Assignment 8: Time Series Analysis

David Liddle

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
#Check working directory
getwd()
```

```
## [1] "/Users/davidliddle/Documents/EDA-Spring2023"
```

```r
#Load necessary packages
library(tidyverse);library(lubridate);library(here);library(ggplot2);
library(cowplot);library(knitr); library(agricolae);library(dplyr);
library(zoo); library(Kendall); library(tseries)
# Build ggplot theme
david_theme <- theme_classic(base_size = 14) +
theme(axis.text = element_text(color = "black"),
legend.position = "right",
panel.grid.minor = element_line(color = "gray", linetype = "solid"),
panel.grid.major = element_line(color = "gray", linetype = "solid"),
legend.background = element_rect(fill = "gray"))
#Set the theme
theme_set(david_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.

```r
#1 Load datasets from Ozone_TimeSeries folder
#Create a list of filepaths for all the raw data
file_paths <- list.files(here("Data/Raw/Ozone_TimeSeries"),
                         pattern = "\\.csv$", full.names = TRUE)
#Read data and combine them into a single data frame
GaringerOzone <- bind_rows(lapply(file_paths, read.csv,
                                  stringsAsFactors = TRUE))
# I used chatGPT to find the lapply function
```

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.

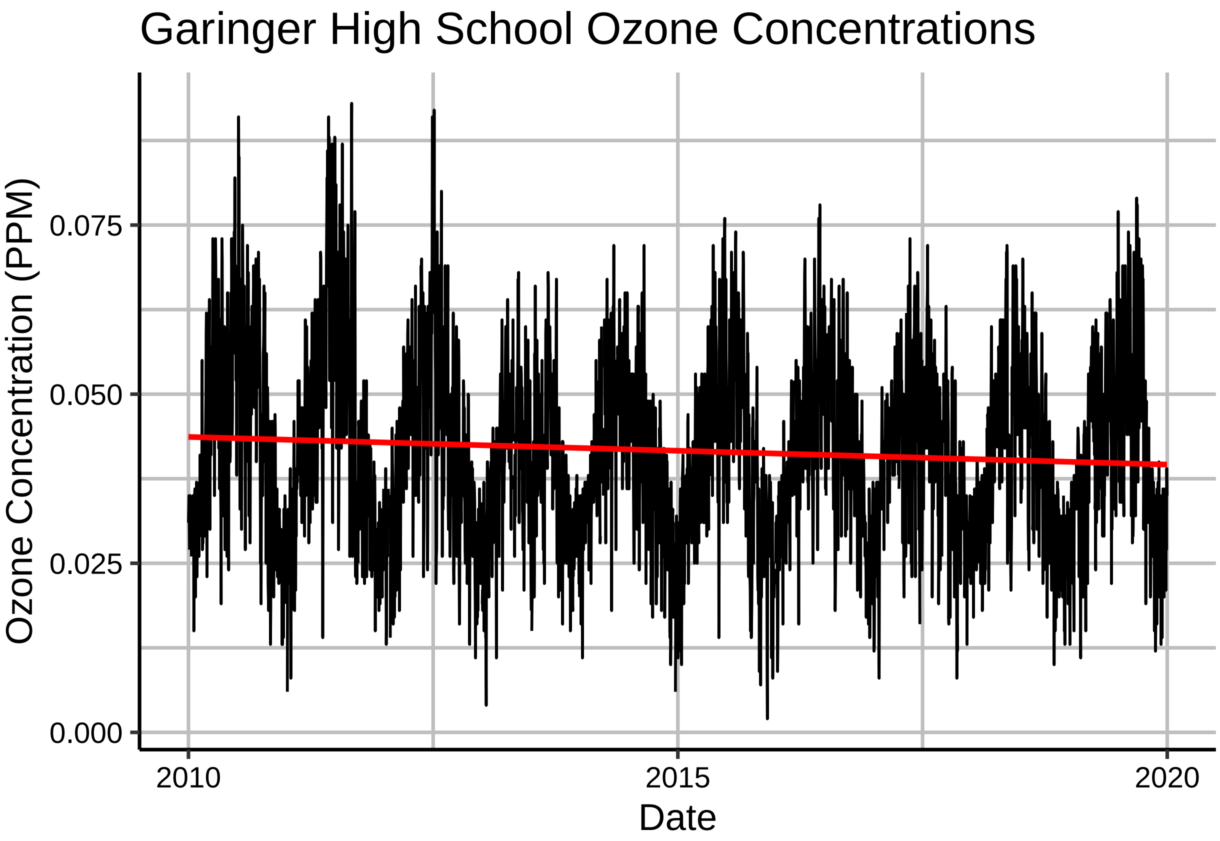
```r
# 3 Set our date column as a date class
GaringerOzone <- GaringerOzone %>% mutate(Date = mdy(Date))
#4 Wrangle GaringerOzone dataset
GaringerOzone_wrangle <- GaringerOzone %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
# 5 Create a sequence of dates from "2010-01-01" to "2019-12-31"
Days <- as.data.frame(seq(from = as.Date("2010-01-01"),
                          to = as.Date("2019-12-31"), by = "1 day"))
# Rename the column to "Date"
colnames(Days) <- "Date"
# 6 Combine data frames
GaringerOzone_1 <- left_join(Days, GaringerOzone_wrangle, by="Date")
```

