Assignment 7: Time Series Analysis

Laura Martinez

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 dataframe named `GaringerOzone` of 3589 observation and 20 variables.

```
# 1 Working directory
setwd("~/Documents/EDA-Fall2022")
getwd()
```

```
## [1] "/Users/laura/Documents/EDA-Fall2022"
```

```
library(tidyverse)
library(lubridate)
# install.packages('zoo')
library(zoo)
# install.packages('trend')
library(trend)

# Set theme
Laurastheme <- theme_classic(base_size = 12) + theme(axis.text = element_text(color = "black"),
    legend.position = "right")
theme_set(Laurastheme)

# 2
```

```
Ozone_2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
    stringsAsFactors = TRUE)
Ozone_2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
    stringsAsFactors = TRUE)
Ozone_2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
    stringsAsFactors = TRUE)
Ozone_2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
    stringsAsFactors = TRUE)
Ozone_2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
    stringsAsFactors = TRUE)
Ozone_2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
    stringsAsFactors = TRUE)
Ozone_2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
    stringsAsFactors = TRUE)
Ozone_2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
    stringsAsFactors = TRUE)
Ozone_2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
    stringsAsFactors = TRUE)
Ozone_2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
    stringsAsFactors = TRUE)

# Combine datasets
GaringerOzone <- rbind(Ozone_2010, Ozone_2011, Ozone_2012, Ozone_2013,
    Ozone_2014, Ozone_2015, Ozone_2016, Ozone_2017, Ozone_2018,
    Ozone_2019)
```

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
GaringerOzone$Date <- as.Date(GaringerOzone$Date, format = "%m/%d/%Y")

# 4
Ozone_wrangled <- select(GaringerOzone, c("Date", "Daily.Max.8.hour.Ozone.Concentration",
    "DAILY_AQI_VALUE"))

# 5

sequence1 <- seq(as.Date("2010-01-01"), as.Date("2019-12-31"),
    by = "1 day")

Days <- as.data.frame(sequence1)
```

```
names(Days)[names(Days) == "sequence1"] <- "Date"

# 6
library(dplyr)
GaringerOzone1 <- left_join(Days, Ozone_wrangled, by = "Date")

summary(GaringerOzone1)
```

```
##       Date            Daily.Max.8.hour.Ozone.Concentration DAILY_AQI_VALUE
##  Min.   :2010-01-01   Min.   :0.00200                      Min.   :  2.00
##  1st Qu.:2012-07-01   1st Qu.:0.03200                      1st Qu.: 30.00
##  Median :2014-12-31   Median :0.04100                      Median : 38.00
##  Mean   :2014-12-31   Mean   :0.04163                      Mean   : 41.57
##  3rd Qu.:2017-07-01   3rd Qu.:0.05100                      3rd Qu.: 47.00
##  Max.   :2019-12-31   Max.   :0.09300                      Max.   :169.00
##                       NA's   :63                           NA's   :63
```

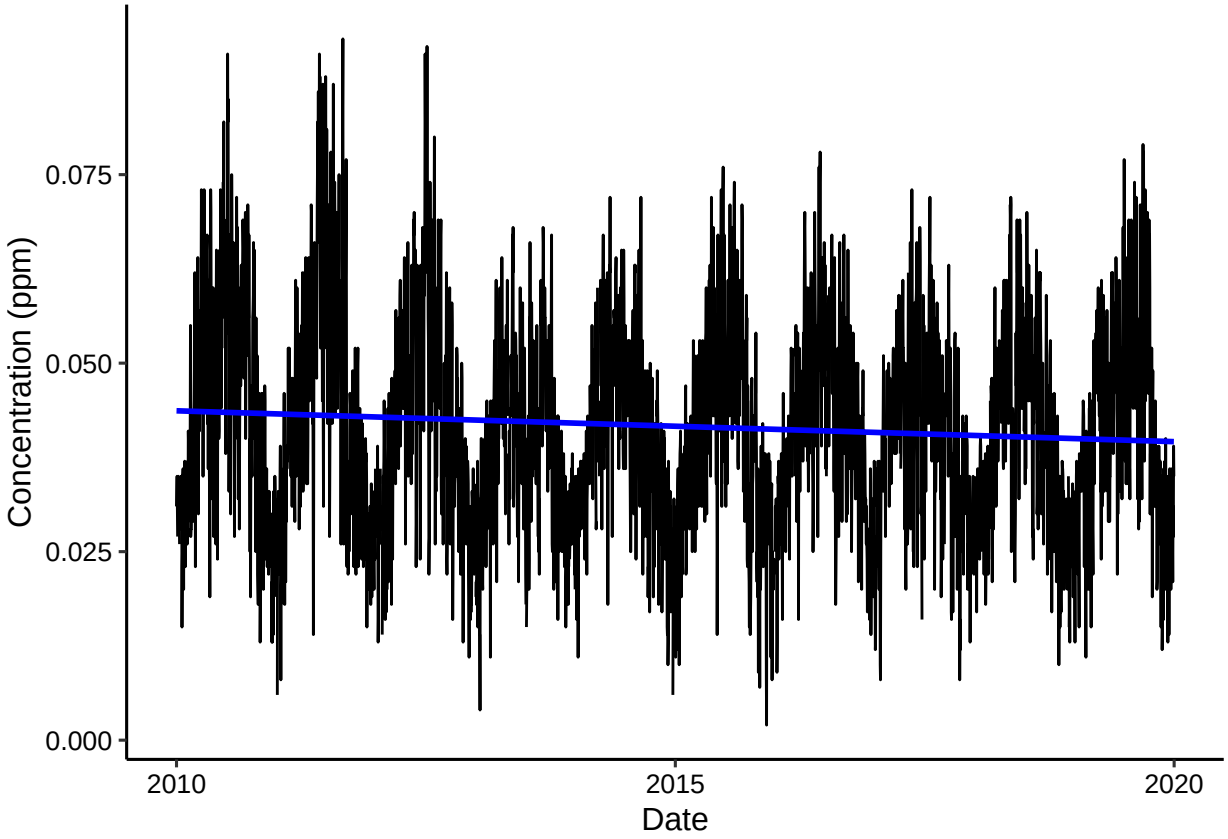
Visualize

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
line.plot <- ggplot(GaringerOzone1, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
    geom_line() + geom_smooth(method = "lm", se = FALSE, color = "Blue") +
    xlab("Date") + ylab("Concentration (ppm)")
print(line.plot)
```

