Assignment 7: Time Series Analysis

Matthew Vining

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

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)


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
wrangle.4 <- GaringerOzone %>%
    select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
Days <- as.data.frame(seq.Date(from = as.Date("2010-01-01"), to = as.Date("2019-12-31"),
    by = 1))
names(Days) <- c("Date")

# 6
Garinger_Ozone <- left_join(Days, wrangle.4, by = c("Date"))
summary(Garinger_Ozone)
```

```
##       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
```

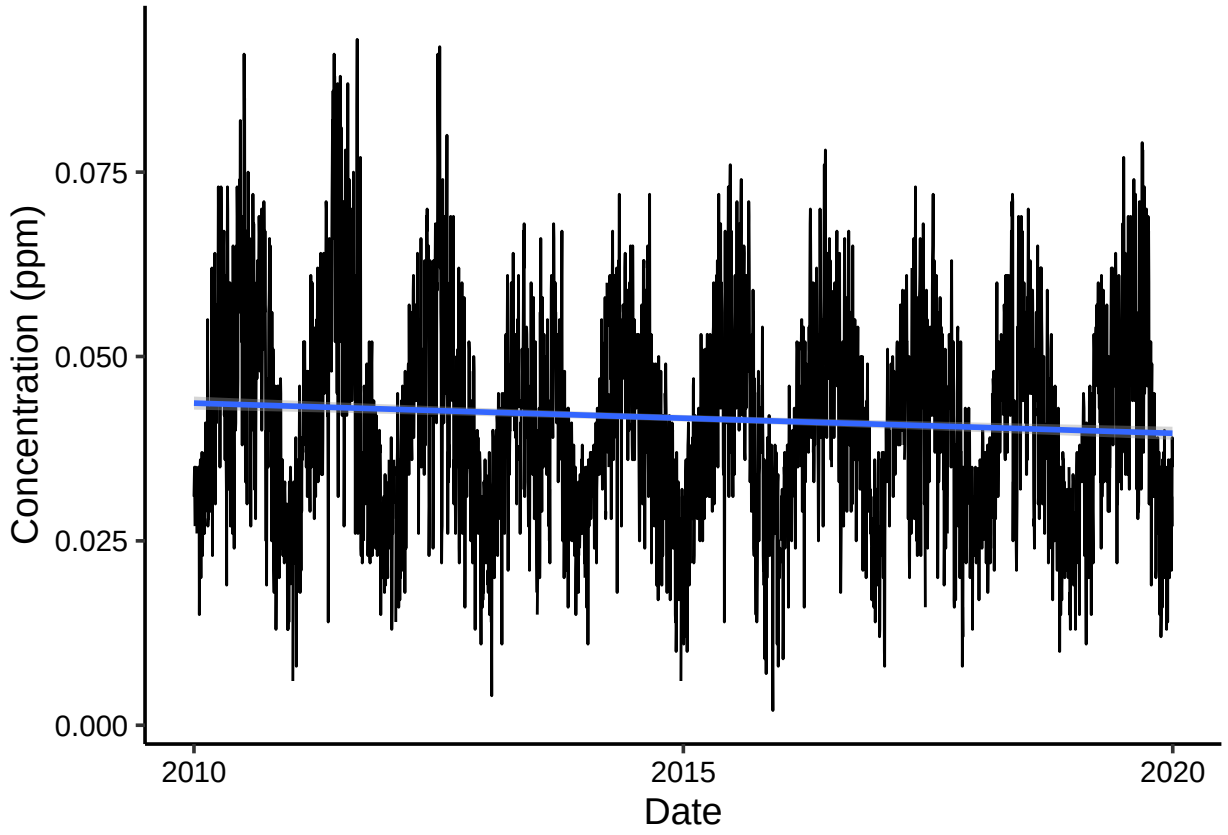
Visualize

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
plot.7 <- ggplot(Garinger_Ozone, aes(x = Date, y = Daily.Max.8.hour.Ozone.Concentration)) +
    geom_line() + geom_smooth(method = "lm") + xlab("Date") + ylab("Concentration (ppm)")
print(plot.7)
```