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 plot
plot <- ggplot(GaringerOzone_1,
               aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
# Create line graph
geom_line() +
geom_smooth(method = "lm", se = FALSE, col = "red") +
  labs(title = "Garinger High School Ozone Concentrations",
       x = "Date",
       y = "Ozone Concentration (PPM)") # Name axes
print(plot)
```

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



Answer: The line graph sugggests that ozone levels fluctuate every year reaching a high value of around 0.075 ppm and a low value of 0.012 ppm. The trend line also indicates a very slight decrease in the average ozone levels from 2010 to 2020.

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 Add linear interpratation to missing data
GaringerOzone_clean <-
  GaringerOzone_1 %>%
  mutate(Daily.Max.8.hour.Ozone.Concentration =
           zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: The linear interpolation would be the best fit for this data rather than a piecewise constant because ozone levels fluctuate between different readings. Therefore we can assume that there would be an intermediate value between one ozone reading and the next. The linear interpolation picks the two data points and extrapolates an intermediate value between them. The piece-wise interpolation takes the two data points and fills the intermediate value with the closest value. In our data, this would lead to the data being skewed and producing more abrupt changes in the data. The spline interpolation is similar to the linear interpolation, but it uses a quadratic equation to joint the two data points together. In our data, this coudl potentially interoduce variablility to the data.

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 Create new data frame
GaringerOzone.monthly <- GaringerOzone_clean %>%
    mutate(Year = year(Date), Month = month(Date)) %>% #Ask about this in class
 group_by(Year, Month) %>%

# Calculate the mean ozone concentration for each month
  summarize(Mean_Ozone = mean(Daily.Max.8.hour.Ozone.Concentration)) %>%

# Create a new Date column with the first day of each month
  mutate(Date = ymd(paste(Year, Month, "01", sep = "-")))
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

```
# Arrange the data by Date if needed
GaringerOzone.monthly <- arrange(GaringerOzone.monthly, Date)

