Assignment 7: Time Series Analysis

Logan Dye

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
getwd()
```

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

```
library(tidyverse)
library(lubridate)
library(zoo)
library(trend)

mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "red"),
        legend.position = "bottom")
theme_set(mytheme)

EPAair2010 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv")
EPAair2011 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv")
EPAair2012 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv")
EPAair2013 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv")
```

```
EPAair2014 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv")
EPAair2015 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv")
EPAair2016 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv")
EPAair2017 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv")
EPAair2018 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv")
EPAair2019 <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv")

GaringerOzone <- rbind(EPAair2010, EPAair2011, EPAair2012, EPAair2013, EPAair2014, EPAair2015, EPAair20
```

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

GaringerOzone1 <- select(GaringerOzone, 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"), "days"))
colnames(Days) <- c("Date")

# 6
GaringerOzone <- left_join(Days, GaringerOzone1, by = c("Date"))
```

Visualize

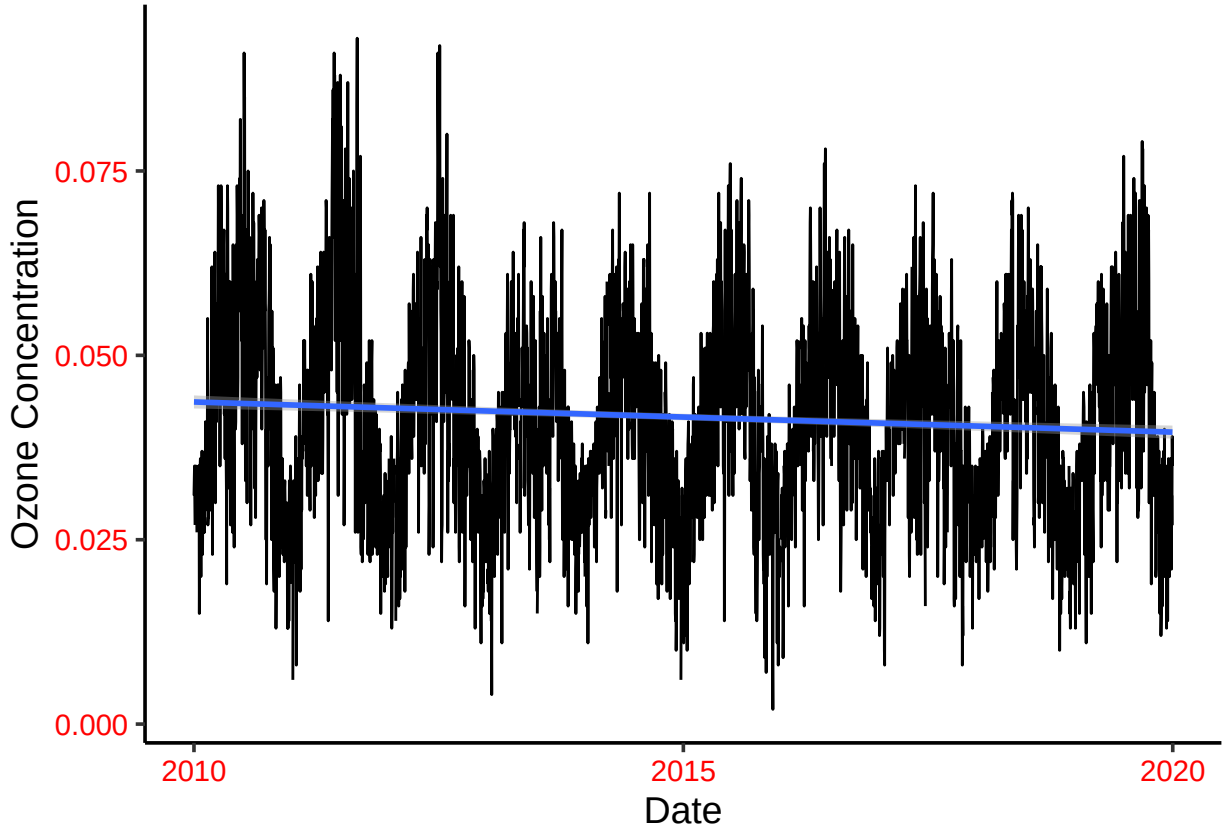
7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7

