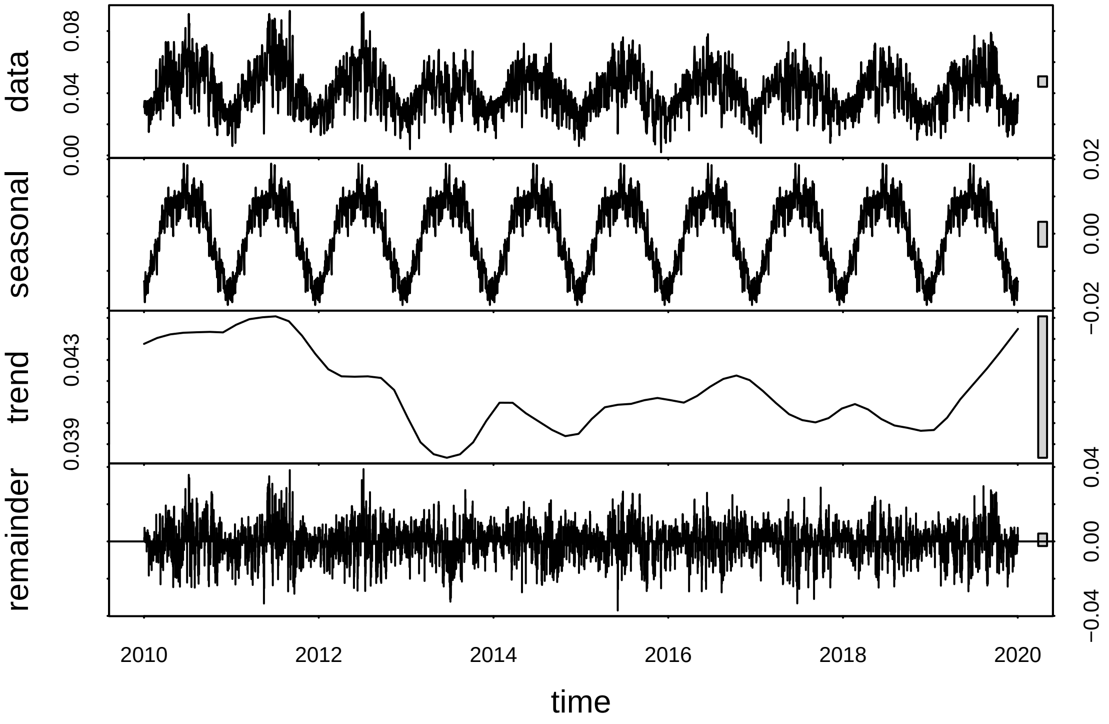
```
#9. Creating a new data frame that contains mean ozone concentration for each month
GaringerOzone.Monthly <- GaringerOzone.Daily %>% #creating the new data frame via a pipe
  mutate(Month = month(Date)) %>% #adding a month column
  mutate(Year = year(Date)) %>%  #adding a year column
  mutate(Month_Year = my(paste(Month, "-", Year))) %>% #combining the month and year into a new column
  group_by(Month_Year) %>% #setting the order of the month-year combination column
                                          #mainly for graphing purposes
  dplyr::summarize(Mean_Monthly_O3 = mean(Daily.Max.8.hour.Ozone.Concentration))
                          #calculating the monthly mean ozone concentrations
```

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.