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

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



Answer:Based on the visualization of data from the plot and blue linear trend line, it appears that there is a negative linear relationship associated with date and ozone concentration with a small rate of change 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
Ozone.inter <- Garinger_Ozone %>%
    mutate(Daily.Max.8.hour.Ozone.Concentration = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

summary(Ozone.inter)
```

```
##       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   :0.04151                      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: Both piece wise and spline interprolations look at connecting dots between existing data where data is missing beyond a linear interprolation, accounting for functions such as exponential or quadratic. Because the plotted relationship appears to have a linear trend, we opt for the linear interprolation to fill in the missing data. If visualization of data looked to have a non-linear trend, then it would make sense to fill in the missing data in a non-linear format.

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)

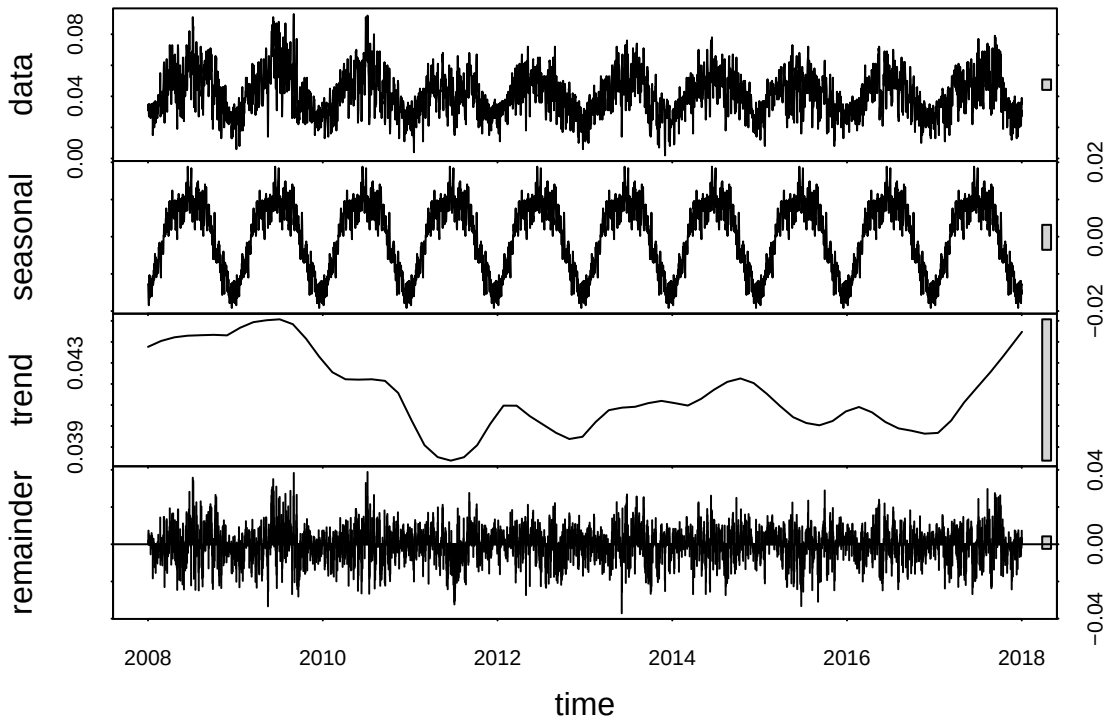
```
# 9
GaringerOzone.monthly <- Ozone.inter %>%
    mutate(month_year = floor_date(Date, "month")) %>%
    group_by(month_year) %>%
    summarise(mean.conc = mean(Daily.Max.8.hour.Ozone.Concentration))
```

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.