OzonePlot <-
  ggplot(GaringerOzone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
  geom_line() +
  geom_smooth(method = lm) +
  ylab("Ozone Concentration")
print(OzonePlot)
```

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

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



Answer: The plot suggests there may be a slight downward trend in the Ozone Concentration over time.

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( Daily.Max.Clean = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration) )

GaringerOzone_Clean <-
  GaringerOzone_Clean %>%
  select(Date, Daily.Max.Clean, DAILY_AQI_VALUE)

summary(GaringerOzone)
```

```
##       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
```

```
summary(GaringerOzone_Clean)
```

```
##       Date            Daily.Max.Clean   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   :0.04151   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
```

Answer: When we looked at the data vizualization there were no large jumps in the data and the shape did not look like a quadratic function. Since we know that the data will not connect with a quadratic function we do not want to us spline. Likewise, since there are no big jumps of missing data points the linear interpolation makes more sense than the piecewise constant interpolation becasue we can assume that it was a relativly linear path from one observed observation to the next.

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(Month = month(Date)) %>%
  mutate(Year = year(Date)) %>%
  group_by(Year, Month) %>%
  summarize(MeanMonthlyO3 = mean(Daily.Max.Clean)) %>%
  mutate(Month_Year = my(paste(Month, "-", Year)))
```

```
## `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.

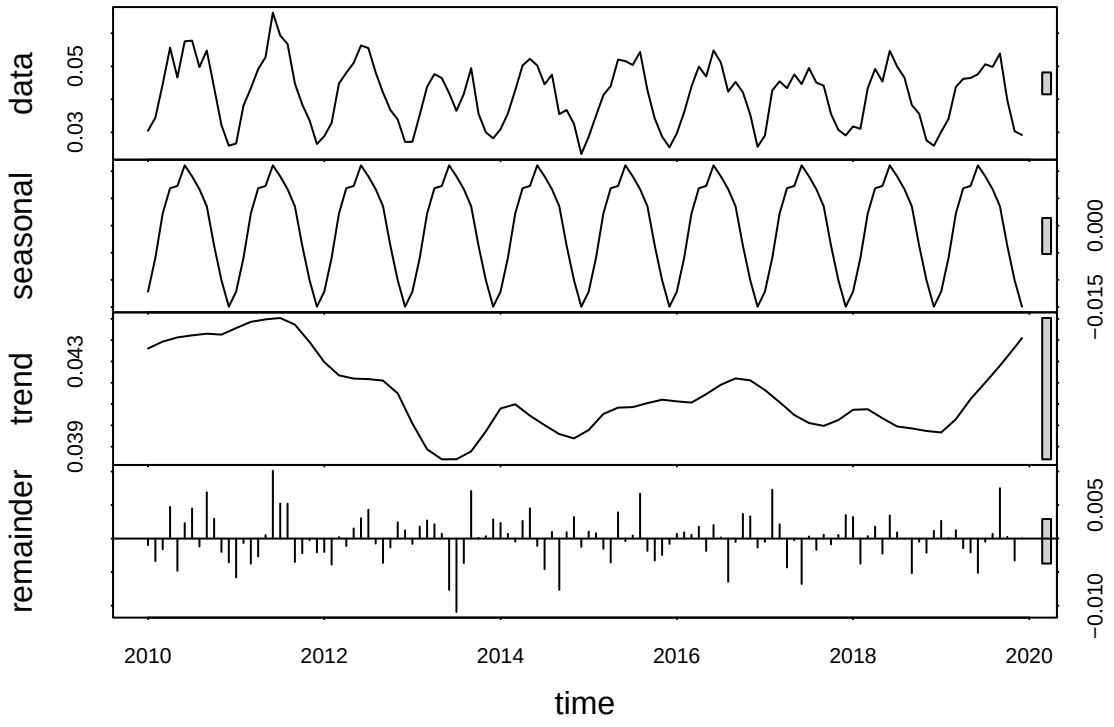
```
#10
f_month <- month(first(GaringerOzone.monthly$Month_Year))
f_year <- year(first(GaringerOzone.monthly$Month_Year))
GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$MeanMonthlyO3,
                   start=c(f_year,f_month),
                   frequency=12)

f_month2 <- month(first(GaringerOzone_Clean$Date))
f_year2 <- year(first(GaringerOzone_Clean$Date))
f_day2 <- day(first(GaringerOzone_Clean$Date))
GaringerOzone.daily.ts <- ts(GaringerOzone_Clean$Daily.Max.Clean,
                             start = c(f_year2, f_month2, f_day2),
```