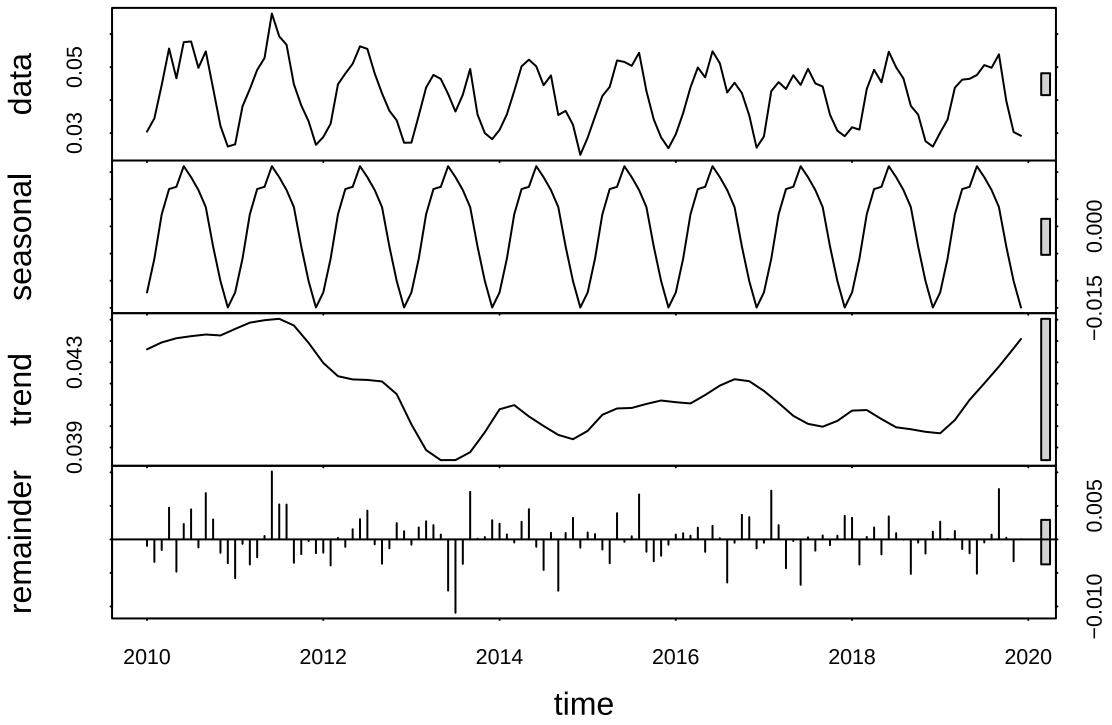
```r
# 10. Generating two time series objects for ozone concentrations over time -
# daily and monthly
GaringerOzone.Daily.ts <- ts(GaringerOzone.Daily$Daily.Max.8.hour.Ozone.Concentration,
    start = c(2010, 1), frequency = 365)

GaringerOzone.Monthly.ts <- ts(GaringerOzone.Monthly$Mean_Monthly_03, start = c(2010,
    1), frequency = 12)
```

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

```r
# 11. Decomposing the daily and monthly time series objects and plotting the
# components
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. Running a monotonic trend analysis, specifically the seasonal
# Mann-Kendall, for the monthly ozone concentration time series
GaringerOzone.Monthly.Seasonal.MannKendall <- Kendall::SeasonalMannKendall(GaringerOzone.Monthly.ts)
summary(GaringerOzone.Monthly.Seasonal.MannKendall)
```

