Assignment 8: Time Series Analysis

Sayra Martinez

Spring 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

```
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=60), tidy=TRUE)
getwd()
#Import libraries and checking my working directory
library(tidyverse); library(lubridate)
#install.packages("trend")
library(trend)
#install.packages("zoo")
library(zoo)
#install.packages("Kendall")
library(Kendall)
#install.packages("tseries")
library(tseries)
library(here)
here

#2 Set theme
sayratheme <- theme_classic(base_size = 11) +
  theme(axis.text = element_text(color = "black"),
```

```
        legend.position = "right",
        legend.direction = "vertical",
        plot.title =
          element_text(color = "royalblue", size = 11))
theme_set(sayratheme)
```

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.

```
# 1
GaringerOzone.files = list.files(path = "./Data/Raw/Ozone_TimeSeries/",
    pattern = "*.csv", full.names = TRUE)

GaringerOzone.complete <- GaringerOzone.files %>%
    plyr::ldply(read.csv, stringsAsFactors = TRUE)
dim(GaringerOzone.complete)  # it is consistent with the given dimensions.
```

```
## [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 Column date as a date format
GaringerOzone.complete$Date <- as.Date(GaringerOzone.complete$Date,
    format = "%m/%d/%Y")
# 4 only contains the columns Date,
# Daily.Max.8.hour.Ozone.Concentration, and
# DAILY_AQI_VALUE. colnames(GaringerOzone.complete) #to be
# sure about the column names
GaringerOzone_processed <- select(GaringerOzone.complete, Date,
    Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5 Create a new data frame 'Days' from 2010-01-01 to
# 2019-12-31, and rename column

Days <- data.frame(Days = seq(as.Date("2010-01-01"), as.Date("2019-12-31"),
    by = "day"))
```

```r
colnames(Days) <- c("Date")

# Days <- data.frame(Date = seq(as.Date('2010-01-01'),
# as.Date('2019-12-31'), by = 'day')) ##Note for me: I can
# directly put the name of the column

# 6
GaringerOzone <- Days %>%
    left_join(GaringerOzone_processed, by = "Date") %>%
    select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)
dim(GaringerOzone)  #It corresponds to the given dimensions.
```

```
## [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?

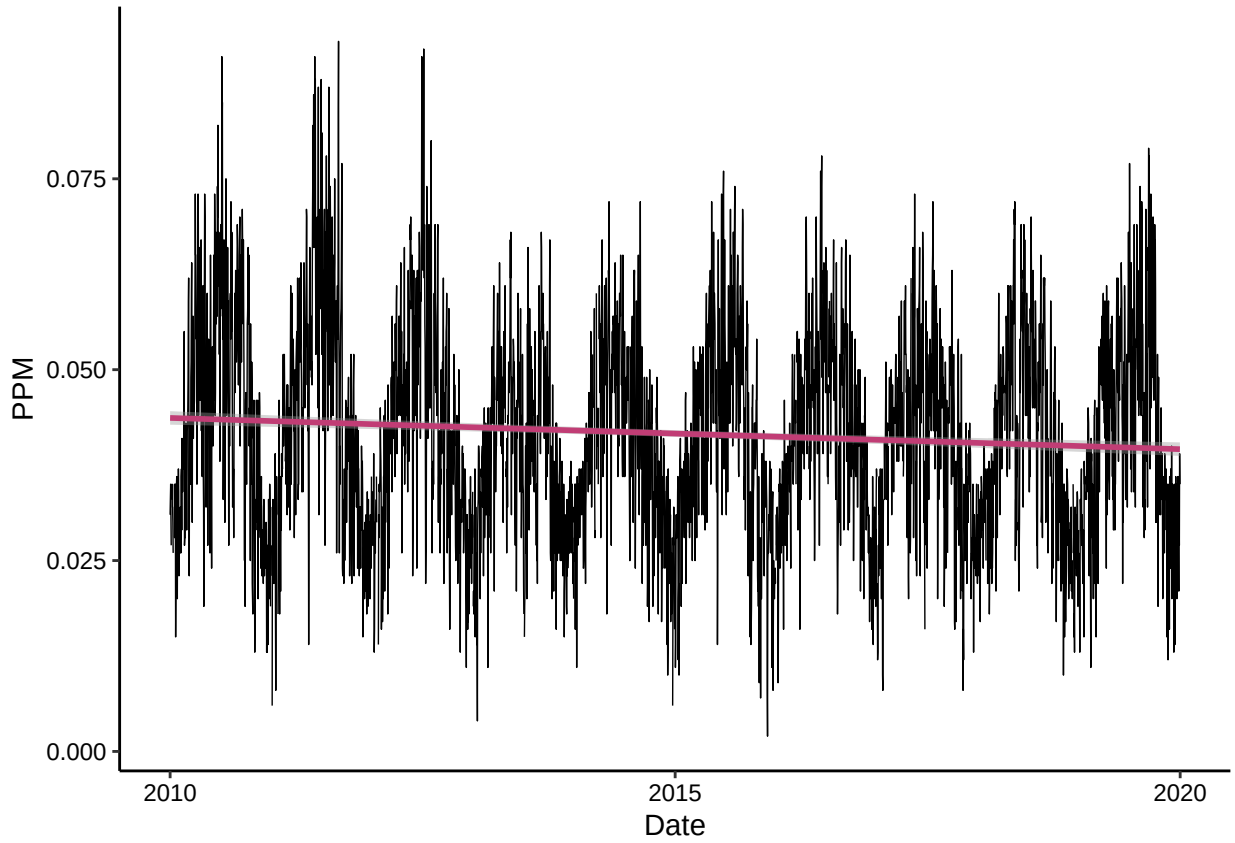
```r
# 7
PPM_plot <- ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
    geom_line(size = 0.25) + ylab("PPM") + geom_smooth(method = lm,
    color = "#c13d75ff")
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```r
print(PPM_plot)
```

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

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



Answer: Yes, the plot shows a slight negative slope, thus, we can infer that ozone concentrations has being slightly reducing along the period.

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

# Replacing missing values in
# Daily.Max.8.hour.Ozone.Concentration
GaringerOzone <- GaringerOzone %>%
    mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

summary(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration)
```