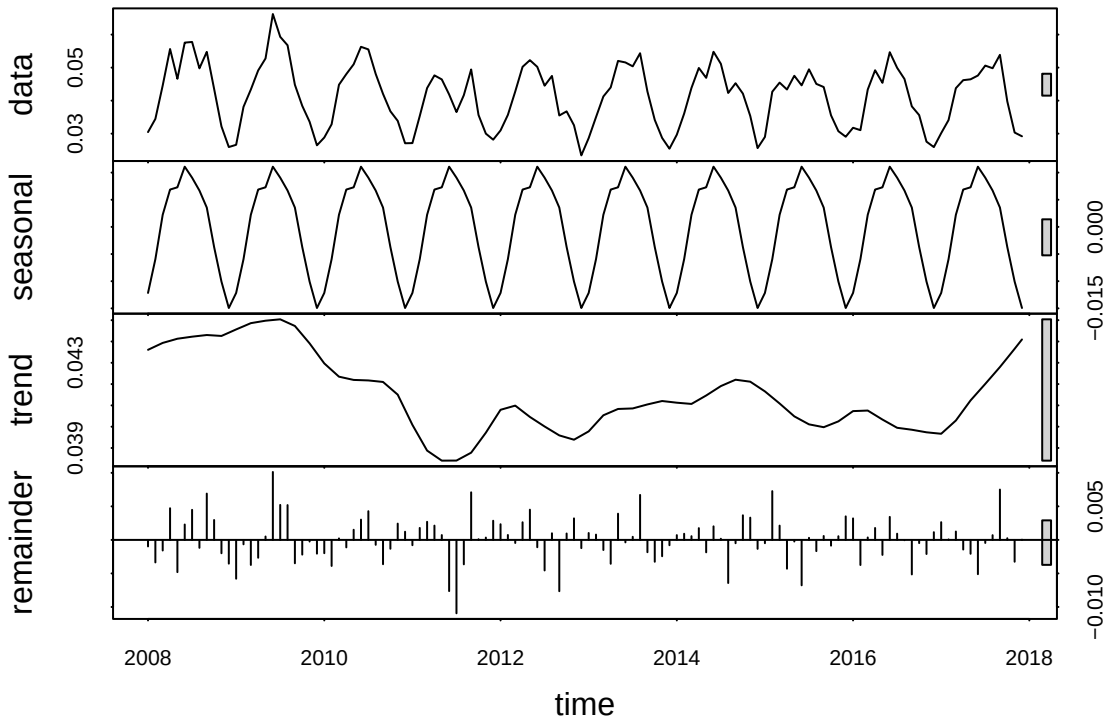
```
# 10
GaringerOzone.daily.ts <- ts(Ozone.inter$Daily.Max.8.hour.Ozone.Concentration, start = c(2010 -
    1 - 1), frequency = 365)

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$mean.conc, start = c(2010 -
    1 - 1), frequency = 12)
```

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

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




```
monthly_decomp <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(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
trend_12 <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
summary(trend_12)
```