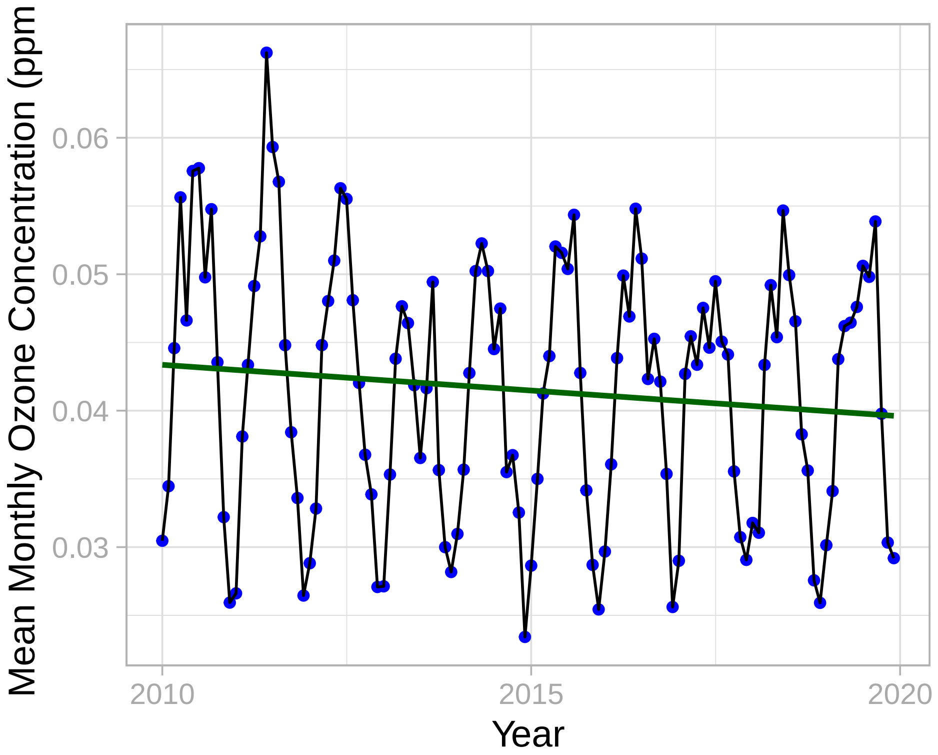
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: In this scenario, running the seasonal Mann-Kendall trend analysis is the most appropriate (as opposed to other methods: linear regression, Mann-Kendall, and Spearman Rho) because although the data is monotonic, it is not linear. In viewing the monthly ozone decomposition plot, you can see that seasonality explains much of the variation in ozone concentration month-to-month. This is emphasized in the seasonal plot by the small grey box to the right of the plot. Since there is distinct seasonality in our data, the other methods of trend analysis are not appropriate since they cannot detect 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. Creating a point & line plot depicting mean monthly ozone concentrations over time
Ozone_monthly_plot <-
  ggplot(GaringerOzone.Monthly, aes(x = Month_Year, y = Mean_Monthly_03)) + #assigning x & y values
  geom_point(color = "blue") + #adding blue points
  geom_line() + #adding a line to connect points
  geom_smooth(method = "lm", se = FALSE, color = "dark green") + #adding a green trend line
  xlab("Year") + #renaming the x-axis
  ylab("Mean Monthly Ozone Concentration (ppm)") #renaming the y-axis
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: In terms of the research question, ozone concentrations have changed over the 2010s at this monitoring station. The p-value in this scenario is 0.046724, which althought it is very close to 0.05, it is still less than 0.05. As a result, we must reject the null hypothesis that states the data is stationary. Further, there is a trend in mean monthly ozone concentrations in ppm from 2010 to 2020. The plot suggests a slight negative trend in ozone concentrations over time from ~0.043ppm in 2010 to ~0.039ppm in 2020 shown via the trendline.

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. Subtracting the seasonal component from the monthly ozone concentration
# time series
GaringerOzone.Monthly.Nonseasonal <- GaringerOzone.Monthly.ts - GaringerOzone.Monthly.Decomp$time.series
    1]

# 16. Running the Mann-Kendall test on the non-seasonal monthly ozone
# concentration time series
```

```
GaringerOzone.Monthly.MannKendall <- Kendall::MannKendall(GaringerOzone.Monthly.Nonseasonal)
summary(GaringerOzone.Monthly.MannKendall)
```

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

> Answer: In comparing the results of the Mann-Kendall test on the non-seasonal monthly ozone series to the seasonal Mann-Kendall test on the complete monthly ozone series, the Mann-Kendall test on the non-seasonal ozone series provided a much lower p-value (0.0075 versus 0.046724). Since this p-value is much less than 0.05, we must reject the null hypothesis that the data is stationary and conclude that there is a trend present in monthly ozone concentration over time. In addition, since this second Mann-Kendall test did not include the seasonality component, it suggests that there is a significant trend in the time series that is not due to seasonality.