# View the resulting data frame
head(GaringerOzone.monthly)
```

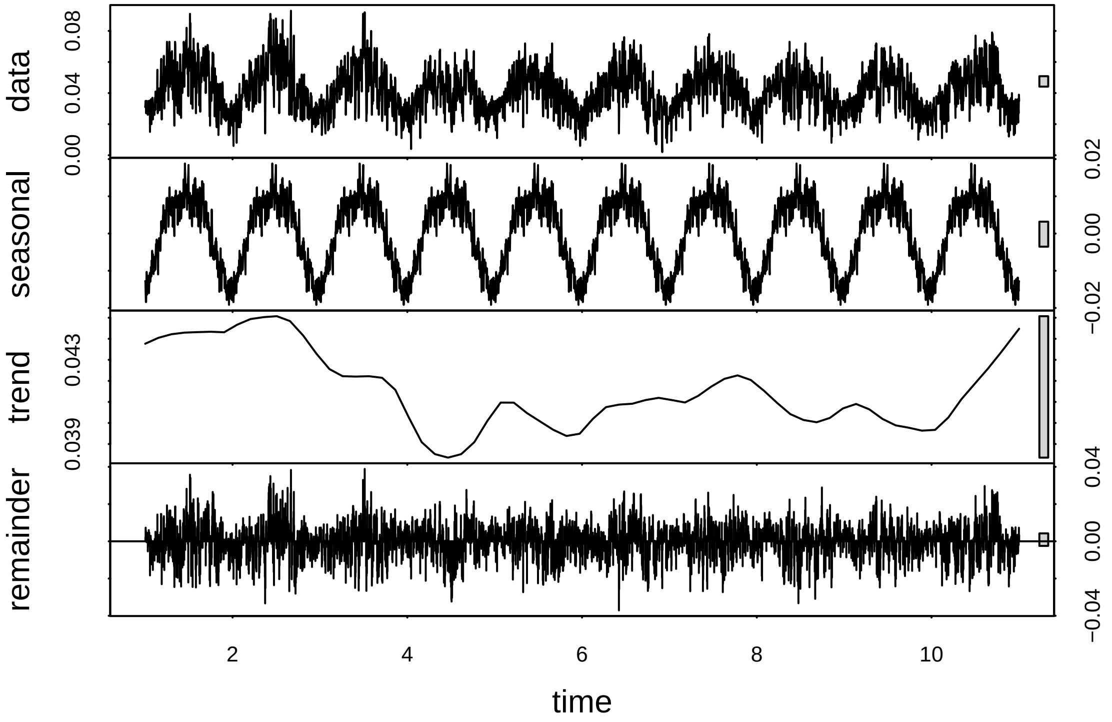
```
## # A tibble: 6 x 4
## # Groups:   Year [1]
##    Year Month Mean_Ozone Date
##   <dbl> <dbl>      <dbl> <date>
## 1  2010     1     0.0305 2010-01-01
## 2  2010     2     0.0345 2010-02-01
## 3  2010     3     0.0446 2010-03-01
## 4  2010     4     0.0556 2010-04-01
## 5  2010     5     0.0466 2010-05-01
## 6  2010     6     0.0576 2010-06-01
```

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.

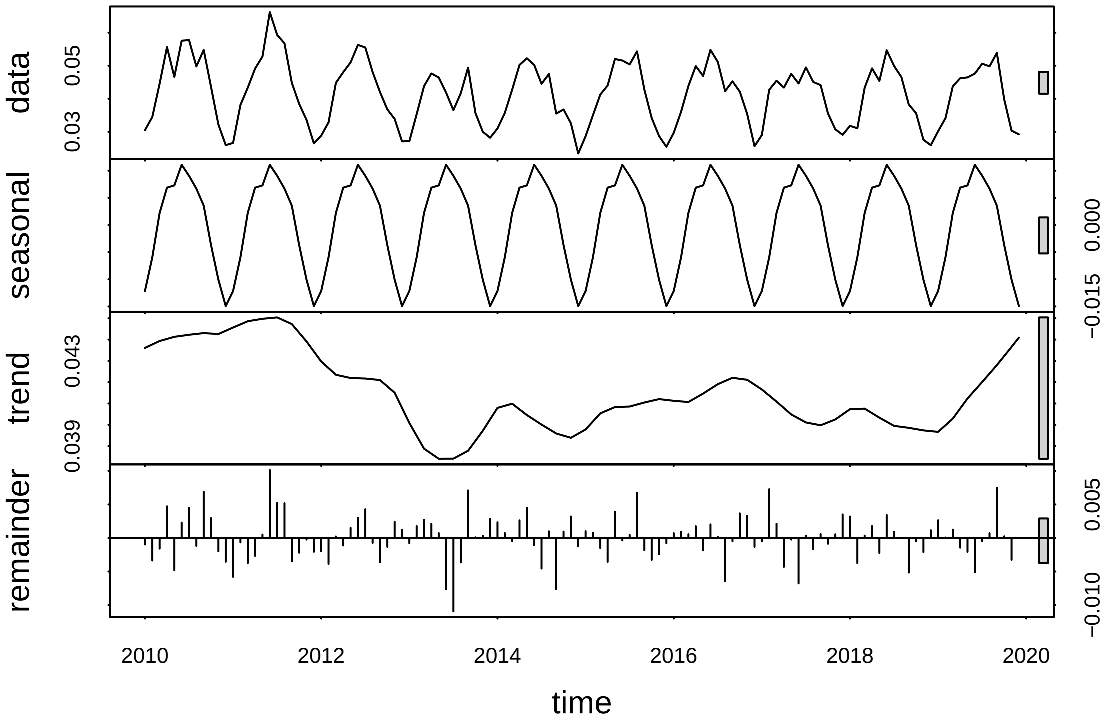
```r
#10 Generate time series for daily
f_date <- day(first(GaringerOzone_clean$Date))
GaringerOzone.daily.ts <- ts(GaringerOzone_clean$Daily.Max.8.hour.Ozone.Concentration,
                   start= f_date,
                   frequency=365)
#10 Generate time series for monthly
f_month <- month(first(GaringerOzone.monthly$Date))
f_year <- year(first(GaringerOzone.monthly$Date))
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.

```r
#11 Decompose daily time series
Ozone.daily_decomp <- stl(GaringerOzone.daily.ts,s.window = "periodic")
plot(Ozone.daily_decomp)
```



```r
# Decompose monthly time series
Ozone.monthly_decomp <- stl(GaringerOzone.monthly.ts,s.window = "periodic")
plot(Ozone.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
# Run SMK test
Ozone_data_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

# Inspect results
Ozone_data_trend
```