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

Answer: Because filling the missing data is simpler, as we assume missing data falls between the previous and next measurement. Besides, if there are not abrupt changes among the known data

there is no need to add complexity, so the linear interpolation keeps smooth transitions between the known points.

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 %>%
    mutate(month = month(GaringerOzone$Date), year = year(GaringerOzone$Date)) %>%
    select(Date, month, year, Daily.Max.8.hour.Ozone.Concentration,
        DAILY_AQI_VALUE) %>%
    group_by(year, month) %>%
    summarize(mean_ozone = mean(Daily.Max.8.hour.Ozone.Concentration,
        na.rm = TRUE)) %>%
    mutate(Date.new = as.Date(paste(year, month, "01", sep = "-")))
```

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

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
start_date_month <- min(GaringerOzone.monthly$Date.new)
start_date_month
```

```
## [1] "2010-01-01"
```

```
end_date_month <- max(GaringerOzone.monthly$Date.new)
end_date_month
```

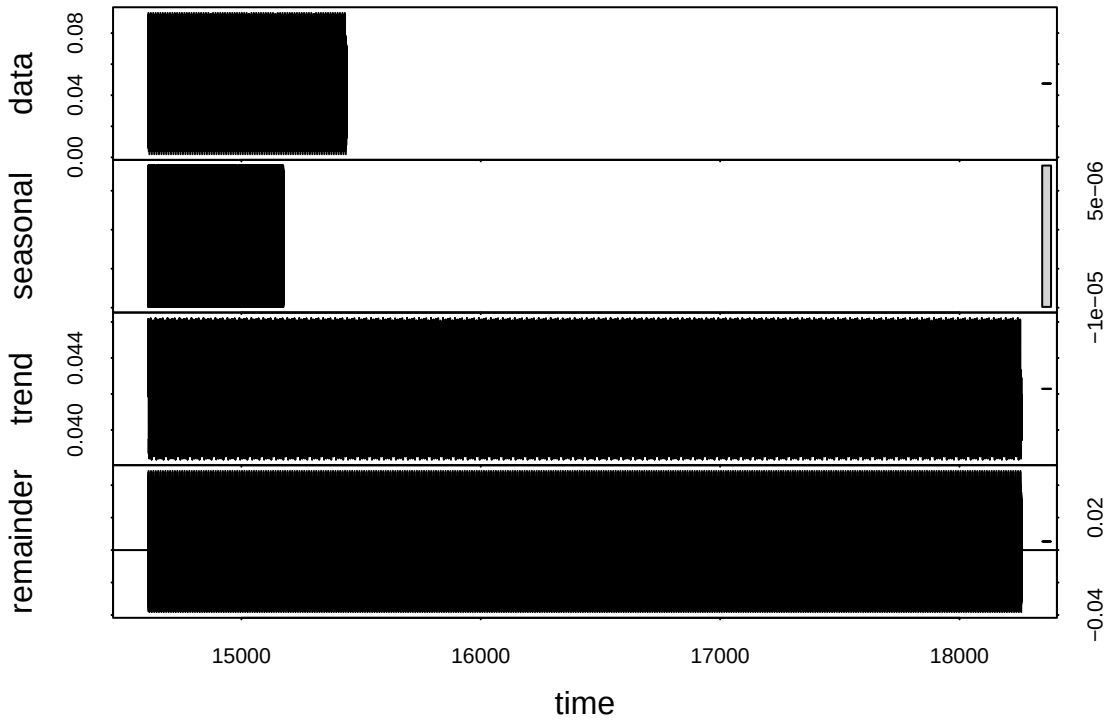
```
## [1] "2019-12-01"
```

```
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean_ozone,
    start = c(2010, 1), end = c(2019, 12), frequency = 12)

start_date_daily <- min(GaringerOzone$Date)
end_date_daily <- max(GaringerOzone$Date)
GaringerOzone.daily.ts <- ts(GaringerOzone$Daily.Max.8.hour.Ozone.Concentration,
    start = start_date_daily, end = end_date_daily, frequency = 365)
```