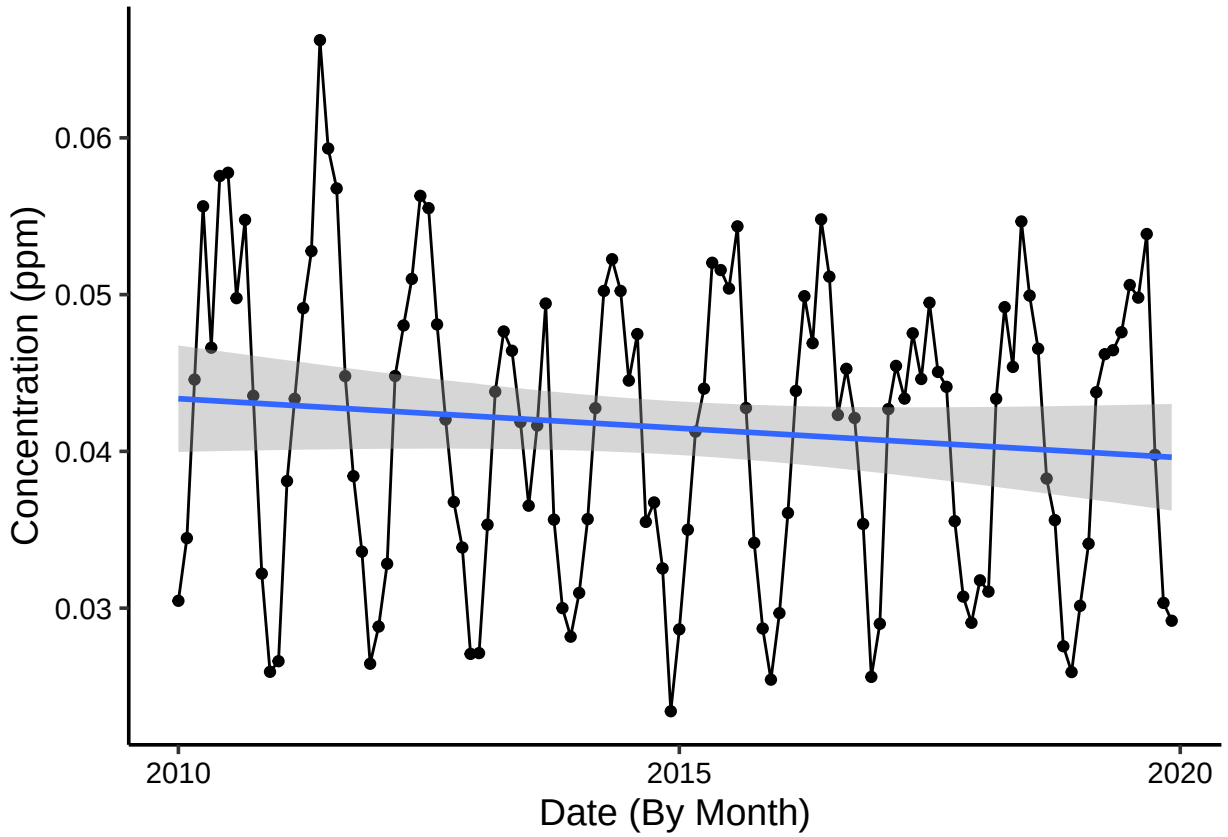
```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

Answer: We want to use the seasonal Mann-Kendall case for this series because our plotted decomposition shows strong signs of a seasonal trend in ozone concentrations over time.

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
plot.13 <- ggplot(GaringerOzone.monthly, aes(x = month_year, y = mean.conc)) + geom_line() +
    geom_point() + geom_smooth(method = "lm") + xlab("Date (By Month)") + ylab("Concentration (ppm)")
print(plot.13)
```

```
## `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: Based on our results, it appears that ozone concentration has in fact changed in teh 2010s at this station. One fact is that our Mann Kendall test shows a negative montomic trend of -77 and has a signfiicant p-velu of 0.046 for a 95% confidence interval. These statistics indidcate significance against a null hypothesis that there is no change. Additionally, the graph shows a clear decreasing linear trend in our data to accompany our statistical output.

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
series_Components <- as.data.frame(monthly_decomp$time.series[, 2:3])

# 16
series.ts <- ts(series_Components$trend, start = c(2010 - 1 - 1), frequency = 12)
trend_16 <- Kendall::SeasonalMannKendall(series.ts)
summary(trend_16)
```

```
## Score =  -164 , Var(Score) = 1500
## denominator =  540
## tau = -0.304, 2-sided pvalue =2.291e-05
```

```
summary(trend_12)
```

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

Answer: After removing the seasonal components in out time series, the Mann Kendall test shows that the non-seasonal series has a lower p-value than the one that contained seasonal components. This indicates that when removing seasonal components, we have a stronger signficance associated with the relationship between monthly ozone concentration over time (linear). We also see that the non-seasonal test has a higher score, indicating a stronger monotomic downward trend. Controlling for seasonal variation in ozone concentration allowed the statistical test to provide a stronger reasonoing for rejecting an null hypothesis that there is no change in the 2010s.