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

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



Answer: Although the data shows a cyclical nature to the data, the overall plot shows a slightly negative trend over time. The smoothed line shows a negative decline from the year 2010 to 2019, which signifies a negative relationship between time and concentration in ppm).

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
GO_interpolate <- GaringerOzone1 %>%
    mutate(Ozone_Cxn = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))
```

Answer: Linear interpolation fills in missing data by computing the value from the previous and next measurement, whereas piecewise constant uses a "nearest neighbor" approach. In doing so, the data is filled by assuming the value of the date nearest to it. This approach may be faulty depending on the availaibilty of data and the nearest data. Spline uses the quadratic formula or polynomials to interpolate missing values instead of drawing a straight line between the closest values, but in this case we do not have to worry about cycles.

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 <- GO_interpolate %>%
    mutate(Month = month(Date), Year = year(Date)) %>%
```

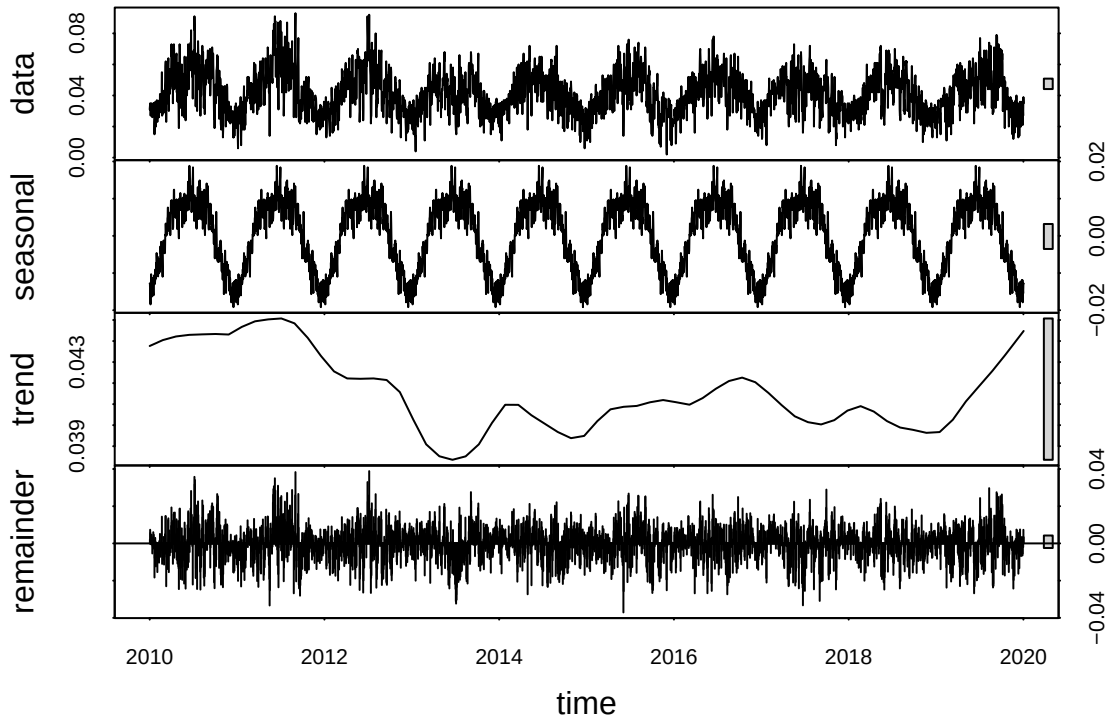
```
    mutate(dateMY = my(paste0(Month, "-", Year))) %>%
    group_by(dateMY) %>%
    summarise(Mean_Ozone = mean(Ozone_Cxn), na.rm = TRUE)
```