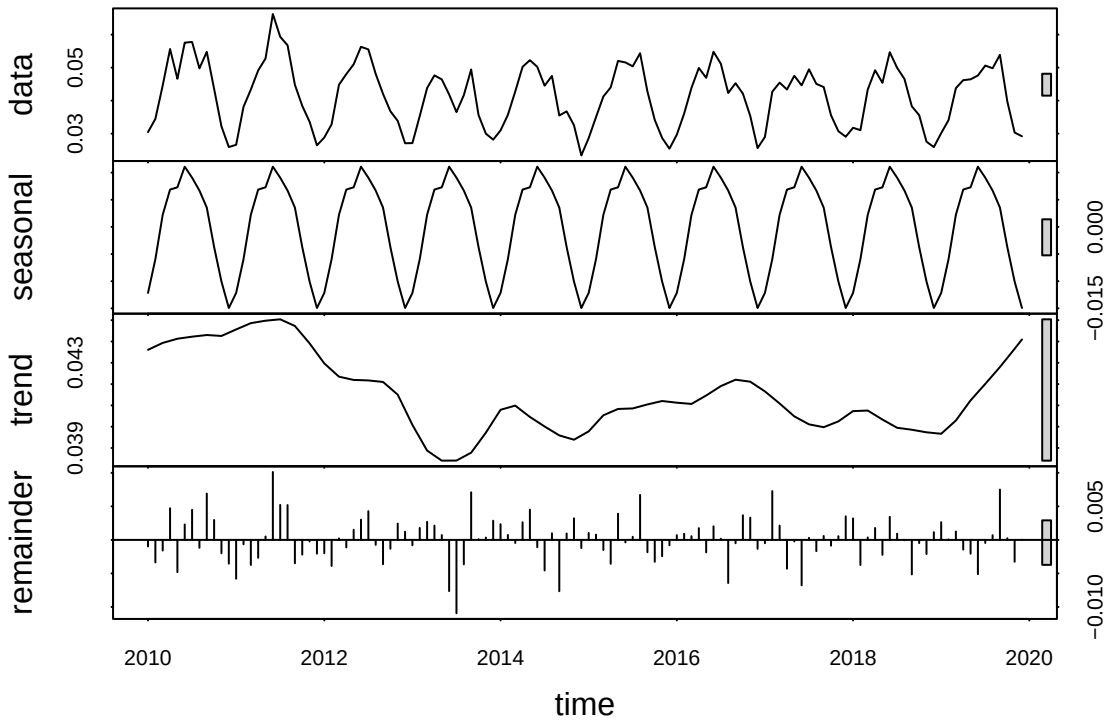
11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

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




```r
# Monthly time series

Decomposed_GaringerOzone.monthly <- stl(GaringerOzone.monthly.ts,
    s.window = "periodic")
plot(Decomposed_GaringerOzone.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 Running trend analysis and inspecting my results
GaringerOzone.monthly.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
GaringerOzone.monthly.trend
```

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

```
summary(GaringerOzone.monthly.trend)
```