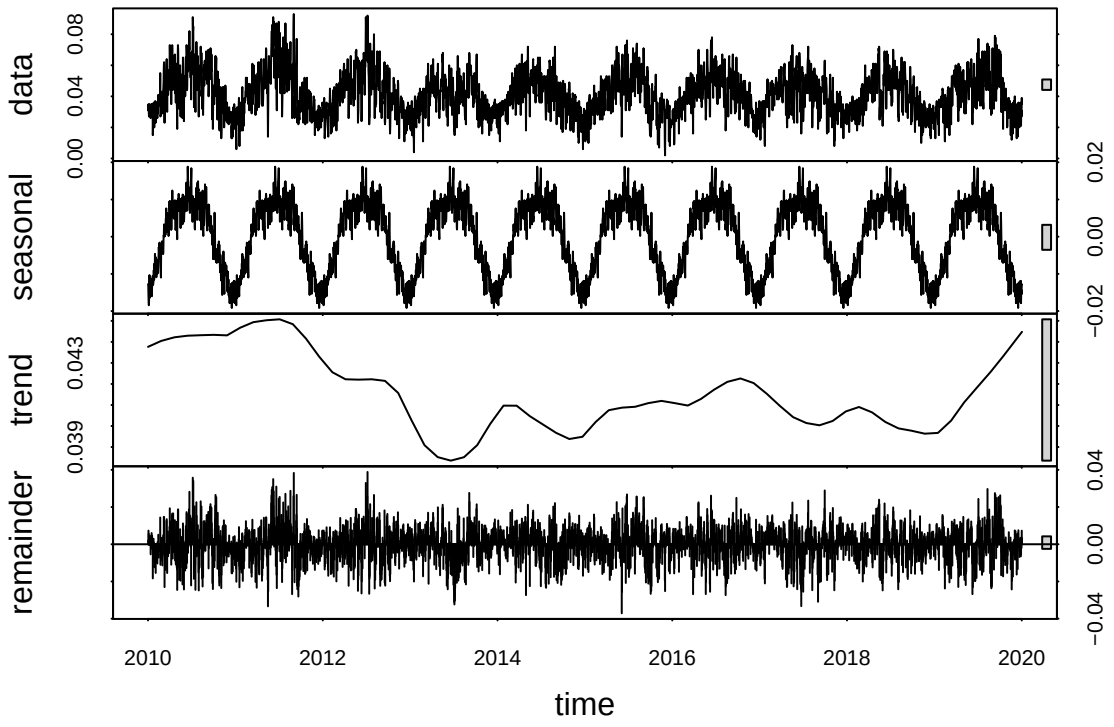
```
                                frequency = 365)
```

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

```
#11
Monthly_decomp <- stl(GaringerOzone.monthly.ts,s.window = "periodic")
plot(Monthly_decomp)
```




```
Daily_decomp <- stl(GaringerOzone.daily.ts,s.window = "periodic")
plot(Daily_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

Monthly_Ozone_Trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)

Monthly_Ozone_Trend
```

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

```
summary(Monthly_Ozone_Trend)
```