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.

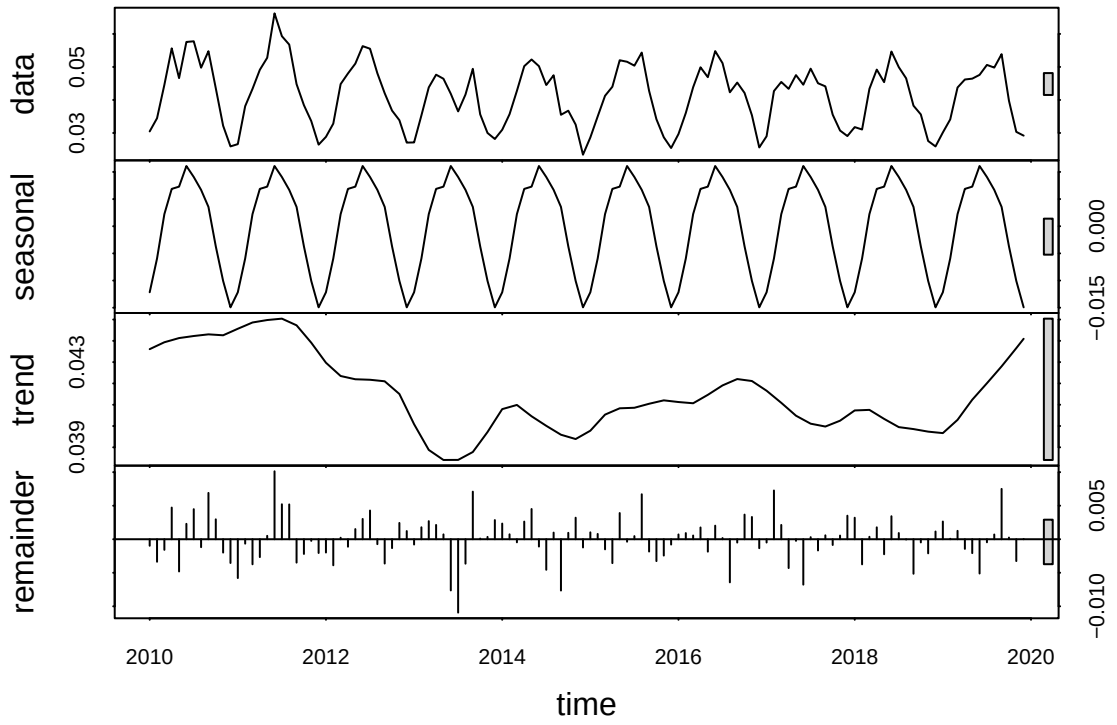
```
# 10
f_month1 <- month(first(GO_interpolate$Date))
f_year1 <- year(first(GO_interpolate$Date))

GaringerOzone.daily.ts <- ts(GO_interpolate$Ozone_Cxn, start = c(f_year1,
    f_month1), frequency = 365)

f_month2 <- month(first(GaringerOzone.monthly$dateMY))
f_year2 <- year(first(GaringerOzone.monthly$dateMY))

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$Mean_Ozone,
    start = c(f_year2, f_month2), frequency = 12)
```

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

```
# 11
GO.daily_decomp <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GO.daily_decomp)
```




```
GO.monthly_decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GO.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
Ozone_trend1 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
```