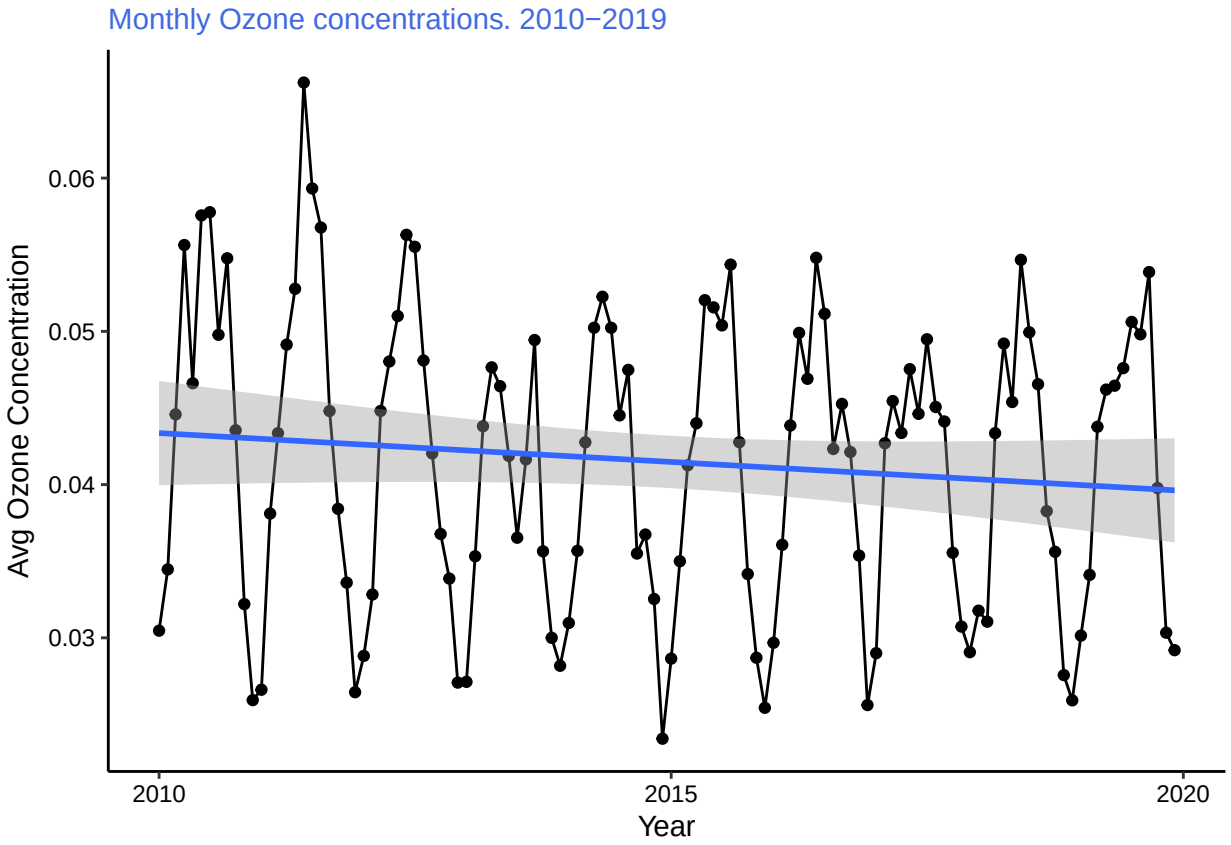
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: I think that seasonal Mann-Kendall is suitable because the information is a monthly data, and we can find seasonal patterns in the ozone concentration.

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
Monthly_plot <- ggplot(GaringerOzone.monthly, aes(x = Date.new,
    y = mean_ozone)) + geom_point() + geom_line() + ggtitle("Monthly Ozone concentrations. 2010-2019") +
    xlab(expression(paste("Year"))) + ylab(expression(paste("Avg Ozone Concentration"))) +
    geom_smooth(method = lm)
print(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: Regarding to the question "Have ozone concentrations changed over the 2010s at this station", evidence shows that yes, there has been a reduction in ozone concentration during the last years. The Seasonal MannKendall showed a tau = -0.143, which refers to that reduction, while the pvalue (0.046724), shows that such change is statistically significant at the 95% confidence level.

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
components.GaringerOzone.monthly <- as.data.frame(Decomposed_GaringerOzone.monthly$time.series[,
    1:3])

components.GaringerOzone.monthly <- mutate(components.GaringerOzone.monthly,
    Observed = GaringerOzone.monthly$mean_ozone, Date = GaringerOzone.monthly$Date.new)

# 16 First I create the time series with the non-seasonal
# monthly series
GaringerOzone.monthly.nonseasonal.ts <- ts(components.GaringerOzone.monthly$Observed,
    start = c(2010, 1), end = c(2019, 12), frequency = 12)

GaringerOzone.monthly.trend1 <- Kendall::MannKendall(GaringerOzone.monthly.nonseasonal.ts)
GaringerOzone.monthly.trend1
```

```
## tau = -0.0594, 2-sided pvalue =0.33732
```

```r
summary(GaringerOzone.monthly.trend1)
```

```
## Score =  -424 , Var(Score) = 194364.7
## denominator =  7139
## tau = -0.0594, 2-sided pvalue =0.33732
```

Answer: when removing seasonality, and running the Mann Kendall test I have that tau is still negative but lower when comparing with results for the complete monthly series. Besides, this time, the p value is 0.33732, so it is not statistically significant.