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

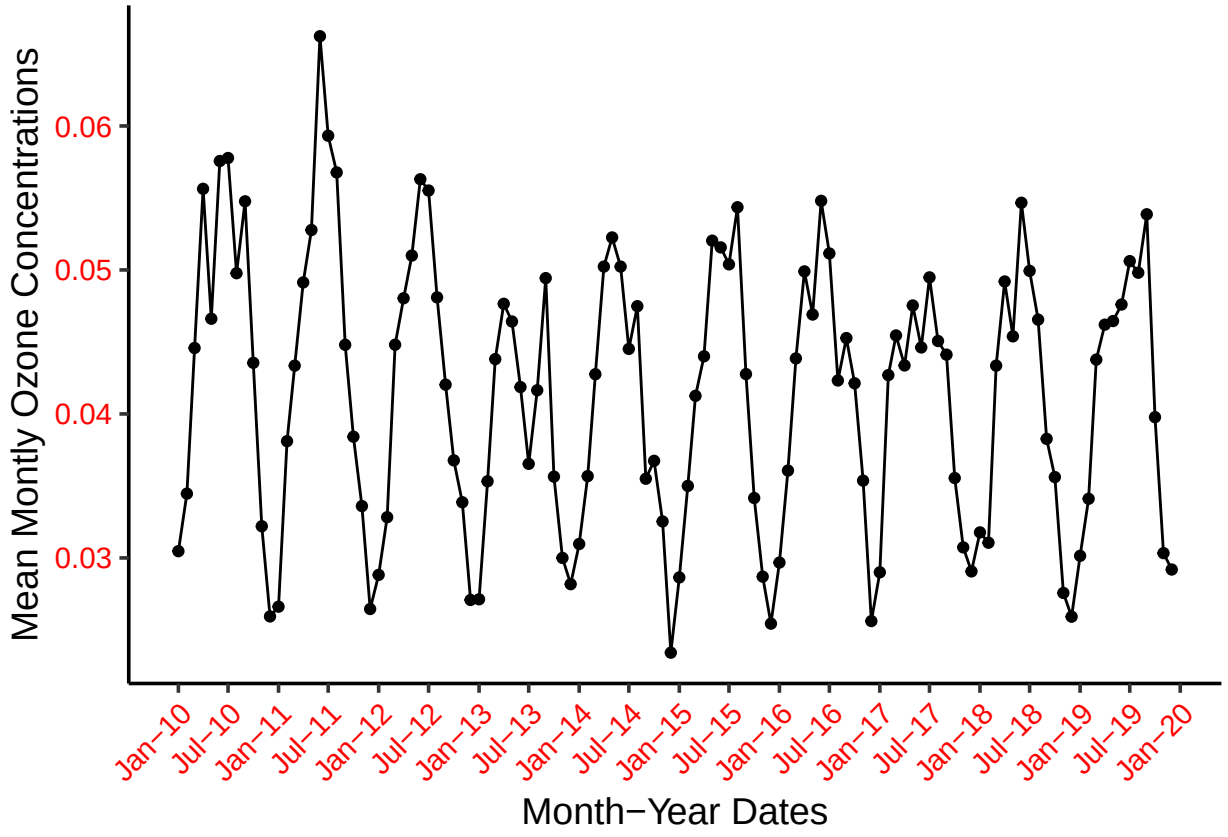
Answer: A seasonal Mann-Kendall trend analysis is more appropriate becasue our Ozone data is showing a large about on seasonality. THe seasonal Mann-Kendall trend analysis is the only one that is able to account for the seasonality without us substituting it out.

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

Monthly_Ozone_Plot <-
  ggplot(GaringerOzone.monthly, aes(x = Month_Year, y = MeanMonthlyO3)) +
  geom_point() +
```

```
  geom_line() +
   scale_x_date(limits = as.Date(c("2010-01-01", "2019-12-01")),
    date_breaks = "6 months", date_labels = "%b-%y") +
  ylab("Mean Montly Ozone Concentrations") +
  xlab("Month-Year Dates") +
   theme(axis.text.x = element_text(angle = 45,  hjust = 1))
print(Monthly_Ozone_Plot)
```



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: Our results show that there is a slight negative trend in ozone concentrations over time. This trend is statistically significant. (Score = -77, 2-sided pvalue = 0.046724)

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

Garinger.Ozone.Monthly_Components <- as.data.frame(Monthly_decomp$time.series[,2:3])

Garinger.Ozone.Monthly_Components <-
  mutate(Garinger.Ozone.Monthly_Components) %>%
  mutate(Date = GaringerOzone.monthly$Month_Year) %>%
```

```
  mutate(NonseasonalData = trend + remainder) %>%
  select(Date, NonseasonalData)

summary(Garinger.Ozone.Monthly_Components)
```

```
##       Date            NonseasonalData
##  Min.   :2010-01-01   Min.   :0.02747
##  1st Qu.:2012-06-23   1st Qu.:0.03932
##  Median :2014-12-16   Median :0.04120
##  Mean   :2014-12-16   Mean   :0.04149
##  3rd Qu.:2017-06-08   3rd Qu.:0.04325
##  Max.   :2019-12-01   Max.   :0.05514
```

```
Garinger.Ozone.Nonseasonal_ts <- ts(Garinger.Ozone.Monthly_Components$NonseasonalData, start = c(2010,1)


#16

Garinger.Ozone.Monthly.MK <- Kendall::MannKendall(Garinger.Ozone.Nonseasonal_ts)

summary(Garinger.Ozone.Monthly.MK)
```

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

Answer: Removing seasonality from the time series creates a much higher score, moving from -77 to -1179, and a much lower pvalue, moving from 0.046724 to 0.0075402. By removing the seasonality we created a much clearer and more confident timeseries, further showing the impact seasonality had on our dataset.