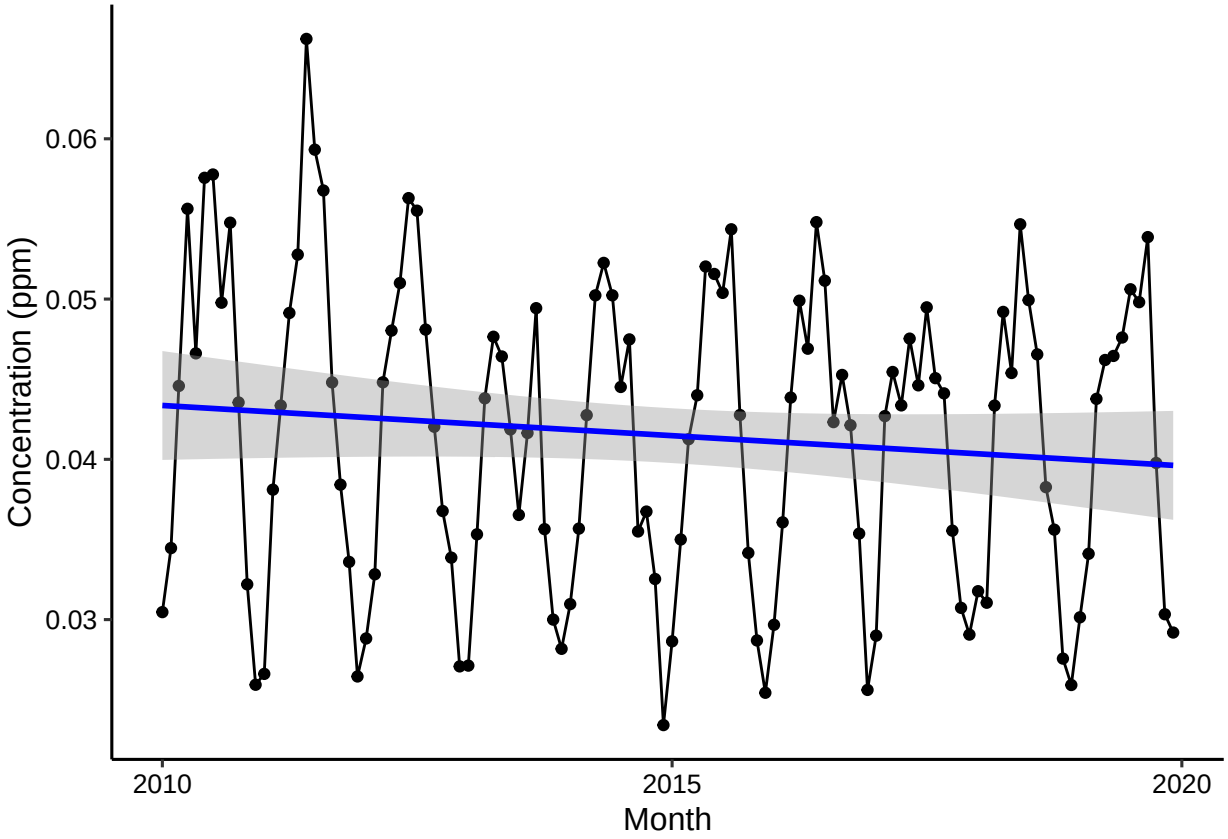
Answer: The seasonal Mann Kendall tests for seasonality, whereas the Mann Kendall requires seasonality has to be removed in order for the test to run.

- **Mann-Kendall**: no seasonality, non-parametric, missing data allowed. Function: `MannKendall()` (package: Kendall)
- **Seasonal Mann-Kendall**: seasonality, non-parametric `SeasonalMannKendall` (package: Kendall)

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
Monly_Ozone.plot <- ggplot(GaringerOzone.monthly, aes(x = dateMY,
    y = Mean_Ozone)) + geom_line() + geom_point() + geom_smooth(method = "lm",
    se = TRUE, color = "Blue") + xlab("Month") + ylab("Concentration (ppm)")
print(Monly_Ozone.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: Ozone concentrations have decreased slightly over the course of 2010 to 2019 as shown by the linear regression. The Mann Kendall test evaluates whether the trend in data is increasing or decreasing over time and if this relationship is statsictically significant. The results from the Mann Kendall further emphasize a negative correlation between monthly ozone and time. The tau value of -0.1427 shows a negative trend that decreases as time goes on. The results from the seasonal Mann Kendall test also have an S value of -77, which indicate a downward trend because $S < 0$. The double-sided p-value is less than 0.5, which means that we cannot reject the null (no relationship) and the relationship between ozone and time is 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.

```
# 15
GO_NonSeasonal <- (GaringerOzone.monthly.ts - GO.monthly_decomp$time.series[,
    1])

# 16
Ozone_trend2 <- Kendall::MannKendall(GO_NonSeasonal)
```

    Answer: The Mann Kendall test reports an S value of -1179, which signifies a negative trend. Although this value is lower than the value of the the Seasonal Mann Kendall, both values indicate a negative trend in the data. Similar to the SMK, the MK tau value is -0.1651, which signifies a negative trend in the data. The p-value of the MK is 0.00754; this like the SMK indicates that the

null hypothesis is rejected and there is a relationship between Ozone and date that is statistically significant.