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

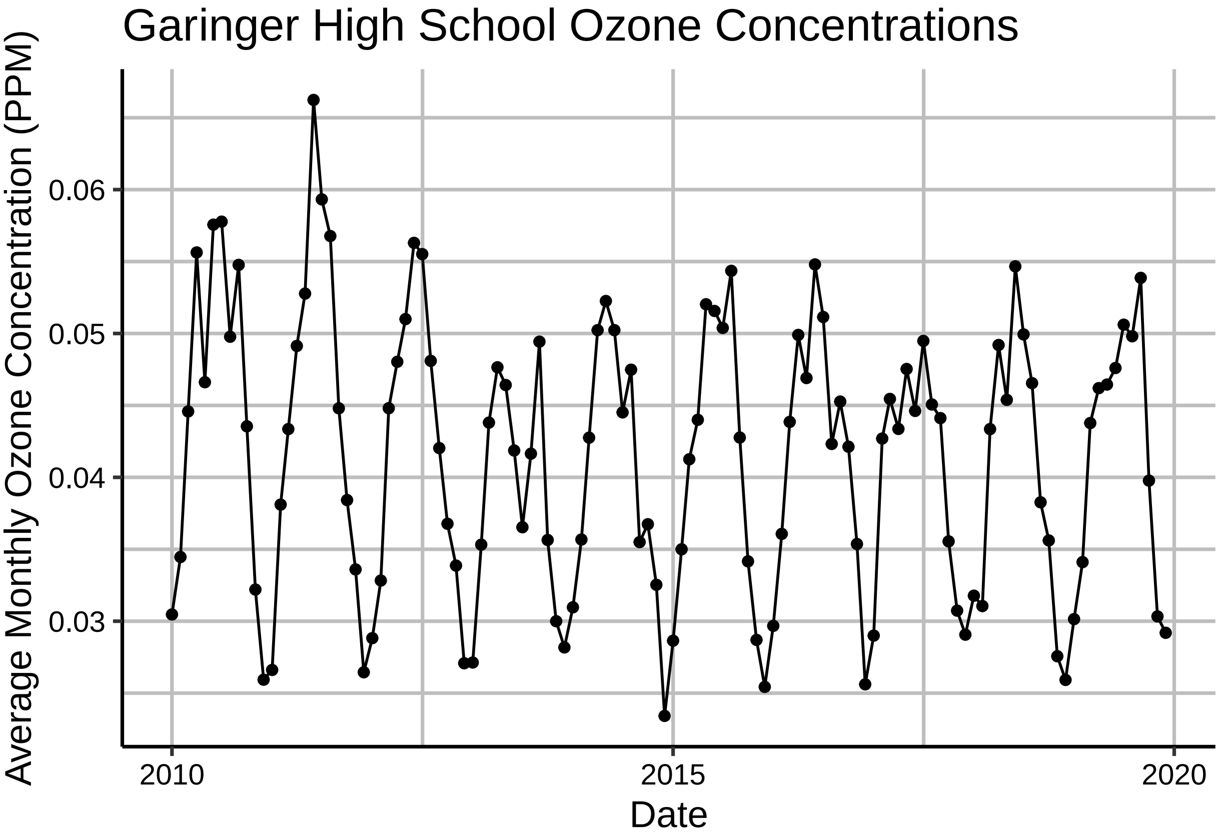
```
summary(Ozone_data_trend)
```

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

Answer: The Seasonal Mann-Kendall Test is the most appropriate trend analysis method as it can detect trends in time series data while accounting for seasonal variations or periodic components. It is particularly useful in situations where you want to assess whether there is a monotonic (increasing or decreasing) trend in a time series with a clear seasonal pattern or cyclic behavior.

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
plot2 <- ggplot(GaringerOzone.monthly, aes(x = Date, y = Mean_Ozone)) +
# Create scatterplot
geom_point() + geom_line() +
  labs(title = "Garinger High School Ozone Concentrations",
       x = "Date",
       y = "Average Monthly Ozone Concentration (PPM)") # Name axes
print(plot2)
```



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 monthly ozone levels followed a consistent pattern of fluctuation during the decade long study period. The results of the Seasonal Mann-Kendall test indicated a slight negative trend in the average monthly ozone levels from 2010 to the end of the 2019 (tau = -0.143) with a 2-sided p-value of 0.046724, which is slightly below our alpha value of 0.05, indicated that the results are 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.

```r
#15
# We can extract the components and turn them into data frames
GaringerOzone.monthlycomponents <- as.data.frame(Ozone.monthly_decomp$time.series[,1:3])
GaringerOzone.monthlycomponents <- mutate(GaringerOzone.monthlycomponents,
                                          Observed = GaringerOzone.monthly$Mean_Ozone,
                                          Date = GaringerOzone.monthly$Date,
                                          Nonseason = Observed - seasonal)

Monthly_components.ts <- ts(GaringerOzone.monthlycomponents$Nonseason,
                            start=c(f_year,f_month),
                            frequency=12)

#16 Running a non-seasonal Mann-Kendall trend analysis
# Run SMK test
Ozone_data_trend2 <- Kendall::MannKendall(Monthly_components.ts)

# Inspect results
Ozone_data_trend2
```

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

```r
summary(Ozone_data_trend2)
```

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

> Answer: When comparing the Seasonal Mann-Kendall test to the nonseasonal, the resulting tau and 2-sided p-value are -0.165 and 0.0075402, respectively. Since a tau value of zero indicates no trend in the data, this lower tau value indicates a more negative trend in the data when the seasonal component is removed. The p-value also is orders of magnitude smaller, showing mucher stronger evidence of this trend. Overall, removing the seasonal component from the data offers more robust evidence of a negative trend.